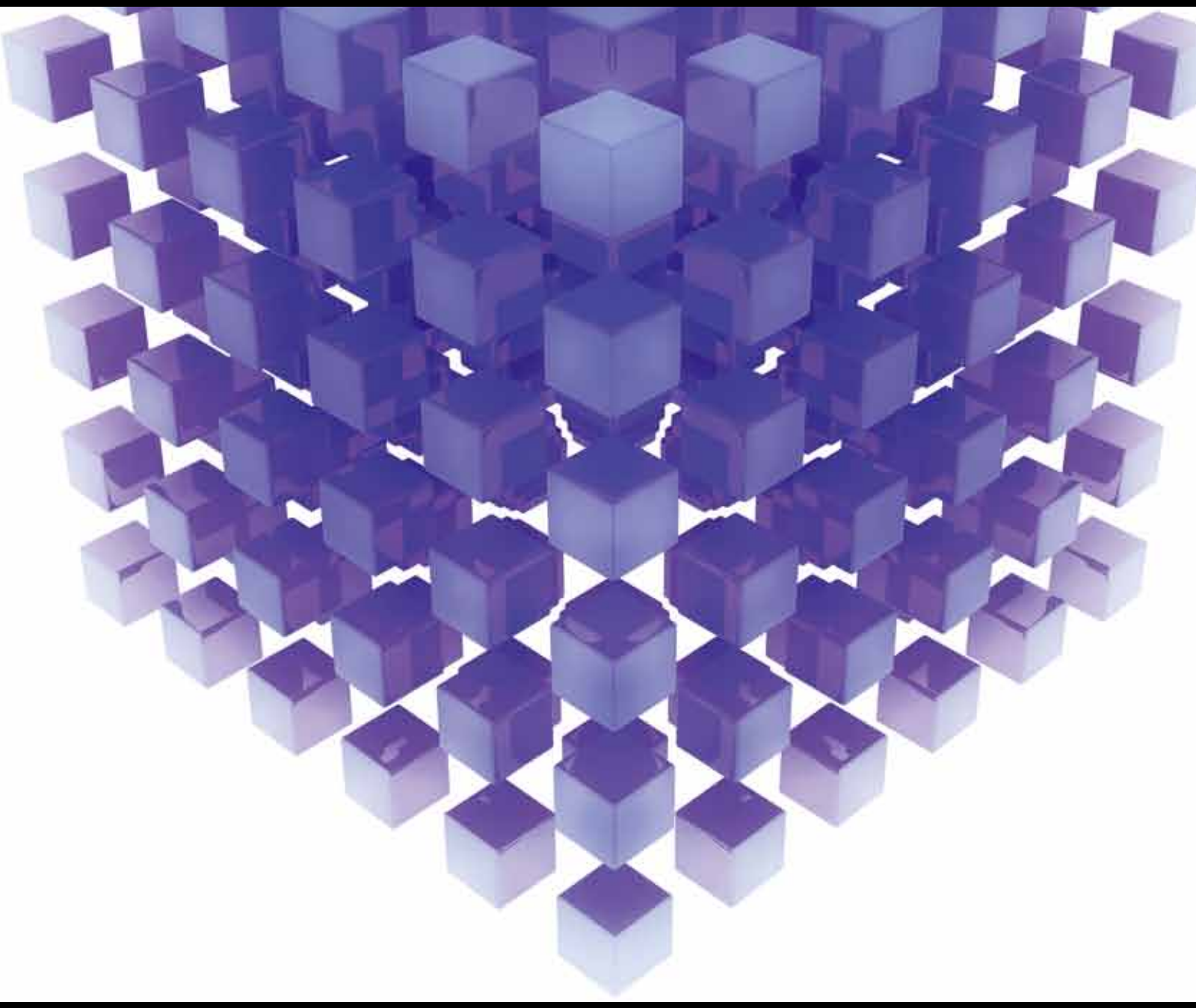
Mathematical Problems in Engineering

# Applied Mathematics and Algorithms for Cloud Computing and IoT

Guest Editors: Yuxin Mao, Vijay Bhuse, Zhongmei Zhou, Pit Pichappan, Mahmoud Abdel-Aty, and Yoshinori Hayafuji



# Applied Mathematics and Algorithms for Cloud Computing and IoT

Mathematical Problems in Engineering

# Applied Mathematics and Algorithms for Cloud Computing and IoT

Guest Editors: Yuxin Mao, Vijay Bhuse, Zhongmei Zhou, Pit Pichappan, Mahmoud Abdel-Aty, and Yoshinori Hayafuji

# Editorial Board

# Contents

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 946860, 2 pages
http://dx.doi.org/10.1155/2014/946860

*Editorial*

# Applied Mathematics and Algorithms for Cloud Computing and IoT

**Yuxin Mao,[1] Vijay Bhuse,[2] Zhongmei Zhou,[3] Pit Pichappan,[4] Mahmoud Abdel-Aty,[5] and Yoshinori Hayafuji[6]**

[1] *School of Computer and Information Engineering, Zhejiang Gongshang University, Hangzhou 310018, China*
[2] *Department of Computing, East Tennessee State University, Johnson City, TN 37614-1266, USA*
[3] *Department of Computer Science and Engineering, Zhangzhou Normal University, Zhangzhou, China*
[4] *School of Information Systems, Al Imam University, Riyadh, Saudi Arabia*
[5] *Scientific Publishing Center, Bahrain University, 32038 Sakhir, Bahrain*
[6] *Graduate School of Science and Technology, Kwansei Gakuin University, 2-1 Gakuen, Sanda, Hyogo 669-1337, Japan*

Correspondence should be addressed to Yuxin Mao; maoyuxin@zjgsu.edu.cn

Received 8 January 2014; Accepted 8 January 2014; Published 13 February 2014



Today's world is drifting in all new-fangled technology revolution due to the influence of cloud computing and Internet of Things (IoT) technologies. Currently, cloud computing and IoT are the hottest issues of future Internet.

Cloud computing provides a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction [1]. Cloud computing can be treated as a long-held dream of computing as a utility and has the potential to transform a large part of the IT industry, shaping the way IT hardware is designed and purchased [2]. The term Internet of Things (IoT) was proposed by Kevin Ashton in 1999 [3].

As another emerging research field, IoT draws up a novel paradigm that is rapidly gaining ground in the scenario of modern wireless telecommunications [4]. The major strength of the IoT idea is the high impact it will bring on several aspects of everyday life and behavior of potential users. The research of IoT is still in its infancy. Therefore, there are not any standard definitions for Internet of Things [5].

We experience altogether a different capability when these two technologies go hybrid. There are plenty of new tools and applications coming up in a day-to-day manner in these two fields for our life. However, there are many complex real-life problems (especially engineering problems) in these fields, which require us to solve by using mathematical methods and efficient algorithms. We shall think about these new challenges from the points of both mathematics and computers.

The papers in this special issue cover a wide range of topics in cloud computing and IoT, from underlying infrastructure and algorithms to high-level systems and applications.

The term *cloud computing* is mostly used to sell hosted services in the sense of application service provisioning at a remote location. At the foundation of cloud computing is the broader concept of shared services and converged infrastructure. Therefore, service is obviously a very important issue in cloud computing (also for IoT). In this special issue, several papers are devoted to this topic. All these papers try to use some kind of mathematical methods to solve the service-oriented problem in cloud computing.

As the major focus of IoT, equipping all objects in the world with minuscule identifying devices or machine-readable identifiers could transform our daily life [6, 7]. IoT technologies can be used in a wide range of applications. A number of papers in this special issue have shown their research efforts on using mathematic methods to build IoT applications in smart grid, industry monitoring, city emergency, and so forth.

Intelligent or optimization algorithms are widely used in both cloud computing and IoT. A dozen of papers in this issue just talk about how to use existing intelligent algorithms, such as classification, self-learning, evidence theory, predicting, evolutionary algorithms, and fuzzy logic, to solve the problems in cloud computing and IoT.

Security is another basic issue for both cloud computing and IoT. Either cloud computing or IoT has a feature of openness in applications. Therefore, how to preserve information security is very important to cloud computing and IoT. The research of information security or network security has an inborn relationship with mathematics. Some papers in this issue also focus on this problem.

Social networking service (SNS) or social media [8], as high-level applications based on cloud computing, have become more and more important in our daily life. SNS also has a close connection to IoT, as many SNS applications are built upon mobile networks. Therefore, how to analyze those large-scale SNS is an interesting topic for both cloud computing and IoT, which relies on quantitive or numerical approaches.

We hope that readers will find in this special issue not only the new ideas, cutting-edge information, new technologies, and applications of cloud computing and IoT, but also a special emphasis on how to solve various engineering problems by using applied mathematics and algorithms.

*Yuxin Mao*
*Vijay Bhuse*
*Zhongmei Zhou*
*Pit Pichappan*
*Mahmoud Abdel-Aty*
*Yoshinori Hayafuji*

## References

[1] P. Mell and T. Grance, "The NIST definition of cloud computing," 2009, http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf.

[2] M. Armbrust, A. Fox, R. Griffith et al., "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[3] A. Kevin, "That "Internet of Things" Thing, in the real world things matter more than ideas," *RFID Journal*, June 2009.

[4] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: a survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[5] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context aware computing for the internet of things: a survey," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 414–454, 2014.

[6] P. Magrassi, A. Panarella, N. Deighton, and G. Johnson, "Computers to acquire control of the physical world," Gartner Research Report T-14-0301, 2001.

[7] C. Associati, "The evolution of Internet of Things," 2011, http://www.casaleggio.it/pubblicazioni/Focus_internet_of_things_v1.81%20-%20eng.pdf.

[8] D. M. Boyd and N. B. Ellison, "Social network sites: definition, history, and scholarship," *Journal of Computer-Mediated Communication*, vol. 13, no. 1, pp. 210–230, 2007.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 604294, 10 pages
http://dx.doi.org/10.1155/2014/604294

*Research Article*

# Predicting the Times of Retweeting in Microblogs

**Li Kuang,[1] Xiang Tang,[2] and Kehua Guo[3]**

[1] *School of Software, Central South University, Changsha 410075, China*
[2] *Hangzhou Institute of Services Engineering, Hangzhou Normal University, Hangzhou 310012, China*
[3] *School of Information Science and Engineering, Central South University, Changsha 410075, China*

Correspondence should be addressed to Kehua Guo; guokehua@csu.edu.cn

Received 8 August 2013; Accepted 20 August 2013; Published 11 February 2014

Academic Editor: Zhongmei Zhou



Recently, microblog services accelerate the information propagation among peoples, leaving the traditional media like newspaper, TV, forum, blogs, and web portals far behind. Various messages are spread quickly and widely by retweeting in microblogs. In this paper, we take Sina microblog as an example, aiming to predict the possible number of retweets of an original tweet in one month according to the time series distribution of its top $n$ retweets. In order to address the problem, we propose the concept of a tweet's lifecycle, which is mainly decided by three factors, namely, the response time, the importance of content, and the interval time distribution, and then the given time series distribution curve of its top $n$ retweets is fitted by a two-phase function, so as to predict the number of its retweets in one month. The phases in the function are divided by the lifecycle of the original tweet and different functions are used in the two phases. Experiment results show that our solution can address the problem of predicting the times of retweeting in microblogs with a satisfying precision.

## 1. Introduction

Microblog is a social network based platform where information can be shared, propagated, and obtained. Users can publish their tweets through SMS, instant messenger, email, web sites, or third-party applications by inputting at most 140 words [1]. Microblog bloomed rapidly due to its numerous advantages such as real-time and high interaction. The number of Sina microblog users in China has reached up to 250 million during 2 years [2], and it has become a very important Internet application for nearly half of Chinese netizens.

Retweeting is a very important user behavior in microblogs. Users can forward the tweets which they are interested in, so that the followers of the users can see the tweets as well. The tweet publishing pattern and propagation form, as well as its concise presentation with multimedia added such as music, video, and pictures, make the information spreading faster in microblog than that in traditional media, with the content and form being more diverse. Therefore, how to predict the times of retweeting in microblogs by analyzing the features of tweets propagation becomes a hot research topic.

The result of the research can be applied in many areas: a tweet that is retweeted largely represents a hot topic, so the prediction on the times of retweeting can help find hot topics in microblog. Second, a hot tweet can represent the focus that most people are concerned about so we can monitor public opinions in a better fashion by predicting the times of retweeting. Moreover, microblog reacts more rapidly compared to traditional media, especially on social emergency, so traditional media like newspaper can draft news based on the latest hot tweets in microblog.

The 13th International Conference on Web Information System Engineering (WISE 2012) [3] organized a challenge on Sina microblog. The organizers collected a number of retweets related to 33 original tweets from Sina microblog. There are about 100 retweeting records corresponding to each original tweet. One of the proposed challenges is to predict the times of retweeting of the 33 original tweets in one month. Motivated by the challenge proposed in WISE 2012, we addressed the significant problem by three steps:

first, the primitive data are divided into 33 groups, where the data in one group correspond to the retweets of an original tweet. For each group, the primitive data are parsed by extracting the values of property tags, so that the time series distribution of top 100 retweets for each original tweet can be derived. Second, calculate the lifecycle of each original tweet according to its content and the characteristic of the time series distribution of top 100 retweets including response time and interval. Third, in order to predict the times of retweeting of the 33 original tweets in one month, the derived time series distribution curves of top 100 retweets are fitted by a two-phase function, where the first phase is the calculated lifecycle of the original tweet and the second phase is the remainder time in one month. The value in the 1st phase is derived by fitting the curve by a lineal function, while the value in the 2nd phase is by a logarithm function. The final predicted value of retweeting times is the sum of the values of two phases. The experiments show that the proposed solution in this paper can greatly address the problem of predicting the times of retweeting in microblogs, and the average error is controlled within 20%.

The paper is organized as follows. Related work is introduced in Section 2. The form and volume of collected microblog data are introduced in Section 3. The detailed solution to predicting the times of retweeting is illustrated in Section 4. The experiment results are presented in Section 5. And finally the conclusions and future work are given.

## 2. Related Work

The blossom of microblog aroused wide attention of many researchers. Presently, they begin to conduct research on the problems related to microblogs, including analyzing the contents of microblogs, mining the association relation between microblogs and real society [4–11], and predicting whether a tweet will be retweeted as well as the characteristic of retweeting behavior [12–21].

In the related work on the analysis of microblog contents, researchers found that microblog plays an important role in many areas, for example, political elections, earthquake disaster, marketing management, and various kinds of information spreading [4–11]. Tumasjan et al. [6] find that the political emotion of tweet users has close relation with election and tweets can reflect voters' inclination in real society by using LIWC text analysis software. Bollen et al. [7] find that society, culture, politics, and economy have a great influence on public sentiment through extended emotional analysis. Sakaki et al. [8] successfully find out the earthquake epicenter from Twitter messages through time probability model, and Qu et al. [9] pointed out that microblogs play an important and positive role in disaster by comparing the content of microblogs before and after Yushu earthquake in 2010. Achananuparp et al. [10] proposed a model for describing users' originating and promoting behaviors so as to detect interesting events from sudden changes in aggregated information propagation behavior of Twitter users.

In the related work in retweeting tweets, many researchers study and analyze what contents and features of a tweet make it be retweeted more easily. For example, Chen and Zhang [12] predict whether a tweet will be retweeted based on its emotional or content keywords, user tags, and historical retweeting frequency. Xiong et al. [13] studied information diffusion on microblogs based on retweeting mechanism and proposed a diffusion model (SCIR) which contains four states, two of which are absorbing. Zhang et al. [14] predict whether a tweet will be retweeted by ranking tweets based on weighted feature model. Hong et al. [15] discuss why and how people retweet messages, as well as what messages will be retweeted by making use of TF-IDF points. Zaman et al. [16] predict the information spreading in Twitter through collaborative filtering algorithm. Petrovic et al. [1] decide whether a tweet will be retweeted by manual experiments and then predict it by improved passive progressing algorithm. However, few works on predicting the times that a message is retweeted are published.

Zhang et al. [22] propose to compute the probability that a user retweets a tweet by considering several features first and then build a retweet model with the probability to predict the number of possible views of a tweet. Unankard et al. [23] compare four different methods, of which the first one is discovering a regression function based on the popularity of messages and network connectivity, the second one is learning a classification model based on users' preferences in different fields of topics, the third one is simulating retweeting paths starting from a root message by employing Monte Carlo method, and the fourth is building a recommendation model based on collaborative filtering. Luo et al. [24] propose to identify most similar message from training data based on the similarity between their time series values in the same length period and then fit the ARMA models over the whole time series of the identified message, and finally the fitted model is applied to the test tweet to predict future values. Compared with their work, in this paper, we propose a new perspective to differentiate the time period when a tweet may be largely retweeted and that when the possibility of retweeting becomes small and propose a new concept, a tweet's lifecycle, which is determined by analyzing the content of the tweet as well as the time series distribution of its top $n$ retweets. Based on the calculated lifecycle, different functions are fitted within and out of its lifecycle, so as to predict the number of retweets of a tweet in one month.

## 3. Dataset

In this paper, we take the Sina microblog data as an example to study the prediction on the times of retweeting. This section will introduce the form and volume of the collected raw data.

*3.1. Data Form.* The basic form of each datum in the collected dataset is as follows:

Tweet:time:A|#|mid:B|#|uid:C\tD\tE...|#|isContain Link:F|#|eventList:G|#|rtTime:H|#|rtMid:I|#|rtUid: J|#|rtIsContainLink:K|#|rtEventList:L.

In which the detailed meaning of each property tag is shown as Table 1.

Table 1: Data tags and their meanings.

| | |
|---|---|
| time | The time when a re-tweeting message is issued, whose form is yyyy-mm-ddhh:mm:ss |
| mid | The unique identification ID of the re-tweeting message |
| uid | The user ID who publishes the re-tweeting message |
| isContainLink | Whether the re-tweeting message contains a link, whose value is of kind Boolean (true or false) |
| eventList | The event tags of the re-tweeting messages, that is, its keywords |
| rtTime | The time when the re-tweeted original tweet is published, whose form is yyyy-mm-ddhh:mm:ss |
| rtMid | The message ID of the original tweet |
| rtUid | The user ID who publishes the original tweet |
| rtIsContainLink | Whether the original message contains a link, whose value is of kind Boolean (true or false) |
| rtEventList | The event tags of the original messages |

In order to illustrate the detailed meaning of every property more clearly, we take the following datum as an example:

time:2011-06-0511:26:56|#|mid:270926510254626223
8|#|uid:67010010610100010184292270218 38|#|is
ContainLink:false|#|rtTime:2011-06-05 08:19:59|#|rt
Mid:2709258383303085289|#|rtUid:9256021720209
2828482|#|rtIsContainLink:false|#|rtEventList:Li Na win French Open in tennis$Francesca Schiavone.

The datum shows the following: the original tweet ID (rtMid) is 2709258383303085289, it was created and published by a user with ID 92560217202092828482 (rtUid) at 2011-06-05 08:19:59 (rtTime), it does not contain a link (rtIsContainLink: false), and it is about Li Na winning French Open in tennis with event tags "rtEventList:Li Na win French Open in tennis$Francesca Schiavone." The original tweet is retweeted by a user with uid 67010010610100010184292270218 38 at 2011-06-05 11:26:56 (Time), its message ID (mid) is 2709265102546262238, and it does not contain a link (isContainLink:false).

Each primitive datum is constructed by such property-value pairs. We can find the retweeting time, retweeting message ID, the original tweet ID, event tags, and so forth from each datum, so as to understand and use each datum.

*3.2. Data Volume.* We eliminate repeated messages and finally got 3292 valid messages by preprocessing data based on integrity constraints. The 33 original tweets are annotated with event tags, and the 33 groups of data are mainly involved in 6 events, including the death of Steve Jobs, the earthquake in Japan, Li Na winning French Open tennis contest, Yao Jiaxin's murder case, bombing in Fuzhou, and the publishing of Xiaomi phones. Each of the 33 groups contains about 100 retweeting messages. The original tweet ID and corresponding number of collected retweeting messages for each group are shown in Table 4.

## 4. Predicting the Times of Retweeting

Given the time series distribution of top $n$ retweets of an original tweet, we aim to predict the number of retweets in the future one month. In order to get a more accurate predicted value, we propose to fit the given time series distribution curve by a two-phase function, whose phases are divided according to the lifecycle of the original tweet.

*4.1. Lifecycle of a Tweet.* Every creature in the earth has its own lifecycle. We think that every tweet has its lifecycle like the creatures on the earth as well. We find that the lifecycle of a tweet plays an important role in predicting the times of retweeting. If the contents of two tweets are similar, the retweeting numbers per day of the two are nearly the same, and meanwhile their publishing time points are close, the tweet with a longer lifecycle will have a larger number of retweets. Hence, in order to predict the retweeting times more accurately, we propose the concept of the lifecycle of a tweet, that is, the time duration when a tweet can be retweeted in a large number.

We find that the lifecycle of a tweet is related to the response time of the first retweet, the importance of the content, and the interval distribution of retweets, and we will illustrate the three factors in the following part.

*4.1.1. The Response Time of the First Retweet.* The response time of the first retweet means the time difference between the time of the first retweet and that of the origin tweet.

Generally speaking, the faster the first retweet is posted, the more attention is paid to the original one. And the more popular the original tweet is, the more likely it will be retweeted. Thus, correspondingly, an original tweet which is retweeted in a short time may get more attention and thus have a longer lifecycle.

According to the 33 groups of retweeting records, we design a formula to calculate the score with respect to response time. We divide them into four levels according to different intervals of response time, and each level corresponds to different functions on the response time. In general, the shorter time the first retweet is posted, the higher score will the original one get. The response time in the high speed group is less than 10 seconds, and the corresponding score in this group is assigned a full score of 10 points. The response time in the 2nd group is between 10 and 100 seconds, and the range of corresponding score in this group is [6, 10] points, and the score declines with a $(\lg x)^{-1}$ speed. The response time in the 3rd group is between 100 and 10000 seconds, and the range of corresponding score in this group is [0.6, 6] points; the score declines with $x^{-1/2}$ speed. The slow ones are over 10000 seconds, some are even more than 70000 seconds, and the range of corresponding score in this group is (0, 0.6]

Table 2: The importance of content.

| Content | Rank | $S_{\text{importance of content}}$ |
|---|---|---|
| High | T3 | 7–9 |
| Middle | T2 | 4–6 |
| Low | T1 | 1–3 |

Table 3: The interval time distribution.

| Interval time distribution | Rank | $S_{\text{interval time distribution}}$ |
|---|---|---|
| Separate and uneven | T3 | 3–5 |
| Even | T2 | 1–3 |
| Grow up highly | T1 | 0.1-0.2 |

points; the score declines slower than the 3rd group with $x^{-1/4}$ speed. The score on response time is proportional to the length of its lifecycle. The score with respect to response time is shown as

$$S_{\text{response time}} = \begin{cases} 10, & 0 < x \ll 10 \\ 2 + 8 \cdot (\lg x)^{-1}, & 10 < x \leq 100 \\ 60 \cdot x^{-1/2}, & 100 < x \leq 10000 \\ 6 \cdot x^{-1/4}, & x \geq 10000. \end{cases} \quad (1)$$

*4.1.2. The Importance of the Content.* The vast amount of retweeting happens only when the content is attractive, which is named as the importance of content. People tend to pay more attention to those tweets with attractive contents, that is, with high grade of importance of content.

The contents of tweets involve all aspects of our lives. According to Sina microblog, tweets can be classified to the categories such as lifestyle, love, entertainment, film, television, sports, finance, science, art, fashion, culture, and media. A tweet will be retweeted by a large number of times only when there is something attractive enough in its content, such as being about a pop star's affair or some big emergency. Take some pieces of news as examples.

(1) Before the death of American singer Michael Jackson was published, there were numerous fans coming into the hospital of the University of California in Los Angeles, where Michael Jackson had been, since they got the news from Facebook and Twitter. Moreover, only one hour later after the announcement of death, there were more than 65000 reply messages and retweets in Twitter; over 5000 of them came out within one minute.

(2) In February 2010, a 93-year-old Mrs. Xiao, who was from Chengdu, needed RH-AB blood because of the fracture. Lacking blood, she was in danger at that time. In that case, her daughter came to send a tweet to ask for help. Only within 12 hours, there were more than 3000 people that helped to retweet it. Fortunately, 3 friends from the Internet donated their blood and she was saved.

To conclude the cases above, the tweet about the death of Michael Jackson received more than 65000 comments and retweets within one hour, and the tweet about seeking RH-AB blood received more than 3000 people's attention within half a day; therefore, we guess that the more attractive the content is, the more chances it would be retweeted.

But what kind of content would be attractive? We believe that if the content is related to the hot issue recently, such as Olympic Games, disaster, or a pop star's affair and big social case, it would be attractive. And moreover, if the time of the tweet issued is close to the time of the occurrence of the event, the tweet would attract much attention and the level of importance of content is high. In comparison, if the tweet is posted in a relatively long time later, or the content is attractive only to some professional people in some specific field, the level of importance of content is in the middle. Finally, if there are few people concentrating on it or the tweet is posted very long time after the event happens, the level of importance of content is low. The rank and corresponding score on the importance of content with respect to different kinds of contents are shown in Table 2. The higher the importance of content is, the more scores the tweet will get on the $S_{\text{importance of content}}$.

For instance, the case of Michel Jackson is about a pop star, and the tweet is issued on time, so that the content of tweet is very attractive, the rank is identified as T3, and the score on the importance of content $S_{\text{importance of content}}$ would be 9.

*4.1.3. The Interval Time Distribution of Retweets.* According to the observation of data, if the number of retweets grows up very fast, for example, the tweet is retweeted for thousands of times in a short time, the retweeting will be in saturation soon; therefore, the lifecycle of the original tweet is relatively short; if the interval time distribution curve is even, that is, the number of retweeting grows up in a peace way, the life cycle of the original tweet would be relatively long; if the distribution curve of retweets is scatter and discrete, the tweet needs more time to get saturation and the lifecycle would be very long. The rank and corresponding score on the interval time distribution with respect to different type of curve are shown in Table 3.

For detailed values, we may make judgments based on the following standards. Divide the interval time distribution of all retweets according to the time equally. (1) If the number of retweets is growing fast, appearing as a linear with high slope (over 60 degrees), or an exponential curve, as Figure 1(a) shows, the curve is of the type dense rise. In general, the score on the interval distribution for this type is [0.1, 0.2]. (2) If the growth of retweets is steady as Figure 1(b) shows, the curve is of the type general steady and the score is [1, 3]. (3) If the growth of the retweets is small and flat, as Figure 1(c) shows, the curve is of the type scatter, and the score is [3, 5]. In addition, if the number of retweets increases sharply at early stage but becomes more and more slow afterwards, which means the trend is subsequent fatigue, the rank for this type of curve is deemed as T1, and the lifecycle would not be long, so the score is set around [0.2,1]. Despite all the criteria, the accurate values need further studies. According to the above

Table 4: 33 original messages and corresponding number of collected re-tweeting messages.

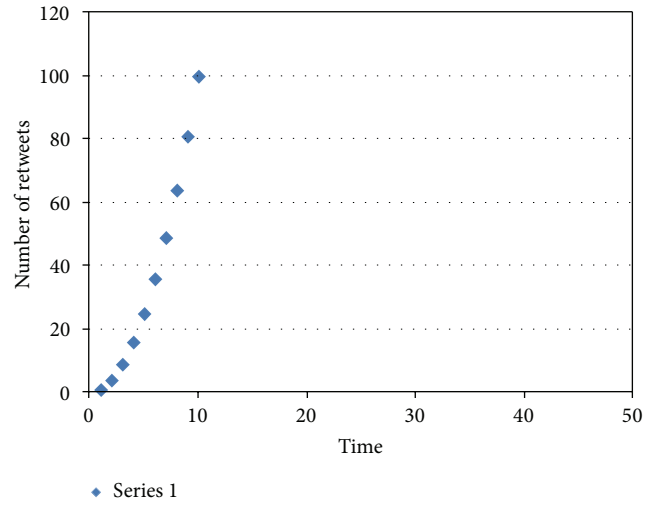
| rtMid | Count(*) |
|---|---|
| 2243526721410152330 | 100 |
| 2243578214587694822 | 100 |
| 2700059958269443492 | 99 |
| 2700117991448817596 | 96 |
| 2700176673306864228 | 100 |
| 2701374467440601577 | 97 |
| 2701431322360449433 | 100 |
| 2709258383303085289 | 100 |
| 2709864654666932643 | 100 |
| 2709870697693881414 | 100 |
| 2709871713230486085 | 100 |
| 2709893077170155796 | 100 |
| 51000180083282169 | 100 |
| 51000180083492814 | 100 |
| 51000180091104384 | 100 |
| 510001856830842390 | 100 |
| 510001856834367317 | 100 |
| 510001904903643837 | 100 |
| 510001908564754698 | 100 |
| 5100019107401880 | 100 |
| 55000180091534860 | 100 |
| 55000180527027036 | 100 |
| 550001906873838396 | 100 |
| 58000180083553705 | 100 |
| 8872263516485596 | 100 |
| 8872961090747701 | 100 |
| 8872983825828431 | 100 |
| 8872990233170214 | 100 |
| 8896800636296312 | 100 |
| 8896822338137478 | 100 |
| 8896858839607761 | 100 |
| 8896889634186199 | 100 |
| 8896952812610010 | 100 |

discussion, we design the rank and corresponding scores of interval time $S_{\text{interval time distribution}}$ as Table 3 shows.

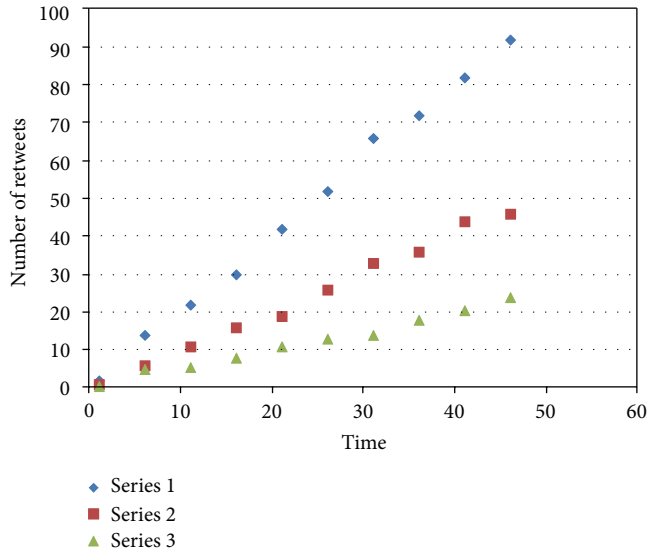
In summary, we make a calculation formula to compute the lifecycle of a tweet considering the above three factors:

$$\begin{aligned}\text{Lifecycle} &= S_{\text{interval time distribution}} \\ &\quad * \left(0.6 * S_{\text{importance of content}} + 0.4 * S_{\text{response time}}\right).\end{aligned} \tag{2}$$

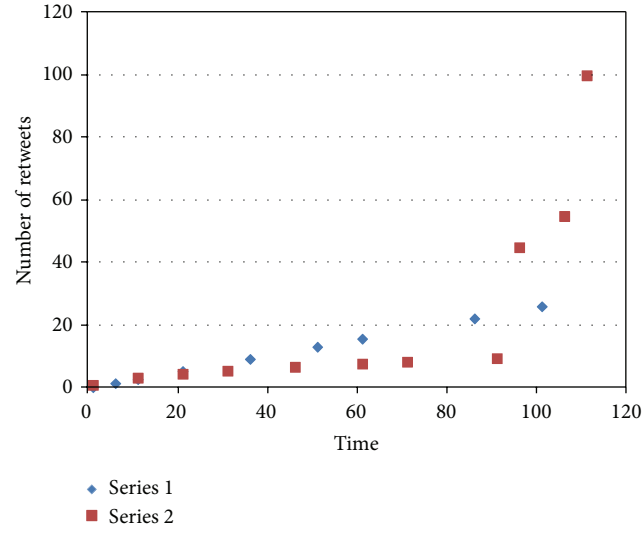
In the formula, the coefficients of the importance of content and response time are 0.6 and 0.4 separately, which are achieved by experiments on training data. The interval time distribution has a direct impact on the whole fitting of function curve, so the score on this part is worked as a product factor.




(a) Dense rise




(b) General steady




(c) Scatter

Figure 1: (a) Dense rise, (b) general steady, and (c) scatter.

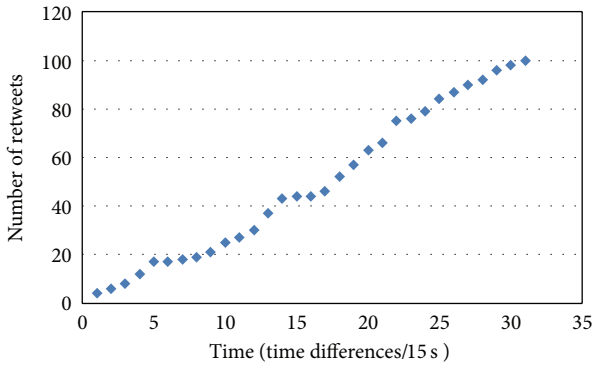



Figure 2: The time distribution scatter diagram of top 100 retweets of an original tweet which is related to the Steven Jobs' death.

Take the retweeting of an original tweet related to Steven Jobs' death issued at 12:07:52 2011/10/6 as an example. First, the event of Jobs' death belonged to the category of a star's affair, so the rank of the importance of the content is T3. Steven Jobs is the ex-CEO and one of the founders of Apple, who has a significant impact on the public, so we set $S_{\text{importance of content}}$ as 9. Second, the response time of the first retweet is 22 seconds, and according to formula 1 we have $S_{\text{response time}}$ as 8. Last, the number of retweets is increasing steady as Figure 2 shows, at the pace of 10 more retweets per minute, and the retweeting saturates within 460 seconds. The interval time distribution is like Figure 1(b), which belongs to general steady type, so $S_{\text{interval time distribution}}$ is set to 1. Therefore, the lifecycle of the original tweet is $1 * (9 * 0.6 + 8 * 0.4) = 8.6$ days.

*4.2. Two-Phase Function Curve Fitting.* The given time series distribution curve of top 100 retweets of an original tweet is then fitted by a two-phase function whose phases are divided according to the lifecycle of the original tweet. Main steps are illustrated as follows.

(1) We make use of Matlab, a mathematical analysis tool, for the purpose of function curve fitting. We need first to make a connection between *mysql* andMatlab and then execute sql statements through *exec* function, so as to import data from *mysql* to Matlab.

(2) Take preliminary analysis and draw scatter diagram based on the imported data. In the diagram, the $x$-axis data item "time" is not accurate time points but calculated by the time difference. In order to make the result more intuitive, we make the points in the scatter diagram more concentrated by dividing time slots. Figure 2 shows the time distribution scatter diagram of top 100 retweets of an original tweet which is related to Steven Jobs' death mentioned in Section 3.1.

In the following part, we will calculate the prediction value by fitting the curve with a two-phase function. In the first phase, that is, within the calculated lifecycle of the original tweet, a linear function is used to fit the curve. Most of the retweets occur within the lifecycle of the tweet, and the remainder appears as slow growing, so a logarithmic function like $a * \lg(x - b) + c$ is used to fit the curve in the 2nd phase. The detailed processes in the 3rd and 4th steps are shown as follows.

(3) In order to minimize error, we select a linear function which has the highest matching degree with the scatter points to fit the curve in the 1st phase. The line passes through as much points as possible. For every two points $(x_1, y_1)$ and $(x_2, y_2)$, a liner function $[(y_2 - y_1)/(x_2 - x_1)](x - x_1) + y_1$ is used to link them, and the whole curve is fitted from the relation among points. The detailed slope and intercept are decided based on the model of double moving average [25] in *Matlab*. It can avoid the lag deviation of single moving average method. The double moving average method adjusts the single one by adding a second moving average and then builds a linear model based on both average values.

The average of first moving is

$$M_t^{(1)} = \frac{1}{N}\left(y_t + y_{t-1} + \cdots + y_{t-N+1}\right). \tag{3}$$

Double moving average is making another moving average based on the first moving average, and the corresponding formula is

$$M_t^{(2)} = \frac{1}{N}\left(M_t^{(1)} + M_{t-1}^{(1)} + \cdots + M_{t-N+1}^{(1)}\right). \tag{4}$$

Since we have analyzed the growth of retweets in the 1st phase which appears as a liner function, we suppose the prediction model in the 1st phase is

$$y_{t+m} = a_t x + b_t, \quad m = 1, 2 \ldots, \tag{5}$$

in which $t$ is the current time and $m$ is the time slots from $t$ to the lifecycle of the tweet; $a_t$ is the slope and $b_t$ is the intercept, and the two are called smooth coefficients.

According to model (5), we can have

$$\begin{aligned} a_t &= y_t, \\ y_{t-1} &= y_t - b_t, \\ y_{t-2} &= y_t - 2b_t, \\ &\vdots \\ y_{t-N+1} &= y_t - (N-1)\, b_t. \end{aligned} \tag{6}$$

So we have

$$\begin{aligned} M_t^{(1)} &= \frac{1}{N}\left(y_t + y_{t-1} + \cdots + y_{t-N+1}\right) \\ &= -\frac{y_t + \cdots + \left[y_t - (\mathrm{N}-1)\, b_t\right]}{N} \\ &= y_t - \frac{N-1}{2} b_t. \end{aligned} \tag{7}$$

Therefore,

$$y_t - M_t^{(1)} = \frac{N-1}{2} b_t. \tag{8}$$

According to model (5) and to make similar inference as (8), we can have

$$y_{t-1} - M_{t-1}^{(1)} = \frac{N-1}{2} b_t. \tag{9}$$

Therefore,

$$\begin{aligned} y_t - y_{t-1} &= M_t^{(1)} - M_{t-1}^{(1)} = b_t, \\ M_t^{(1)} - M_t^{(2)} &= \frac{N-1}{2} b_t. \end{aligned} \tag{10}$$

Then the smooth coefficients can be calculated by

$$\begin{aligned} a_t &= 2M_t^{(1)} - M_t^{(2)}, \\ b_t &= \frac{2}{N-1}\left(M_t^{(1)} - M_t^{(2)}\right). \end{aligned} \tag{11}$$

According to the fitting curve, the function value when the $x$-axis value reaches the lifecycle of the original tweet is the predicted number of retweets in the 1st phase. An example scatter diagram and its corresponding fitting curve in the 1st phase are shown in Figure 3.

(4) For the remaining part that is beyond the lifecycle while being within one month, a logarithm function is used to fit the curve. The coefficients in the logarithm function $a * \lg(x - b) + c$ can be achieved by fitting the scatter points, and we can get the predicted value in the 2nd phase by passing the value of rest time into the function.

Take retweeting of the original tweet about Steven Jobs' death issued at 12:07:52 2011/10/6 as an example. Its lifecycle is 8.6 days as calculated in Section 4.1. In the 1st phase, the fitted linear function is $[(y_2 - y_1)/(x_2 - x_1)](x - x_1) + y_1 = (100 - 98/31 - 30)(x - 30) + 98 = 2(x - 30) + 98$, which can be derived from Matlab. We should translate the metric from day to seconds before the following calculation; that is, 8.6 days is equal to 743040 seconds (8.6 day $*$ 86400 sec/day = 743040 seconds). As we mentioned in step 2, the accurate seconds are divided into time slots by every 15 seconds. So here $x$ is equal to 743040/15 = 49536, and then we can get the predicted retweeting number in the 1st phase by passing the value of $x$ into the linear function; that is, $2 * (49536 - 30) + 98 = 99110$. In the 2nd phase, the logarithm function $(a * \lg(x - b) + c)$ is used to predict the retweeting number in the remaining 21.4 days. The coefficients can be achieved directly by *Matlab*; here $a$ is 2432, $b$ is −714, and $c$ is −1.599$e$ + 004, and the value of the 2nd phase by passing $x$ into the logarithm function is 117. Finally, the values of the two phases are summed up and the final result of the prediction on the retweeting number in 30 days is 99110 + 117 = 99227. Compared to actual retweeting number 110904, the deviation of our result is

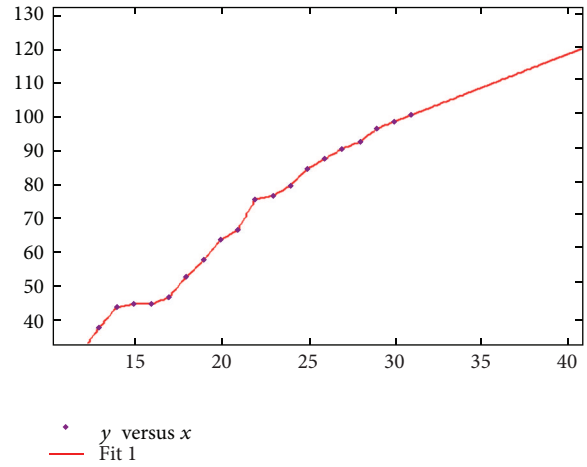



Figure 3: An example diagram and its corresponding fitting curve in the 1st phase.

$$\begin{aligned} &\frac{\left|\text{real number} - \text{prediction number}\right|}{\text{real number}} * 100\% \\ &= \frac{110904 - 99227}{110904} * 100\% = 10.52\%. \end{aligned} \tag{12}$$

## 5. Experiment Analysis

The result of prediction on the times of retweeting of the 33 original tweets is presented in Table 5.

In this table we can find out that the average error is less than 20%; we can conclude that our prediction is almost close to the real number of retweeting. Although different events have different lifecycle, we can get that the prediction values in the 1st phase play a dominate role, while those in the 2nd phase account for a smaller proportion.

## 6. Conclusions and Future Work

The prediction on the times of retweeting in microblog is to quantize the speed of information spread in microblogs and to find out the focus of public attention at all times, which is the key point of our research. In this paper, we analyze the behavior characteristics of retweeting in microblog and predict the times of retweeting of an original tweet in one month by a two-phase function curve fitting. The experiment shows that our approach can work out the prediction on retweeting times, and the average error is controlled within 20%.

Even so, our work still has some improvement to do, which is the direction in the future. First, the selected function may not be proper in some time, which leads to some exceptional results, so we may try some other function model. Second, we may do experiments on big data in order

Table 5: Result of prediction on re-tweeting of the 33 original tweets and comparison to real values.

| rtmid | rttime | Event | Lifecycle | 1st value | 2nd value | Prediction value | Actual value | Deviation |
|---|---|---|---|---|---|---|---|---|
| 8872263516485596 | 2011/10/6 19:17:17 | death of Steve Jobs$mourn Steve Jobs | 9.6 | 4969 | 23 | 4992 | 5587 | 10.65% |
| 8872961090747701 | 2011/10/6 12:07:52 | death of Steve Jobs$mourn Steve Jobs | 8.6 | 99110 | 117 | 99227 | 110904 | 10.52% |
| 8872983825828431 | 2011/10/6 10:01:33 | death of Steve Jobs$mourn Steve Jobs | 10.2 | 22786 | 84 | 22870 | 27514 | 16.88% |
| 8872990233170214 | 2011/10/6 10:14:24 | death of Steve Jobs$mourn Steve Jobs | 9.8 | 11560 | 37 | 11597 | 13768 | 15.77% |
| 8896800636296312 | 2011/8/17 9:33:03 | Xiaomi release$Xiaomi | 24.4 | 67256 | 42 | 67298 | 72021 | 6.56% |
| 8896822338137478 | 2011/8/17 12:28:16 | Xiaomi release$Xiaomi | 24.2 | 484 | 17 | 501 | 587 | 14.65% |
| 8896858839607761 | 2011/8/17 10:19:08 | Xiaomi release$Xiaomi | 25.2 | 40506 | 34 | 40540 | 47297 | 14.29% |
| 8896889634186199 | 2011/8/17 11:04:27 | Xiaomi release$Xiaomi | 24.8 | 39810 | 46 | 39856 | 38017 | 4.84% |
| 8896952812610010 | 2011/8/17 17:04:08 | Xiaomi release$Xiaomi | 26.5 | 5415 | 5 | 5420 | 6903 | 21.48% |
| 51000180083282169 | 2011/3/11 15:09:44 | House prices$ houseJapan Earthquake$Miyagi-ken | 0.6 | 7468 | 2 | 7470 | 5972 | 25.08% |
| 51000180083492814 | 2011/3/11 15:45:08 | House prices$ houseJapan Earthquake$Miyagi-ken | 0.64 | 4495 | 4 | 4499 | 5538 | 18.76% |
| 51000180091104384 | 2011/3/11 16:31:18 | Japan Earth-quake$magnitude 9.0 earthquake | 1.5 | 9709 | 26 | 9735 | 11699 | 16.79% |
| 55000180091534860 | 2011/3/11 16:31:55 | Japan Earth-quake$magnitude 9.0 earthquake | 0.97 | 14611 | 47 | 14658 | 16891 | 13.22% |
| 55000180527027036 | 2011/3/12 9:19:52 | Japan Earth-quake$magnitude 9.0 earthquake | 1.6 | 6888 | 25 | 6913 | 8022 | 13.82% |
| 58000180083553705 | 2011/3/11 15:08:16 | Japan Earth-quake$magnitude 9.0 earthquake | 0.8 | 25645 | 52 | 25697 | 30174 | 14.84% |
| 5100019107401880 | 2011/4/1 12:56:42 | Yao Jiaxin murder case$Zhang Miao | 2.4 | 10819 | 77 | 10896 | 12400 | 12.13% |
| 510001856830842390 | 2011/3/27 11:54:27 | Yao Jiaxin murder case$Yao Jiaxin | 8.3 | 4439 | 23 | 4462 | 4873 | 8.43% |
| 510001856834367317 | 2011/3/27 18:55:52 | Yao Jiaxin murder case$Yao Jiaxin | 11.8 | 756 | 6 | 762 | 776 | 1.80% |
| 510001904903643837 | 2011/4/19 10:19:43 | Yao Jiaxin murder case$Yao Jiaxin | 1.12 | 7244 | 108 | 7352 | 9779 | 24.82% |
| 510001908564754698 | 2011/4/13 14:36:44 | Yao Jiaxin murder case$Yao Jiaxin | 12 | 33846 | 32 | 33878 | 36298 | 6.67% |
| 550001906873838396 | 2011/4/17 10:40:19 | Yao Jiaxin murder case$Yao Jiaxin | 9.2 | 47524 | 58 | 47582 | 53385 | 10.87% |
| 2243526721410152330 | 2011/4/22 12:18:52 | Yao Jiaxin murder case$Yao Jiaxin | 1.4 | 24181 | 92 | 24273 | 27906 | 13.02% |

Table 5: Continued.

| rtmid | rttime | Event | Lifecycle | 1st value | 2nd value | Prediction value | Actual value | Deviation |
|---|---|---|---|---|---|---|---|---|
| 2243578214587694822 | 2011/4/22 12:41:36 | Yao Jiaxin murder case$Yao Jiaxin | 10.5 | 12138 | 41 | 12179 | 14462 | 15.79% |
| 2700059958269443492 | 2011/5/27 4:28:31 | Fuzhou bombings$Qian Mingqi$Fuzhou | 1.5 | 2451 | 22 | 2473 | 2813 | 12.09% |
| 2700117991448817596 | 2011/5/26 20:50:41 | Fuzhou bombings$Qian Mingqi$Fuzhou | 1.6 | 6919 | 14 | 6933 | 7979 | 13.11% |
| 2700176673306864228 | 2011/5/27 0:48:40 | Fuzhou bombings$Qian Mingqi$Fuzhou | 1.62 | 6994 | 17 | 7011 | 7876 | 10.98% |
| 2701374467440601577 | 2011/5/26 12:01:08 | Fuzhou bombings$Qian Mingqi$Fuzhou | 8.1 | 4806 | 15 | 4821 | 5465 | 11.78% |
| 2701431322360449433 | 2011/5/26 19:11:24 | Fuzhou bombings$Qian Mingqi$Fuzhou | 1.3 | 8956 | 34 | 8990 | 10772 | 16.54% |
| 2709258383303085289 | 2011/6/5 8:19:59 | Li Na win French Open in tennis$Francesca Schiavone | 10.2 | 1726 | 7 | 1733 | 1927 | 10.07% |
| 2709864654666932643 | 2011/6/4 23:00:46 | Li Na win French Open in tennis$Francesca Schiavone | 0.7 | 36267 | 67 | 36334 | 43146 | 15.79% |
| 2709870697693881414 | 2011/6/4 21:50:59 | Li Na win French Open in tennis$Francesca Schiavone | 0.9 | 116374 | 112 | 116486 | 136544 | 14.69% |
| 2709871713230486085 | 2011/6/4 21:46:34 | Li Na win French Open in tennis$Francesca Schiavone | 1.38 | 39670 | 72 | 39742 | 48925 | 18.77% |
| 2709893077170155796 | 2011/6/4 20:58:17 | Li Na win French Open in tennis$Francesca Schiavone | 1.6 | 3493 | 8 | 3501 | 3983 | 12.10% |

to optimize and adjust the curve fitting, so as to reduce the error.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## Acknowledgments

The work is supported in part by the following funds: the National Natural Science Foundation of China under the Grant no. 61202095 and 61173176 and the Scientific Research Project of Central South University under the Grant no. 7608010001.

## References

[1] S. Petrovic, M. Osborne, and V. Lavrenko, "RT to win! Predicting message propagation in twitter," in *Proceedings of 5th International AAAI Conference on Weblogs and Social Media*, pp. 586–589, 2011.

[2] China Internet Network Information Center (CNNIC), *The 29th Internet Development Statistics Report in China*, 2012.

[3] "WISE 2012 challenge," http://www.wise2012.cs.ucy.ac.cy/challenge.html.

[4] B. J. Jansen, M. Zhang, K. Sobel, and A. Chowdury, "Twitter power: tweets as electronic word of mouth," *Journal of the American Society for Information Science and Technology*, vol. 60, no. 11, pp. 2169–2188, 2009.

[5] R. Long, H. F. Wang, Y. Q. Chen, O. Jin, and Y. Yu, "Towards effective event detection, tracking and summarization on microblog data," in *Web-Age Information Management*, H.

Wang, S. Li, S. Oyama, X. Hu, and T. Qian, Eds., vol. 6897 of *Lecture Notes in Computer Science*, pp. 652–663, 2011.

[6] A. Tumasjan, T. O. Sprenger, P. G. Sandner, and I. M. Welpe, "Predicting elections with twitter: what 140 characters reveal about political sentiment," in *Proceedings of 4th International AAAI Conference on Weblogs and Social Media*, pp. 178–185, 2010.

[7] J. Bollen, H. Mao, and A. Pepe, "Determining the public mood state by analysis of microblogging posts," in *Proceedings of the 12th International Conference on the Synthesis and Simulation of Living Systems*, pp. 667–668, 2010.

[8] T. Sakaki, M. Okazaki, and Y. Matsuo, "Earthquake shakes twitter users: real-time event detection by social sensors," in *Proceedings of the 19th International World Wide Web Conference (WWW '10)*, pp. 851–860, April 2010.

[9] Y. Qu, C. Huang, P. Zhang, and J. Zhang, "Microblogging after a major disaster in China," in *Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW '11)*, pp. 25–34, March 2011.

[10] P. Achananuparp, E. P. Lim, J. Jiang, and T. A. Hoang, "Who is retweeting the tweeters? Modeling, originating, and promoting behaviors in the twitter network," *ACM Transactions on Management Information Systems*, vol. 3, no. 3, article 13, 2012.

[11] J. Tang, X. Wang, H. Gao, X. Hu, and H. Liu, "Enriching short text representation in microblog for clustering," *Frontiers of Computer Science in China*, vol. 6, no. 1, pp. 88–101, 2012.

[12] J. Chen and C. Zhang, "Research on prediction of comprehensive forwarding probability based on emotional word content, user tags, historical forward rate in MicroBlogging community," 2012, http://www.paper.edu.cn/releasepaper/content/201111-371.

[13] F. Xiong, Y. Liu, Z. J. Zhang, J. Zhu, and Y. Zhang, "An information diffusion model based on retweeting mechanism for online social media," *Physics Letters A*, vol. 376, no. 30-31, pp. 2103–2108, 2012.

[14] Y. Zhang, R. Lu, and Q. Yang, "Predicting retweeting in microblogs," *Journal of Chinese Information Processing*, vol. 26, no. 4, pp. 109–114, 2012.

[15] L. Hong, O. Dan, and B. D. Davison, "Predicting popular messages in twitter," in *Proceedings of the 20th International Conference Companion on World Wide Web (WWW '11)*, pp. 57–58, April 2011.

[16] T. R. Zaman, R. Herbrich, J. V. Gael, and D. Stern, "Predicting information spreading in twitter," in *Proceedings of the Workshop on Computational Social Science and the Wisdom of Crowds (NIPS '10)*, 2010.

[17] Y. Zhang, R. Lu, and Q. Yang, "Prediction of the micro-blog retweet behavior," in *Proceedings of the National Conference on Information Retrieval*, 2011.

[18] D. Boyd, S. Golder, and G. Lotan, "Tweet, tweet, retweet: conversational aspects of retweeting on twitter," in *Proceedings of the 43rd Annual Hawaii International Conference on System Sciences (HICSS-43 '10)*, January 2010.

[19] R. Lahan, *The Economics of Attention*, University of Chicago Press, 2006.

[20] B. Suh, L. Hong, P. Pirolli, and E. H. Chi, "Want to be retweeted? Large scale analytics on factors impacting retweet in twitter network," in *Proceedings of the 2nd IEEE International Conference on Social Computing (SocialCom '10)*, pp. 177–184, August 2010.

[21] J. Berger and K. L. Milkman, "Social transmission, emotion, and the virality of online content," Wharton Research Paper, 2010.

[22] H. B. Zhang, Q. Zhao, H. Y. Liu, J. He, X. Y. Du, and H. Chen, "Predicting retweet behavior in weibo social network," in *Web Information Systems Engineering—WISE 2012*, X. S. Wang, I. Cruz, A. Delis, and G. Huang, Eds., vol. 7651 of *Lecture Notes in Computer Science*, pp. 737–743, 2012.

[23] S. Unankard, L. Chen, P. Li et al., "On the prediction of retweeting activities in social networks—a report on WISE 2012 challenge," in *Web Information Systems Engineering—WISE 2012*, X. S. Wang, I. Cruz, A. Delis, and G. Huang, Eds., vol. 7651 of *Lecture Notes in Computer Science*, pp. 744–754, 2012.

[24] Z. L. Luo, Y. Wang, and X. T. Wu, "Predicting retweeting behavior based on autoregressive moving average model," in *Web Information Systems Engineering—WISE 2012*, X. S. Wang, I. Cruz, A. Delis, and G. Huang, Eds., vol. 7651 of *Lecture Notes in Computer Science*, pp. 777–782, 2012.

[25] C. T. Ragsdale, *Spreadsheet Modeling and Decision Analysis*, Cengage Learning, 6th edition, 2010.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 826068, 7 pages
http://dx.doi.org/10.1155/2014/826068

*Research Article*

# From Pixels to Region: A Salient Region Detection Algorithm for Location-Quantification Image

**Mengmeng Zhang, Zhi Liu, Huan Zhou, and Jian Wang**

*College of Information Engineering, North China University of Technology, No. 5 Jinyuanzhuang Road, Shijingshan District, Beijing 100144, China*

Correspondence should be addressed to Mengmeng Zhang; zhangmengmeng.1@hotmail.com

Received 5 September 2013; Accepted 23 September 2013; Published 29 January 2014

Academic Editor: Zhongmei Zhou



Image saliency detection has become increasingly important with the development of intelligent identification and machine vision technology. This process is essential for many image processing algorithms such as image retrieval, image segmentation, image recognition, and adaptive image compression. We propose a salient region detection algorithm for full-resolution images. This algorithm analyzes the randomness and correlation of image pixels and pixel-to-region saliency computation mechanism. The algorithm first obtains points with more saliency probability by using the improved smallest univalue segment assimilating nucleus operator. It then reconstructs the entire saliency region detection by taking these points as reference and combining them with image spatial color distribution, as well as regional and global contrasts. The results for subjective and objective image saliency detection show that the proposed algorithm exhibits outstanding performance in terms of technology indices such as precision and recall rates.

## 1. Introduction

Image saliency detection is the key to the extraction of image information. Extracting image saliency regions is required in most image processing methods that are based on image content because important image components provide the most comprehensive information on an entire image. Therefore, precisely, extracting the salient regions of images effectively facilitates many image applications such as image retrieval [1–3], adaptive image compression [4, 5], object recognition [6], and content-aware image resizing [7].

Humans can easily focus on the salient parts of images according to experience and judgment, but machines are unable to precisely replicate such an ability. Many scholars have studied this matter on the basis of biology, physiology, and neurobiology. In these studies, some features that the salient regions should have, including uniqueness, randomness, and surprising characteristics, are acquired.

In this paper, we present salient region detection as a random distribution problem of image pixels' binary labels. To concretely analyze an image, we represent all the image pixel sets as a sequence of only 1, 0; that is, each pixel of the image has only two kinds of attribution, which either belong to the salient region or not. The two important properties in this distribution are randomness and correlation. Randomness refers to already known pixel sets belonging to the salient region but with unknown number, location, and pixel combination, that is, the size, position, and shape of the salient regions of an image. Correlation refers to random pixel distribution, but this distribution is not an irregularly complete "free random distribution." These pixels always influence one another through some associated features such as image contrast, multiscale features, and color distribution.

We establish a computation mechanism, called location-quantification. The algorithm begins with corner points that have abundant information, and then the corner points are obtained by improving the smallest univalue segment assimilating nucleus (SUSAN) operator. Then, the salient region is located by using the corner points combined with their global features. Finally, the salient region of the image is quantified by spatial weighted similarity. The image saliency detection results are evaluated by two technology indices: the precision

for single images and the success rate for large images. We test our algorithm on publicly available benchmark image data sets and compare it with existing representational algorithms. The results show that our method excellently performs in terms of the above-mentioned technology indices.

## 2. Related Work

Image visual attention detection has rapidly developed in the last 20 years, and various outstanding methods have been developed. Different approaches have varied points of emphases and perform well in certain respects.

The biological vision-inspired model proposed by Koch and Ullman [8] affects a large amount of image saliency detection algorithms that are based on basic image features. Itti et al. [9] define image saliency in which the intensity, color, and orientation of images are combined to compute an image saliency map. This method analyzes only the global features of an image to effectively locate salient regions. However, the identified region is not sufficiently effective for salient region quantization. Goferman et al. [10] synthetically consider the image's regional and global features. This method exhibits an improvement in image region details, unlike that proposed by Itti et al. [9], but this technique remains inadequate for homogeneously highlighting an entire salient region.

Some algorithms incorporate mathematical models on the basis of biological vision. Harel et al. [11] and Gopalakrishnan et al. [12] use medium range forecast to process basic image features. Duan et al. [13] employ principal component analysis to transform image color space and decrease dimension. Li et al. [14] propose a method based on sparse coding length to compute the salient region of an image. This method views image saliency as a direct reflection of coding length and presents a probable explanation for the influence of visual saliency. These methods can provide good results up to a certain extent and extend the scope of image types. Nevertheless, the introduction of mathematical models increases computational complexity.

Liu et al. [15] and Judd et al. [16] propose systematic methods of saliency detection for mass images. All their methods introduce the advantages of machine learning and collect the features of the salient region labeled by tested participants. This kind of machine learning method is robust for both image type and precision of saliency detection. The aforementioned techniques can precisely estimate images with complicated scenes. However, these approaches are unsuitable for real-time image saliency detection because the processes require complicated equipment and incur high computational costs. The increase in complexity is nonlinear with the performance improvement of saliency detection for common nature images.

Hou and Zhang [17] analyze image saliency in the frequency domain. The authors found that the average value of the log frequency spectrum of mass images is directly proportional to frequency. The salient parts of an image are obtained by subtracting the average log amplitude spectrum from the log amplitude spectrum of one image. Guo et al. [18] suggest that better results can be obtained by using the phase spectrum of the Fourier transform.

Our method is based on basic image features, which are analyzed by location-quantification. We first obtain the probable salient points and then quantify these points to reconstruct the entire image using the salient region. We comprehensively consider the regional and global image features; thus, our method sufficiently highlights the entire salient region. We do not compare and compute every pixel of an image, so that the algorithm results in less computational complexity.

## 3. Our Method

The essential problem in saliency region detection is the need to decide which pixels belong to the salient region and which do not belong in an image. Therefore, we define $A = \{a_{\mathrm{pix}(w,h)}\}$ as a given image. For a random pixel $\mathrm{pix}(w,h)$, $a_{\mathrm{pix}(w,h)} \in \{1,0\}$ represents whether this pixel is a salient point. $P(w,h)$ is used to indicate the saliency probability of one pixel coordinate $(w,h)$ and then $P(w,h) \in (0,1)$.

The algorithm is intended for common nature images, whose salient regions contain most of the information that the image expresses. The image background also contributes to showing the entire image content, with relationships such as beautiful flowers and green leaves. The energy of different pixels varies, indicating that $P(w,h)$ is also different. The algorithm therefore begins from some pixels with high $P(w,h)$ values.

*3.1. Computing Reference Points.* The information on a pixel in an image commonly depicts that the frequency of pixel color value is low and its energy is high. Various methods, such as the SUSAN corner [19], Harris corner [20], and scale-invariant feature transform points, have been used to detect high energy and abundant information points in an image. In the current work, we use the SUSAN corner [19] to detect the points with high $P(w,h)$ in an image to reduce the computational effort required in the early stage.

SUSAN corner detection [19] uses a circle template to move in an image. The intensity of every pixel in the template is compared with that of the pixel at the template nucleus. This comparison is expressed by

$$D(r,r_0) = \begin{cases} 1 & \text{if } |p(r) - p(r_0)| \le t \\ 0 & \text{if } |p(r) - p(r_0)| > t. \end{cases} \quad (1)$$

The SUSAN region of every pixel is defined as

$$n(r_0) = \sum_r D(r,r_0). \quad (2)$$

Then, we obtain the initial corner response of this pixel by

$$R(r_0) = \begin{cases} g - n(r_0) & \text{if } n(r_0) < g \\ 0 & \text{others,} \end{cases} \quad (3)$$

where $g$ is the geometric threshold and commonly assigned the value $(1/2)N$, which indicates half of the number of pixels in the template.

To guarantee that the SUSAN algorithm for gray images is suitable for color image corner detection, we improve

the algorithm as follows. Equation (1) only considers pixel intensity as the measurement standard, which is insufficient for more complicated color images. Thus, we change $p(r) - p(r_0)$ in (1) into

$$D'(r, r_0) = \left\| p(r) - p(r_0) \right\|_2, \quad \forall P(r) \in I, \tag{4}$$

indicating the norm of the color vector in the CIELab color space.

The value of $t$ in (1) is usually 25 according to experimental results, but this invariable value lacks flexibility. In our algorithm, we define self-adaption $t$ as

$$t = \left\| \frac{\sum_N (p_x - \overline{p})}{N} \right\|. \tag{5}$$

Thus, we use (6) to compare the template nucleus pixels with the other pixels in the template in the CIELab color space:

$$D(r, r_0) = \begin{cases} 1 & \text{if } \left\| P(r) - P(r_0) \right\| \le \left\| \dfrac{\sum_{n(r_0)_{\text{Max}}} (p_x - \overline{p})}{n(r_0)_{\text{Max}}} \right\|, \\ 0 & \text{if } \left\| P(r) - P(r_0) \right\| > \left\| \dfrac{\sum_{n(r_0)_{\text{Max}}} (p_x - \overline{p})}{n(r_0)_{\text{Max}}} \right\|, \end{cases} \tag{6}$$

where $\overline{p}$ is the average value of all the pixel vectors in the template.

The third column in Figure 1 shows the results of the improved SUSAN operator, in which (a) represents the source images, (b) represents images with the salient regions identified by participants, and (c) represents images with reference points computed by the improved SUSAN operator. We analyze these points in terms of two aspects. First, most of the obtained reference points are located in the labeled salient region, a finding that is consistent with our goal and shows that the obtained points' $P(w, h)$ values are large in all the pixels. Second, not all reference points are located in the labeled salient regions because of the correlation among the pixels. We consider the correlation of only one pixel with tens of other pixels around it in computing the reference points. All the pixels in an image are correlative; this relationship is called global correlation. We process the aforementioned reference points by their global correlation to obtain the saliency location map, as discussed in the next section.

*3.2. From Point to Region.* We have already obtained some reference points according to the principle of "larger probability," but some of these points are not located in the labeled salient region in as Figure 1(c). The absence of the points in the labeled region is attributed to the fact that we choose only the limited-length neighborhood features that take these points as the center in the computation process. Their global features are disregarded.

*3.2.1. Global Point Processing.* We use the global contrast method introduced by Cheng et al. [21] to compute the global features of the reference points because this method effectively separates a large-scale object from its surroundings. This technique is also preferred over local contrast-based methods that produce high saliency values at near-object edges. In the method of Cheng et al. [21], every pixel should be compared with others, but the analysis in the preceding section indicates that this comparison is not necessary for all pixels because the saliency probability of image pixels is different.

We choose the large probability points and compute their global contrast by

$$S_A = \sum_{w \times h} D\left(P_{x,y}, P_i\right), \tag{7}$$

where $P_{x,y}$ denotes the vector with coordinates $(x, y)$; $P_i$ is the vector of the random points in the image; and $D(P_{x,y}, P_i)$ represents the distance of the pixels in the CIELab color space, which is expressed by (4).

We define a threshold $T$ to sift all the points:

$$T = \overline{S_A} = \frac{\sum_N S_A}{N}, \tag{8}$$

where $T$ is the average saliency value of the reference points. All the reference points that show $S_A < T$ are removed; then, the isolated points are also removed. The result is shown in Figure 1(d). The fourth column of Figure 1 shows that most of the reference points are located in the labeled salient region after processing.

*3.2.2. Point Diffusion by Spatial Weighted Similarity.* As shown in Figure 1, the remaining points after global contrast are mostly located in the labeled salient region or boundary. The next step is obtaining the entire salient region according to the reference points. The color and texture of the pixels inside the salient region are similar but differ from those of the background. A random pixel search method centered on one reference point is proposed to complete salient region detection.

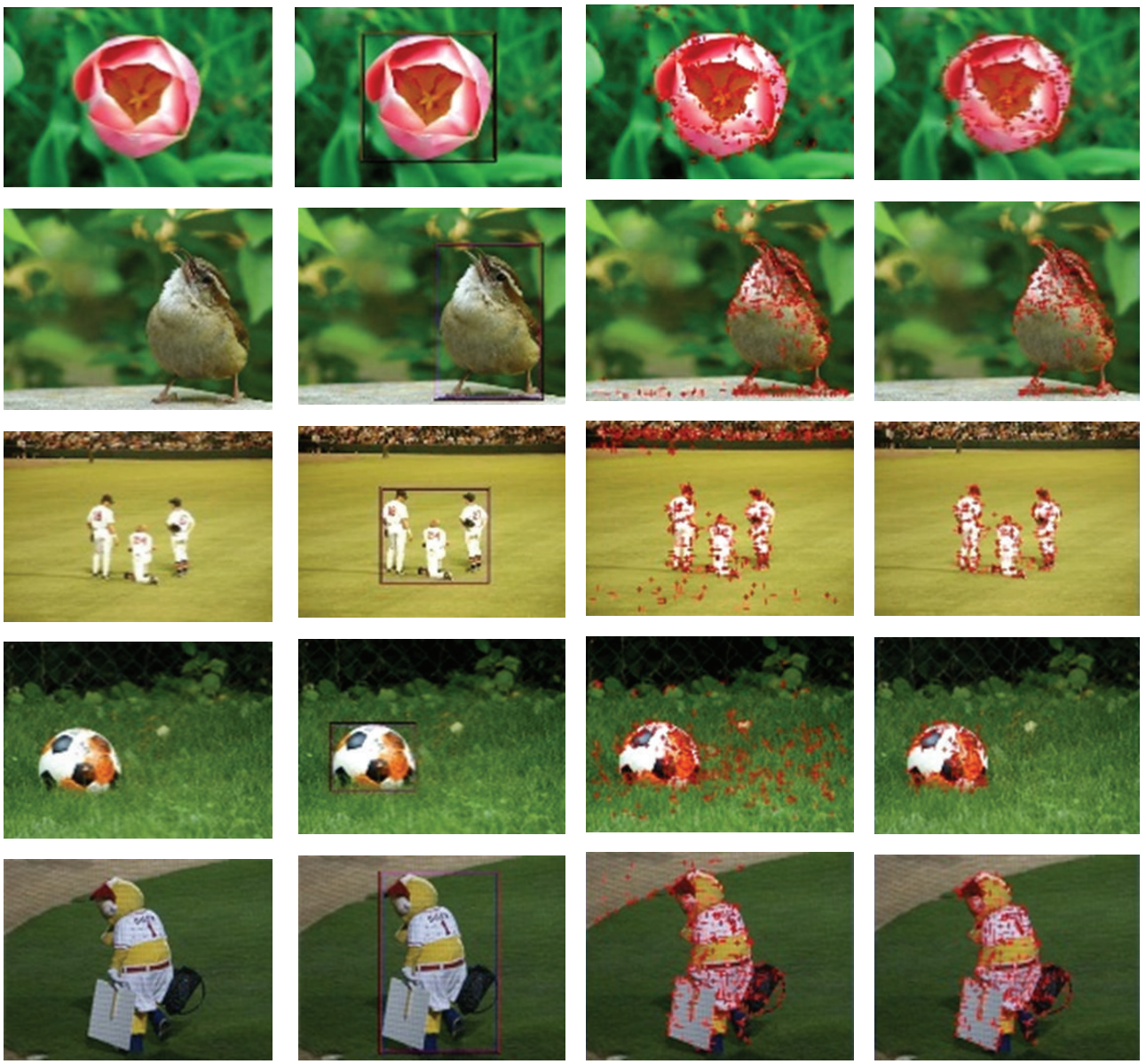
We suggest that the computation of every pixel's saliency value follows the four basic principles according to the relationship of the reference and other pixels.

(1) Both of the regional and global saliency probability of the reference points after processing are high; thus, the saliency probability of pixels with similar features should also be high.

(2) The saliency probability of the pixels is related to distance [10, 13]. That is, some pixels' features are similar to those of reference points, but their saliency probability decreases as a result of large distances.

(3) All saliency pixels are centered before the entire image is subjected to saliency detection [16]. That is, the position of the salient object is near the center of the image for a nature image.

(4) The amount of pixel information is a ratio that is inversely related to frequency, but their saliency probability is low, regardless of whether the frequency is a maximum or minimum value.



(a) (b) (c) (d)

Figure 1: Reference points in different images. From left to right: source images, images with labeled salient regions, images with preliminarily reference points, and images with reference points after global processing.

Every reference point is successively chosen as the center point from all $M$ reference points according to the aforementioned four principles. Every pixel in the circle of its circle neighborhood is searched until all the pixels are chosen. The saliency value is computed by

$$\begin{aligned}\text{Saliency}\,(c,r) = &\sum_{M,\forall p_c\in I,\forall p_{c+r}\in I} \omega_3\,(c+r)\cdot\omega_4\,(c+r)\\ &\cdot\omega_2\,(c,r)\cdot\exp\left(-\left\|p_c-p_{c+r}\right\|_2\right),\end{aligned} \tag{9}$$

where $c$ is the reference point coordinate and $r$ is the circle radius. Weight $\omega_2(c,r)$, defined according to principle 2, is

$$\omega_2\,(c,r) = \frac{1}{1+\lambda\cdot r}, \tag{10}$$

where the algorithm of Goferman et al. [10] is used as the reference for argument $\lambda$ and $\lambda = 3$.

Weight $\omega_3(c,r)$ indicates the offset from pixel to the image center. The expression in the study of Judd et al. [16] is used as the reference and is defined as follows:

$$\omega_3\,(c+r) = 1 - \frac{\text{DistToCenter}\,(p_{c+r})}{\max_c\left\{\text{DistTocenter}\,(p_c)\right\}}, \tag{11}$$

where $\text{DistToCenter}(p_{c+r})$ is the spatial distance from pixels to the image center and $\max_c\{\text{DistTocenter}(p_c)\}$ denotes the normalizing factor.

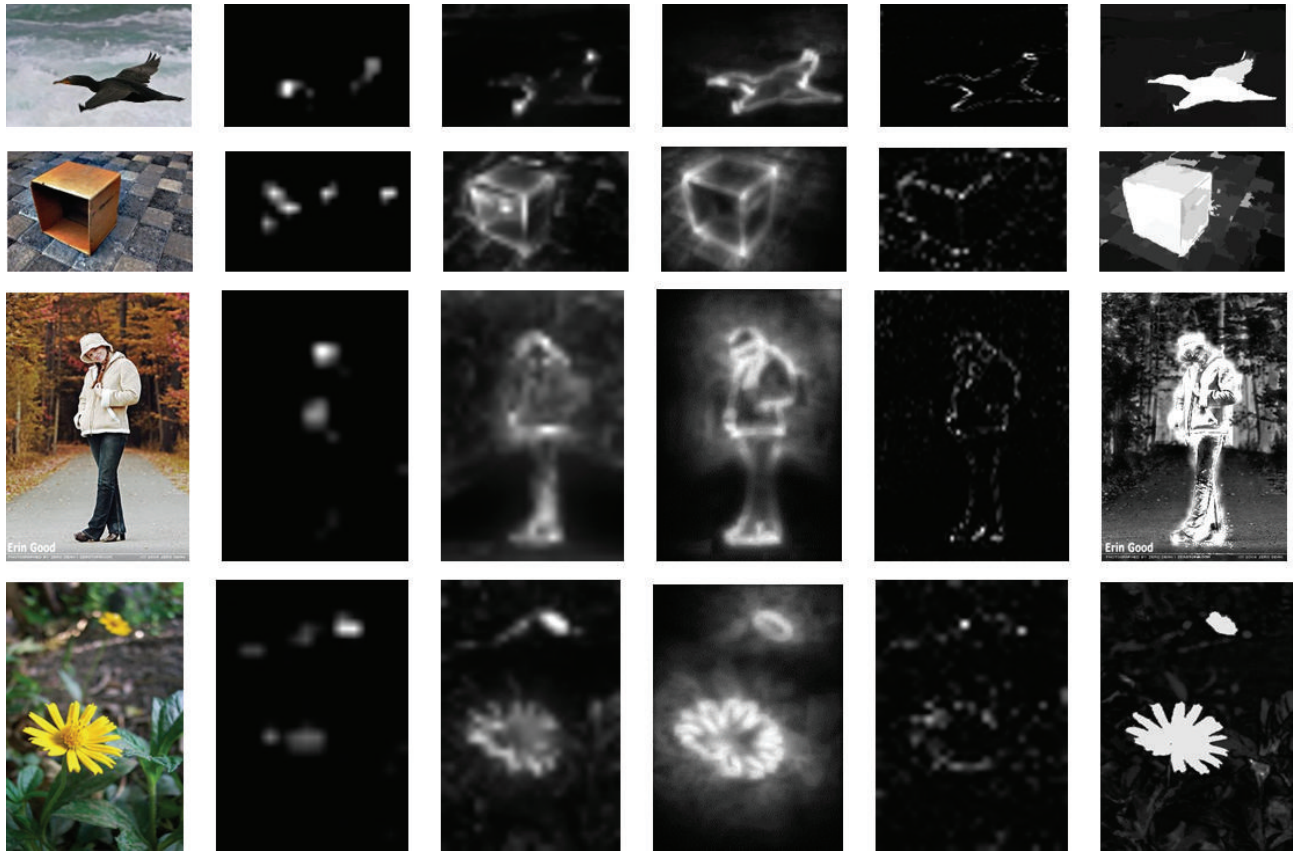



Figure 2: Comparison of our method with others. From left to right: source image, the methods of Itti, Harel, Goferman, Hou, and our method.

We use the Gauss function to show the influence of the frequency of one pixel on its saliency probability. We define $\omega_4(c, r)$ as

$$\omega_4(c+r) = A \exp\left[-\frac{\left(P_{c+r} - \overline{P}\right)^2}{\sigma_s^2}\right], \tag{12}$$

where $P_{c+r}$ is the frequency of pixel $p_{c+r}$; $\overline{P}$ is the average frequency of pixels in the image; and $\sigma_s^2$ controls the strength of pixel frequency weight. Larger values of $\sigma_s^2$ are more influential in saliency probability. In our experiment, $\sigma_s^2 = 0.4$.

To compute (9), we usually first choose the center point of all the reference points. If one point is similar to another that has been computed, we disregard it. Our experimental results are shown in Figure 2.

*3.2.3. Multiscale Enhancement.* Multiple scales of an image play an important role in enhancing the algorithm effect in the area of image recognition and object detection. We compute (9) at scale $\{s, (1/2)s, (1/4)s\}$. We merge the three saliency maps of different scales to obtain the final saliency map. Merging necessitates transforming the three saliency maps to the same scale because of their different sizes. The merging method based on the weighted saliency maps is used to generate the final saliency map. We use the reference points' number proportions at different scales as weight standards:

$$\begin{aligned}\text{SaliencyMap}|_{\text{final}} = {} & \frac{N_s}{N} \cdot \text{SaliencyMap}|_s \\ & + \frac{N_{s/2}}{N} \cdot \text{SaliencyMap}|_{(1/2)s} \\ & + \frac{N_{s/4}}{N} \cdot \text{SaliencyMap}|_{(1/4)s},\end{aligned} \tag{13}$$

where $N_s$, $N_{s/2}$, and $N_{s/4}$ are the reference points' numbers at different scales and $N$ is the sum of all the reference points at three scales.

## 4. Experimental Results and Evaluation

Various methods for evaluating the results of salient region detection are available, and each method emphasizes different elements. We generally classify these methods as subjective and objective evaluations. The first problem in evaluating an algorithm is the selection of an image data set. We test our algorithm in the data set provided by Cheng et al. [21] and Achanta et al. [22]. We then show our subjective and objective evaluation results.

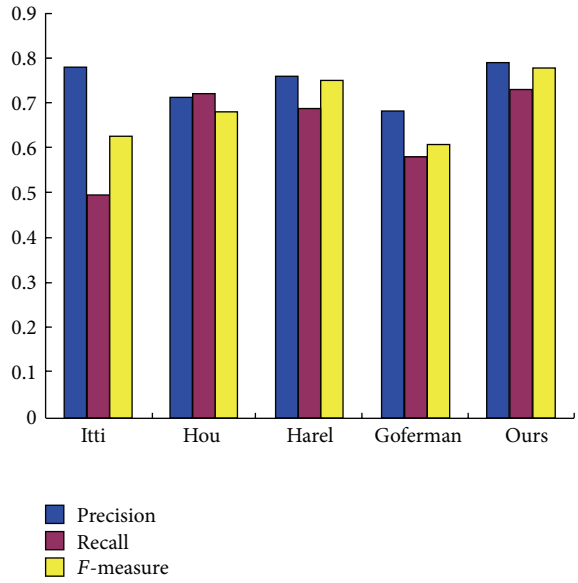



Figure 3: Precision, recall, and $F$-measure value histogram of the different methods.

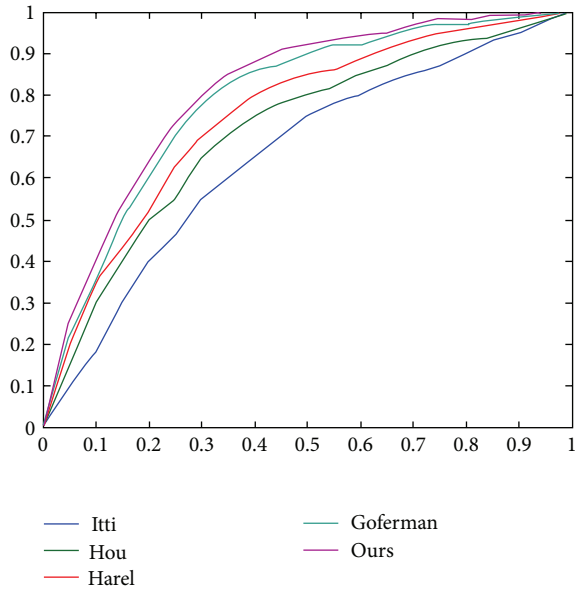



Figure 4: ROC curves of our method and those of the other four approaches.

*4.1. Subjective Evaluation Results.* Subjective evaluation results are usually reflected by comparing the results of different algorithms. In our paper, some representative methods are chosen to compare results. The chosen methods are those of Itti et al. [9], Harel et al. [11], Goferman et al. [10], and Hou and Zhang [17]. The method of Itti et al. [9] is a classic image saliency detection method, from which more effective methods emerged. The method of Harel et al. [11], called graph-based visual saliency, possesses features of biology motivation and mathematical computation. The method proposed by Goferman et al. [10] uses the regional and global features of an image and is similar to ours. The spectral residual method of Hou and Zhang [17] analyzes image saliency in the frequency domain, serving a new route to saliency detection. Figure 3 shows the comparison of our results with those of the aforementioned methods. The figure shows that our method improves the precision and integrity of salient regions.

Table 1: Average cost time for computing test data sets. The resolution of most test images is 400 × 300. The test platform is a machine with a Dual Core 2.10 GHz CPU and 2G memory. The test system is Microsoft Windows XP.

| Method | Itti | Harel | Goferman | Hou | Our method |
|---|---|---|---|---|---|
| Time (s) | 0.812 | 0.145 | 56 | 0.136 | 10.368 |
| Code type | Matlab | Matlab | Matlab | Matlab | Matlab |

*4.2. Objective Evaluation Results.* Objective evaluation results are usually reflected by quantitative data. Evaluation results are reflected in two aspects. The applicability of the algorithm should be considered, which is indicated by the run time and memory space of the algorithm. If the algorithm is applied under high real-time requirements, then the requirement of run time and memory space for the algorithm is high. The recall ratio and precision should also be taken into account. These factors can be generated by the confusion matrix of results.

Table 1 shows the comparison of the run time of our algorithm and those of others. The resolution of most test images is 400 × 300. The test platform is a machine with a Dual Core 2.10 GHz CPU and 2G memory. The test system is Microsoft Windows XP.

Under similar principles, our method consumes more time than do the comparison methods but spends less time than that observed in the method of Goferman (Table 1). However, we achieve better saliency detection results at the cost of time. In special situations, our algorithm satisfies the real-time requirement of an efficient programming language such as C++.

Figure 3 shows the comparison histogram of our method with those of others in terms of precision and recall rates. The $F$-measure is computed by

$$F_\beta = \frac{\left(1+\beta^2\right)\text{Precision} \times \text{Recall}}{\beta^2 \times \text{Precision} + \text{Recall}}. \tag{14}$$

As in [20], we use $\beta^2 = 0.3$ to weight precision more than recall ratio. Our method exhibits higher precision, recall ratio, and $F_\beta$ (Figure 3).

We also show the receiver operating characteristic curves of our method and compare them with those of the aforementioned approaches. Figure 4 indicates that our method achieves higher hit rates and lower false positive rates.

## 5. Conclusion

We propose a salient region detection algorithm for location-quantification images. The algorithm considers points to

regions. First, the salient region is preliminarily located by corner. The region is then precisely located on the basis of image global features. Finally, the salient region is determined by spatial weighted similarity. Our algorithm does not compare pixels individually; only a few are compared, thereby reducing computational complexity. Our algorithm exceeds the performance of other well-known methods.

Nonetheless, we note that our method excessively depends on image color features, making it nonideal for images with complicated scenes and background textures. Future plans include incorporating advanced factors such as human face detection and image symmetry to obtain better results. We believe that exact saliency region detection can improve the results of image scene analysis, image classification, and image retrieval based on image content.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## Acknowledgments

This work is partially supported by the National Natural Science Foundation of China (nos. 61370111 and 61103113) and Beijing Municipal Education Commission General Program (KM201310009004).

## References

[1] G. T. Özyer and F. Y. Vural, "A content-based image retrieval system using visual attention," in *Proceedings of the 18th IEEE Signal Processing and Communications Applications Conference (SIU '10)*, pp. 399–402, Diyarbakır, Turkey, April 2010.

[2] D. Zhuang and S. Wang, "Content-based image retrieval based on integrating region segmentation and relevance feedback," in *Proceedings of the International Conference on Multimedia Technology (ICMT '10)*, pp. 1–3, Ningbo, China, October 2010.

[3] Z. Chen, Z. Feng, and Y. Yu, "Image retrieve using visual attention weight model," in *Proceedings of the IEEE 2nd International Conference on Software Engineering and Service Science (ICSESS '11)*, pp. 718–721, Beijing, China, July 2011.

[4] H. Bai, C. Zhu, and Y. Zhao, "Optimized multiple description lattice vector quantization for wavelet image coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 7, pp. 912–917, 2007.

[5] L. Qin, C. Zhu, Y. Zhao, H. Bai, and H. Tian, "Generalized gradient vector flow for snakes: new observations," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 5, pp. 883–897, 2013.

[6] U. Rutishauser, D. Walther, C. Koch, and P. Perona, "Is bottom-up attention useful for object recognition?" in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '04)*, vol. 2, pp. II37–II44, July 2004.

[7] S. S. Lin, I. C. Yeh, C. H. Lin, and T. Y. Lee, "Patch-based image warping for content-aware retargeting," *IEEE Transactions on Multimedia*, vol. 15, no. 2, pp. 359–368, 2013.

[8] C. Koch and S. Ullman, "Shifts in selective visual attention: towards the underlying neural circuitry," *Human Neurobiology*, vol. 4, no. 4, pp. 219–227, 1985.

[9] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1254–1259, 1998.

[10] S. Goferman, L. Zelnik-Manor, and A. Tal, "Context-aware saliency detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '10)*, pp. 2376–2383, San Francisco, Calif, USA, June 2010.

[11] J. Harel, C. Koch, and P. Perona, "Graph-based visual saliency," in *Advances in Neural Information Processing Systems*, pp. 545–552, 2006.

[12] V. Gopalakrishnan, Y. Hu, and D. Rajan, "Random walks on graphs to model saliency in images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '09)*, pp. 1698–1705, Miami, Fla, USA, June 2009.

[13] L. Duan, C. Wu, J. Miao, L. Qing, and Y. Fu, "Visual saliency detection by spatially weighted dissimilarity," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '11)*, pp. 473–480, Providence, RI, USA, June 2011.

[14] Y. Li, Y. Zhou, L. Xu, X. Yang, and J. Yang, "Incremental Sparse Saliency Detection," in *Proceedings of the 16th IEEE International Conference on Image Processing (ICIP '09)*, pp. 3093–3096, Cairo, Egypt, 2009.

[15] T. Liu, Z. Yuan, J. Sun et al., "Learning to detect a salient object," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 2, pp. 353–367, 2011.

[16] T. Judd, K. Ehinger, F. Durand, and A. Torralba, "Learning to Predict Where Humans Look," in *Proceedings of the IEEE 12th International Conference on Computer Vision (ICCV '09)*, pp. 2106–2113, Kyoto, Japan, 2009.

[17] X. Hou and L. Zhang, "Saliency detection: a spectral residualapproach," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '07)*, pp. 1–8, Minneapolis, Minn, USA, June 2007.

[18] C. Guo, Q. Ma, and L. Zhang, "Spatio-temporal saliency detection using phase spectrum of quaternion fourier transform," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '08)*, pp. 1–8, Anchorage, Alaska, USA, June 2008.

[19] S. M. Smith and J. M. Brady, "SUSAN—a new approach to low level image processing," *International Journal of Computer Vision*, vol. 23, no. 1, pp. 45–78, 1997.

[20] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proceedings of the Alvey Vision Conference*, University of Manchester, 1988.

[21] M. Cheng, G. Zhang, N. J. Mitra, X. Huang, and S. Hu, "Global contrast based saliency region detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR '11)*, pp. 409–416, Providence, RI, USA, June 2011.

[22] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk, "Frequency-tuned salient region detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '09)*, pp. 1597–1604, Miami, Fla, USA, June 2009.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 720416, 7 pages
http://dx.doi.org/10.1155/2013/720416

*Research Article*

# A Distributed Intrusion Detection Scheme about Communication Optimization in Smart Grid

**Yunfa Li and Qili Zhou**

*School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou, China*

Correspondence should be addressed to Yunfa Li; yunfali@hust.edu.cn

Received 1 September 2013; Accepted 26 November 2013

Academic Editor: Yuxin Mao



We first propose an efficient communication optimization algorithm in smart grid. Based on the optimization algorithm, we propose an intrusion detection algorithm to detect malicious data and possible cyberattacks. In this scheme, each node acts independently when it processes communication flows or cybersecurity threats. And neither special hardware nor nodes cooperation is needed. In order to justify the feasibility and the availability of this scheme, a series of experiments have been done. The results show that it is feasible and efficient to detect malicious data and possible cyberattacks with less computation and communication cost.

## 1. Introduction

Smart grid, which is composed of some sensors, digital smart meters, and digital controls, can efficiently and intelligently manage energy supply and consumption. Based on this characteristic, each consumer or energy supplier can monitor and control two-way energy flow. And each consumer or energy supplier can real time manage energy supply and consumption by using smart gird. In fact, the factor that each consumer or energy supplier can monitor and manage energy supply and consumption is that it can get the information of energy supply and consumption by using some management and communication tools in smart grid. In general, a smart grid communication network includes home area network (HAN), neighborhood area network (NAN), and wide area network (WAN). NAN is a network of multiple HANs to deliver metering data to the data concentrator and deliver control date to corresponding HANs. WAN is the largest network for communication to/from data center. In a HAN, each appliance (such as electricity, gas, water, heat, solar panels, etc.) is equipped with a smart meter. And these smart meters connect corresponding smart appliances. Finally, these meters connect a metering gateway. In a NAN, many metering gateways of different home areas connect each other to form a possible wireless mesh network. A WAN connects smart metering gateways with utility and the distribution control system.

Based on the description above, we can find that the smart grid is a hybrid of the power system and the communication network. Therefore, there are a lot of security problems which need people to resolve. These problems include unauthorized smart metering data access, distributed turning off of all devices by an attacker, smart metering data repudiation, stealing power without notice, and attacking the infrastructure of smart grid to cause power outage. Although there are a lot of traditional computer network security mechanisms and methods which can be used in a smart grid, these mechanisms and methods do not take into account the real-time nature of the smart grid and lack of the risk management function when the smart grid is attacked. As a matter of fact, these traditional security mechanisms and methods do not meet the security requirement of smart grid.

Based on the security requirement of smart grid, we will propose an optimization routing algorithm of communication and build an intrusion detection scheme for smart grid in this paper. In the scheme, we will integrate a lot of cybersecurity into the communication of smart grid. This paper is organized as follows. In the next section, we review some related work. In Section 3, we first propose an efficient communication optimization algorithm. Then, we present

an intrusion detection algorithm for smart grid. The two part algorithms compose a distributed intrusion detection scheme about communication optimization in smart grid. In Section 4, we describe a series of experiments and analyze the results of experiments in detail. The time complexity of this distributed intrusion detection scheme is analyzed in Section 5. Finally, the conclusions are drawn in Section 6.

## 2. Related Works

With the growth of the application of smart grid, the security of communication has widely been concerned. More and more people begin to present some methods and mechanisms for protecting the security of communication in smart grid. These methods and mechanisms can be simply shown as follows.

In [1], Ren et al. proposed a provable secure scheme PASS for privacy-protected yet accountable communications between smart meters and smart grid control center. In the PASS scheme, some formal definitions and requirement analysis of privacy and accountability are provided in smart grids. Corresponding data can be hidden and the privacy of customer can be protected by using the PASS scheme. In [2], a wireless communication architecture is first proposed for a smart distribution grid (SDG) based on wireless mesh networks (WMNs). Then, the security framework under this communication architecture is analyzed and potential security attacks and possible counterattack measures are studied. In order to demonstrate the effectiveness of the security framework, a smart tracking firewall was developed to address the intrusion detection and response issue in a WMN-based SDG system. In [3], Wei et al. first discussed the major challenges and strategies to protect smart grid against cyberattacks. Then, they proposed a conceptual layered framework for protecting power grid automation systems against cyberattacks.

In [4], Li and Cao first introduce multicast communications in the smart grid, analyze the requirements on multicast authentication, and review related work. Then, they describe their one-time signature scheme. At last, they present the multicast authentication protocol. In [5], Luo et al. first described the background of smart grid. Then, they analyzed some key technologies, like wide-area measurement system and wide-area communication system. These technologies accelerate the development of wide-area protection. Finally, they prospected the development trends of wide-area protection in smart grid. In [6], Xia and Wang first analyze Wu-Zhou's key management scheme for the smart grid and show it is easily broken by man-in-the-middle attacks. Then, they consider the communication model used by Wu-Zhou and give a detailed description of the components in the model. With the components, they propose a new secure key distribution scheme for the smart grid with high efficiency as well as high security.

In [7], Yan et al. present the background and requirements for smart grid communication security. After discussing the challenge of smart grid communication security, the current research and solutions are surveyed. This paper gives an insight into smart grid communication security in architecture features, system designs, and technical development. In [8], Li et al. discussed the design of a secure access gateway (SAG) for home area network and provided a framework on how to improve the system security, capacity, flexibility, and scalability through cognitive networking.

In [9], Wang and Lu presented a comprehensive survey of cybersecurity issues for the smart grid. Specifically, they focused on reviewing and discussing security requirements, network vulnerabilities, attack countermeasures, secure communication protocols, and architectures in the smart grid. They aim to provide a deep understanding of security vulnerabilities and solutions in the smart grid and shed light on future research directions for smart grid security. In [10], Knapp and Samani explained how to secure the smart grid by using the security methodologies and practices described in earlier chapters. Many methods were explained, including end-point protection, securing individual "zones" within the smart grid architecture, data and application security, and situational awareness.

In [11], Huang et al. first discuss the important security problem of bad data injection in smart grid. Then, from the defenders' point of view, the authors study the quickest detection techniques to detect the bad data injection attack as quickly as possible. And they also demonstrate that the proposed attack can be accomplished by learning the topology structure of the power system and is difficult to detect. In [12], Bou-Harb et al. discuss the security and feasibility aspects of possible communication mechanisms that could be adopted on that subpart of the grid. By accomplishing this, the correlated vulnerabilities in these systems could be remediated, and associated risks may be mitigated for the purpose of enhancing the cybersecurity of the future electric grid.

Though these security methods for smart grid are derived from theoretical analyzing and deducing, they do not take into account the real-time nature of smart grid and lack of the risk management function when the smart grid is attacked. Therefore, these methods are limited in practical application. In order to solve this problem, we propose a new intrusion detection scheme for smart grid, which is based on our proposed communication optimization algorithm.

## 3. Intrusion Detection Scheme

As mentioned above, a typical communication network infrastructure consists of home area network, neighborhood area network, and wide area network in a smart grid. In order to describe the network infrastructure, we define

$$G = (V, E, D), \tag{1}$$

where $G$ is a directed graph used to describe a multihop multichannel network. $V$ is a smart grid component, which denotes some nodes and can measure electricity consumption and manage communications with higher layer components. $E$ is the wireless links between the nodes. $D$ denotes the information flow which can be transmitted in different channels smoothly.

In order to enhance the security of smart grid, the intrusion detection technology is usually used. The fundamentals of intrusion detection is to gather and analyze communication information from various areas within a computer or a network to identify possible security breaches, which include both intrusions (attacks from outside the organization) and misuse (attacks from within the organization). If an obvious deviation between monitored information values is found, an alarm will be issued and the corresponding outlier should be identified and segregated from the communication network of smart grid. If the transmission of communication is not constrained or optimized in the smart grid, the instruction system may achieve a higher false alarm rate because the power requirement of system is dynamic. Therefore, it is necessary to consider how to control and optimize the transmission of communication information in the smart grid. Since the smart grid network is a hybrid of the power system and a communication network, intrusions should be detected that concern either the power system or the communication network or both. Because there are three different layer communication networks (HAN, NAN, and WAN), our proposed intrusion detection algorithm should have the distributed characteristics which can be applied in the three different layer communication networks.

*3.1. Communication Optimization Algorithm.* In order to describe the communication optimization algorithm conveniently, all the notation and definitions used in the rest of this paper are summarized in Table 1, respectively.

In the multihop multichannel network $G$, the links or channels and all flows should be set appropriately when the data flow can be transmitted in the channels smoothly. Here, we let the Boolean variable $v_i(e \mid t) = 1$ when data is transmitted on link channel $i$ of link $e$during time slot $t$. Thus, we can get the following inequality because the number of active channels of link $e$ during time slot $t$ is less than the maximum number of all channels of the link $e$:

$$\sum_{i \in C} v_i(e \mid t) \leq C(e). \tag{2}$$

Because $S_i$ is the constraining set belonging to channel $i$ of link $e$ and $H(S_j)$ is the right hand side constant of the pair which includes the set $S_j$, we can get the following inequality:

$$\frac{1}{H(S_j)} \sum_{(e,j) \in S_j} \frac{d_j(e)}{C_j(e)} \leq 1. \tag{3}$$

Because $p_i(e \mid m)$ is the number of information flow $m$ allowed on channel $i$ of link $e$ and $r(m)$ is the total number of information flows $m$ allowed on the link, we can get $\sum_{e=s(m)} \sum_{i \in C} p_i(e \mid m) = r(m)$. As we all know, the flow allowed on a channel is smaller than the total flow transmitted on this channel, and we can get the following inequality:

$$\frac{1}{H(S_j)} \sum_{S_j} \frac{\sum_{m \in M} p_j(e \mid m)}{C_j(e)} \leq 1. \tag{4}$$

The basic method of the communication optimization algorithm is to find the optimal path for communication flows. Based on this method, the collector first counts all the routes from the source node to the destination node. Then, it begins to calculate the accumulation of the constraining pairs' weights. In the following step, it will label the route which has the lowest value of weight accumulations as the shortest path. Based on these shortest paths, the desired information flow $r(m)$ is distributed to the corresponding node. Thus, the source destination pair, which is also called a commodity pair, can find the optimal route with the shortest distances and link weights.

Table 1: The notation and definitions used in the rest of this paper.

| Notation | Definition |
|---|---|
| $e$ | A link of the network |
| $C(e)$ | All channels of a link $e$ |
| $C(a \mid e)$ | The maximum number of active channels that can be activated on the link $e$ |
| $C_i(e)$ | The transmission capacity of the $i$th channel in data link $e$ |
| $d_i(e)$ | The information flow currently transmitting on channel $i$ of link $e$ |
| $v_i(e \mid t)$ | The Boolean variable when information is transmitted on channel $i$ of link $e$ during time slot $t$ |
| $S_i$ | The constraining set belongs to channel $i$ of link $e$ |
| $n$ | The number of sets of link and channel constraining pairs $(S_1, S_2, S_3, \ldots, S_n)$ between each pair of nodes which are allowed to communicate with each other directly |
| $(s(m), d(m))$ | The communication pair of information $m$ between source node and destination node |
| $H(S_j)$ | The right hand side constant of the pair which includes the set $S_j$ |
| $M$ | The number of communication pairs |
| $p_i(e \mid m)$ | The number of information flow $m$ allowed on channel $i$ of link $e$ |
| $r(m)$ | The total number of information flows $m$ allowed on a link |
| $w(j)$ | The weight of the constraining pair of channel $i$ of link $e$ |
| $b(j)$ | The weight of each set $S_i$ whose value is calculated by the distance between two nodes |
| $F$ | The maximum scaling factor where its value denotes the total slack capacity needed in the network |

Based on the above constraints and optimization method, we propose a communication optimization algorithm in smart grid. The shortest path of communication can be found by using the communication optimization algorithm. The communication optimization algorithm is described as follows.

*Algorithm 1* (communication optimization algorithm). Consider the following.

*Step 1.* Initialize $G = (V, E, D)$, and detect whether there is some source information in certain node. If there is some

source information in the node, it will be labeled as a source node.

*Step 2.* Detect the destination node of the source information and find out all adjacent links of the source node.

*Step 3.* If link $e$ is one of all adjacent links of the source node, the collector counts all the possible transmission routes from the source node to the destination node by the depth-first search method which includes link $e$.

*Step 4.* Count all sets of link and channel constraining pairs $(S_1, S_2, S_3, \ldots, S_n)$ between each pair of nodes which are allowed to communicate with each other directly.

*Step 5.* Initialize $W(j) \leftarrow \delta$, $j \in \{1, 2, 3, \ldots n\}$.

*Step 6.* Initialize $a \leftarrow 0$.

*Step 7.* Judge whether all information flows transmitted on channel $i$ of link $e$ are smaller than the allowed maximum (namely, $\sum_{j=1} w(j) < 1$). **If** $\{\sum_{j=1} w(j) < 1\}$, **Then** {the collector begins to calculate the lowest value of weight of the route in terms of the following computation process:

{ For $m = 1$ To $M$

{$r \leftarrow r(m)$.

/* assign the total number of information flows $m$ allowed on a link to the variable $r$ */:

While $(r > 0)$

$\{\Delta w(j) \leftarrow (\sum_{j \in S_i} (b(j)/H(S_j)))/C_i(e)$

$P_{\min}(s(m), d(m)) \leftarrow \min \sum_{m \in M} \Delta w(j)$.

/* calculate the lowest value of weight accumulations about the communication pair of information $m$ between source node and destination node */:

$\xi \leftarrow \min(d(P_{\min}))$.

/* assign the lowest value of weight accumulations to the variable $\xi$ */:

$\eta \leftarrow \min\{r, \xi\}$.

/* the route which has the lowest value of weight accumulations will be labeled as the shortest path */:

$r \leftarrow r - \eta$.

/* the total number of the variable $r$ should minus the route of information flow which has the lowest value of weight accumulations */:

$d_i(e) \leftarrow d_i(e) + \eta$.

/* the information flow currently transmitted on channel $i$ of link $e$ should append the route of information flow which has the lowest value of weight accumulations */:

$w(j) = w(j) * (1 + (\varepsilon\eta/d(P_{\min})))\}\}$.

/* calculate the current weight of the constraining pair of channel $i$ of link $e$ */:

$a = a + 1\}$

}

**Else** {go to Step 8}.

*Step 8.* $\rho \leftarrow \max(\sum_{j \in S}(d_i(e)/C_i(e)))$.

*Step 9.* $F \leftarrow a/\rho$.

/* calculate the maximum scaling factor */

*Step 10.* End.

*3.2. Intrusion Detection Algorithm.* Based on the communication optimization algorithm, we will design an intrusion detection algorithm to detect malicious data and possible cyberattacks which have considerable influence on the communication flow. In the intrusion detection algorithm, we let the smart grid be completely safe during its deployment phase. And the existing security agreement of different layers holds the same assumptions, which include HAN, NAN, and WAN. Moreover, the monitor can detect the following information, which is shown in Table 2, to collect the malicious network attack behavior.

In the smart grid, there is HAN, NAN, and WAN. Each area network may have multiple links and each link may have multiple channels. Each channel has its communication flow for a link. These communication flows form different monitoring attributes of link. Here, let the multiple monitoring attributes of link $e_i$ in certain area network form a multiple dimension vector $A(e_i) = \{A_1(e_i), A_2(e_i), A_3(e_i), \ldots, A_k(e_i)\}$, where $k$ is the number of the monitoring attributes. And all the monitoring vectors of link $e_1, e_2, e_3, \ldots, e_n$ form a matrix $A(e) = \{A(e_1), A(e_2), A(e_3), \ldots, A(e_n)\}$, where $n$ is the number of links in the area network.

In order to detect any possible intrusion and maintain the security of each area network, we let the information flow sent from the source node to the destination node keep to a normal distribution in each channel during intrusion detection. Thus, all $A(e_i)$ $(e_i \in \{e_1, e_2, e_3, \ldots e_n\})$ in certain area network form a sample of a multivariate normal distribution. And $A(e_i)$ is distributed as $N_k(\mu, \sigma)$, following a multivariate normal distribution with mean vector $\mu$ and variance-covariance matrix $\sigma$. Therefore, the probability that $A(e_i)$ satisfies $(A(e_i) - \mu)^T \sigma^{-1} (A(e_i) - \mu) \succ \chi_k^2(\alpha)$ is $\alpha$, where $\chi_k^2(\alpha)$ is the upper $(100\alpha)$th percentile of a chi-square distribution with $k$ degrees of freedom.

If we assume that the estimate of $\mu$ is $\widehat{\mu}$ and the estimate of $\sigma$ is $\widehat{\sigma}$, then, we can get the probability that $A(e_i)$ satisfies $(A(e_i) - \widehat{\mu})^T \widehat{\sigma}^{-1} (A(e_i) - \widehat{\mu}) \succ \chi_k^2(\alpha)$, which is expected to be roughly $\alpha$. Let $\phi(e_i) = ((A(e_i) - \widehat{\mu})^T \widehat{\sigma}^{-1} (A(e_i) - \widehat{\mu}))^{1/2}$. Link $e_i$ will be regarded as an outlier if $\phi(e_i)$ or $\phi^2(e_i)$ is unusually large. In our algorithm, link $e_i$ is regarded as an abnormal link if $\phi^2(e_i) \succ \chi_k^2(\alpha)$.

Table 2: The detected information.

| Detected information | Collected attack behavior |
|---|---|
| Sensor sensed data | Fabricate information attack |
| Information sending rate | Energy exhausting attack |
| Information mismatch rate | Message alter attack |
| Information receiving rate | Sink hole attack |
| Information dropping rate | Black hole attack, select forward attack |
| Information sending power | Worm hole attack, hello attack |

Rather than estimating $\mu$ and $\sigma$ by the simple mean and the simple variance-covariance matrix:

$$\widehat{\mu} = \frac{1}{n}\sum_{i=1}^{n} A(e_i),$$
$$\widehat{\sigma} = \frac{1}{n-1}\sum_{i=1}^{n} (A(e_i) - \widehat{\mu})(A(e_i) - \widehat{\mu})^T \quad (5)$$

in which the values from outlying links can easily distort the estimates of $\mu$ and $\sigma$ and the detection via Mahalanobis distances may fail to identify true abnormal links, we adopt the orthogonalized Gnanadesikan-Kettenring estimators $\widehat{\mu}$ and $\widehat{\sigma}$ [13]. Therefore, the intrusion detection algorithm can be described as follows.

*Algorithm 2* (the intrusion detection algorithm). Consider the following.

*Step 1.* The system sends a "detect" command to the controller of HAN, the controller of NAN, and the controller of WAN, respectively.

*Step 2.* After each controller receives the "detect" command, it begins to execute our proposed communication optimization algorithm and finds the shortest path of communication from the source node to the destination node, respectively.

*Step 3.* The controller of each area network, respectively, controls the information flow of corresponding channel and lets the information flow sent from the source node to the destination node keep to a normal distribution in the channel during intrusion detection.

*Step 4.* After the destination node receives the information flow sent from the source node, corresponding controller begins to calculate $\widehat{\mu}$ and $\widehat{\sigma}$ in terms of the following computation process.

(1) Compute $B(e) = \{b(e_1), b(e_2), \ldots, b(e_n)\}$, where $b(e_1) = P^{-1}A(e_1)$, $P = \mathrm{diag}(\lambda(\widetilde{A}_1(e)), \lambda(\widetilde{A}_2(e)), \ldots, \lambda(\widetilde{A}_k(e)))$, and $\lambda(\widetilde{A}_j(e))$ is the $j$ row of $A(e)$.

(2) Calculate $k \times k$ matrix $R$, where $\psi_{i,j} = \begin{cases} (1/4)\widehat{\lambda}^2(B_i+B_j) - \widehat{\lambda}^2(B_i-B_j) & j \neq k \\ 1 & j=k \end{cases}$

(3) Apply the spectral decomposition to obtain $\psi = \theta\Lambda\theta^T$, where $\theta$ is $\psi$'s eigenmatrix and $\Lambda$ is the diagonal matrix composed of $\psi$'s eigenvalues.

(4) Compute $H = \{h(e_i) \mid h(e_i) = \theta^T A(e_i)\}$. Then calculate $\Delta = (\widehat{\mu}(H_1(e)), \widehat{\mu}(H_2(e)), \ldots, \widehat{\mu}(H_k(e)))$ and $\Phi = \mathrm{diag}(\widehat{\lambda}^2(H_1(e)), \widehat{\lambda}^2(H_2(e)), \ldots, \widehat{\lambda}^2(H_k(e)))$.

(5) Let $V = P\theta$. Then the robust multivariate estimates are $\widehat{\mu} = V\Delta$ and $\widehat{\sigma} = V\Phi V^T$.

*Step 5.* If the controller finds obvious deviation between the information data sent by the source node and its monitored data, it will raise the alarm and show corresponding warning information.

*Step 6.* The controller of each area network, respectively, judges that the information data of each source node has completely been transmitted or not within its own jurisdiction. **If** {there is some information data which has not completely been transmitted}, **Then** {go to Step 3}; **Else** {go to Step 7}.

*Step 7.* End.

In fact, we choose the Mahalanobis distance measurement because it includes the interattribute dependencies. Thus, we can compare the attribute combinations and get more precise results [14]. The reason why we decide to choose the orthogonalized Gnanadesikan-Kettenring estimator is because it ensures a high breakdown point with some missing data and can compute quickly with a lower computational cost [15].

## 4. Experiments and Results Analysis

In this section, we first introduce our experiments. Then, we analyze the results. The specific process can be described as follows.

*4.1. Experiments.* In our experiments, IEC 61850, ZigBee, and IEEE 802.11s are used to build a complex communication system for smart grid. In the complex communication system, there are three layers which include HAN, NAN, and WAN. Moreover, there are 20 nodes in HAN, 3 nodes in NAN, and 1 node in WAN. In order to ensure the security of each component, every node has an intrusion detection system belonging to its corresponding network, which can use our proposed intrusion detection scheme and the inside r attacker detection scheme [15]. By a series of experiments, we can get the false alarm ratio, the detection accuracy ratio, and the power consumption in the intrusion detection system belonging to WAN when the intrusion detection system uses the two different detection schemes, respectively. The results are shown in Figures 1, 2, and 3.

The similar situation can also get in another node's intrusion detection system by using the above two different detection schemes, respectively.

*4.2. Results Analysis.* Figure 1 shows the false alarm ratio between our proposed intrusion detection scheme and the

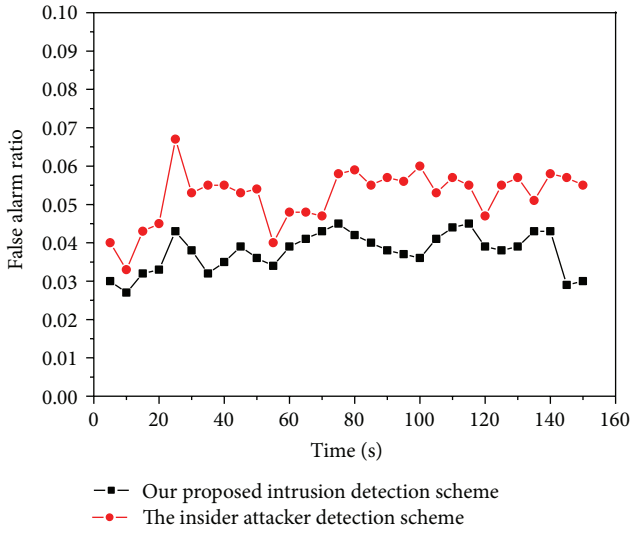



Figure 1: Comparison of false alarm ratio between our proposed intrusion detection scheme and the insider attacker detection scheme.

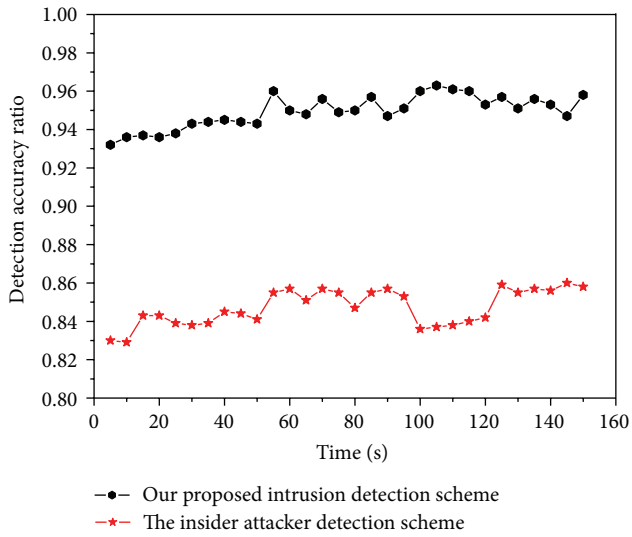



Figure 2: Comparison of detection accuracy ratio between our proposed intrusion detection scheme and the insider attacker detection scheme.

insider attacker detection scheme. According to the results of Figure 1, we will find that our proposed intrusion detection scheme produces less number of false alarm alerts than that of the insider attacker detection scheme.

Figure 2 shows the detection accuracy ratio between our proposed intrusion detection scheme and the insider attacker detection scheme. According to the results of Figure 2, we will find that the detection accuracy of our proposed intrusion detection scheme is much higher than that of the insider attacker detection scheme.

Figure 3 is the comparison of power consumption between our proposed intrusion detection scheme and the insider attacker detection scheme. According to the results

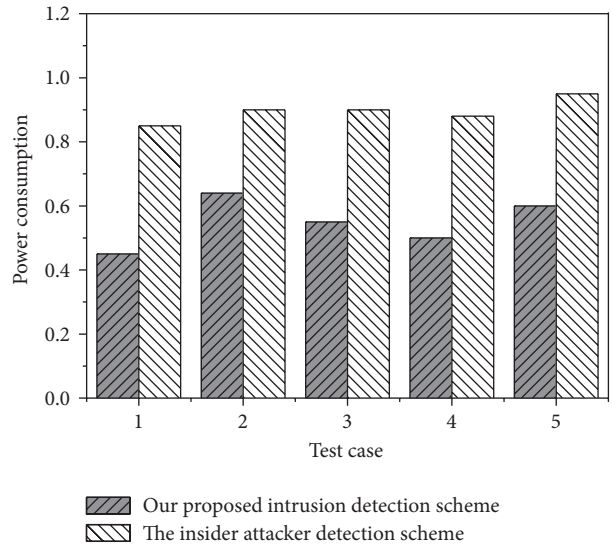



Figure 3: Comparison of power consumption between our proposed intrusion detection scheme and the insider attacker detection scheme.

of Figure 2, we find that the power consumption of our proposed intrusion detection scheme is much less than that of the insider attacker detection scheme.

The main reason generating the three situations is that the detection results are less influenced in our proposed intrusion detection scheme than those in the insider attacker detection scheme.

Because the similar situation can also get in another node's intrusion detection system by using the different detection schemes, respectively, our proposed intrusion detection scheme is feasible and available.

## 5. Complexity Analysis

In this section, we will analyze the time complexity of our proposed distributed intrusion detection scheme about communication optimization in smart grid. The analysis processes are described as follows.

There are two main algorithms in the distributed intrusion detection scheme. One is the communication optimization algorithm and the other is the intrusion detection algorithm. The time complexity of our proposed distributed intrusion detection scheme mainly focuses on the two algorithms.

(1) In the communication optimization algorithm, we have the following.

(a) In order to detect whether there is some source information in certain node, this algorithm takes $O(|V|)$ time to find out all the nodes, where $|V|$ denotes the total number of nodes in $V$.

(b) In order to count all the possible transmission routes from the source node to the destination node, this algorithm takes $O(|V| + |E|)$ time

to realize the goal, where $|E|$ denotes the total number of nodes in $E$.

(c) In order to calculate the lowest value of weight of the route, this algorithm takes $O(n * M)$ time to realize the goal, where $n$ denotes the set total number of link and channel constraining pairs and $M$ denotes the total number of information flows.

Hence, the total time complexity in the communication optimization algorithm is $O((|V| + |E|) * |V| * n * M)$.

(2) In the intrusion detection algorithm, we have the following.

(a) In order to calculate $\widehat{\mu}$ and $\widehat{\sigma}$, this algorithm takes $O(n_1^3 * K)$ time to realize the goal, where $n_1$ denotes the set total number of links and $K$ denotes the total number of all monitoring attributes in an information flow. Therefore, it will take $O(n_1^3 * K * M)$ time to calculate all $\widehat{\mu}$ and $\widehat{\sigma}$ when the total number of information flows is $M$.

Thus, the time complexity of our proposed distributed intrusion detection scheme is $O((|V| + |E|) * |V| * n * M^2 * n_1^3 * K)$, where $|V|$ denotes the total number of nodes in $V$,$|E|$ denotes the total number of nodes in $E$, $n$ denotes the set total number of link and channel constraining pairs, $M$ denotes the total number of information flows, $n_1$ denotes the set total number of links, and $K$ denotes the total number of all monitoring attributes in an information flow.

## 6. Conclusion

In this paper, we proposed an intrusion detection scheme for smart grid which is based on our proposed communication optimization algorithm. In the scheme, each node acts independently when it processes the communication flows and handles cybersecurity threats. And neither special hardware nor nodes cooperation is needed. Our experiment results show that our scheme can achieve a lower false alarm rate and a higher detection accuracy rate than the existing detection schemes. At the same time, it can also reduce the monitoring power consumption with the requirement of grouping the nodes in the network.

## Acknowledgments

This paper is supported by Zhejiang Provincial Natural Science Foundation of China under Grant nos. Y14F020186 and Y14F020194 and Startup Foundation of School Grant no. KYS055608103.

## References

[1] W. Ren, J. Song, Y. Yang, and Y. Ren, "Lightweight privacy-aware yet accountable secure scheme for SM-SGCC communications in smart grid," *Tsinghua Science and Technology*, vol. 16, no. 6, pp. 640–647, 2011.

[2] X. Wang and P. Yi, "Security framework for wireless communications in smart distribution grid," *IEEE Transactions on Smart Grid*, vol. 2, no. 4, pp. 809–818, 2011.

[3] D. Wei, Y. Lu, M. Jafari, P. M. Skare, and K. Rohde, "Protecting smart grid automation systems against cyberattacks," *IEEE Transactions on Smart Grid*, vol. 2, no. 4, pp. 782–795, 2011.

[4] Q. Li and G. Cao, "Multicast authentication in the smart grid with one-time signature," *IEEE Transactions on Smart Grid*, vol. 2, no. 4, pp. 686–696, 2011.

[5] L. Luo, N. Tai, and G. Yang, "Wide-area protection research in the smart grid," *Energy Procedia*, vol. 16, pp. 1601–1606, 2012.

[6] J. Xia and Y. Wang, "Secure key distribution for the smart grid," *IEEE Transactions on Smart Grid*, vol. 3, no. 3, pp. 1437–1443, 2012.

[7] Y. Yan, Y. Qian, H. Sharif, and D. Tipper, "A survey on cyber security for smart grid communications," *IEEE Communications Surveys and Tutorials*, vol. 14, no. 4, pp. 998–1010, 2012.

[8] T. Li, J. Ren, and X. Tang, "Secure wireless monitoring and control systems for smart grid and smart home," *IEEE Wireless Communications*, vol. 19, no. 3, pp. 66–73, 2012.

[9] W. Wang and Z. Lu, "Cyber security in the smart grid: survey and challenges," *Computer Networks*, vol. 57, no. 5, pp. 1344–1371, 2013.

[10] E. D. Knapp and R. Samani, "securing the smart grid," in *Applied Cyber Security and the Smart Grid*, pp. 125–145, 2013.

[11] Y. Huang, M. Esmalifalak, H. Nguyen et al., "Bad data injection in smart grid: attack and defense mechanisms," *IEEE Communications Magazine*, vol. 51, no. 1, pp. 27–33, 2013.

[12] E. Bou-Harb, C. Fachkha, M. Pourzandi, M. Debbabi, and C. Assi, "Communication security for smart grid distribution networks," *IEEE Communications Magazine*, vol. 51, no. 1, pp. 42–49, 2013.

[13] R. A. Maronna, R. D. Martin, and V. J. Yohai, *Robust Statistics: Theory and Methods*, John Wiley & Sons, Chichester, UK, 2006.

[14] V. J. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artificial Intelligence Review*, vol. 22, no. 2, pp. 85–126, 2004.

[15] F. Liu, X. Cheng, and D. Chen, "Insider attacker detection in wireless sensor networks," in *Proceedings of the 26th IEEE International Conference on Computer Communications (IEEE INFOCOM '07)*, pp. 1937–1945, May 2007.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 619029, 6 pages
http://dx.doi.org/10.1155/2013/619029

*Research Article*

# Warehouse Optimization Model Based on Genetic Algorithm

**Guofeng Qin,[1] Jia Li,[2,3] Nan Jiang,[1] Qiyan Li,[1] and Lisheng Wang[1]**

[1] *The Computer Science & Technology Department, Tongji University, Shanghai 200092, China*
[2] *Shanghai Bao-Steel Logistics Co. Ltd., Shanghai 200940, China*
[3] *The School of Business Administration, Northeastern University, Shenyang 110819, China*

Correspondence should be addressed to Guofeng Qin; qingguofeng.cn@gmail.com

Received 25 July 2013; Accepted 1 September 2013

Academic Editor: Yuxin Mao



This paper takes Bao Steel logistics automated warehouse system as an example. The premise is to maintain the focus of the shelf below half of the height of the shelf. As a result, the cost time of getting or putting goods on the shelf is reduced, and the distance of the same kind of goods is also reduced. Construct a multiobjective optimization model, using genetic algorithm to optimize problem. At last, we get a local optimal solution. Before optimization, the average cost time of getting or putting goods is 4.52996 s, and the average distance of the same kinds of goods is 2.35318 m. After optimization, the average cost time is 4.28859 s, and the average distance is 1.97366 m. After analysis, we can draw the conclusion that this model can improve the efficiency of cargo storage.

## 1. Introduction

In the process of cargo storage, it requires to assign a position for each cargo. An appropriate position for each cargo is important and influent the efficiency of cargo storage. So it is very important to manage the position of warehouse automatically. The optimization of warehouse is necessary for high efficiency [1, 2].

Xiangling Xu in Xi'an University of Technology has established a knowledge library based on expert system, through analysis and summary of the job schedule and the assignment of the position in automatic warehouse system. They also discussed the feasibility of the schedule of task using expert system, through simulation by computer.

Xiangli Shi in Taiyuan Heavy Machinery Institute proposed a simple model of warehouse and a new algorithm of schedule for this model. Take the task of getting or putting goods as a task which is associated with the library into a car by the task number and it does the task of cargo storage by task number.

H. Brynzer proposed a principle of partition. This principle divides the whole warehouse into different pieces according to different goods, different selection method, and different type of work. As a result, it will improve the efficiency of cargo storage by putting different goods in different position. Of course, the online schedule theory cannot match the current needs because these papers only discussed this problem with single row [3].

A 3D model is built in this paper, in order to analyses this problem through row, column, and layer. Perhaps, different goods could be laid in different position. When goods are got and put, there are many different roadways and multi-operation to making the stacker load balance in order to improve the storage efficiency, an optimization method will be studied as follow.

## 2. 3-Dimensional Model

Generally, In order to meet the needs of the shelf stability and improve security, heavy cargo should be kept on the ground or the lower position on the shelf, and light cargo should be put in the higher position on the shelf. As a result, it can reduce the height of the whole shelf. The advantage of this mathematical model is to ensure that the focus of the whole shelf is below half of the height of the shelf.

*2.1. Analysis Problem.* The warehouse storage totally has $N$ rows, each row of shelves has $P$ layers and $Q$ Columns, see Figure 1(a). The $k$th row, $i$th layer, and $j$th column position is

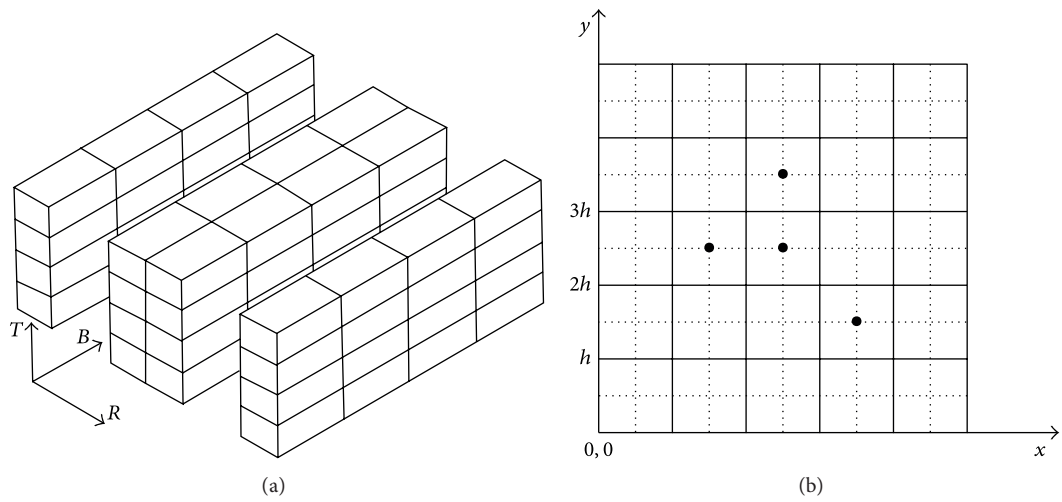



Figure 1: Shelf Figure. (a) Shelf Figure. (b) The focus of Shelf.

on the shelf with $X_{kij}$, the goods' weight is $m_{kij}$ which is put on the position, the turnover is $I_{kij}$. Each position's height is $h$ and the width is $l$, $k \in [1, N]$, $i \in [1, P]$, $j \in [1, Q]$. The closest row near to the shipping area is the first row, the closest column near to the shipping area is the first column, and the closest layer near to the shipping area is the first layer. In this paper, the lower row, the lower layer, and the lower column are encoded. As a result, the smaller number is encoding for the lower position in the warehouse.

*2.2. Build Model.* A single row of shelves of warehouse can be seen in Figure 1(b). For each cargo, the overall focus of the goods is in the center of the cargo. As shown above, the intersection of dotted lines is the focus of goods. For any cargo $(i, j)$, the focus of the goods is $((i - 1/2)l, (j - 1/2)h)$.

Each position's height is $h$, and the width is $l$; the goods' weight is $G_{kij}$, excluding the width of column between positions. The focus of the whole warehouse is as follow:

$$S_x = \frac{\sum_{i=1}^{P}\sum_{k=1}^{N}\sum_{j=1}^{Q}\left(G_{kij} * (i - 1/2)\, l\right)}{\sum_{i=1}^{P}\sum_{k=1}^{N}\sum_{j=1}^{Q} G_{kij}}, \quad G_{kij} \leqslant G_{\max},$$

$$S_y = \frac{\sum_{i=1}^{P}\sum_{k=1}^{N}\sum_{j=1}^{Q}\left(G_{kij} * \left(j - 1/2\right) h\right)}{\sum_{i=1}^{P}\sum_{k=1}^{N}\sum_{j=1}^{Q} G_{kij}}, \quad G_{kij} \leqslant G_{\max}, \tag{1}$$

where $G_{\max}$—each shelf can bear maximum weight of cargo (kg); $G_{kij}$—weight of goods on the cargo (kg); $S_x$, $S_y$—the focus of the whole shelf on $x$ and $y$ direction position (m).

In order to meet the demand for the stability of shelf, the objective function is to make the focus the smallest one. We only consider the $y$-axis direction in the center of the shelf.

Consider

$$S_{y\min} = \min\left(S_y\right) = \min\left(\frac{\sum_{i=1}^{P}\sum_{k=1}^{N}\sum_{j=1}^{Q}\left(G_{kij} * \left(j - 1/2\right) h\right)}{\sum_{i=1}^{P}\sum_{k=1}^{N}\sum_{j=1}^{Q} G_{kij}}\right). \tag{2}$$

It needs to meet the following condition:

$$S_{y\min} \leq \frac{1}{2}\sum_{i=1}^{P} h. \tag{3}$$

## 3. Object Function

For improving the efficiency of automatic warehouse system, allocation strategy for database storage includes improved FIFO, (Light underweight) LUW, (Divided Roadway Store) DRS, (Goods Relation) GR, (Cargo Partition), CP and (Shortest Route) SR. This paper uses the distance of the same goods and the cost time of cargo as the two parameters of analysis.

*3.1. Efficiency Model.* The operation cycle of stacker is a main impact factor to decide efficiency of warehouse. It is the main cost time of warehouse. Stacker moves from the shipping area to the specified location; it gets goods on the shelf and comes back to the shipping area. After running these tasks, it finishes shipping goods once. This paper only discusses the stacker moves from the shipping area to the specified location as it costs time. The speed of stacker on the $x$-axis and $y$-axis direction can be seen in Figure 2. $V_{\max}$ is stacker's max speed.

For a given distance $S_x$ on the $x$-axis direction, the running time on the horizon is as follows:

$$t_x = \begin{cases} 2\sqrt{\frac{s_x}{a_x}} & S_x \le S_{x\max}, \\ 2\frac{v_{x\max}}{a_x} + \frac{\left(s_x - v_{x\max}^2/a_x\right)}{v_{x\max}}, & S_x \ge S_{x\max}, \end{cases} \quad (4)$$

$$S_{x\max} = \frac{v_{x\max}^2}{2a_x}.$$

The same for a given distance $S_y$, the cost time is as follows:

$$t_y = \begin{cases} 2\sqrt{\frac{s_y}{a_y}} & S_y \le S_{y\max}, \\ 2\frac{v_{y\max}}{a_y} + \frac{\left(s_y - v_{y\max}^2/a_y\right)}{v_{y\max}}, & S_y \ge S_{y\max}, \end{cases} \quad (5)$$

$$S_{y\max} = \frac{v_{y\max}^2}{2a_y}.$$

The total cost time is $t_{kij} = \max(t_x, t_y)$.

The efficiency of the warehouse is calculated, including the cost time of a stacker, it also has relationship with the frequency of goods. Inventory turnover is also called inventory turnover. It is a measure of acquired inventory and evaluation of business, production, marketing, and other aspects of management to recover the status of a comprehensive index. The optimization of cargo within the warehouse can be used to measure the speed of movement of a product. It is the cost of goods sold by average inventory ratio obtained by the addition.

$$\text{Ratio} = \left(\frac{\text{inventory frequency}}{\text{inventory time}}\right) * 100 \quad (6)$$

A formula of efficiency is built as follows:

$$E_{\min} = \min E = \min \sum_{i=1}^{P} \sum_{k=1}^{N} \sum_{j=1}^{Q} \left(f_{kij} * t_{kij}\right) \quad (7)$$

*3.2. Classification Model.* When goods are put on the shelf, the relevance of the goods needs to be considered. According to this, an object function is built. At first, the average distance should be calculated between the same kinds of goods on one shelf. In Figure 2, every black point expresses the focus of every good. It is assumed that these goods belong to the same type. We need to calculate the distance between every two goods compute the sum of these distances.

Consider

$$d_{ij} = \sqrt[2]{\left(dx_i - dx_j\right)^2 + \left(dy_i - dy_j\right)^2}. \quad (8)$$

The sum of these distances is as follows:

$$D_{\text{sum}} = \sum_{i=1}^{T-1} \sum_{j=i+1}^{T} d_{ij}. \quad (9)$$

$T$—count of same kinds of goods.

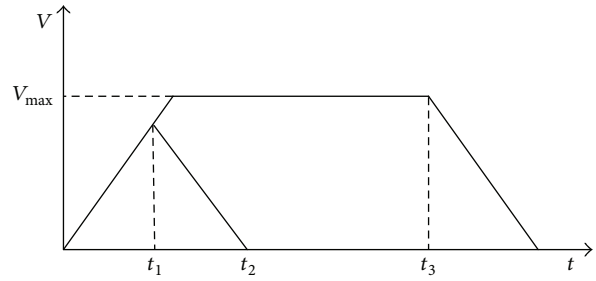



Figure 2: Relation between speed and time.

$T(T+1)/2$ numbers are added. The average value is as follows:

$$D_e = \frac{2D_{\text{sum}}}{T(T+1)} = \frac{2\sum_{i=1}^{T-1}\sum_{j=i+1}^{T} d_{ij}}{T(T=1)}. \quad (10)$$

The min goal distance is as follows:

$$D_{\min} = \min D_e = \min\left(\frac{2\sum_{i=1}^{T-1}\sum_{j=i+1}^{T} d_{ij}}{T(T=1)}\right). \quad (11)$$

For object Function, we need to discuss the two models: efficient model and classification model. Deal with these two models into one, so we use weight distribution. Finally, the objective function is $f = qE_{\min} + (1-q)D_{\min}$.

So the assignment of the position on shelves should be considered in both efficiency and classification models. This is a combination of multiobjective optimization problem. For multiobjective optimization problem, the goal is conflicting in many cases. Generally, there is no unique global optimal solution, but there is an optimal solution set. Concentration of elements in the set of optimal solutions is not comparable. The optimal performance of a target solution is likely to mean poor performance of other objectives. Only one pursuit of one objective optimization has not much practical significance. It has great importance to seek a solution which makes the objective function of each dimension in good performance [4].

## 4. Warehouse Optimization Genetic Algorithm

Warehouse optimization genetic algorithm is complex combinatorial optimization problem of data. Genetic algorithm is based on evolutionary theory which has good adaptive performance and whose condition is not strict. It is suitable for solving complex combinatorial optimization problems.

*4.1. Algorithm Analysis.* Adaptor function value can be known as an ability which measures an individuation to survive in a set. It determines the degree to understand the pros and cons. It determines an individuation to survive or to die.

Because the goal is to find the min value of the objective function and the two models above is to find max value, so

$$\begin{bmatrix} a_{n1} & a_{n2} & \cdots & a_{nn} \\ a_{(n-1)1} & a_{(n-1)2} & \cdots & a_{(n-1)n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{11} & a_{12} & \cdots & a_{1n} \end{bmatrix}$$

Figure 3: The general matrix.

$$\begin{bmatrix} 1 & 2 & 5 & 6 \\ 3 & 4 & 7 & 8 \\ 9 & 10 & 13 & 14 \\ 11 & 12 & 15 & 16 \end{bmatrix} \begin{bmatrix} 1 & 2 & 5 & 6 \\ 3 & 14 & 7 & 8 \\ 9 & 10 & 13 & 4 \\ 11 & 12 & 15 & 16 \end{bmatrix}$$

Figure 4: Relationship between shelf and matrix.

we take the following measures. A suitable degree function is as follows:

$$f_e = \begin{cases} f_{\max} - f & f_{\max} - f > 0 \\ 0 & \text{other.} \end{cases} \tag{12}$$

Because the optimization problem of cargo storage is complexity and the scale of the solution is large, it is hard to encode the position using 0 and 1. As a result, we encode positions using $N \times P \times Q$ matrix. $x_{kij}$ is number of goods in the matrix. Information of goods is defined with weight, count, and time of turnover. The general matrix corresponding is in Figure 3. An example matrix in experience is in Figure 4.

The individuals in GA group are randomly generated when initialized. It is very important to determine the initialization group. If the difference between every two individuals is too small, it will make the group loss diverse and lead this algorithm to early convergence. As a result, it cannot lead this genetic algorithm to get the global optimization.

Firstly, a certain number of individuals are generated randomly. Secondly, the best one is picked and added to the initial group. Lastly, this process is continue to iterate until the count of individuals in initial population is big enough.

Consider

$$\begin{aligned} d_s &= \left\| S_{mi} - S_{ni} \right\| = \sqrt[2]{\sum_{j=1}^{n} \left( S_{mij} - S_{nij} \right)}, \\ d_e &= \left\| E_{mi} - E_{ni} \right\| = \sqrt[2]{\sum_{j=1}^{n} \left( E_{mij} - E_{nij} \right)}. \end{aligned} \tag{13}$$

The differences between individuals are used to determine which individual should be chosen.

*4.2. Genetic Operators.* Genetic operators decide which parents should be chosen into the next generation to continue this algorithm. The method of gamble selection is used, including three steps. Firstly, some locations are selected randomly from the first parent individual and the elements corresponding to these locations are stored. Secondly, these elements are deleted from the second parent individual. Lastly, the other elements of the second parent individual are copied to an empty string.

$$\begin{bmatrix} 1 & 2 & 5 & 6 \\ 3 & 4 & [7] & 8 \\ 9 & 10 & 13 & 14 \\ 11 & 12 & [15] & 16 \end{bmatrix} \times \begin{bmatrix} 15 & 7 & 5 & 6 \\ 8 & 4 & [2] & 3 \\ 9 & 11 & 13 & 14 \\ 10 & 12 & [1] & 16 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 & 5 & 6 \\ 3 & 4 & 2 & 8 \\ 9 & 10 & 13 & 14 \\ 11 & 12 & 1 & 16 \end{bmatrix} \times \begin{bmatrix} 15 & 7 & 5 & 6 \\ 8 & 4 & 7 & 3 \\ 9 & 11 & 13 & 14 \\ 10 & 12 & 15 & 16 \end{bmatrix}$$

Figure 5: Crossover.

$$\begin{bmatrix} 15 & 7 & 5 & 6 \\ 3 & 4 & 2 & 8 \\ 9 & 10 & 13 & 14 \\ 11 & 12 & 1 & 16 \end{bmatrix} \times \begin{bmatrix} 1 & 2 & 5 & 6 \\ 8 & 4 & 7 & 3 \\ 9 & 11 & 13 & 14 \\ 10 & 12 & 15 & 16 \end{bmatrix}$$

Figure 6: Crossover result.

$$\begin{bmatrix} 1 & 2 & 5 & 6 \\ 3 & 4 & [7] & 8 \\ 9 & 10 & 13 & 14 \\ 11 & 12 & [15] & 16 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 2 & 5 & 6 \\ 3 & 4 & [15] & 8 \\ 9 & 10 & 13 & 14 \\ 11 & 12 & [7] & 16 \end{bmatrix}$$

Figure 7: Mutation.

$$\begin{bmatrix} 1 & 2 & 5 & 6 \\ 3 & 4 & 0 & 8 \\ 9 & 10 & 13 & 14 \\ 0 & 12 & 15 & 16 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 2 & 0 & 0 \\ 3 & 4 & 6 & 8 \\ 9 & 10 & 13 & 14 \\ 5 & 12 & 7 & 16 \end{bmatrix}$$

Figure 8: Repair result.

The elements are exchanged in the first parent individual with the same order. According to crossover the two individuals, the corresponding gene mapping is determined to exchange 7 with 2 and 15 with 1. The details can be seen in Figure 5. The matrix is legitimized with the corresponding gene mapping. The crossover result can be seen in Figure 6.

*4.3. Mutation.* Mutation is randomly changing the value of some elements in one individual in a small probability. The goal is to increase the diversity of the group.

In probability, some individual mutation with experience is used. The individuals are chosen to mutate in Figure 7. Two positions are exchanged in the matrix with the probability of $P_m$, which is to exchange the two goods. If this mutation happens in a small range, $P_m$ is always between 0.1–0.4.

Only the goods are adjusted on the shelf; they might be allocated on the position which is far away from the shipping area. The optimized result is to put goods to the empty position which is near the shipping area. The repair result can be seen in Figure 8.

*4.4. Termination Condition.* Some randomly individuals are selected to initial population for the crossover and mutation. What is the termination condition? As the algorithm runs until it reachs a certain extent, the structure of individual between every two individuals is very similar. It will be

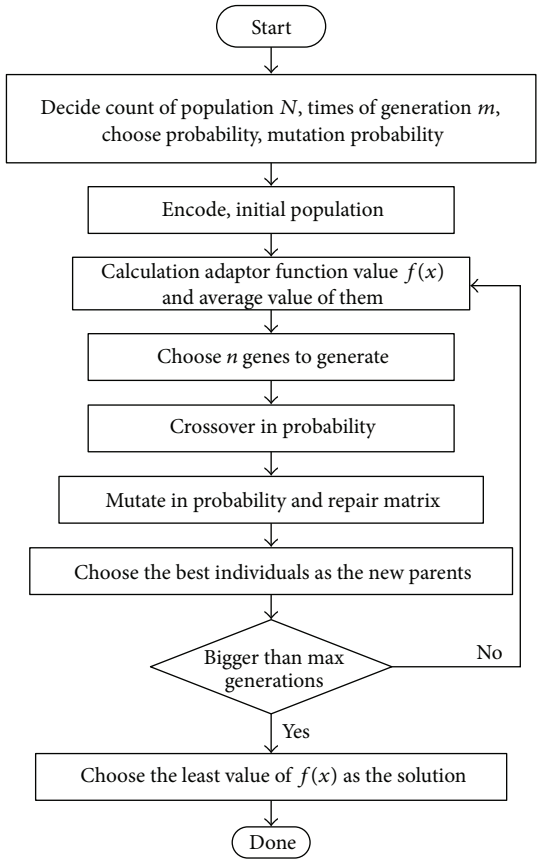



Figure 9: Algorithm flowchart.

difficult to find a better individual if the algorithm still runs. Under this condition, we think this algorithm is convergent. In this paper, we think the population is convergent if the standard deviation of this population is little, less than certain values, which is usually small.

Consider

$$D = \sqrt[2]{\sum_{i=1}^{n} \left(f_i - \overline{f}\right)}. \tag{14}$$

In the formula, $i$ is the $i$th individual in the population, $n$ is the count of the population, $f_i$ is the adaptor function value of the $i$th individual, and $f$ is the average adaptor function value.

In the process of calculation, if the value of $D$ is less than 0.001, this population is convergent. Algorithm flowchart can be seen in Figure 9.

## 5. Results on Simulation

A class of GA algorithm is setup, which is the main class of the program. In this class, Firstly, it will create HW_COUNT = 125 goods then save all these information of goods into a list m_hw. Then allocate all these goods on the shelf randomly to simulate the work of cargo storage. Before optimization, cost time is 4.52996 s and distance of same good is 2.35318 m.

Table 1: Some parameters in optimization.

| Shelf | Skater | GA |
|---|---|---|
| Row: 4<br>Layer: 6<br>Column: 8<br>Length: 1 m<br>Height: 0.8 m | Horizon speed: $V_{xmax} = 1.2$ m/s<br>Horizon acceleration: $a_x = 0.4$ m/s$^2$<br>Vertical speed: $V_y = 0.4$ m/s<br>Vertical acceleration: $a_y = 0.25$ m/s$^2$ | Population count: 30<br>Gene count: 192<br>Crossover: 0.4<br>Mutation: 0.1 |

Table 2: Simulation data.

| ind | $F(x)$ | Rate (s) | Dis (m) | Total | Foc (m) |
|---|---|---|---|---|---|
| 1 | 0.291076 | 4.52996 | 2.35318 | 3.87693 | 2.33338 |
| 2 | 0.352275 | 4.536 | 2.36011 | 3.88323 | 2.29361 |
| 3 | 0.358204 | 4.62787 | 2.27533 | 3.92211 | 2.40088 |
| 4 | 0.367442 | 4.60357 | 2.30125 | 3.91287 | 2.20633 |
| 5 | 0.397199 | 4.64631 | 2.12435 | 3.88972 | 2.20108 |
| 6 | 0.409661 | 4.54302 | 2.32383 | 3.87726 | 2.1698 |
| 7 | 0.452732 | 4.39787 | 2.51894 | 3.83419 | 2.18738 |
| 8 | 0.506566 | 4.41311 | 2.41126 | 3.81256 | 2.1628 |
| 9 | 0.5593 | 4.53966 | 2.03329 | 3.78775 | 2.20494 |
| 10 | 0.585234 | 4.46537 | 2.1202 | 3.76182 | 2.09798 |
| 11 | 0.597896 | 4.4522 | 2.10872 | 3.74915 | 2.11846 |
| 12 | 0.600551 | 4.38193 | 2.26382 | 3.7465 | 2.15979 |
| 13 | 0.631733 | 4.36109 | 2.20852 | 3.71532 | 2.22436 |
| 14 | 0.647764 | 4.32488 | 2.23957 | 3.69929 | 2.14683 |
| 15 | 0.668034 | 4.37444 | 2.05635 | 3.67902 | 2.09141 |
| 16 | 0.726188 | 4.32223 | 2.31069 | 3.71876 | 2.08546 |
| 17 | 0.739863 | 4.29818 | 2.34995 | 3.71371 | 1.97941 |
| 18 | 0.749038 | 4.36351 | 2.16693 | 3.70454 | 2.05676 |
| 19 | 0.764096 | 4.25177 | 2.37747 | 3.68948 | 1.93628 |
| 20 | 0.813277 | 4.2236 | 2.27926 | 3.6403 | 1.93039 |
| 21 | 0.865533 | 4.27263 | 1.99067 | 3.58804 | 1.97837 |
| 22 | 0.86788 | 4.28859 | 1.97366 | 3.59411 | 2.07889 |

In order to make the test more accurately, the parameters are defined in Table 1.

The results on simulation can be seen in Table 2.

In global, the cost time and the distance are reduced. Before optimization, the cost time of cargo storage is 4.52996 s and the distance between same kinds of goods is 2.35318 m; after optimization, the cost time of cargo storage is 4.28859 s and the distance between same kinds of goods is 1.97366 m. In the process of optimization, the focus of shelves is always under half of the height of shelves. But in this process, some point is higher than previous one, the reason is that the objective function use $f_{\max}$ to calculate $f_{\max}$ has changed in this process. As a result, the optimize solution is worse than the previous one. After optimize again, the result is getting better and better globally. The curves of the results can be seen in Figure 10.

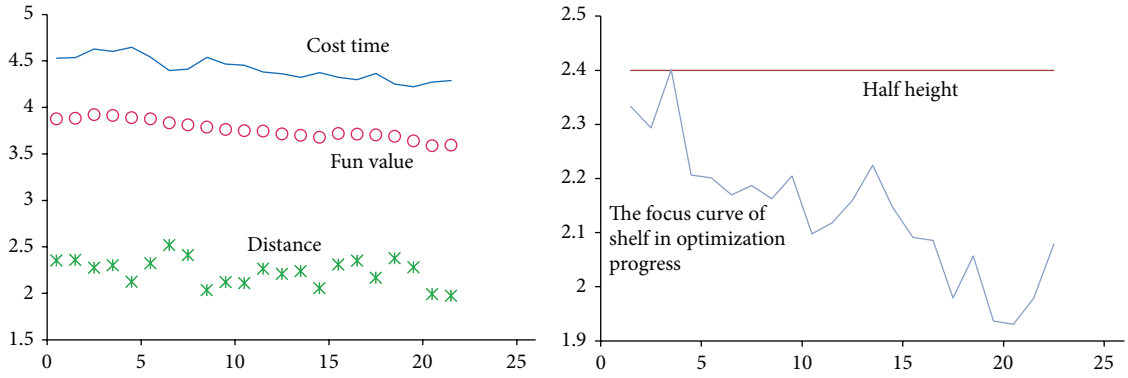



Figure 10: Result curves.

## 6. Discussion

This paper discusses the problem on how to assign a position for goods on shelf in the automated warehouse system and bring out two models: efficiency model and classification model. A genetic algorithm is used to solve and optimize the multiobjective problem. A location optimal solution should be studied and improved in the global scalability in the future.

## Acknowledgment

The researcher is supported by the national 863 program in the Ministry of Science and Technology of China. Project no. 2013AA040302.

## References

[1] G. Nikolakopoulou, S. Kortesis, A. Synefaki, and R. Kalfakakou, "Solving a vehicle routing problem by balancing the vehicles time utilization," *European Journal of Operational Research*, vol. 152, no. 2, pp. 520–527, 2004.

[2] M. Lang and S. Hu, "Study on the optimization of physical distribution routing problem by using hybrid genetic algorithm," *Journal of the Society of Management Science of China*, vol. 10, pp. 51–56, 2002.

[3] G. Zhou and M. Gen, *Journal of Engineering Design and Automation*, vol. 3, p. 157, 1997.

[4] C. A. Coello, "A comprehensive survey of evolutionary-based multiobjective optimization techniques," *Journal of Knowledge and Information Systems*, vol. 3, pp. 269–308, 1999.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 521094, 11 pages
http://dx.doi.org/10.1155/2013/521094

*Research Article*

# An Approach for Composing Services Based on Environment Ontology

**Guangjun Cai[1,2,3] and Bin Zhao[1,3]**

[1] *The Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China*
[2] *Electronics Information Engineering College, Henan University of Science and Technology, Luoyang 471023, China*
[3] *Graduate University of Chinese Academy of Sciences, Beijing 100049, China*

Correspondence should be addressed to Guangjun Cai; guangjuncai.cn@gmail.com

Received 22 May 2013; Accepted 11 July 2013

Academic Editor: Yuxin Mao



Service-oriented computing is revolutionizing the modern computing paradigms with its aim to boost software reuse and enable business agility. Under this paradigm, new services are fabricated by composing available services. The problem arises as how to effectively and efficiently compose heterogeneous services facing the high complexity of service composition. Based on environment ontology, this paper introduces a requirement-driven service composition approach. We propose the algorithms to decompose the requirement, the rules to deduct the relation between services, and the algorithm for composing service. The empirical results and the comparison with other services' composition methodologies show that this approach is feasible and efficient.

## 1. Introduction

Service-oriented computing (SOC) [1], with the standardized, cross-platform and transparency characteristics, can effectively reuse and compose the existing software and system resources. It comes to become a new milestone in distributed system, to be a mainstream computing model in heterogeneous environments, and to improve the competitive advantage in the field of e-commerce [2]. Service composition, as a method to use the existing services and generate new services, is one of the key factors of the success of the service-oriented computing [3].

Many institutions and enterprises have been currently engaged in the research of this area, and a series of standard language and method [3, 4] are published. The separation between the service description and the service implement, and the standardization of the service description and the service process provide the foundation to interoperate among services in various platforms. However, there are still many open problems need to be addressed for the reasons that the service is open, dynamic and changeable and that the composition problem with high complexity [4]. We believe that by solving the following problems the efficiency of the service composition can be improved.

(i) Lacking of effective model's support. Some methods (such as [5]) do not use the model, while others (such as [6]) are based on a service-oriented process model changed with the change of services.

(ii) Few requirements and domain knowledge. Many methods (e.g., [2]) need no constraints on the behaviour of the composition services, and the domain knowledge is generally only used in the detection of match between services (such as [7]).

(iii) Lacking support to the composition of the large-grained or precise services. For example, [8] can only composite the atomic services as a composite object.

The environment model completely independent of the service implement provides a basis to acquire a precise and on-demand requirement. Based on it, a requirement driven approach is proposed which with the requirement descriptions and the service descriptions based on environment ontology as inputs. This approach firstly reduces

Table 1: Meanings of the concepts.

| Concept | Meaning |
|---|---|
| Environment entity | Entities which a system will interact with |
| Causal entity | Entities whose properties include predictable causal relationships among its shared phenomena |
| Autonomous entity | Entities which can receive or send message autonomously usually consist of people |
| Symbolic entity | Entities which are a physical representation of data |
| Attribute | A named variable for characterizing a static property of an entity |
| Static attribute | Characterize the static properties of the environment entity |
| Dynamic attribute | Characterize the dynamic properties of the environment entity |
| Value | Intangible individual entity that exists outside time and space, and is not subject to change |
| Tree-like hierarchical state machine | A hierarchical state graph for characterizing dynamic properties of a causal entity |
| Finite state machine | A directed state graph for characterizing a dynamic property |
| Divide | Relation between the basic state machine and a state |
| State | Value of an entity at a given time |
| Transition | A state change in a state machine represent the relation between two states |
| Message | The content that is sent or received from one environment entity to another |
| Command message | Contain the action command |
| Data message | Message that has no command contains only the parameters |
| Interaction | Observable shared phenomenon between a system and its environment entity |
| Message interaction | Interaction that occurs through the messaging |
| Value interaction | Interaction that occurs through reading or assigning the value of the static attribute |

the complexity of the problem through the decomposition of the requirement in order to improve the degree of parallelism of the service composition. Then, the rules to determine the relations between the services are proposed to improve the granularity of the service. Finally, it can present the composition result in the WS-CDL [9].

The rest of this paper is structured as follows. In Section 2, we introduce environment ontology and a usage scenario and present the environment ontology-based descriptions of some services and requests. Section 3 proposes some algorithms decomposing the requirement, some rules to determine the relations between services, and the method to compose the services. Section 4 presents some implement results and analyse it. After discussing some related work in Section 5, we conclude in Section 6.

## 2. Environment Ontology and the Description Based on Environment Ontology

The idea of environmental modelling originates from requirements engineering. Reference [10] pointed out that the semantics of software systems is in the environment of the software rather than in the software itself. Then, [11] introduces it in service-oriented computing and uses it as the basis to characterize the services and the user requirement.

*2.1. Environment Ontology.* The environment ontology describes the nature of the entity of the service problem, as well as its possible changes, the process of the change, the conditions, and the result of the change. Environment the service entities interact with consists of various environment entities. Environment entities, either concrete or abstract, shared the same semantic with all the relevant service and the user requirements rather than relying on a service or requirement. The top-level concepts are shown in Table 1.

The various types of environmental entities require a different form of the description except for static properties and the interaction. Causal entities have a description of the dynamic properties in the form of the tree-like hierarchical state machines; symbolic entities and autonomous entities are with a data set and a set of events, respectively. Part of the concept is defined as follows.

*Define 1.* Finite state machine (Fsm) describing the changes in one causal entity is a four-tuple ⟨States, Trans, init, fin⟩, where *States* is a set of the state; *Trans*, a set of transitions, is defined as $\{\langle ss, mess, ts\rangle \mid ss, ts \in$ States, *mess* denote the message set$\}$; *Mess*, a set of messages, is defined as $\{\langle dir, mes\rangle \mid dir = \downarrow|\uparrow$, *mes* is a data message *data* or a command message *event*$\}$, Finally, $init, fin \in States$, is the initial state and the target state, respectively.

*Define 2.* Tree-like hierarchical state machine (Hsm), to describe the state change of an individual controllable entity, is a triple ⟨Fsms, Divs, rootfsm⟩, where *Fsms* is a set of *fsm*; $Divs = \{\langle s, fsm\rangle \mid s$ is a state, $s \notin fsm \wedge s \in hsm\}$, where *s* is called the parent state of *fsm*; $rootfsm \in Fsms$.

*Define 3.* The effect of the environment entity denoted *Eff*, describe the changes of the environmental entity, is defined as $Events \mid Datas \mid Hsm$, where $Eff := Events := \{\langle dir, event\rangle \mid dir = \downarrow|\uparrow$, *event* is a command message$\}$ when the entity is an autonomous entity; $Eff := Datas := \{\langle dir, data\rangle \mid dir = \downarrow|\uparrow$, *data* is a data message$\}$ when the entity is a symbolic entity; $Eff := Hsm$ when the entity is a causal entity.

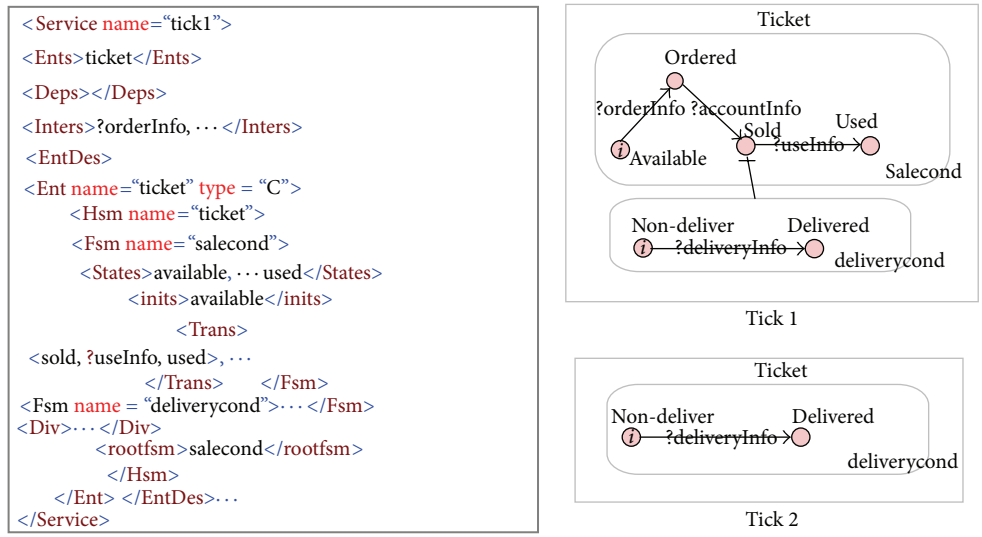



Figure 1: The description of the available service based on environment ontology.

*Define 4.* Environmental entity (EnvEnt) defined as a four-tuple ⟨Name, Type, AttrSet, Eff⟩, where; *Name* is the name of the environment entity; Type := "S" | "C" | "A" represents the environment entity, respectively, representing a symbolic entity, the causal entity, and autonomous entity; *AttrSet* := {⟨attrn, attrv⟩ | *attrn* is an attribute name and *attrv* is an attribute value}; *Eff* describes the change of the environment entity.

### 2.2. Describing the Service Based on Environment Ontology

*Define 5.* The service description based on environment ontology characterizes the semantics of the services through the physical changes the service causes. It can be defined as ⟨name, Ents, Inters, Deps, EntDes, Sers, and Funcs⟩, where *name* is a service name; *Ents* is a set of the environment entities; *Inters* is a set of interactions and it is a subset of the messages; $Deps = \{\langle se, inter, te\rangle \mid se, te \in Ents$ is, respectively, the source entity and the target entity; $inter \in Inters\}$ is generated by the effect of the service between the environment entities; $EntDes := \{e' \mid e'$ describes part of one environment entity $e \in Ents\}$ is the description of the environmental entity under the effect of the service; $Sers = \{ser \mid ser$ is a service$\}$ is a set of services; $Funcs := \{\langle func, ser\rangle \mid func \in Ents \cup Hsms \cup Fsms \cup Trans \cup Datas \cup Events, ser \in Sers\}$ is a set of functions describing the relation between each functionality and the services provide it.

The process of the service description is a process projecting service functionality to the environment entities. The input of the process consists of the Ents, Inters, and the environment ontology; the output of the process consists of Deps and EntDes. The specific algorithm can see [11, 17]. Figure 1 lists the text description and the graphics description of the two ticketing services.

### 2.3. Describing the Requirement Based on Environment Ontology

*Define 6.* The requirement description based on environment ontology can be seen as a service description without the service's implementation. It is defined as a quintuple ⟨name, Ents, Inters, Deps, EntDes⟩, where the meaning of the elements is the same with that in Define 5.

The process to describe the requirement is similar to that which depicts services. There are two results when the requirement is described: the requirement is described with only one option, or description with more than one option. The latter case needs more user constrain. Figure 2 presents a specific illustration of the requirement, wherein the text description is presented in (a) and the effect of changes of the ticket in (b).

## 3. Composing the Services Based on Environment Ontology

The task of web service composition is to search appropriate service and to arrange them in an appropriate manner. For the requirements which are depicted in the same level with the composition result, we present a method through decomposing the requirement to search the available service based on the service discovery method in [11]. The task is divided into three parts: decomposing the requirement, determining the relation between the services, and generating the composition service.

*3.1. Decomposing the Requirement.* The nature of the composition problem is that one problem cannot be addressed by a single service. Furthermore, our method is based on a requirement described in a formal detailed determine way.

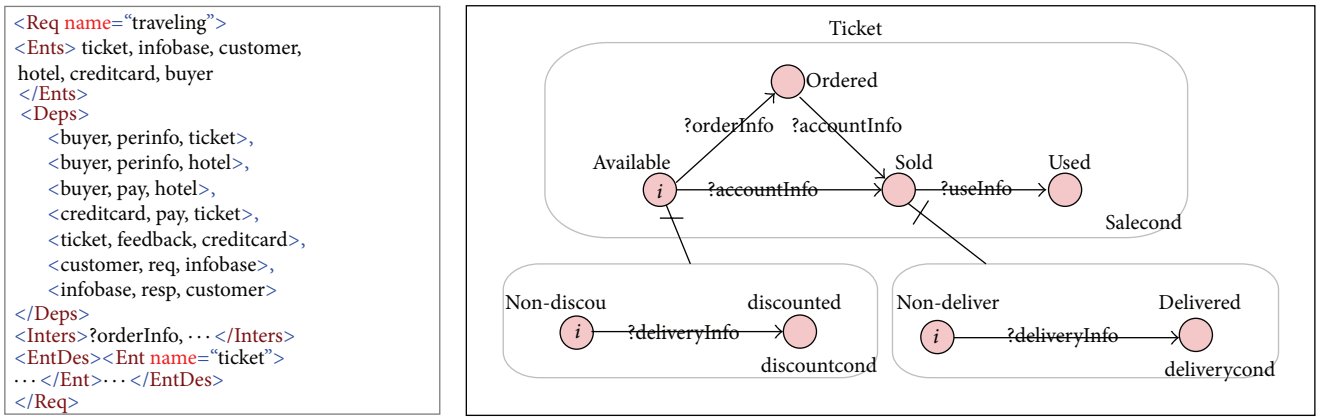



Figure 2: The requirement description based on environment ontology.

Thus we can acquire a composite service through a way which firstly decomposes the request into various parts and then composes services according to the requirement.

*3.1.1. Decomposing the Requirement Involving Multienvironment Entities.* In this section, we decompose the requirement among environment entities and do not consider the internal description of the environment entity. The type of dependency among the environment entities can be classified into four classes: no dependence, dependent on the environment entities in one direction, and dependent on a symbol entity or autonomous entity, and dependent on each other. The corresponding decomposition algorithms are shown in Algorithms 1, 2, 3, and 4.

The elements in *ReqSet* produced by Algorithm 1 have no dependence on other entities. Algorithm 1 satisfies consistency and has order of $n^2$ time complexity where $n$ is the number of entities in the requirement. Algorithm 2 decompose the requirement according to whether the interaction between the entities is in one direction. When multiple environment entities meet the conditions of the decomposition, the consistency can only satisfy all the decompositions with the same order. Its time complexity is the same with Algorithm 1. Algorithm 3 decomposes the requirement according to the type of the entity and has the same consistency and time complexity with Algorithm 2. Algorithm 4 is used when there is only message dependence among different entities. It can meet the consistency and its time complexity is equal to $O(n)$ where $n$ is the number of entities. Taking the requirement in Figure 2(a) as the input to present how to use these algorithms. The decomposition results are listed in Table 2.

*3.1.2. Decomposing the Requirement Involving One Environment Entity.* This section decomposes the requirement in one environment entity. Autonomous entity or symbolic entity having no constrains on its interaction can be directly divided into individual event or data by Algorithm 7, and no further discussion is required. But in a causal entity, which is specified as a tree-like hierarchal state machine, there exists the dependence between a state and a tree-like hierarchal state machine, the dependence between a state and a finite state machine, and the dependence between states. The decomposition algorithms are shown in Algorithms 5, 6, and 7.

Algorithm 5 decomposes the requirement containing a single *hsm* based on the couple between the finite state machines with the same state as an ancestor greater than the couple between the finite state machines with the different state as ancestor. The result of it is a set of the requirement each involving one *hsm*, and it can be called repeatedly several times until each includes only one *fsm*. Algorithm 5 satisfies consistency and has the time complexity $O(f)$, where $f$, less than or equal to the number of *fsm* in the *hsm,* is the number of the requirements the decomposition produces.

Algorithm 6 decomposes the requirement containing a *fsm* based on the key state into more than one *fsm*. The key state can be specified by the user, the initial state of *fsm*, and the termination state of *fsm* or the parent state of the finite state machine contain one key state. It satisfies consistency and has the time complexity $O(m^2)$, where $m$ is the number of states in the *fsm*.

Algorithm 7 can be used to decompose the requirement containing a single autonomous entity, a single symbolic entity, or an individual *fsm* based on the basic functionality unit of the requirement. It satisfies consistency and has the time complexity $O(k)$, where $k$ is the number of the basic functionalities.

Taking the requirement in Figure 2(b) as the input to present how to use Algorithms 5–7. The decomposition results are listed in Table 3.

*3.2. Determining the Relations of the Services.* The second task of web services composition is to determine the relations among services so that they can collaborate with each other for satisfying the request. To determine the relations among services is to derive the relationship between the various services based on the relations of the functionalities in the

```
DecByDep(Req, ReqSet)    //input Req, output ReqSet
(1)  while(Req.Ents ≠ ∅)
(2)     e:= get (Req.Ents);
(3)     if(Relate(e) ≠ Req.Ents)//Relate(e) refer to a set include e and the entities depend on e
(4)        Set_e:= create (Relate(e)); //create a set include the elements in Relate(e)
(5)        ReqSet:= ReqSet ∪ create(Set_e, Req); //create(Set_e, Req) is to acquire a requirement corresponding to the Set
(6)        Req:= Req.Remove(Set_e); //Removing a requirement corresponding to the Set from Req
(7)     end if
(8)  end while
(9)  ReqSet:= ReqSet ∪ Req;
```

Algorithm 1: Decomposition algorithm among unrelated environmental entities.

```
DecByDepDir(Req, ReqSet)
(1)  if(∃e((e ∈ Req.Ents)∧∀dep(des ∈ Req.Deps∧(dep.se ≠ e∨dep.te ≠ e))))
(2)     ReqSet:= create (e, Req);
(3)     ReqSet:= ReqSet ∪ DecByDep (Req.Ents-{e});
(4)  end if
```

Algorithm 2: Decomposition algorithm according to the dependent type between environmental entities.

```
DecByEntTyp(Req, ReqSet)
(1)  if(∃e(e ∈ Req.Ents∧(type(e) = A∨type(e) = S)))
(2)     ReqSet:= create(e, Req);
(3)     ReqSet:= ReqSet∪DecByDep(Req.Ents-{e});
(4)  end if
```

Algorithm 3: Decomposition algorithm according to the type of environmental entities.

requirement description as well as the relations between the functionalities and services. The rules are given in this section in two layers: to determine the relations between the services among multienvironment entities and to determine the relations between services in one environment entity. They will help to use the service in more appropriate granularity, to reduce complexity of the external control flow, and to reduce the number of services in composition services and the communications overhead between the services.

*3.2.1. Determine the Service Relations among Multiple Environment Entities.* If the requirement involving multienvironment entities needs to be accomplished by several services instead of one available service, we need to identify the relations between the services.

Let *inter* and *inters* be, respectively, an interaction and a set of the interactions between environment entities and *type(inter)* the type of *inter*, whose value could be *msg* or *val*. *R(inter)* and *S(inter)*, respectively, represent the environment entity receiving *inter* and that sending *inter*. |*inters*| and |*con(inters)*| represent the number of interactions in *inters* and the number of the messages in *inters*, respectively. *Type(ent)* denotes the type of the environment entity *ent*. Determining the relations between the services involving multienvironment entities are shown in Box 1.

Rule 1 shows that when the multiple value interactions act on the same property of the same environment entity and wherein the received value interaction, the order of the value interactions is subject to the constraints of the received value interaction. Rule 2 shows that when the multiple message interactions act on one causal entity, the order of the message interactions is subject to the constraints of the causal entity. Rule 3 shows that when the multiple message interactions act on one autonomous entity, the order of the message interactions is subject to the constraints of the autonomous entity. Rule 4 shows that when the multiple message interactions act on one symbol entity and wherein one received interaction, the order of the message interactions is subject to the constraints of the received interaction. The time complexity of these rules is up to $O(h^2)$, where $h$ is the number of interactions.

Back to the example in Figure 2(a), there is no interaction set to meet Rules 1 and 4. The result produced by Rule 2 is {⟨{⟨creditcard, pay, ticket⟩, ⟨ticket, feedback, creditcard⟩}, ticket⟩, ⟨{⟨creditcard, pay, ticket⟩, ⟨ticket, feedback, creditard⟩}, creditcard⟩}. The result produced by Rule 3 is {⟨{⟨customer, req, infobase⟩, ⟨infobase, resp, customer⟩}, customer⟩}.

*3.2.2. Determine the Service Relations in One Environment Entity.* The task of this section focused on the causal entity described by the tree-like hierarchical state machine. We need to determine the relations among the functionalities which associate with the services acting on one causal entity. If the relation is between the services, we need to divide the functionality in some time. If the relation is in a service, we need to merge the functionalities into one service. According to the structure of the causal entity, this task can be divided

```
DecByEnt(Req, ReqSet)
(1)  for(each e in Req.Ents){ // the number of environmental entities in Req.Ent greater than 1
(2)      ReqSet:= ReqSet ∪ create(e, Req);
(3)  end for
```

Algorithm 4: Decomposition algorithm among the various environmental entities.

Table 2: Decomposition result of each step by Algorithms 1–4.

| Algorithm | Decomposition result |
|---|---|
| Algorithm 1 | {infobase, customer}, {hotel, creditcard, ticket, buyer} |
| Algorithm 2 | {infobase, customer}, {buyer}, {hotel}, {creditcard, ticket} |
| Algorithm 3 | {infobase}, {customer}, {buyer}, {hotel}, {creditcard, ticket} |
| Algorithm 4 | {infobase}, {customer}, {buyer}, {hotel}, {creditcard}, {ticket} |

```
DecByHsmStr(Req, ReqSet) // input Req, which can be represented by a Tree-like hierarchal state machine(thsm), output ReqSet
(1)  rootfsm:= GetRootfsm(Req);
(2)  for(each subhsm of rootfsm)//subhsm.rootfsm.superstate ∈ rootfsm
(3)      ReqSet:= ReqSet ∪ create(subhsm, Req);
(4)  end for
(5)  ReqSet:= ReqSet ∪ create(rootfsm, Req);
```

Algorithm 5: Decomposition algorithm based on the structure of the *hsm*.

```
DecByState(Req, ks, ReqSet)// input Req which can be represented by a finite state machine and ks, output ReqSet
(1)  if((Req.State-(From(ks) ∪ To(ks))) = ∅) //Req.State is a set of the states in Req
(2)      From(ks):= From(ks)-{ks}; //From(ks) is a set of the states which ks can reach
(3)      To(ks):= To(ks)-{ks}; //To(ks) is a set of the states which can reach ks
(4)  end if
(5)  Req1 = create(Req.State-(From(ks) ∪ To(ks)), Req);
(6)  Req2:= create(From(ks)-To(ks), Req);
(7)  Req3:= create(To(ks)-From(ks), Req);
(8)  Req4:= create(From(ks) ∩ To(ks), Req);
(9)  ReqSet:= Req1 ∪ Req2 ∪ Req3 ∪ Req4;
```

Algorithm 6: Decomposition algorithm based on the key state.

```
DivByBeha(Req, ReqSet)// input Req which can a autonomous entity, a symbolic entity or a fsm, output ReqSet
(1)  for(each b in Req)// b is the basic unit of the functionality, which can be a data, an event or transition
(2)      ReqSet:= create({b}, Req);
(3)  end for
```

Algorithm 7: Decomposition algorithm based on the basic unit of the functionality.

Table 3: Decomposition result of each step by the decomposition Algorithms 5–7.

| Algorithm | Inputs | Decomposition result |
|---|---|---|
| Algorithm 5 | Figure 2(b) | {salecond}, {discountcond}, {deliverycond} |
| Algorithm 6 | Salecond in Figure 2(b), ordered | {<available,. . .,ordered>, <ordered,. . .,sold>, <available,. . .,sold>}, {<sold,. . .,used>} |
| Algorithm 7 | Salecond in Figure 2(b), | {<available,. . .,ordered>}, {<ordered,. . .,sold>}, {<available,. . .,sold>}, {<sold,. . .,used>} |

Rule 1: (|inters|>1) ∧ (type(inters)=val) ∧ ∀inter((inter∈inters) → (R(inter)=ent ∨ S(inter)=ent)) ∧ (|con(inters)|=1) ∧ ∃inter ((inter∈inters) ∧ R(inter)=ent) => constrain(inters,inter) ∧ (R(inter)=ent).
Rule 2: (|inters|>1) ∧ (type(inters)=msg) ∧ ∀inter((inter∈inters) → (R(inter)=ent ∨ S(inter)=ent) ∧ (type(ent)=C)) => constrain (inters,ent).
Rule 3: (|inters|>1) ∧ (type(inters)=msg) ∧ ∀inter((inter∈inters) → (R(inter)=ent ∨ S(inter)=ent) ∧ (type(Ent)=A)) => constrain (inters, Ent).
Rule 4: (|inters|>1) ∧ (type(inters)=msg) ∧ ∀inter((inter∈inters) → (R(inter)=ent ∨ S(inter)=ent)) ∧ (type(ent)=S)) ∧ (|con(inters)| =1) ∧ ∃inter((inter∈inters) ∧ R(inter)=ent) => constrain(inters, inter) ∧ (R(inter)=ent).

Box 1

Rule 5: <s,fsm>∈Divs ∧ Sers(Pres(s))=(Sers(Sucs(s))) ∧ |Sers(Sucs(s))|=1 ∧ (Sers(Sucs(init(fsm))) ≠ Sers(Pres(s)) ∨ Sers (Pres(fin (fsm))) ≠ Sers(Sucs(s))) => Divide(Sers(Sucs(s)), s).
Rule 6: <s, fsm>∈Divs ∧ Sers(Sucs(init(fsm))) ≠ Sers(Pres(s)) => Divide(Sers(Sucs(init(fsm)))∩Sers(Pres(s)),fsm); <s,fsm>∈Divs ∧ Sers(Pres(fin(fsm))) ≠ Sers(Sucs(s)) => Divide(Sers(Pres (fin (fsm)))∩Sers(Sucs(s)), fsm).
Rule 7: <s,fsm>∈Divs ∧ Sers(Pres(s))=Sers(Sucs(init(fsm))) ∧ |Sers(Pres (s))|=1 => Merge(Pres(s), Sucs(init(fsm))); <s,fsm>∈Divs ∧ Sers (Sucs(s))=Sers(Pres(fin(fsm))) ∧ Sers(Sucs(s))|=1 => Merge (Sucs(s), Pres (fin(fsm))).
Rule 8: Sers(Pres(s))=Sers(fsm) ∧ |Sers(Pres(s))|=1 ∧ ∀fsm(<s, fsm>∈Divs → (Sers(fsm)=Sers(Pres(s)))) => Merge(Pres(s),fsm); (Sers (Sucs(s))=Sers(fsm)) ∧ |Sers(Sucs(s))|=1 ∧ ∀fsm(<s,fsm>∈Divs → (Sers(fsm) =Sers (Sucs(s)))) => Merge (Sucs(s), fsm).

Box 2

Rule 9: (|Pres(s)|=|Sucs(s)|=1) ∧ (Ser(Pres(s))=Ser(Sucs(s)))) => Merge(Pres(s),Sucs(s)).
Rule 10: BranchEntry(b) ∧ BranchExit(e) ∧ (source(t) =source(t′)=b) ∧ (target(t) =target(t′)=e) ∧ (Ser(t)=Ser(t′)) => Merge(t, t′).
Rule 11: BranchEntry(b) ∧ |Sers(Sucs(b))|=1 => Move(Sers (Sucs(b)), b).
Rule 12: (|Pres(s)| >1 ∨ (|Sucs(s)| >1)) ∧ (Ser(Sucs(s))∩Ser(Pres(s)) ≠ ∅) ∧ (|Sers(Sucs(s))| >1 ∨ (|Ser(Pres(s))| >1)) => Divide (Ser(Sucs(s))∩Ser(Pres(s)),s).
Rule 13: s∈From(s) ∧ t∈Pres(s) ∧ source(t)∉From(source(t)) ∧ t′∈Pres(s) ∧ source(t′)∈From(source(t′)) ∧ Ser(t)=Ser(t′) => Divide (Ser (t), s).
Rule 14: s∈From(s) ∧ ((t∈Sucs(s)∧target(t)∈From(target (t))) ∧ (t′∈Sucs(s) ∧ target(t′)∉From(target(t))) ∧ (Ser(t) =Ser(t′)) => Divide (Ser(t),s).

Box 3

into the transformation between the finite state machine and the transformation in a finite state machine.

Let $s$ a state, $t$ and $t'$ a transition, *Pres(*s*)* a set of transitions which ending with $s$, and *Sucs(*s*)* a set of transitions which starting from $s$. *Ser(*t*)* and *Sers(fsm),* respectively, represent $\langle t, ser\rangle \in funcs$ and $\langle fsm, ser\rangle \in funcs$. $|set|$ indicates the number of elements in the *set*, such as $|Pres(s)|$ which is the number of the transitions in *Pres(*s*)*. *init(fsm)* and *fin(fsm),* respectively, denote the initial state and the final state of the *fsm,* and *source(t)* and *target(t)* represent the source state and the target state of the transition $t$. $Merge(t, t')$ denotes merging the two transitions, *merge(t, fsm)* denotes merging $t$ and *fsm*, *divide(ser,s)* represents dividing the *ser* by s, *divide(ser, fsm)* represents dividing the *ser* on the boundary of *fsm*, *move(ser, b)* represents move of the $b$ to the end of *ser*.

*(1) The Transformation between the Finite State Machines*. The transformation rules determining the relations of the services acting on a tree-like hierarchical state machine are shown in Box 2.

Rules 5 and 6 are used to determine, respectively, whether the service needs to be divided by the parent state of a finite state machine and whether the service acting on several finite state machines needs to be divided by the boundary of the finite state machine. Rules 7 and 8 can be used to merge the services acting on several finite state machines. All the rules of this section can be completed in polynomial time $n$ which equals the number of *states* plus the number of *transitions* contained in the relevant *fsm*.

*(2) The Transformation in a Finite State Machine*. When a finite state machine associates with several services, we determine the relations between these services. Thus, our task in this section is to conclude the relations between the services according to the relation between transitions and the relation between the transition and the services. Before introducing the transform rules, we give some definitions firstly.

*Define 7*. The branch entry point is denoted as $BranchEntry(s)$iff $(|Sucs(s)| > 1) \wedge (|Pres(s)| = 1) \wedge (s \notin To(s))$.

```
Input: Req //the requirement description based on environment ontology
Output: Ser //the composition service
(1)  ReqSet:= {Req}; //ReqSet is a set of the requirement needing to be implemented by the services;
(2)  while(ReqSet ≠ ∅)
(3)    for(each req in ReqSet)
(4)      ser:= discovery(req, SerSet);
(5)      if(ser ≠ ∅){//discovery(req, SerSet) successes
(6)        LabReq:= LabReq ∪ label(req, ser); //label the req using the ser
(7)        continue;
(8)      end if
(9)      ReqSet:= ReqSet ∪ dec(req, rule)-{req}; // dec() is used to decomposition the requirement
(10)   end for
(11) end while
(12) Ser.{name, Ents, Deps, EntDes}:= req{name, Ents, Deps, EntDes}
(13) determin the Sers and Funcs of the Ser according to the req and LabReq;
(14) for(each ent in LabReq.Ent)
(15)   if(ent.type = C&&|Sers(ent)| > 1)
(16)     for(each fsm in rhsm)
(17)       if(|Sers(fsm)| > 1&&hasproced(subFsm(fsm)))
(18)         adjust the Sers and Funcs of the Ser using the rule 9–14;
(19)         adjust the Sers and Funcs of the Ser using the rule 5–8;
(20)       end if
(21)     end for
(22)   end if
(23)   adjust the Sers and Funcs of the Ser using the rule 1–4;
(24) end for
(25) generateCDL(Ser);//generate the description in WS-CDL of the ser
```

Algorithm 8: Composition algorithm based on environment ontology.

*Define 8*. The branch exit point is denoted as $BranchEixt(e)$ if and only if the state $e$ satisfies $BranchEntry(b) \wedge ((From(b)\text{-}(To(e) \cup From(e))) = \varnothing) \wedge (\forall s(s \in (To(e) \cap From(b)) \rightarrow ((From(b)\text{-}(To(s) \cup From(s))) \neq \varnothing)) \wedge (e \notin From(e))$.

The rules are shown in Box 3. Rule 9 orients the sequence structure mergers the services in a state which has a single pervious transition and a single successor transition. Rules 10, 11, and 12 orient the choice structure, in which Rules 10 and 11 are used to merge the branches, and the Rule 12 is used to divide the service. Finally, Rules 13 and 14 orient the loop structure, in which Rule 13 is used at the entrance of the loop and Rule 14 is used at the exit of loop when one service acts on both in the loop and outside the loop. All of them belong to the polynomial time complexity class.

*3.3. Generating the Composition Service.* The algorithm generating the composition service is shown in Algorithm 8. It consists of two main phases. The first phase decomposes the requirement and discovery of the corresponding services in lines 1–11. The discovery process can be accomplished by the method proposed in [11] or some other service discovery or composition method. The second constructs the composition service and adjusts the relation between the services in lines 12–25.

```
<Service name="traveling">
  <Ents>ticket, hotel</Ents>
  <Deps><buyers, perinfo, ticket>,··· </Deps>
  <Inters>?orderInfo, ··· </Inters>
  <EntDes>
    <Ent name="ticket" type="C">··· </Ent>···
  </EntDes>
  <Sers>tick41, tick42, tick2, tick3,··· </Sers>
  <Funcs><Func> <Ent name="ticket"> <Hsm name="ticket">
        <Fsm name="salecond">
      <Tran> <available, {?orderInfo, ··· }, sold> </Tran>
    </Fsm> </Hsm> </Ent>
  <Sers>tick41</Sers>
  </Func></Funcs>···
</Service>
```

Figure 3: Part of the description of the composition service.

As that analyzed in previous part, the time complexity of Algorithms 1–7 and the Rules 1–14 belongs a polynomial complexity. In Algorithm 8, the execution number of Algorithms 1–4 does not exceed the number of the environmental entities in the requirement, the execution number of Algorithms 5–7 does not exceed the total number of the basic behaviors, the execution number of the Rules 1–4 does not exceed the total number of the interaction, and the execution number of Rules 5–14 does not exceed the number of states. Thus, Algorithm 8 is a polynomial complexity algorithm. Part of the description of the composition service generated from the requirement in Figure 2 is shown in Figure 3.

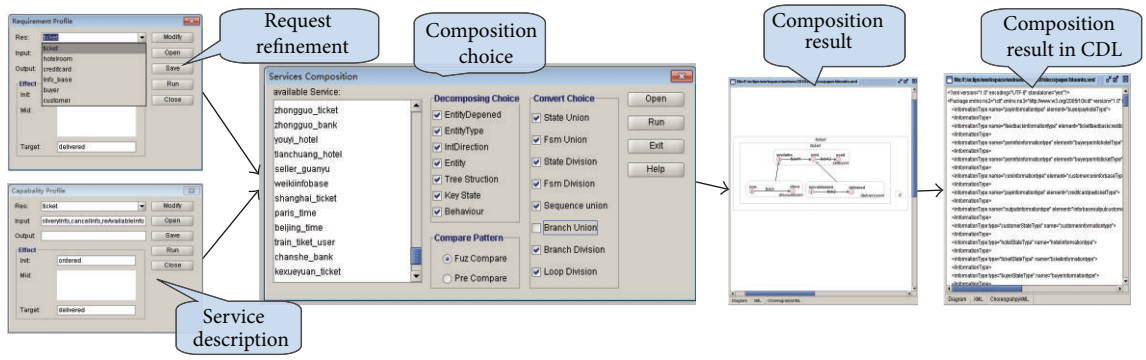
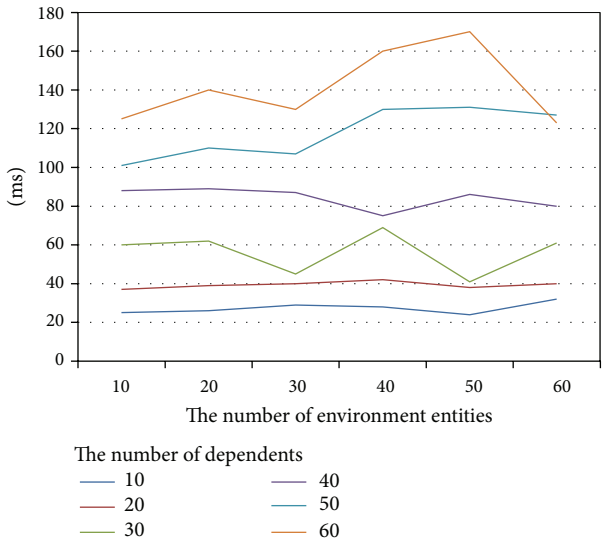
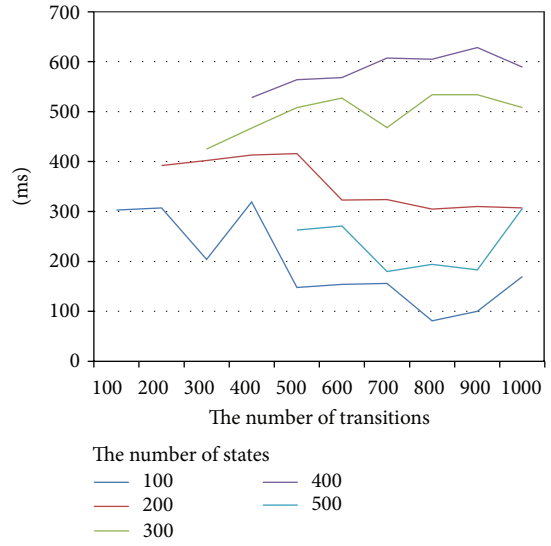


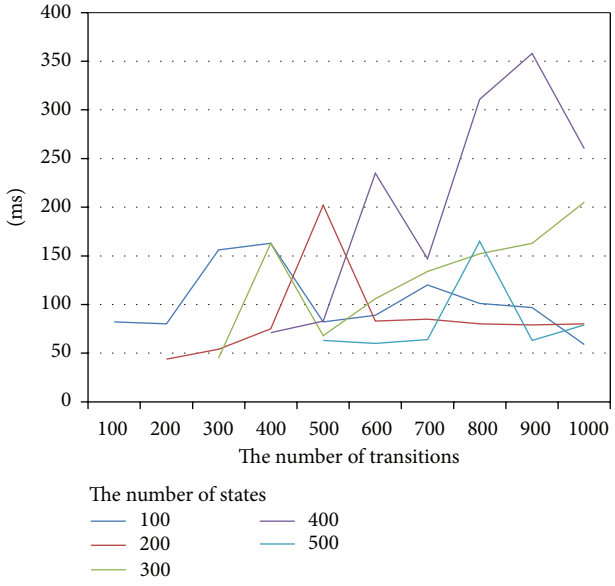

Figure 4: The main interface and the composition result of a prototype.



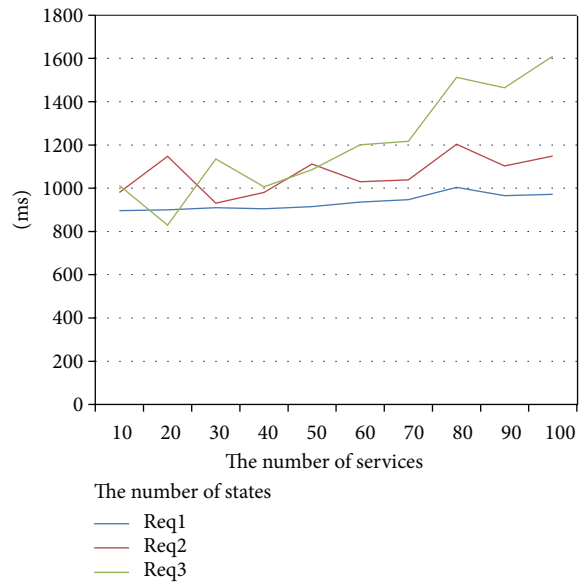

(a) The time to decompose the requirement among environment entities




(b) The time to decompose the requirement in an environment entity




(c) The time to transform the service model




(d) The time to compose services

Figure 5: The figure of composition time.

Table 4: Comparison of various approaches to service composition.

| Approach | Service description | Requirement description | Description object | Service type | Approaches | Others | Complexity |
|---|---|---|---|---|---|---|---|
| Berardi et al. [12] | Behavior | B | Service | B, A | Behavior equivalence | Single community | Exponential |
| McIlraith and Son [2] | IOPEB | OE | — | A | Agent | User constrain | — |
| Bultan et al. [13] | B | B | Conversation | B, A | Behavior equivalence | — | — |
| Roman et al. [14] | IOPEB | IOPEB | Service, goal | — | Mediator | Domain ontology | Exponential |
| Maamar et al. [5] | IOB | O | Context | A | Agent | Multilevel context | — |
| Sirin et al. [8] | IOPE | OE | — | A | Complex service-based decomposition | Complex and simple service | — |
| Brogi et al. [6] | IOB | O | — | C, A | Graph constructing and coloring | — | Exponential |
| Arpinar et al. [7] | IO | IO | Domain, service and process | A | Matching and searching | — | — |
| Liang et al. [15] | Qos | Qos | Service | A | Matching | Constrain | Polynomial |
| Tao et al. [16] | IO, Qos | IO, Qos | Service, task | A | Decomposition and discovery | Qos | Polynomial |
| Cai [17, 18] | IOPEB | IOPEB | Environment | A, B, C | Decomposition and transformation | Environment ontology | Polynomial |

## 4. The Implementation and Analysis

To achieve the service composition described in this paper, the six modules need to be implemented. The first one is used to refine the request into the requirement based on the environment ontology. The second is used to describe a service based on the environment ontology. The third is to decompose the requirement. The fourth is used to discovery and matching of the service. The fifth is to determine the relationship between the services, and the final is to generate the composition result described in the appropriate manner. The main interface and the composition result of a prototype we implemented in java are shown in Figure 4.

The required times the algorithms run on our prototype platform are shown in Figure 5. It is consistent with our previous analysis. The reasons the proposed method could improve the efficiency of the composition can be concluded as follows.

(i) The first one is the enrichment of the composition knowledge introduced by the description of the requirement and the service based on environment ontology. The former limits the position of the service to be used, while the latter can help to eliminate some paths that are just possible in theory but not allowed by the domain knowledge or the user.

(ii) The second lies in the hierarchal structure. The position and the order to decompose the requirement to some extent determine the location and the priority to composite the services. The decomposition also helps to improve degree of parallelism.

(iii) Thirdly, this approach can support the composition of the composite service and behavior services.

Besides these, this work contributes to expand the range of the service description into the problem supporting to introduce the more precise Qos constrain in the future work, and wherein some work can be used to other domains such as the rules determining the relationship between services which can be used for processing control flow.

## 5. Related Work

This section briefly discusses the relationships between our works with the existing service composition approaches. The differences between ours with others are illustrated in Table 4, where I, O, P, E, B and "—" denote input, output, precondition, effect, behavior, and unspecified content explicitly, respectively. Instead of focusing on the description of Web services of their own, we give attention to the effects imposed by the services on their environment entities and state that all the capabilities are based on the environment entities, whose characteristics and interconnections are observable and applicable during service discovery and composition. And for the character of requirement decomposition and discovery, the approach can adapt for composing different type services by changing the discovery constrains.

## 6. Conclusion and Future Work

In order to solve the problems of the service composition efficient in the single problem domain, this paper proposes the methods to generate requirement, to describe the service, and to compose the service based on environment ontology. Compared with the existing efforts in this field, this work advances the state of art in the following aspects.

(i) More efficient composition: the domain knowledge described in environment ontology and the method to decompose the requirement not only can reduce the size of the problem, but also help to improve the degree of parallelism.

(ii) More types of services composition: the method to determine the service relationship according to the problem model not only can support a composition of atomic services but also support the composition of the composite services and the behavior services.

(iii) Optimized composition model: the method adjusting the relation between the services based on the composition structure provides the foundation to schedule localization and optimizes the execution model.

In addition, the prototype implementation can generate a composition result described by the WS-CDL. This paper presents an ongoing work for tackling the issue of automatic service composition. Subsequent work will extend the ontology for supporting the service composition in various domains and the multilevel problem. And then we will enhance the service composition procedure for considering the nonfunctional concerns and the correctness of the composite services. Moreover, as stated in [1], the transaction and security must be considered if we would use it in reality. Finally, some other application domains will be researched to use this method.

## Acknowledgments

This work is partially supported by the National Natural Science Fund for Distinguished Young Scholars of China under Grant no. 60625204, the Key Project of National Natural Science Foundation of China under Grant nos. 60736015 and 90818026, and the National 973 Fundamental Research and Development Program of China under Grant no. 2009CB320701.

## References

[1] M. P. Papazoglou and D. Georgakopoulos, "Service-oriented computing," *Communications of the ACM*, vol. 46, no. 10, pp. 25–29, 2003.

[2] S. McIlraith and T. C. Son, "Adapting Golog for composition of semantic Web services," in *Proceedings of the 8th International Conference on Knowledge Representation and Reasoning*, pp. 482–496, Toulouse, France, 2002.

[3] N. Milanovic and M. Malek, "Current solutions for Web service composition," *IEEE Internet Computing*, vol. 8, no. 6, pp. 51–59, 2004.

[4] R. Jinghai and S. Xiaomeng, "A survey of automated Web service composition methods," in *Semantic Web Services and Web Process Composition*, pp. 43–54, Springer, Heidelberg, Germany, 2005.

[5] Z. Maamar, S. K. Mostéfaoui, and H. Yahyaoui, "Toward an agent-based and context-oriented approach for Web services composition," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 5, pp. 686–697, 2005.

[6] A. Brogi, S. Corfini, and R. Popescu, "Semantics-based composition-oriented discovery of Web services," *ACM Transactions on Internet Technology*, vol. 8, no. 4, pp. 1–39, 2008.

[7] I. B. Arpinar, R. Zhang, B. Aleman-Meza, and A. Maduko, "Ontology-driven Web services composition platform," *Information Systems and e-Business Management*, vol. 3, no. 2, pp. 175–199, 2005.

[8] E. Sirin, B. Parsia, D. Wu, J. Handler, and D. Nau, "HTN planning for Web service composition using SHOP2," *Web Semantics*, vol. 1, no. 4, pp. 377–396, 2004.

[9] N. Kavantzas, D. Burdett, G. Ritzinger, T. Fletcher, Y. Lafon, and C. Barreto, Web Services Choreography Description Language Version 1.0, 2005, http://www.w3.org/TR/ws-cdl-10/.

[10] M. Jackson, *Problem Frames: Analyzing and Structuring Software Development Problems*, Addison-Wesley, 2001.

[11] P. Wang, Z. Jin, L. Liu, and G. Cai, "Building toward capability specifications of Web services based on an environment ontology," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 4, pp. 547–561, 2008.

[12] D. Berardi, D. Calvanese, G. D. Giacomo et al., "Automatic composition of eservices that export their behavior," in *Proceedings of the 1st International Conference on Service-Oriented Computing (ICSOC '03)*, pp. 43–58, Springer, Trento, Italy, 2003.

[13] T. Bultan, X. Fu, R. Hull, and S. Jianwen, "Conversation specification: a new approach to design and analysis of e-service composition," in *Proceedings of the 12th International Conference on World Wide Web*, pp. 403–410, 2003.

[14] D. Roman, J. Scicluna, C. Feier, D. Fensel, A. Polleres, and J. de Bruijn, "Ontology-based Choreography and Orchestration of WSMO Services," WSMO Final Draft, 2005, http://www.wsmo.org/TR/d14/v0.4/.

[15] Z. Liang, H. Zou, F. Yang, and R. Lin, "A hybrid approach for the multi-constraint Web service selection problem in Web service composition," *Journal of Information & Computational Science*, vol. 9, no. 13, pp. 3771–3781, 2012.

[16] F. Tao, L. Zhang, K. Lu, and D. Zhao, "Research on manufacturing grid resource service optimal-selection and composition framework," *Enterprise Information Systems*, vol. 6, no. 2, pp. 237–264, 2012.

[17] G. Cai, "Web service composition on the environment level," in *Proceedings of the 6th International Conference on Semantics, Knowledge and Grid*, pp. 243–250, 2010.

[18] G. Cai, "Requirement driven service composition: an ontology-based approach," in *Proceedings of the 6th International Conference on Intelligent Information Processing*, pp. 16–25, 2010.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 542618, 6 pages
http://dx.doi.org/10.1155/2013/542618

*Research Article*

# Evaluation of the City Emergency Capacity Based on the Evidence Theory

**Jiang-hua Zhang,[1] Wen-hui Huang,[2] and Jin Xu[3]**

[1] *School of Management, Shandong University, Jinan, Shandong 250100, China*
[2] *School of Management, University of Chinese Academy of Sciences, Beijing 100049, China*
[3] *School of Mathematics, Shandong University, Jinan, Shandong 250100, China*

Correspondence should be addressed to Jiang-hua Zhang; zhangjianghua28@gmail.com

Received 5 September 2013; Accepted 24 October 2013

Academic Editor: Yuxin Mao



The evidence theory is a decision-making method used to solve incomplete information and uncertain problems. This paper proposes to use evidence theory to evaluate city emergency capacity on the basis of the existing evaluation system. And when scoring each index, experts give the trust degrees of different evaluation ranks, including the trust degrees that cannot be given. It can be seen from this study that the application of evidence theory in city emergency capability evaluation has its feasibility and superiority.

## 1. Introduction

In the 21st century, a variety of unexpected accidents occur frequently, along with the ever-accelerating process of urbanization, testing the city emergency response capacity. American "9.11" incident in 2001 sounded the alarm to the world's countries, and city emergency response capacity construction is on the agenda. City emergency capability refers to the capabilities of disaster prevention and mitigation when dealing with the possible disasters and accidents of manpower, technology, organization, and resource in a city, which also refers to the comprehensive disposal abilities of a city on unexpected disasters during the entire process. With the expansion of human activity and the improvement of the modernization, the impact of the unexpected disasters on city economic development, social stability, and public safety is becoming more and more serious. As to social management and public service, the government has the responsibility of dealing with emergencies, protecting people's lives and property, and safeguarding the public security. But the 2008 Wenchuan earthquake in China, the 3.11 earthquakes in Japan, the 2012 "Sandy" hurricane in the United States, and a series of city disasters, have exposed the disadvantage of the government administration in dealing with emergency disasters, which directly or indirectly affect the disaster relief activities. Therefore, in order to better cope with sudden disasters and accidents, it is necessary to strengthen the research of city emergency response capacity.

## 2. Evaluation Index System of City Emergency Capacity and Method Reviews

Carrying out emergency capacity assessment is the basis work of strengthening city disaster emergency management, and it is a motivation for the government to improve emergency management capabilities. In recent years, various national, regional governments, and academics have made many contributions on establishing the city emergency evaluation index system and methods.

The United States is one of the countries to first study the city emergency capability and is currently walking in the forefront of this field. It established a comprehensive evaluation index system which has been widely referred to as the standard on a global scale. In June 1997, the Federal Emergency Management Agency (FEMA) and the National Emergency Management Association (NEMA) jointly proposed a set of capability assessment for readiness (CAR) [1], including 13

emergency management functions, 209 attributes, and 1014 evaluation indexes. In addition, state and local governments also set 5–8 different indexes to evaluate their city emergency capacities, such as the command and management, emergency preparedness, resource management, communications and information management, and supporting technologies.

Japan is located in the Western Pacific volcanic seismic belt, and is a multivolcano and earthquake-prone country. Therefore, establishing the city emergency capacity is particularly important to this island nation. Japan's emergency capacity index system adopted "four-level" government disaster emergency system and disaster emergency assistance system. The "four-level" disaster emergency systems include central government disaster response, local government disaster response, community disaster response, and residents' associations and self-help disaster education. Disaster emergency assistance systems include disaster emergency preparedness system, disaster emergency information systems, and disaster emergency government and social forces alliance system [2]. Especially after the Great Hanshin earthquake, the original emergency management plans were modified by the Japanese local governments in order to accommodate the city emergency management needs. They conducted a comprehensive analysis and assessment on the local government's emergency management in accordance with the emergency command, emergency information systems, refuge facilities, and storage of relief supplies, emergency medical system, and other 39 projects.

Taiwan area's disaster emergency capability evaluation sets different systems according to different objects. Its emergency work performance has eleven major categories, including general, disaster potential analysis, distribution of relief resources, disaster response units for each stage of the division of labor and responsibilities, disaster case investigation and analysis, storm and flood emergency response, earthquake emergency response, disaster response common to other types of measures, disaster management, attachments, and others [3].

Because urbanization in China started late, the city emergency capacity is gradually taken seriously in recent years. Now, it has formed management system and corresponding legal norms, but it still needs scientific and effective evaluation index system and evaluation methods. Deng et al. [4] first propose the city emergency capability assessment system framework in China, including 18 classes, 76 properties, and 405 features, which comprehensively reflected all aspects of the current city emergency capacity. Zhang et al. [5] refer the U.S. Federal Emergency Management Agency (FEMA), building the city emergency capacity evaluation index system and using the fuzzy AHP to determine the weight of each index. Cheng et al. [6] change the analysis of assessment index system from qualitative analysis to quantitative analysis with the application of SEM and analyze relations of each index. X.-T. Wu and L.-P. Wu [7] build the evaluation system and a comprehensive evaluation model to the fire emergency capability in city community based on the fuzzy analytic hierarchy process. Wang et al. [8] use logistic curve alternative, the traditional linear curve analysis, and the city emergency capability development process and propose the city emergency capability assessment model based on logistic curve.

Table 1: The evaluation index system of emergency capability.

| Target layer | Primary indexes | Secondary indexes |
|---|---|---|
| City emergency capability | Predisaster crisis prevention and early-warning capabilities | The law and the plan establishment |
| | | Monitoring and early-warning |
| | | The emergency management organizations |
| | | Training, exercises, and education |
| | Disaster response and disposal capabilities | Risk analysis and other emergency response systems |
| | | Professional teams and volunteers |
| | | Command, control, and communications on the scene |
| | | Planning and management of logistics |
| | Postdisaster recovery and reconstruction capabilities | Postdisposal |
| | | Funds to support recovery and reconstruction |

In addition, Simpson [9] uses a number of research methods, including case studies, observation, surveys, and interviews to study neighborhood and local emergency capability, which are applied to several fields of inquiry, including neighborhood and community planning and nature hazards, in particular earthquakes. Simonoff et al. [10] present a risk-based approach that can potentially be used to help emergency planners to improve the capabilities of public safety organizations to respond to terrorist attacks, accidents, or natural disasters. Ju et al. [11] present a hybrid fuzzy method consisting of FAHP and a 2-tuple fuzzy linguistic approach to evaluate emergency capacity.

But so far, there is not a unified, comprehensive mathematical theory for city emergency capacity evaluation. According to people's understanding on the evidence and knowledge, the evidence theory (*D*-*S* theory) gives uncertainty measure for the uncertain events, which can be better to deal with the fuzzy and uncertain information synthesis problem. This paper will use the evaluation index system and weight in the literature [5], as shown in Table 1. We will try to use *D*-*S* theory to make the evaluation of city emergency capacity, hoping to provide a theoretical basis for city emergency capacity.

## 3. Evidence Theory

In 1968, Dempster discussed the problem of statistical reasoning generalization and provided the synthesis principle of two groups of evidence (i.e., two independent information sources) in the book "*A Generalization of Bayesian Inference.*" On this basis, Shafer published the book "*The Mathematical Theory of Evidence*" in 1976, announcing the birth of the evidence theory, which is also called the *D*-*S* theory [12, 13].

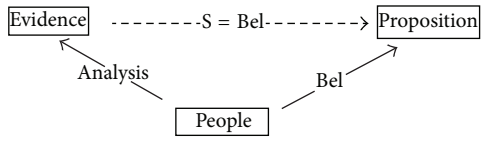



Figure 1: The constructive explanation of probability.

Evidence theory is the ideal method to solve the uncertainty problem, and it causes wide interest in the artificial intelligence field. Evidence theory has a good application in target recognition, the military command, comprehensive control, fault diagnosis, prospecting, and other research fields [14, 15].

*3.1. Basic Concepts.* Suppose a frame of discernment $H = \{H_1, H_2, \ldots, H_N\}$, which is a set of all possible results of a certain question. Shafer thought that the selection of $H$ depended on people's knowledge and understanding level, and the elements in $H$ were independent and should contain all possible results.

Shafer gave a new explanation of probability constructive explanation. Namely, probability is the trust degree of people in a true proposition on the basis of evidence, referred to as the reliability. In his view, there were not certain objective contacts between a batch of given evidence and a given proposition that could determine a precise support; we cannot always use a fairly accurate real numbers to express a down-to-earth person's psychological description of a proposition. The understanding of reliability to a proposition according to evidence can be illustrated in Figure 1.

*Definition 1.* Suppose a frame of the discernment $H$, if set function $m : 2^\Theta \to [0, 1]$ ($2^\Theta$ is the power set of $\Theta$) satisfies (1) $m(\phi) = 0$; (2) $\sum_{A\subset\Theta} m(A) = 1$.

One calls $m$ as the basic probability assignment on frame $\Theta$. When $A \subset \Theta$, $m(A)$ is called the basic probability number of $A$; When $A = \Theta$, $m(A)$ refers to the part of the reliability that does not know how to allocate.

The basic probability number reflects the size of the reliability of $A$ itself. Condition (1) reflects that it does not produce any reliability for empty set (empty proposition). Condition (2) reflects that although one can give a proposition to a reliability value of any size, one has to subject to the constraints that the sum of the reliability value of all propositions is equal to 1; that is, the total reliability is 1.

*Definition 2.* Suppose that a frame of the discernment $H$, $m : 2^\Theta \to [0, 1]$ is the basic probability assignment on frame $H$; then, one calls function $\mathrm{Bel} : 2^\Theta \to [0, 1]$ which is defined by $\mathrm{Bel}(A) = \sum_{B\subset A} m(B)$ ($\forall A \subset \Theta$) as the reliability function on $\Theta$.

*3.2. Synthesis Principle.* The following is Dempster synthesis principle, which has the most application value in evidence theory.

**Theorem 3.** *Suppose that* $\mathrm{Bel}_1, \mathrm{Bel}_2, \ldots, \mathrm{Bel}_n$ *are the reliability functions on the same frame of discernment* $H$, $m_1, m_2, \ldots, m_n$ *are the corresponding basic probability assignments, if* $\mathrm{Bel}_1 \oplus \mathrm{Bel}_2 \oplus \cdots \oplus \mathrm{Bel}_n$ *exists, and the basic probability assignment is* $m$*; then,*

$$\forall A \subset \Theta, \quad A \neq \phi,$$

$$m(A) = K \cdot \sum_{\substack{A_1,\ldots,A_n\subset\Theta \\ A_1\cap\cdots\cap A_n = A}} m_1(A_1)\cdots m_n(A_n). \tag{1}$$

*Among which,*

$$K = \left( \sum_{\substack{A_1,\ldots,A_n\subset\Theta \\ A_1\cap\cdots\cap A_n \neq \phi}} m_1(A_1)\cdots m_n(A_n) \right)^{-1}. \tag{2}$$

*Or*

$$K = \left( 1 - \sum_{\substack{A_1,\ldots,A_n\subset\Theta \\ A_1\cap\cdots\cap A_n = \phi}} m_1(A_1)\cdots m_n(A_n) \right)^{-1}. \tag{3}$$

*This formula is the Dempster principle which is synthesized by multiple reliability functions.*

## 4. The Application of *D-S* Theory in the City Emergency Capability Evaluation

The city emergency capability is a complex and comprehensive problem which includes multiple indexes. When evaluating a city, we should not only consider the existence of the objective evidence of city emergency capability but also the subjective factors caused by the insufficient evidence and personal knowledge and experience restrictions of evaluation experts. *D-S* theory is developing on the basis of considering these factors. Therefore, the application of *D-S* theory in the city emergency capability evaluation is feasible and has its certain application value.

*4.1. The Description of the Problem*

*Target Layer (E).* Evaluation of city emergency capability.

*Frame of Discernment.* $H = \{H_k \mid k = 1, 2, 3, 4, 5\}$ = {Best, Good, Average, Poor, Worst}.

*Index System.* $E = \{E_i \mid i = 1, 2, \ldots, n\}$, of which $E_i = \{E_{ij} \mid j = 1, 2, \ldots, m_i\}$, $m_i$ means that the primary index $E_i$ has $m_i$ secondary indexes. The index system in this paper is as Table 1 and we will use the weight in the literature [5]. After the aggregation and unification, index weights of various levels are as follows:

$$w = \{w_i \mid i = 1, 2, 3\} = \{0.3846, 0.4381, 0.1773\};$$

$$w_1 = \{0.3193, 0.2624, 0.1368, 0.2816\};$$

$$w_2 = \{0.2223, 0.1678, 0.3454, 0.2646\};$$
$$w_3 = \{0.5561, 0.4439\}. \tag{4}$$

Evaluation experts $L = \{L_q \mid q = 1, 2, \ldots, s\}$.

Experts give the confidence $S_{ij,k}$ ($i = 1, 2, \ldots, n$; $j = 1, 2, \ldots, m_i$; $k = 1, \ldots, 5$), among which $0 \le S_{ij,k} \le 1$ and $\sum_{k=1}^{5} S_{ij,k} \le 1$, to get the evaluation $H_k$ under the secondary index $E_{ij}$ of a city on the basis of objective evidence according to their own knowledge, experience, personal preference, and so forth.

*4.2. Calculation Steps.* First of all, establish the *mass* function of secondary indexes according to the confidence $S_{ij,k}$ and the secondary index weights given by experts. The specific method is to suppose that the index with the maximum weight in secondary indexes is the key index, while the rest are nonkey indexes. Suppose that the key index is $E_{iq}$ and its weight is $w_{iq}$, and then, the *mass* function is

$$m\left(E_{iq} \mid H_k\right) = \alpha \cdot S_{ij,k};$$
$$m\left(E_{iq} \mid H_\Theta\right) = 1 - \sum_{k=1}^{5} m\left(E_{iq} \mid H_k\right). \tag{5}$$

Suppose that the nonkey index is $w_{it}$; then, the corresponding *mass* function is

$$m\left(E_{it} \mid H_k\right) = \left(\frac{w_{it}}{w_{iq}}\right) \alpha \cdot S_{ij,k};$$
$$m\left(E_{it} \mid H_\Theta\right) = 1 - \sum_{k=1}^{5} m\left(E_{it} \mid H_k\right), \tag{6}$$

among which $\alpha$ is the discount coefficient and its value range is $(0, 1]$ and let $\alpha = 0.9$ in this paper. Due to the different weights of different indexes, the supporting strengths on the higher level index produced by their confidences are different. For instance, suppose that there are two secondary indexes $E_{i1}$ and $E_{i2}$ under the primary index $E_i$ and their relative weights $w_{i1} > w_{i2}$. Expert $L$ gives the same confidence $\beta_{i1} = \beta_{i2}$ to the city A. Then, it is obvious that the supporting strength produced by $\beta_{i1}$ is greater than that produced by $\beta_{i2}$ on the primary index $E_i$, although the confidence $\beta_{i1} = \beta_{i2}$. Thus, we set the key index as the benchmark to which other indexes can refer. Discount coefficient reflects the extent to which the key index and nonkey indexes can support the higher level index.

Secondly, according to the Dempster principle, namely, $m(A) = K \cdot \sum_{A_1, \ldots, A_n \subset \Theta, A_1 \cap \cdots \cap A_n = A} m_1(A_1), \ldots, m_n(A_n)$, we will get a *mass* function of the experts on the primary index after synthesizing the *mass* function of the two indicators. According to the weight of the primary index, we will use the weighting method to calculate the evaluation of target layers of this city.

Thirdly, we will make Dempster synthesize on experts, that is, a concentrative process of the opinions of experts, to obtain the total evaluation of target layers of this city.

Finally, we quantize the frame of discernment, that is, to determine the evaluation value of the comments using the ratio scale method. Here, we will make the following values: $P(H) = \{P(H_1), P(H_2), P(H_3), P(H_4), P(H_5)\} = \{0.9, 0.7, 0.5, 0.3, 0.1\}$. So, we will get the final evaluation value of the city emergency capacity.

## 5. Example Analysis

This section uses a specific example to illustrate the effectiveness and practicality of city emergency capability evaluation with the application of evidence theory. Setting a certain city in China as an evaluation object, we invite three experts to evaluate them. Table 2 shows the initial evaluation datum of the city by experts $X$, $Y$, and $Z$. According to the initial datum, we construct the *mass* function of secondary indexes, that is, we carry out the calculation based on the key index, nonkey indexes, and discount coefficient.

The results of the first synthesis of the secondary indexes in the evaluation index system according to the synthesis principle of the *D-S* theory are shown in Table 3.

According to the weight of the primary indexes, namely, $w = \{w_i \mid i = 1, 2, 3\} = \{0.3846, 0.4381, 0.1773\}$, we use the weight method to calculate the evaluation of the city target layer. Then, base on the weight and use the weight method to calculate the evaluation of the city target layer. The results are shown in Table 4.

The weights of the three experts $X$, $Y$, and $Z$ are, respectively, 0.2, 0.5, and 0.3. Then, Dempster synthesize three experts $X$, $Y$, and $Z$ according to the Dempster synthesis principle, that is, a concentrative process of the opinions of experts, to obtain the total evaluation of three experts. The results are shown in Table 5.

Finally, according to the ratio scale method mentioned above, namely, $P(H) = \{P(H_1), P(H_2), P(H_3), P(H_4), P(H_5)\} = \{0.9, 0.7, 0.5, 0.3, 0.1\}$, we obtain the final evaluation value after the calculation, so the final result of the city emergency response capacity is 0.6447.

From the calculation procedure and the results above, we can find out the bottleneck in city emergency capability construction through city emergency capability evaluation based on evidence theory. For example, the initial datum given by experts is the different trust degrees of the secondary indexes in city emergency capability evaluation. Through a Dempster synthesis, we can know the evaluation of the primary index. In addition, the synthetic process also takes into account the evaluation that cannot be given due to uncertainty. Focusing on improving the bottleneck with lower scores, we can enhance and improve the city emergency capability construction faster and better, so as to minimize the harm caused by disasters and emergencies.

## 6. Conclusion

The evidence theory is a decision-making method used to solve information-complex and uncertain problems, and the application of evidence theory in city emergency capability evaluation has its feasibility and superiority. This paper

Table 2: The initial evaluation datum of the city table by experts $X$, $Y$, and $Z$.

| The primary index | The secondary indexes | $X$ | $Y$ | $Z$ |
|---|---|---|---|---|
| $E$ | $E_{11}$ | $H_2(0.7)\ H_3(0.1)\ H_4(0.2)$ | $H_2(0.7)\ H_3(0.2)\ H_4(0.1)$ | $H_2(0.6)\ H_3(0.3)\ H_4(0.1)$ |
| | $E_1$ $E_{12}$ | $H_2(0.5)\ H_3(0.4)\ H_4(0.1)$ | $H_2(0.8)\ H_3(0.2)\ H_4(0)$ | $H_2(0.5)\ H_3(0.2)\ H_4(0.3)$ |
| | $E_{13}$ | $H_2(0.2)\ H_3(0.6)\ H_4(0.2)$ | $H_2(0.9)\ H_3(0.1)\ H_4(0)$ | $H_2(0.8)\ H_3(0.1)\ H_4(0.1)$ |
| | $E_{14}$ | $H_2(0.4)\ H_3(0.4)\ H_4(0.3)$ | $H_2(0.8)\ H_3(0.2)\ H_4(0)$ | $H_2(0.4)\ H_3(0.4)\ H_4(0.2)$ |
| | $E_{21}$ | $H_2(0.5)\ H_3(0.4)\ H_4(0.1)$ | $H_2(0.6)\ H_3(0.2)\ H_4(0.2)$ | $H_2(0.7)\ H_3(0.2)\ H_4(0.1)$ |
| | $E_2$ $E_{22}$ | $H_2(0.4)\ H_3(0.5)\ H_4(0.1)$ | $H_2(0.8)\ H_3(0.2)\ H_4(0)$ | $H_2(0.8)\ H_3(0.1)\ H_4(0.1)$ |
| | $E_{23}$ | $H_2(0.1)\ H_3(0.2)\ H_4(0.7)$ | $H_2(0.8)\ H_3(0.2)\ H_4(0)$ | $H_2(0.6)\ H_3(0.2)\ H_4(0.2)$ |
| | $E_{24}$ | $H_2(0.2)\ H_3(0.4)\ H_4(0.4)$ | $H_2(0.7)\ H_3(0.1)\ H_4(0.2)$ | $H_2(0.7)\ H_3(0.1)\ H_4(0.2)$ |
| | $E_{31}$ | $H_2(0.2)\ H_3(0.6)\ H_4(0.2)$ | $H_2(0.7)\ H_3(0.1)\ H_4(0.2)$ | $H_2(0.6)\ H_3(0.2)\ H_4(0.2)$ |
| | $E_3$ $E_{32}$ | $H_2(0.3)\ H_3(0.4)\ H_4(0.3)$ | $H_2(0.6)\ H_3(0.4)\ H_4(0)$ | $H_2(0.6)\ H_3(0.3)\ H_4(0.1)$ |

Table 3: The primary index evaluation of the city by experts $X$, $Y$, and $Z$.

| General goal | The primary index | $X$ | $Y$ | $Z$ |
|---|---|---|---|---|
| $E$ | $E_1$ | $H_2(0.7147)\ H_3(0.1594)$ $H_4(0.1115)\ H_\Theta(0.0144)$ | $H_2(0.9430)\ H_3(0.0452)$ $H_4(0.0056)\ H_\Theta(0.0062)$ | $H_2(0.7319)\ H_3(0.1847)$ $H_4(0.0708)\ H_\Theta(0.0126)$ |
| | $E_2$ | $H_2(0.1568)\ H_3(0.3209)$ $H_4(0.4914)\ H_\Theta(0.0309)$ | $H_2(0.9121)\ H_3(0.0604)$ $H_4(0.0126)\ H_\Theta(0.0149)$ | $H_2(0.8550)\ H_3(0.0620)$ $H_4(0.0663)\ H_\Theta(0.0167)$ |
| | $E_3$ | $H_2(0.1894)\ H_3(0.5731)$ $H_4(0.1894)\ H_\Theta(0.0480)$ | $H_2(0.7560)\ H_3(0.1228)$ $H_4(0.0779)\ H_\Theta(0.0433)$ | $H_2(0.6708)\ H_3(0.1741)$ $H_4(0.1110)\ H_\Theta(0.0441)$ |

Table 4: Emergency capability evaluation of the city.

| Target layer | Expert | Synthesis results |
|---|---|---|
| $E$ | $X$ | $H_2(0.3771)\ H_3(0.3035)$ $H_4(0.2917)\ H_\Theta(0.0276)$ |
| | $Y$ | $H_2(0.8963)\ H_3(0.0656)$ $H_4(0.0215)\ H_\Theta(0.0166)$ |
| | $Z$ | $H_2(0.7750)\ H_3(0.1291)$ $H_4(0.0760)\ H_\Theta(0.0200)$ |

Table 5: The city's evaluation of the target layer by three experts.

| General goal | Synthesis results |
|---|---|
| City emergency | $H_2(0.8736)\ H_3(0.0520)$ |
| Capability | $H_4(0.0240)\ H_\Theta(0.0505)$ |

proposes to use evidence theory to evaluate city emergency capacity on the basis of the existing evaluation system and then shows that it has practical value with the combination of an example. If we let experts to evaluate the emergency capability of a certain city, it is difficult to give a specific evaluation value. We allocate the city emergency capability evaluation into a number of small indexes, and experts only need to score these specific, detailed evaluation indexes. And when scoring each secondary index, experts give the trust degrees of different evaluation ranks, including the trust degrees that cannot be given, so as to better reflect the uncertainty due to evidence confusion or incompletion. In the future, we will research key evaluation index and auxiliary evaluation index of different stages and different institutions, and using different evaluation theories and methods, we will evaluate the typical regional government emergency capability and the main disaster accident emergency capability.

## Acknowledgments

This paper was partially supported by the National Natural Science Foundation of China (NSFC) (Grant no. 71201093), Humanities and Social Sciences Foundation of Ministry of Education of China (Grant no. 10YJCZH217), Promotive Research Fund for Excellent Young and Middle-Aged Scientists of Shandong Province (Grant no. BS2012SF012), and Independent Innovation Foundation of Shandong University, (IIFSDU) (Grant no. 2012TS194).

## References

[1] Emergency Support Function Annexes: Introduction, http://www.fema.gov/pdf/emergency/nrf/nrf-esf-intro.pdf.

[2] The Japanese disaster emergency system for reference, http://www.csstoday.net/Item.aspx?id=5261.

[3] Y. Deng, S. Zheng, and T. Liu, "Review of disaster capability assessment and emergency system," *China Journal of Safety Science and Technology*, vol. 1, no. 5, pp. 56–58, 2005 (Chinese).

[4] Y. Deng, S. Zheng, G. Liu, and T. Liu, "Study on city emergency capability assessment system," *China Journal of Safety Science and Technology*, no. 12, pp. 33–36, 2005 (Chinese).

[5] J. Zhang, X. Zheng, and J. Peng, "Research on emergency capacity evaluation based on fuzzy analytic hierarchy process," *Safety and Environmental Engineering*, vol. 14, no. 3, pp. 80–82, 2007 (Chinese).

[6] L. Cheng, S. Li, and H. Lin, "Establishment of assessment index system for the emergency capability of the coal mine based on SEM," *Procedia Engineering*, vol. 26, pp. 2313–2318, 2011.

[7] X.-T. Wu and L.-P. Wu, "Evaluation of the fire emergency rescue capability in urban community," *Procedia Engineering*, vol. 11, pp. 536–540, 2011.

[8] Z. Wang, B. Wang, and H. Zhang, "Research on urban emergency capability evaluation based on logistic curve," *China Safety Science Journal*, vol. 21, no. 3, pp. 163–169, 2011 (Chinese).

[9] D. M. Simpson, *Building Neighborhood and Local Emergency Capability: The Role of Community-Based Disaster Preparedness Programs*, University of California, Berkeley, Calif, USA, 1997.

[10] J. S. Simonoff, C. E. Restrepo, R. Zimmerman, Z. S. Naphtali, and H. H. Willis, "Resource allocation, emergency response capability, and infrastructure concentration around vulnerable sites," *Journal of Risk Research*, vol. 14, no. 5, pp. 597–613, 2011.

[11] Y. Ju, A. Wang, and X. Liu, "Evaluating emergency response capacity by fuzzy AHP and 2-tuple fuzzy linguistic approach," *Expert Systems with Applications*, vol. 39, no. 8, pp. 6972–6981, 2012.

[12] G. Shafer, *A mathematical Theory of Evidence*, Princeton University Press, Princeton, NJ, USA, 1976.

[13] J. Kacprzyk, *Studies in Fuzziness and Soft Computing*, Springer, Berlin, Germany, 2008.

[14] M. Shoyaib, M. Abdullah-Al-Wadud, and O. Chae, "A skin detection approach based on the Dempster-Shafer theory of evidence," *International Journal of Approximate Reasoning*, vol. 53, no. 4, pp. 636–659, 2012.

[15] M. A. Boujelben, Y. de Smet, A. Frikha, and H. Chabchoub, "A ranking model in uncertain, imprecise and multi-experts contexts: the application of evidence theory," *International Journal of Approximate Reasoning*, vol. 52, no. 8, pp. 1171–1194, 2011.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 867167, 13 pages
http://dx.doi.org/10.1155/2013/867167

*Research Article*

# C-Aware: A Cache Management Algorithm Considering Cache Media Access Characteristic in Cloud Computing

**Zhu Xudong,[1,2] Yin Yang,[2] Liu Zhenjun,[2] and Shao Fang[3]**

[1] *School of Computer and Information Engineering, Zhejiang Gongshang University, Hangzhou 310018, China*
[2] *Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080, China*
[3] *Zhejiang University of Technology, Hangzhou 310014, China*

Correspondence should be addressed to Yin Yang; yinyang80@gmail.com

Received 10 August 2013; Accepted 5 September 2013

Academic Editor: Yoshinori Hayafuji



Data congestion and network delay are the important factors that affect performance of cloud computing systems. Using local disk of computing nodes as a cache can sometimes get better performance than accessing data through the network. This paper presents a storage cache placement algorithm—C-Aware, which traces history access information of cache and data source, adaptively decides whether to cache data according to cache media characteristic and current access environment, and achieves good performance under different workload on storage server. We implement this algorithm in both simulated and real environments. Our simulation results using OLTP and WebSearch traces demonstrate that C-Aware achieves better adaptability to the changes of server workload. Our benchmark results in real system show that, in the scenario where the size of local cache is half of data set, C-Aware gets nearly 80% improvement compared with traditional methods when the server is not busy and still presents comparable performance when there is high workload on server side.

## 1. Introduction

In cloud computing, data congestion and network delay are the important factors that affect performance of systems. Since the hosts for storing data and processing data are often different, the demand for data transmission between tasks is very huge. Zhou et al. [1] shows that the transfer time of data intensive applications accounts for a larger proportion of the overall running time. For example, the system data and user data of PaaS services are usually stored in a centralized way, so when the computing nodes run, they need to handle the data from the storage servers. Obviously, network communication delay becomes the bottleneck of computing performance. What is worse, when multiple tasks need to transmit data across a network at the same time, they would compete for bandwidth, further worsening network delay, and it also increases the complexity of the prediction performance of dynamic model's network congestion.

The traditional way to solve the above question is to schedule the tasks so as to reduce the data transmission. As for data-intensive workflow computing, the main idea is to allocate the calculation task to the nodes that are closer to the data source, thus to reduce the time of network communication and to avoid congestion. Some researches [2] make the dynamic data replication participate as task scheduling, seeking to find the optimal scheduling scheme. It can be proved that the method based on task scheduling is NP-hard problem, which usually uses greedy method or heuristic algorithm to obtain the approximate optimal solution. This kind of method usually applies to data processing and calculation in a short term. The long-term system and service deployment, such as Amazon, Openstack, Sond, is rarely movable once being deployed to computing nodes. Therefore, it is hard to be optimized in scheduling way. Also, because of the centralized storage of data, the computing tasks in different nodes are unable to avoid the storage server bandwidth congestion.

In view of the system and service deployment applications, our group proposes the cache hierarchy architecture based on cheap disk medium [3]. The traditional concept of

cache is to use high speed and relatively expensive, volatile data storage medium to hold hot spots on the upper deck of the access level, thus improving the efficiency of data access. However, it is difficult to meet the demand of data access in the cloud computing system. (1) Computing node uses memory as a cache; the capacity is several GB, but the current cloud computing system has a large data set, of which the volume is usually hundreds of TB even petabytes; also the active data set is greater than the cache capacity. (2) A lot of access to data on a regular basis. Take the system or service startup as an example, the computing node requires access to the storage server to get the startup data, and these data usually will not be used again in a long time. So it is difficult to improve the performance of data access by using the cache. The cheap-disk-based cache hierarchy architecture uses the low-cost and high-capacity disk as the cache media, and the standard block-level interface to provide transparent data caching service for the upper application in the I/O path. The caching system is characterized as low cost, high capacity, the persistent storage, and it can solve internetwork access problems in cloud computing, as well as make it easy to expand to the complex network environment. For example Sun NFS [4], the IBM AFS, Coda [5, 6], xFS, and CAPFS [7, 8] are on the client side using a local disk as a cache to improve system performance, availability, and reliability, so as to alleviate the pressure of the back-end server load and network bandwidth.

The disk-media-based cache architecture contains disk, network, and the cache with different access properties, which the traditional cache management strategy cannot do. For example, under high load condition, the system using large capacity disk medium at low-speed cache data can improve overall system performance, but in low load cases, the emergence of a high-performance network can direct access to the back-end network storage system, and it can obtain higher overall performance than local disk storage. Therefore, the cache effect is not only associated with application access mode, but also the local cache speed characteristic and the overall load of storage system. But most of the current cache management algorithm, such as LRU, ARC [9], 2Q [10], LIRS [11], and UBM [12], each consider different factors in the management, but they are all based on the hypothesis that the local cache performance is much higher than that of medium dielectric properties. It stores the hot data in the cache to pursue high cache hit ratio to improve the performance without considering the performance difference between accessing the data source directly and the cache. In D-Cache [3], it is found that these cache management methods cannot well adapt to the changing of the cache media and data load in some cases and even reduce the overall performance of the system in some cases. How to manage the cache and obtain the overall ideal performance according to the cache medium characteristics, the current load of storage systems and application access mode is an urgent problem.

This paper proposes a storage cache management algorithm called C-Aware as shown in Figure 1. It considers the speed of the cache media and network load conditions, analyzes the access cache and historical information of network data, and also predicts the future access to the cache and storage server performance according to the historical information. Additionally, it decides whether the current request of the data should be stored in the cache. When this algorithm is applied to computing nodes' caches, under the condition of high load, it can use local cheap low-speed cache media to cache data and improve the overall performance of the system; otherwise, it directly accesses the network storage systems to reduce the performance loss caused by access of low-speed media. Through the benchmark test and load trace simulation, the results show that the C-Aware algorithm, compared with other cache algorithms, is able to obtain a better overall performance and better adaptability under different speed of the cache media and storage server load cases.

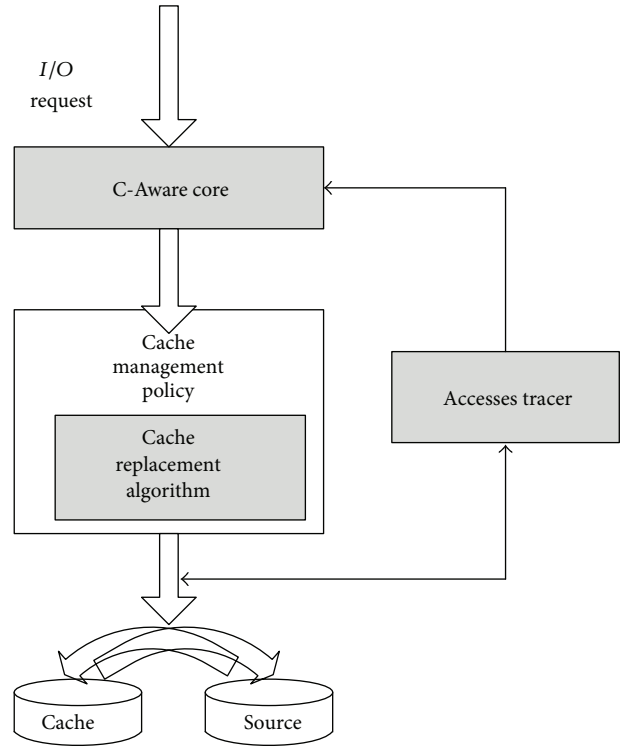



Figure 1: C-Aware framework.

Section 2 briefly introduces the related research on current cache management; Section 3 introduces the basic idea, specific design and implementation of C-Aware algorithm; Section 4 presents the trace simulation test and benchmark assessment results and analysis; the last section summarizes the full text and puts forward the existing problems and the suggestions for future work.

## 2. Related Work

Cache management mainly consists of two parts [13]: cache replacement strategy and place strategy. The former decides to choose a block of data cache as new cache space when

the old one is full; the latter determines the timing of the block data being stored in the cache. Because the current cache system design is based on the assumption that the performance of accessing the cache media is higher than that of direct access to the data source, the majority of cache management methods use cache placement strategy. That means when accessing the data block, it saves data blocks in the cache, and hot data must be kept in the cache as much as possible. So the current cache management algorithm research mostly focuses on the cache replacement problems.

The current cache replacement algorithm develops from LRU and LFU single strategy to the one that can be adaptive to application access patterns, such as the ARC [9], UBM [12], MQ [1], DULO [14], and so forth. With the development and wide use of network technology and storage system, research on the cache algorithm changes from the single stage to multistage multilevel cache coordination management. According to the cache management strategy, its algorithm can be divided into simple single management strategy, the access-frequency-and-access-time-based balance strategy, application-based strategy, detection-based caching strategies, and so forth. Simple single cache management strategy generally uses a single fixed standard to do replace management, such as LRU, MRU, LFU, and so on. They replace either based on cache block that has recently been used or on the number of cache block that has been accessed. These methods are ancient and simple, but particularly effective for a specific application. And they are simple in the system design and implementation, thus being the most used method at present.

The strategy of access frequency and access time balance is adjusted mainly based on the access frequency and a recent visit time, such as LRFU [15], ARC, CAR [16], MQ [1], and so on, but they use different balance adjustment methods. The characteristic of this strategy is that it can be used in two different access patterns but not for more access modes. The strategy implementation is relatively simple, like the ARC algorithm in IBM's high-end storage systems. Application-based strategy is usually based on some kind of special application cache management optimization or according to some specific information, such as DBMIN [17], Application-Controlled File Caching Policies [18, 19]. This strategy has good effects on the specific applications, and the defect is the poor generality. It adopts the way of clues to display information; application program interface is required to provide support. Cache management strategy which is based on detecting generally tracks the access mode, identifies what the current application accesses belong to, such as the circulation and sequential or random pattern, and then adaptively manages according to the predetermined method. The representative algorithms are DEAR [20], PCC [21], UBM [12], and AMP [22]. This strategy has strong adaptability to different access modes, but the design implementation is more complex; therefore it is seldom used in the actual production system. From the classification of cache management level, it can be divided into single-stage and multistage cache management algorithm. The former is only for the machine system's cache management, not considering the influence of other levels of cache, such as traditional LRU, MRU, and LRU. But along with the development of network, distributed systems and independent storage system, the data access level increases, different levels of cache form a multilevel system. They influence each other, and the single-stage single-level cache management is not able to meet this need; therefore, multistage cache management becomes a hot spot recently. Multistage cache management algorithm can be divided into two categories: one is the radical collaborative; one is the level perceptual. The former uses information displayed or management interface to coordinate between different cache hierarchies, such as TQ [23], DEMOTE [2], ULC [24], and other algorithms. This type of algorithm can more accurately coordinate the cache management. But it may need to change the existing program interface and need additional traffic load. According to the evaluation [25], the actual effect is not ideal. The latter, level perceptual algorithm, usually predicts and judges based on the implicit information left by upper level cache to decide the management in this level. Some studies [26] propose cache algorithms for cloud computing.

These management algorithms try to improve the performance of the system mainly from the influence of application's access modes on the cache hit ratio, and they are all based on the strong premise—visit from cache must be far higher than from direct access to the data source. But for disk or similar common low-speed buffer medium in the distributed system, these algorithms have no specific considerations. Disk can hold data persistently and generally be used as an agent of cache mediation in local or I/O access path in the distributed system [27]. On one hand, it can reduce the performance loss caused by network delay or server under high load; on the other hand, it also reduces the number of requests to concurrently access the storage system, which indirectly improves the response speed of the server. But because of the disk performance limitations, the cost of waiting for cache replacement is very large. With the high-speed network and server providing high performance, the traditional cache management methods that keep the data in the cache and pursue high cache hit ratio approach cannot necessarily get good performance, in some cases even harmful. But these algorithms take no consideration of the performance of caching management and data source itself, and in the current distributed system, these factors are very important to the cache performance. C-Aware algorithm, different from the traditional algorithms, takes the cache speed characteristics of the medium itself and the data source current response performance into consideration in the cache management. C-Aware in the cache placement decision is not based on the access-based placement strategy, but the current cache and data source access conditions to dynamically decide whether to cache data or not. It does not pursue high cache hit ratio as the goal but aims to enhance the overall performance of the system and the adaptability in different media performance and service load. For the distributed cache system and network storage system that use low-speed medium such as disk on the client and the I/O access path, this algorithm has stronger practical significance in improving the overall performance of the multilevel cache.

## 3. Description of the C-Aware Algorithm

*3.1. Basic Idea of C-Aware.* The main difference between C-Aware and other traditional algorithms is that C-Aware does not try to cache every request (in this paper, cache means the disk media to store data copy in the computing node) but decides whether to cache data according to current access environment. It gives full consideration to the cache replacement, cache data source, current load, and other factors in cache management. Its idea is simple: C-Aware records the response times of history accesses to cache or source device and guides the cache management decision through a heuristic method based on the past information.

C-Aware framework consists of three components: (1) the original cache management algorithm; (2) C-Aware core, which guides the cache management through heuristic anticipation with history information; and (3) a tracer, which records the processing time for each request.

Based on the history information, C-Aware core heuristically anticipates the future access situation and implements the decision of cache data. C-Aware core is implemented separately for each cache replacement algorithm. It decides whether to cache the data block needed in current access before the replacement algorithm. Once caching the current data block, C-Aware will utilize a traditional replacement algorithm to put the data block into the cache, otherwise, it would not cache the data block. And C-Aware will bypass cache and directly get the data from the network in the next accesses. In short, C-Aware considers access characteristics and workload of cache media, improves the cache performance under heavy server's workload, and tries to eliminate the side effect of caching data on low-speed cache media.

*3.2. C-Aware Tracer.* In the C-Aware algorithms, the response time of cache and network access is adopted to serve as parameter to assess the network access speed characteristic and the storage network workload. The tracer traces and records every access type and response time and uses them as the basis for C-Aware core decision. Based on different access objects and interface in linux system, the I/O access requests can be divided into six types.

(1) Cache Write Request. This type of request refers to all I/O write requests which are sent to cache by C-Aware.

(2) Direct Cache Write Request. This kind of request refers to the write request which is sent directly to cache media without waiting for cache replacement, write-back, and prereading operation. These requests are a subset of Cache Write Requests.

(3) Source Write Request. This category of request refers to the write request which is sent to source media by C-Aware.

(4) Cache Read Request. This type of request refers to all read requests which are sent to cache media.

(5) Direct Cache Read Request. This kind of request refers to the read request which is sent directly to cache media without waiting for cache replacement, write-back, and prereading operation. These requests are a subset of Cache Read Requests.

(6) Source Read Request. This category of request refers to the read request which is transferred to source media by C-Aware.

C-Aware organizes cache space according to cache block as the unit, and I/O requests are usually less than the size of the cache block size. In order to maintain the cache management, a missing data reads into the cache in accordance with the whole cache blocks. Before the entire cache block of data is read into, although cache blocks have been reported in the current buffer index, the data in the cache block is not immediately available. At this time, other I/O access request needs to wait for the cache block read operation to be completed. Under the simplified D-Cache multilevel model, the read of Cache block in the layer disk needs a process; its time influenced by the response time of the network and lower storage. Therefore, in this case, a cache hit delay is more possible than that of direct access to the cache. So the cache hit access is divided into two cases: (1) cache directly hits and does not need to wait because of the cache management; (2) cache hits but needs to wait for the cache block fetches. The two are different in performance.

In C-Aware, we obtain the average response times to predict the response time of current response. The internal storage for recording the total time and access frequency of every request is limited. However, this would not reflect the current access situation accurately due to the different characteristics at different stages. For example, the office system has periodic fluctuation. In the morning of Monday, it has a vast number of visits, but on weekends, that would be a few. Suppose that the current access characteristics can reflect the current system situation, C-Aware gives different weights to different response time for the access requests at different times. As for the specific design, we refer to the I/O scheduling algorithm in the Linux kernel to count I/O response time. C-Aware tracer records the three parameters of each type of requests:

(1) sample, presents the total requests of each type;

(2) totaltime, records the total handle time until the last request is finished by C-Aware;

(3) meantime, presents average response time for different types of request, the basis for C-Aware to judge the current load.

The formulas are

$$\text{totaltime}_{n+1} = \frac{7 \times \text{totaltime}_n + 256 \times \text{last\_request\_handle\_time}}{8}, \quad (1)$$

$$\text{samples}_{n+1} = \frac{7 \times \text{samples}_n + 256}{8}, \quad (2)$$

$$\text{meantime}_{n+1} = \frac{\text{totaltime}_{n+1} + 128}{\text{samples}_{n+1}}. \quad (3)$$

The last_request_handle_time is the handle time of the last I/O request of this type. It can be seen from (1) that the handle time for the most current request has the biggest weight. However, along with the arriving of new requests, the former request weights are reducing. This statistical method is simple, but it considers the influence of past requests by giving a different weight to history total time and it consumes few resources of the system. Therefore, the C-Aware method is easy to apply and has strong practicality. The experimental results also show its validity.

*3.3. C-Aware Core.* Through the tracer, C-Aware obtains the average response times of different types of request, predicts the cache and storage situation, and makes management decision. When the current access data hit the cache, C-Aware transfers the request to the cache; otherwise, it handles the request as follows.

*Rule 1.* If the current request is read and the cache is full, $\text{Meantime}_{\text{Cache\_Read\_Request}} < \text{Meantime}_{\text{Source\_Read\_Request}}$, C-Aware decides to cache data and selects a cache block to replace according to predefined replacement algorithm.

*Rule 2.* For write request, if cache is full, $\text{Meantime}_{\text{Cache\_Write\_Request}} < \text{Meantime}_{\text{Source\_Write\_Request}}$, C-Aware will make a decision to cache current request and then make a cache block replacement.

*Rule 3.* If the current request is read, there are still free cache blocks, and $\text{Meantime}_{\text{Direct\_Cache\_Read\_Request}} > \text{Meantime}_{\text{Source\_Read\_Request}}$, C-Aware would not cache the data request.

*Rule 4.* For write request, there are still free cache blocks, and $\text{Meantime}_{\text{Direct\_Cache\_Write\_Request}} > \text{Meantime}_{\text{Source\_Write\_Request}}$; C-Aware would not cache the data request.

*Rule 5.* If the cache is full and it does not meet Rules 1 and 2, C-Aware would not cache the data request.

*Rule 6.* If there are still free cache blocks and it does not meet Rules 3 and 4, then C-Aware allocates a cache block to cache the data request.

C-Aware is very easy to combine with common cache replacement algorithm, as shown in Algorithm 1.

C-Aware renews the access after handling every I/O request. If there are no more requests of the same type, the request information will keep the same. Therefore, the problem emerges: if the response time of the type cannot be renewed, the system load situation will not be true. For example, when the storage network load rises, the value of $\text{Meantime}_{\text{Source\_Write\_Request}}$ may be much bigger than the value of $\text{Meantime}_{\text{Cache\_Write\_Request}}$. So the C-Aware can choose to cache data, and all the requests should be completed in the cache as much as possible. In this case, because the following request reaches the priority access to the cache, $\text{Meantime}_{\text{Source\_Write\_Request}}$ values may not be updated for a long time. When the storage network load drops, the algorithm will not be able to reflect the current situation, still accessing the data from the cache. In order to solve this problem, C-Aware's solution is to regularly update the average response time for each type of request. Specifically, C-Aware sets minimum update interval time. If the average response time of a particular type of requests is longer than that, C-Aware will update automatically. The current practice is to set the average response time half of the old values, so after a certain time, the response time of this type of request will be very low. Therefore, according to the rules set by the C-Aware, it will take the initiative to choose the type of request, thus updating the average response time.

## 4. Evaluation

We first evaluate the C-Aware algorithms by using caching system simulator based on the trace and test the effectiveness of C-Aware with different applications' traces. Then, we set up a D-Cache system in multiple computing nodes and a web storage server test environment. The D-Cache System executes the prototype C-Aware algorithm. This paper uses iozone test tools to compare the D-Cache system integrating C-Aware algorithms with the one using only LRU algorithm and at the same time compare the performance of IBM dm-Cache [28] caching system.

*4.1. Simulation Results.* The cache simulator of cloud computing system architecture is realized based on disksim [29]. It simulates disk-based cache in client by disksim and simulates the access pattern between computing nodes and storage servers in cloud computing by setting network delay, access delay of storage server, and other parameters. In the simulation environment, we realize the disk-based cache management algorithm like D-Cache and dm-Cache as the real prototype system. It uses the LRU replacement algorithm to compare simulation results. In simulation, the simulation parameters are shown in Table 1. In order to simplify the design of the simulator, we set the value of write-back cache read data, proofread write cache data, network delay, and storage server access delay as fixed average time.

During the test, we use the block level of OLTP and WebSearch. The OLTP test is a test of read/write hybrid and WebSearch test is a test of read operation, with the write operation rate being extremely low, which is related with the network characteristics of the search request. In simulation test, the average value of load access delay of storage server ranges from 1 ms to 800 ms. Figures 2 and 3 show the simulation results of OLTP and WebSearch trace when the storage server load changes. The test results are in accordance with the various algorithms' average response time per 100 requests.

The "direct" means the test results without accessing the disk Cache storage server. Because other parameters are fixed average, the change of direct response time will reflect the storage server load changes. For each trace, we test the cache performance of two disks: one is higher, disksim simulator sets the average seek time as 2 ms; another performance is low,

```
Cache_Handle (Request){
    Get cache_block from data cache according to current request
    if (cache_block ==NULL){
        if (Cache is full){
            if (Rule 1 or Rule 2 is satisfied)
                Cache_Miss (Request);
            else
                Access the storage network directly;
        } else if (Rule 3 or Rule 4 is satisfied){
            Access the storage network directly;
        } else {
            Cache_Miss (Request);
        }
    } else
        Cache_Hit (Request);
    }
```

Algorithm 1: Algorithm of C-Aware core.

Table 1: Simulation test parameters.

| Parameter | Value |
|---|---|
| Storage server access latency under normal circumstances | 0.1 ms |
| Access delay with increasing of storage server load | According to specific tests |
| Network delay | 0.01 ms~0.1 ms |
| Write-back cache operation delay | 10 ms |
| Proofread write cache operation delay | 10 ms |
| Disk cache block size | 256 K |
| Cache size | 1.25 G, 2.5 G, 12.5 G |

the average seek time is 3.5 ms. Specific test results are shown in Figures 2 and 3: (a, b) is the situation when the average seek time is 2 ms. (c, d) is the situation of 3.5 ms. In (a, c) the cache size is 1.25 G, while in (b, d) the cache size is 12.5 G.

It can be seen from the test that when cache access performance is higher (Figures 2(a), 2(b), 3(a), and 3(b)), as D-Cache and dm-cache, the C-Aware algorithm can improve the overall performance of the system. But compared with the D-Cache and dm-Cache, the C-Aware algorithm performance will lose slightly when the cache is larger. This is because the C-Aware will forward requests to the storage server from time to time to detect the storage server's current situation, so as to decide whether to place the data blocks to the local disk cache or not. Therefore, it leads to a higher average response time when the system load is high. On the other hand, when the cache response performance is low (Figures 2(c), 2(d), 3(c), and 3(d)), C-Aware algorithm has stronger adaptability. It can improve the performance in both OLTP and WebSearch's trace tests. But in the reading-dominant WebSearch tests, the average response time of I/O requests is very high when using D-Cache and dm-Cache system. Its performance is much lower than accessing storage system directly.

From Figures 2(c) and 2(d), we can see that C-Aware can significantly improve the speed of the I/O processing at high load with the change of the storage server load. Instead, the speed change of D-Cache and dm-cache is very large without C-Aware algorithm, in many cases, far more than the time required to access storage server directly. The phenomenon is particularly prominent when the cache performance is not high, and the space is little but requires a lot of cache, as shown in Figure 2(c). This is because the D-Cache and dm-Cache use the traditional Cache management algorithm which does not consider cache speed characteristics of the media itself, resulting in a large number of I/O being directed to the low performance cache, thereby reducing the performance of the whole system. C-Aware algorithm will adjust based on the current cache and the storage network access and decide whether to store the following data. Thereby, it obtains a better performance as to the access balance between the cache and the storage server.

In Table 2, we can see that when combined with high-performance cache in the OLTP tests, the cache hit ratio difference in the three types of caching system with different cache size is not big. C-Aware declines a bit 1%~2%, which shows that the C-Aware can achieve better performance by finding cache, thus improving the hit ratio of cache by saving data as far as possible. However, when the performance of the cache is not good, in Table 2, we can observe that the C-Aware will take the initiative to reduce the cache hit ratio and the access to the cache system appropriately, thus improving the overall performance of the system. With decreasing cache size, C-Aware forwards more I/O requests directly to the storage system for processing, and the corresponding cache hit ratio will be lower. But in WebSearch tests, C-Aware also shows the similar phenomenon, which is more obvious. Because the read operation is more time-consuming, in WebSearch tests, when the cache is small, C-Aware will take

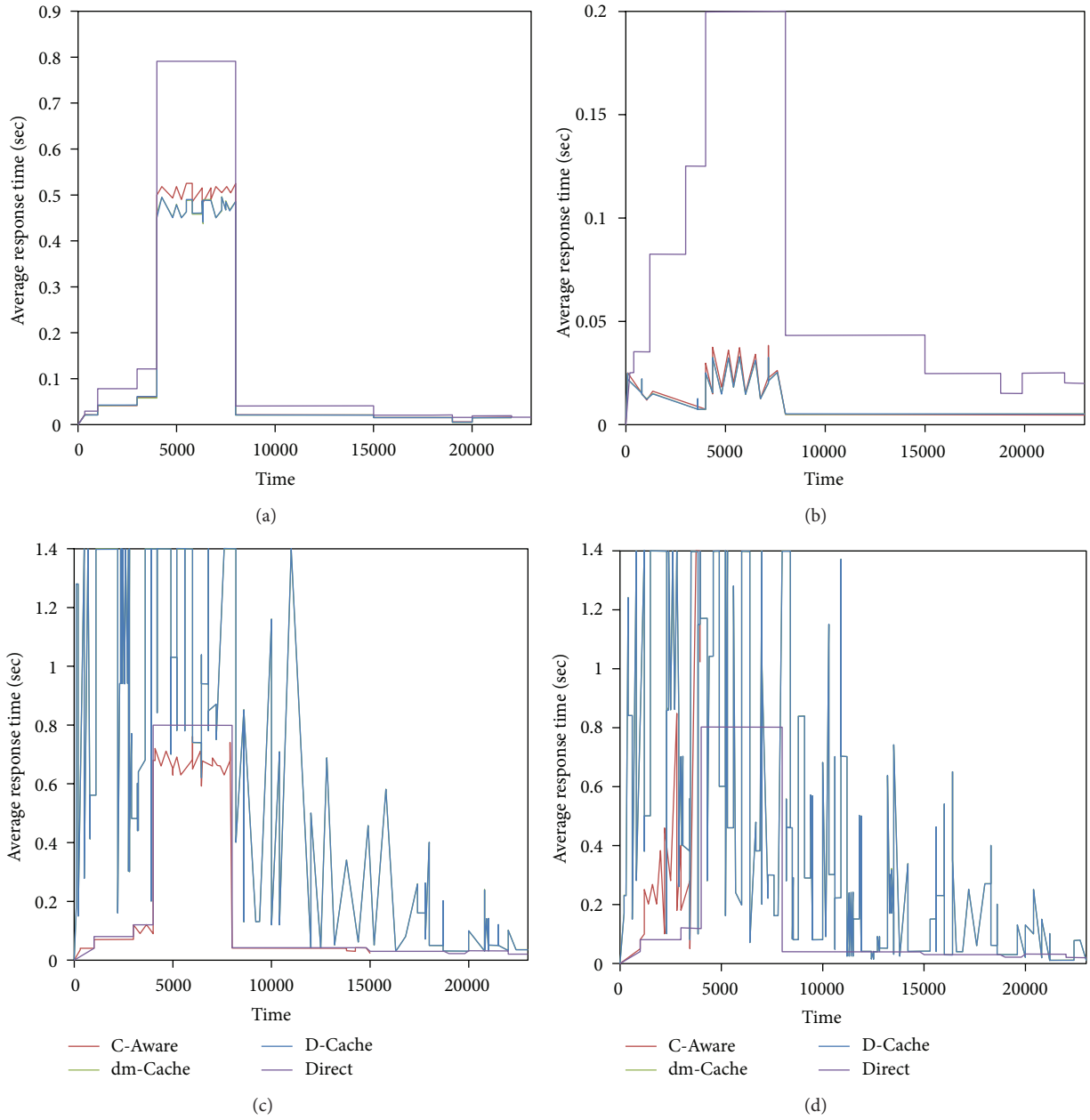



Figure 2: WebSearch test results.

the initiative to reduce the frequency with the cached data and send more I/O requests to the server in order to reduce the load of the disk cache and obtain a better overall performance. Table 3 clearly illustrates the problem with the I/O numbers corresponding to the C-Aware algorithm. When the cache size is 1.25 G, average seek time of cache disk is 3.5 ms, C-Aware has 1130607 requests sent directly to the data source for processing, which account for a quarter of the total number of requests, while under the traditional algorithm there are only 176 requests. Due to the increasing number of requests, the average response time of C-Aware in this test reduces nearly 80%.

We also test the cache storage server under high-load and low-load conditions (for publication reason, we do not list the results here). The test results also show that the C-Aware has better adaptability. Especially when the cache is small and the storage server is able to provide high performance access, C-Aware can significantly reduce the cache data operation and tries to send the request directly to the storage server to complete the processing. In the end, it can achieve better performance. Through the simulation tests, we can see that the C-Aware can dynamically decide whether to cache data or not according to the current access condition and speed properties of the cache media, so that it can in most cases ensure a better overall performance.

*4.2. Prototype System Test Results.* The following experiments have been done on one storage server and eight computing

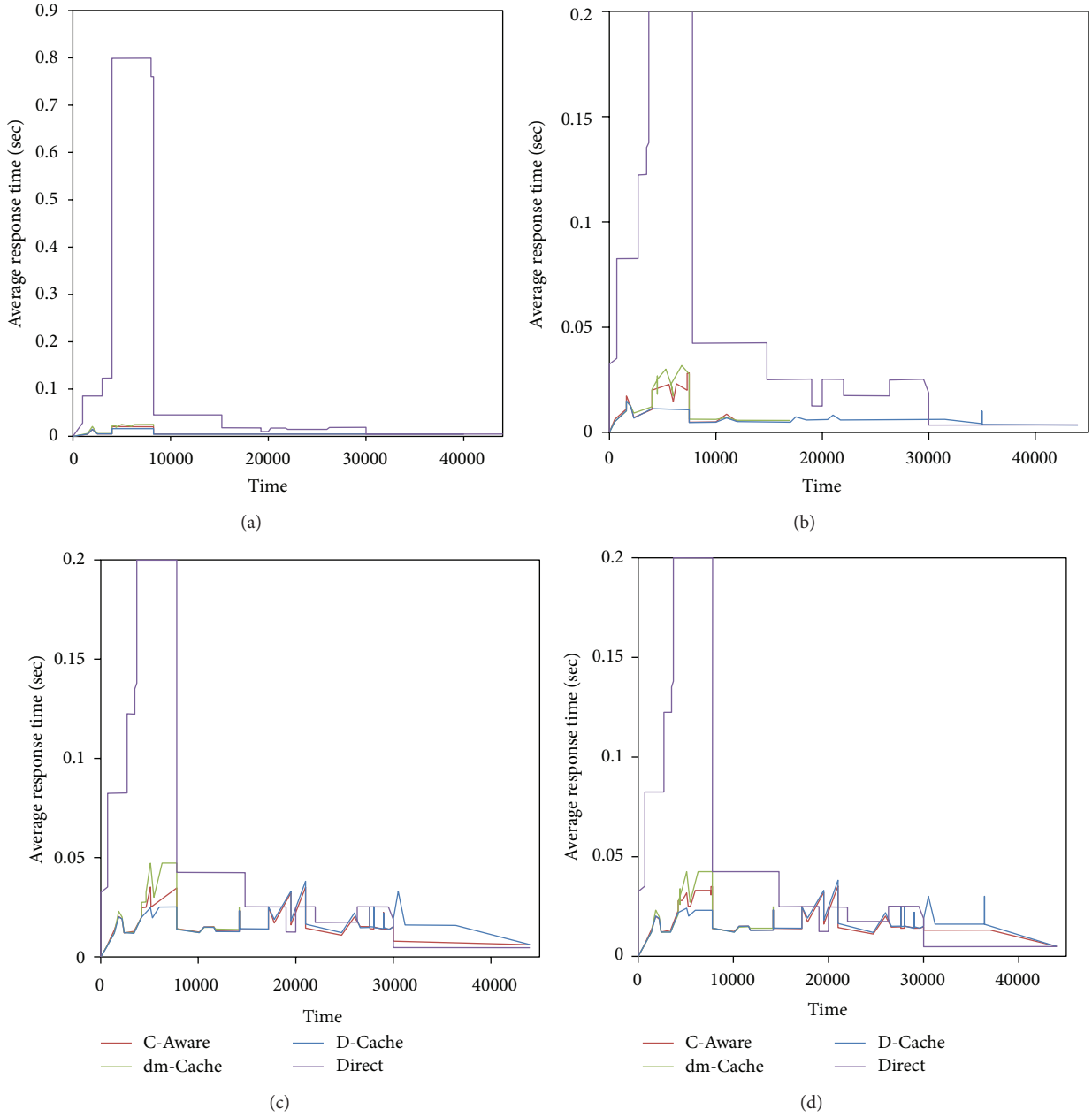



Figure 3: OLTP (finance1) test results.

nodes. The storage server adopts the Intel Xeon(TM) CPU 3.20 GHz × 2.4 GB main memory, Adaptec (formerly DPT) SmartRAID V RAID Controller and six 74 GB SCSI disks to construct a RAID0 storage system, and Intel 82546 GB Gigabit ethernet. The storage server runs 32 bit Fedora Core 4 with 2.6.11-1.1369FC4smp kernel. All AS are equipped with an Intel Xeon(TM) CPU 3.20 GHz × 2.4 GB main memory, Adaptec (formerly DPT) SmartRAID V RAID Controller and three 74 GB SCSI disks to construct a RAID0 storage system, and Intel 82546 GB Gigabit ethernet. Each AS runs 64 bit Fedora Core 4 with x8664 2.6.17-1.2142FC4smp kernel. Both the application servers and storage server are interconnected via Giga-bit Ethernet, and NBD agreement [1] to access the data. We use iozone as the major benchmark tool for this evaluation and 8 G as the size of the test-file, so as to reduce the influence of internal memory on the results. In order to investigate the effects of each algorithm on the scalability of storage system, the benchmark was executed with 1, 2, 4, 8's computing nodes. During the process, each computing node starts the iozone to read/write concurrent access to the storage server at the same time. The tests include:

(1) sequential, random, and mix read/write iozone,

(2) cold cache and warm cache. The test file is already in the cache or not. Before each test, the application server will restart in order to ensure no valid data

Table 2: OLTP trace simulation test results.

| | | Average seek time 2.0 ms | | | Average seek time 3.5 ms | | |
|---|---|---|---|---|---|---|---|
| | Direct | C-Aware | D-Cache | dm-Cache | C-Aware | D-Cache | dm-Cache |
| Cache size 1.25 G | | | | | | | |
| Average response time (ms) | 102.948 | 7.477 | 6.852 | 8.310 | 13.767 | 16.004 | 17.629 |
| Hit ratio (%) | | 96.57 | 98.57 | 98.91 | 89.19 | 98.57 | 98.91 |
| Cache replacement | | 14283 | 14989 | 14989 | 12502 | 14989 | 14989 |
| Direct I/O | 5394885 | 136497 | 25677 | 25677 | 545242 | 25677 | 25677 |
| Cache size 2.5 G | | | | | | | |
| Average response time (ms) | | 7.225 | 6.440 | 7.835 | 14.268 | 15.739 | 17.296 |
| Hit ratio (%) | | 98.12 | 99.26 | 99.46 | 92.77 | 99.26 | 99.46 |
| Cache replacement | | 6141 | 6374 | 6376 | 5298 | 6374 | 6376 |
| Direct I/O | | 71676 | 9021 | 9026 | 363569 | 9021 | 9025 |
| Cache size 12.5 G | | | | | | | |
| Average response time (ms) | | 7.304 | 6.448 | 7.840 | 14.979 | 15.904 | 17.460 |
| Hit ratio (%) | | 99.00 | 99.52 | 99.69 | 95.12 | 99.52 | 99.69 |
| Cache replacement | | 0 | 0 | 0 | 0 | 0 | 0 |
| Direct I/O | | 28490 | 0 | 0 | 239552 | 0 | 0 |

Table 3: Web search simulation test results.

| | | Average seek time 2.0 ms | | | Average seek time 3.5 ms | | |
|---|---|---|---|---|---|---|---|
| | Direct | C-Aware | D-Cache | dm-Cache | C-Aware | D-Cache | dm-Cache |
| Cache size 1.25 G | | | | | | | |
| Average response time (ms) | 200.587 | 162.068 | 266.522 | 266.527 | 171.770 | 967.141 | 967.145 |
| Hit ratio (%) | | 52.24 | 63.80 | 63.80 | 49.59 | 63.80 | 63.80 |
| Cache replacement | | 936005 | 1217932 | 1217932 | 874800 | 1217932 | 1217932 |
| Direct I/O | 4381687 | 928748 | 176 | 176 | 1130607 | 176 | 176 |
| Cache size 2.5 G | | | | | | | |
| Average response time (ms) | | 140.017 | 242.872 | 242.873 | 151.863 | 941.943 | 941.943 |
| Hit ratio (%) | | 63.92 | 71.94 | 71.94 | 61.74 | 71.94 | 71.94 |
| Cache replacement | | 714641 | 866160 | 866160 | 669189 | 866160 | 866160 |
| Direct I/O | | 593114 | 104 | 104 | 753693 | 104 | 104 |
| Cache size 12.5 G | | | | | | | |
| Average response time (ms) | | 74.244 | 129.943 | 129.944 | 355.029 | 829.776 | 829.777 |
| Hit ratio (%) | | 96.23 | 98.02 | 98.02 | 95.39 | 98.02 | 98.02 |
| Cache replacement | | 8024 | 9959 | 9959 | 7304 | 9959 | 9959 |
| Direct I/O | | 93425 | 46 | 46 | 136094 | 46 | 46 |

in the application server memory, thus reducing the influence of memory cache data on testing. The cache replacement algorithms adopted by D-Cache, dm-Cache, and C-Aware are based on the LRU algorithm, and the size of the Cache block is 128 k. Cache capacity can accommodate 50% of test data.

Figure 4 shows the result of Iozone Sequential Test on warm cache. It is the rewrite, read, and reread test immediately after the finishing of iozone's first write test. From Figure 4, we can see that D-Cache that integrated C-Aware has better performance compared with dm-cache which only has LRU algorithm. In the first read test, when the number of clients is less than 8, the D-Cache performs best. The reason is that the cost of cache replacement leads to the decrease of performance. However, the C-Aware can adjust according to the load situation. At the start of the test, since there is no replacement operation, C-Aware would believe the local cache can improve the performance. With the reading full, the cost of cache replacement can decrease the performance; the C-Aware will forward the subsequent requests directly to the storage devices in order to achieve a good overall performance.

In cold cache test, as for pure write test, the difference between writing the already existing file and writing the new one is not that big. The test result should be similar to Figure 4(a), so we did not test again for this kind of situation, and only tested the read and reread as shown in Figure 5.

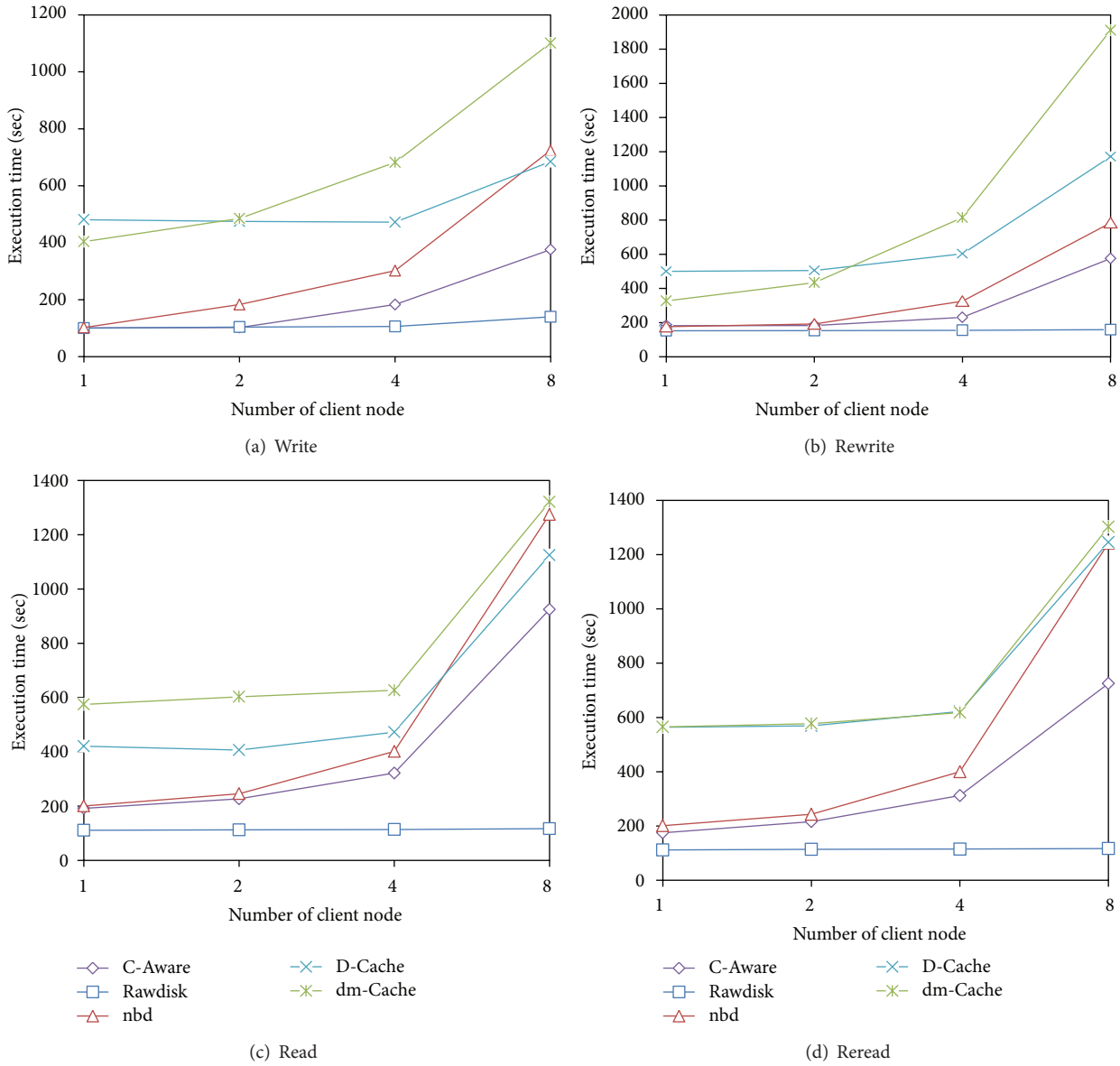
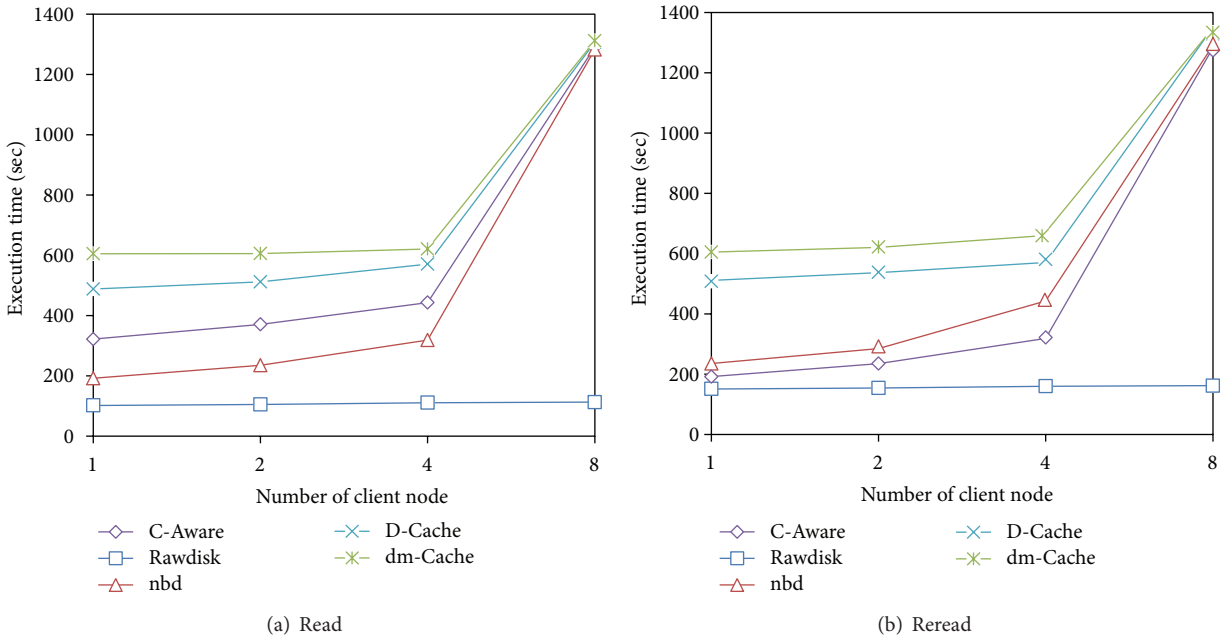



Figure 4: Result of iozone sequential test on warm cache with 50% test data size.

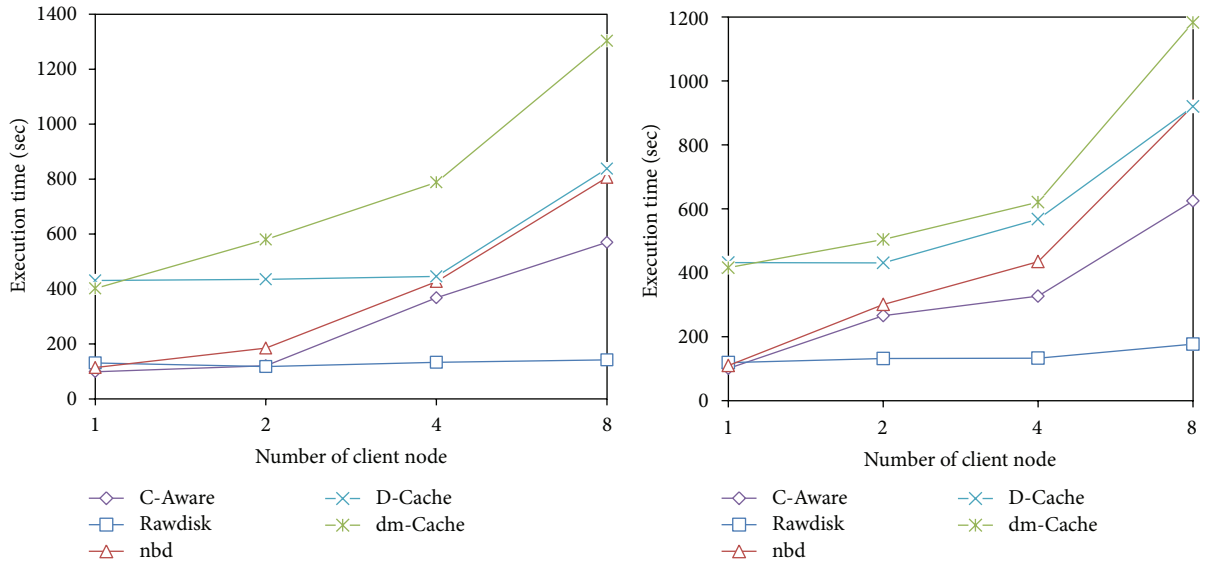
We can see the performance of first read. When the storage server load is low, the performances of dm-cache, D-Cache, and C-Aware are far lower than direct network access. This is because the read request needs to wait for the prereading generated by the cache strategies, and after being written to the cache they can continue to reduce the performance of the read access. Meanwhile, the cache space is limited; cache block replacement further reduces the performance of the system in the reading test process. Compared with the dm-cache and D-cache, the performance of C-Aware is increased by nearly 25%; this is because the C-Aware will try to buffer cache data to reduce the cost and improve the performance. Since it caches part of the data in the cache, it provides a better performance in the subsequent reread. We can also find that with the increase of the storage server system load, the performance difference of dm-cache, D-cache, and C-Aware in the first read data test and direct network access is becoming smaller. This is because the relative continuity and cache prefetching have played an important role in improving performance.

*4.2.1. Iozone Mix Read/Write.* In the mix tests, we mainly test when the read/write percentage is 30% and 80%. Figure 6 reflects warm cache test results. It can be found that in the tests where read occupies a larger scale, the system performance is better. This is because the read operation needs to be synchronized but write can be asynchronous. Figure 7 reflects test results in cold cache. We can find that, in the case of a cold cache, the cache system achieves relatively better performance. We guess this is because the warm cache,




(a) Read (b) Reread

Figure 5: Result of iozone sequential test on cold cache with 50% test data size.



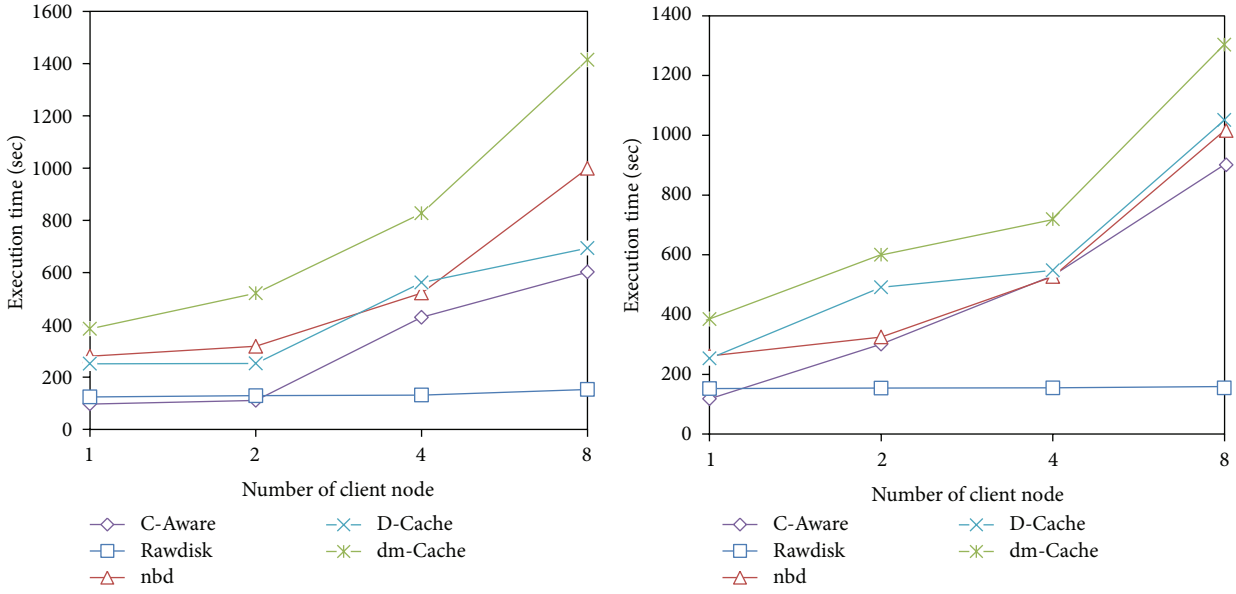

(a) 30% read (b) 80% read

Figure 6: Result of iozone mix test on warm cache with 50% test data size.

after the first write, caches part of the test file, and that will generate cache replacement costs in the mix tests, which can reduce the performance. However, the buffer is empty in the cold cache test at the beginning, thus the write requests can be directly written to the cache, which improves the test performance. From mix test results, it can be seen that the caching system combined with C-Aware algorithm has better performance in most cases.

## 5. Conclusion

This paper presents a storage cache placement algorithm—C-Aware, which considers the speed characteristics and network access situation of computing nodes' cache media. It adaptively decides whether to cache data block by history access information of data source. It mainly concerns the influence of cache media speed characteristic, prefetch cost,




(a) 30% read (b) 80% read

Figure 7: Result of iozone mix test on cold cache with 50% test data size.

and network transmitting situation on the overall system performance. As a result, it achieves good adaptability to workload difference and cache media characteristics. The benchmark and trace simulation tests have verified the conclusion. Suggestions for future research include first, realize C-Aware at the first document level and further test its validity; second, currently, C-Aware just considers the response time of I/O handling, and more system parameters should be taken into account.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## Acknowledgments

This work is supported in part by Natural Science Foundation of China "Research on the snapshot data security storage technology for authorization of release," no. 61100057, and the National Basic Research Program of China, no. 2004CB318205.

## References

[1] Y. Zhou, Z. Chen, and K. Li, "Second-level buffer cache management," *IEEE Transactions on Parallel and Distributed Systems*, vol. 15, no. 6, pp. 505–519, 2004.

[2] M. Theodore Wong and J. Wilkes, "My cache or yours? making storage more exclusive," in *Proceedings of the USENIX Annual Technical Conference*, pp. 161–175, 2002.

[3] Y. Yang, *Research on multi-level and low-cost cache for network storage, [Ph.D. thesis]*, Institute of Computing Technology, Chinese Academy of Science, 2009.

[4] R. Sandberg, D. Goldberg, S. Kleiman, D. Walsh, and B. Lyon, "Design and implementation of the sunnetwork file system," in *Proceedings of the Summer USENIX*, pp. 119–130, June 1985.

[5] J. H. Howard, "An overview of the andrew file system," in *Proceedings of the Winter USENIX Conference*, pp. 23–26, Dallas, Tex, USA, February 1988.

[6] M. Satyanarayanan, "Scalable, secure, and highly available distributed file access," *IEEE Computer Society*, vol. 23, no. 5, pp. 9–20, 1990.

[7] T. Anderson, M. Dahlin, J. Neefe, D. Patterson, D. Roselli, and R. Wang, "Serverless network file systems," in *Proceedings of the 15th Symposium on Operating Systems Principles*, ACM Transactions on Computer Systems, 1995.

[8] M. Vilayannur, P. Nath, and A. Sivasubramaniam, "Providing tunable consistency for a parallel file store," in *Proceedings of the 4th USENIX Conference on File and Storage Technologies*, December 2005.

[9] N. Megiddo and D. S. Modha, "ARC: a self-tuning, low overhead replacement cache," in *Proceedings of the USENIX Conference on File and Storage Technologies (FAST '03)*, San Francisco, Calif, USA, March 2003.

[10] T. Johnson and D. Shasha, "2Q: a low overhead high performance buffer management replacement algorithm," Proceedings of the VLDB-20, September 1994.

[11] S. Jiang and X. Zhang, "LIRS: an efficient low inter-reference recency set replacement policy to improve buffer cache performance," in *Proceedings of the ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, pp. 31–42, June 2002.

[12] J. M. Kim, J. Choi, J. Kim et al., "A low-overhead high-performance unified buffer management scheme that exploits sequential and looping references," in *Proceedings of the 4th conference on Symposium on Operating System Design & Implementation*, San Diego, Calif, USA, 2000.

[13] Z. Chen, Y. Zhou, and K. Li, "Eviction-based cache placement for storage caches," in *Proceedings of the USENIX Annual Technical Conference*, pp. 269–282, 2003.

[14] S. Jiang, X. Ding, F. Chen, E. Tan, and X. Zhang, "DULO: an effective buffer cache management scheme to exploit both temporal and spatial locality," in *Proceedings of the 4th USENIX Conference on File and Storage Technologies*, San Francisco, Calif, USA, 2005.

[15] D. Lee, J. Choi, J.-H. Kim et al., "LRFU: a spectrum of policies that subsumes the least recently used and least frequently used policies," *IEEE Transactions on Computers*, vol. 50, no. 12, pp. 1352–1361, 2001.

[16] S. Bansal and D. S. Modha, "CAR: clock with adaptive replacement," in *Proceedings of the USENIX File and Storage Technologies (FAST '04)*, San Francisco, Calif, USA, 2004.

[17] H. -T. i Chou and J. D. David, "An evaluation of buffer management strategies for relational database systems," in *Readings in Database Systems*, pp. 174–188, 1988.

[18] P. Cao, E. W. Felten, and K. Li, "Application-controlled file caching policies," in *Proceedings of the Technical Conference on Summer USENIX*, Boston, Mass, USA, 1994.

[19] R. H. Patterson, G. A. Gibson, E. Ginting, D. Stodolsky, and J. Zelenka, "Informed prefetching and caching," in *Proceedings of the 15th ACM Symposium on Operating Systems Principles*, pp. 79–95, ACM Press, 1995.

[20] J. Choi, S. H. Noh, S. L. Min, and Y. Cho, "An implementation study of a detection-based adaptive block replacement," in *Proceedings of the Annual USENIX Technical Conference*, pp. 239–252, 1999.

[21] C. Gniady, A. R. Butt, and Y. C. Hu, "Program counter based pattern classification in buffer caching," in *Proceedings of the 6th Symposium on Operating Systems Design and Implementation (OSDI '04)*, 2004.

[22] F. Zhou, R. von Behen, and E. Brewer, "Program context specific buffer caching with AMP," Tech. Rep. UCB CSD-05-1379.

[23] X. Li, A. Aboulnaga, K. Salem, A. Sachedina S, and Gao, "Second-tier cache management using write hints," in *Proceedings of the 4th conference on USENIX Conference on File and Storage Technologies*, San Francisco, Calif, USA, 2005.

[24] S. Jiang and X. Zhang, "ULC: a file block placement and replacement protocol to effectively exploit hierarchical locality in multi-level buffer caches," in *Proceedings of the 24th International Conference on Distributed Computing Systems*, pp. 168–177, March 2004.

[25] Z. Chen, Y. Zhang, Y. Zhou, H. Scott, and B. Schiefer, "Empirical evaluation of Multi-level Buffer Cache Collaboration for Storage System," in *Proceedings of the ACM International Conference on Measurement and Modeling of Computing Systems (SIGMETRICS '05)*, June 2005.

[26] H. Lee, "loudCache: Expanding and shrinking private caches," in *Proceedings of the IEEE 17th International Symposium on High Performance Computer Architecture (HPCA '11)*, pp. 219–230, San Antonio, Tex, USA, 2011.

[27] H. Kllapi, E. Sitaridi, M. M. Tsangaris, and Y. Ioannidis, "Schedule optimization for data processing flows on the cloud," in *Proceedings of the ACM SIGMOD International Conference on Management of data*, pp. 289–300, New York, NY, USA, 2011.

[28] E. V. Hensbergen and M. Zhao, "Dynamic policy disk caching for storage networking," Tech. Rep. RC24123, IBM Research Division Austin Research Laboratory, 2006.

[29] http://www.pdl.cmu.edu/DiskSim/.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 674057, 7 pages
http://dx.doi.org/10.1155/2013/674057

*Research Article*

# An Analysis and Design of the Redirection Schema in ForCES

**Ming Gao,[1,2] Shiju Li,[1] and Weiming Wang[2]**

[1] *Department of Information Science and Electronic Engineering, Zhejiang University, No. 38, Zheda Road, Hangzhou 310027, China*
[2] *Department of Information and Electronic Engineering, Zhejiang Gongshang University, No. 18, Xuezheng Street, Hangzhou 310018, China*

Correspondence should be addressed to Ming Gao; gaoming19790508@126.com

Received 4 July 2013; Accepted 29 July 2013

Academic Editor: Yoshinori Hayafuji



The idea of Forwarding and Control Element Separation has widely accepted by next generation network researchers, the regain attention of IETF (The Internet Engineering Task Force) ForCES (Forwarding and Control Element Separation) is the best proof. An IP tunnel-based redirection schema was proposed to solve the problem of routing protocol messages interaction between ForCES router and the external merchant routers. The technology of network virtualization is introduced to map network interface from ForCES FE (Forwarding Element) to CE (Control Element) which collaborating with the redirect schema.

## 1. Background and Related Work

ForCES (forwarding and control element separation) [1] is a working group of routing area in IETF (the internet engineering task force), which devotes itself to study the architecture, standard, and gordian technique. In this paper, for convenience, we always call IETF ForCES ForCES elliptically.

In ForCES, network devices such as ip router, firewall, and ethernet switch will be called by a joint name of NE (network element) [1]. Every NE is composed of two separated plane: forwarding and control. Forwarding plane consists of several FEs (forwarding element) [2] which are dedicated to fast packet routing lookup and forwarding. CE (control element) on the control plane is designed mainly for calculating path, producing routing table and synchronizing them to every FE.

The purpose of ForCES is to provide an open, standard, and modularized platform for network device's design and manufacture to bring more innovation. Although the concept of forwarding and control element separation has been proposed by IETF ForCES for 10 years, there is still no authoritative research result appearing on the design and analysis of ForCES implementation. Another interesting thing is that the word of ForCES never disappears from our sights in the past 10 years, and now the idea of ForCES becomes the common sense for more and more researchers in next generation network area, especially for open network researches.

In open network area, XORP [3] constructs an open and extensible router platform which provides various routing protocols such as OSPF, RIP, and BGP with open interfaces. These features are very interesting for a ForCES researcher because they can greatly reduce the work of developing routing protocols in ForCES implementation. Similar to the ideas of LFB (logical function block) [4] in ForCES, Click [5] partitions route into a number of abstract function modules and packet's content and transmission direction will be changed when traversing these modules; different combinations of these abstraction modules would generate different types of router function, for example, IPv4 and MPLS. The most notable character of Click is the modularity and high flexibility of architecture; however, Click does not show us how to quickly develop new routing protocols.

In the past two years, a large number of open network researchers turn to the next generation network architecture. The most representative one is SDN [6] which is powered by ONF [7] and IRTF [8], and now it becomes the focus and clearly claims that its core idea comes from IETF ForCES. In ONF, different from traditional network devices, OpenFlow [9] considers flows in a network no longer to be individual packets but a bundle of data flows. The processing of data flow can be defined and directed by controller via configuring

flow tables; every entry of a flow table that is maintained by a controller has a strategy. To replace the pattern of per-packet routing, OpenFlow switch utilizes flow strategies to make decision of packet's outgoing, with just the controller being very similar to ForCES CE.

Although OpenFlow has been very successful and obtained great attention, its route map of clean slate will bring a great shock to the existing IP network. Therefore, the NVO3 (network virtualization over 3-layer) [10, 11] that emerged just last year tries another way of IP tunnel to realize the separation of business logical network and physical network. Every VN (virtual network) presenting as a specified business network has independent topology and address space. The routing between VNs can be done by BGP with the direction of controller because only controller has the overview of VN distribution.

From the previous, we can definitely say that the separation of forwarding and control elements separation is a universal phenomenon now in architecture design and network management. This paper will focus on two problems: (1) How can NE exchange routing protocol information with the outside. (2) How to make CE learn FE's physical network interfaces and operate them locally.

For the problem (1) above, the key is to guarantee the integrality of routing protocol information to flow freely between CE and FE. Wang et al. [12] designed a ForCES-based Ip router and told us how to implement an ip-forwarding-enabled FE but didi not mention the exchanging of routing protocol information between CE and FE. In Wang's implementation, routing table is just added manually into FE and CE did not run any routing protocol. So, we propose an IP tunnel-based redirection mechanism as the carrier to hold all of the routing protocol information.

For the problem (2), the technology of interface virtualization [13] is adopted to number the physical network interfaces on FEs uniformly and then map them to CE. CE will operate these virtual network interfaces just like its own, but in fact they are located in FE.

## 2. The Design of Redirection Schema

Wang et al. [12] supplies an open and standard implementation of ForCES protocol layer (PL) [14] and transport mapping layer (TML) [15] and packages the implementation as a middle box. The middle box is composed of two parts: *ForCES protocol* and *ForCES function middleware*. The former undertakes the tasks of encapsulation, de-encapsulation, and so forth of ForCES PDU (packet data unit), which is defined by PL. The latter on FE is responsible for registration and attribute management of relevant LFBs. The middleware obtains operating information from *ForCES protocol stack*, calls registered processing functions to manipulate LFB attribute, manages LFB event lists and generates event reports. *ForCES function middleware* on CE is designed mainly for saving all the attribute information of LFBs, analyzing user operating commands from user, and distinguishing redirect messages from messages received by

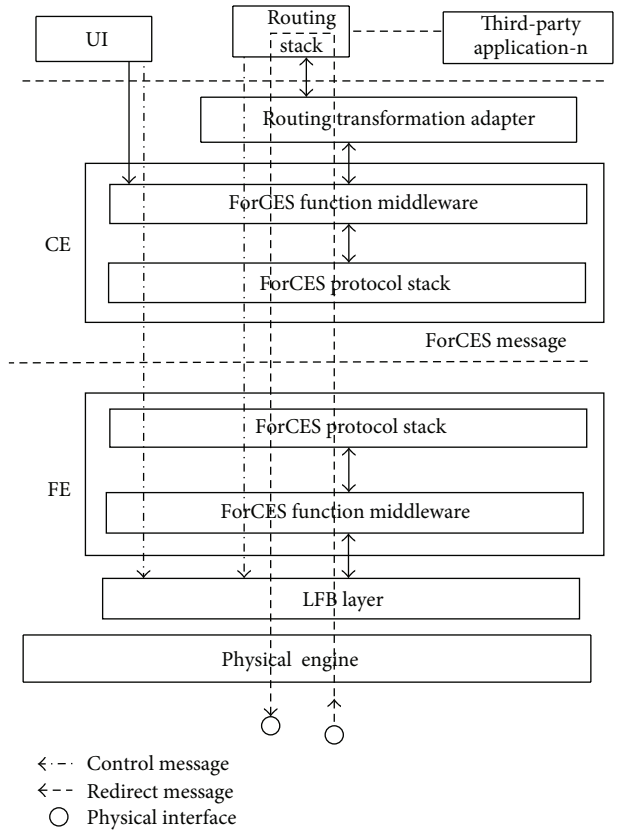



Figure 1: Software structure of ForCES NE.

*ForCES protocol stack* and forwarding them to a third-party software, for example, Xorp, zebra.

Based on Wang's contribution, we further propose a software framework of ForCES NE shown in Figure 1 and the detailed introduction as follows.

When FEs have received routing protocol messages such as OSPF and SNMP from physical network interfaces and cannot handle them by themselves, then FEs will send these messages to CE to request for further processing. After the succeeded processing, CE sends them back to the corresponding FE. Whether from FE to CE or from CE to FE, all the routing protocol messages must be encapsulated as ForCES PDU. We call the process redirection and the type of ForCES PDU *ForCES redirection message*.

The format of PDU of ForCES *redirection message* is defined and implemented by PL [14]; also PL defines another important message named *ForCES control message* which will be used by CE to control and manage FEs including attribute configuration, query, and report of capacity and event.

For the sake of convenient stating, in the remainder of this chapter the words of message and packet will appear alternatively and have the same meaning.

In Figure 1, commonly *routing stack* supplied by third-software Xorp is located in CE. Now we suppose that routing protocol packets (i.e., RIP, OSPF) are originated from outside routers and now have been received in FE; then methods taken by FE must solve problems as follows: (1) settle down which type of ForCES message, redirection, or control to be

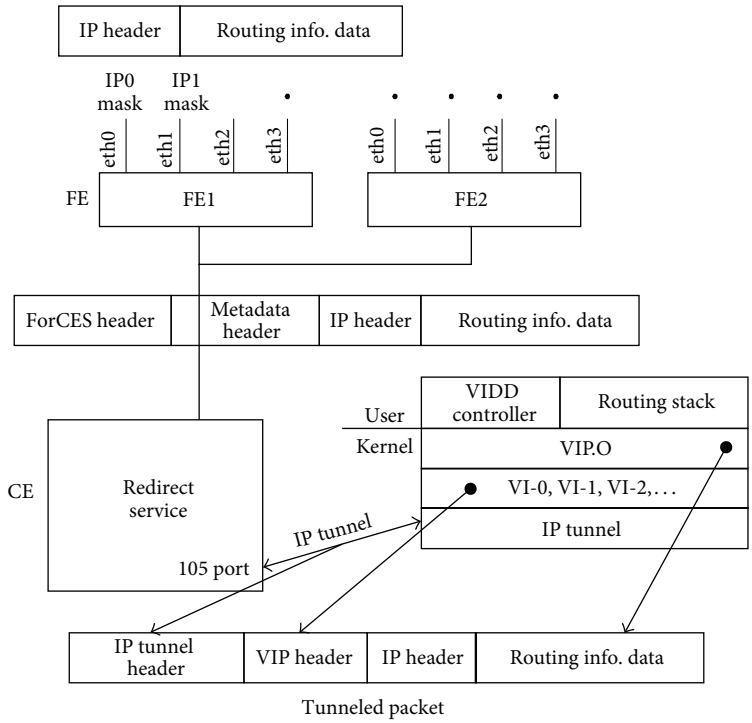



Figure 2: Routing protocol packet transmitting in ForCES NE.

the carrier to transport routing protocol packets from FE to CE and (2) determine what *Metadata* are needed by CE to handle these routing protocol packets properly and then deliver them to CE.

Because of the separation of FE and CE, CE has no way to make sense of where these routing protocol packets come from and go to, so we must try to realize the information synchronization between CE and FE, and the *routing transformation adapter* is introduced which will be carefully detailed in part III. Much information needs to be synchronized; the most important one is the network interface. Physical network interfaces of NE mainly exist in FE, and in order to map these network interfaces from FE to CE, virtual interface (VI) technology [13] is needed. With the help of VI, CE can operate these virtual network interfaces just like its own, but in fact they are located in FE.

According to the definition of ForCES framework [1], an incoming packet in FE will be processed by LFBs from ingress-EtherPHYCop to egress-EtherPHYCop; all of the LFBs must located in datapath defined by FE-Intra topology [16]. Obviously during the processing of LFB, some data maybe produced for next usage. For example, when IPv4UcastLPM LFB is doing the prefix matching, a key data *NexthopID* will be produced which is necessary for subsequent IPv4NextHop LFB is processing and we call these key data *Metadata*. To be noted is that all of the types of LFB mentioned earlier in this paragraph can be found in LFB library [17].

Figure 2 shows an efficient method of how to transfer routing protocol packets between CE and FE. When FE receives the packets which need to be processed by CE, *metadata* will be added to IP packet header and the IP packet is just the routing protocol packet.

To facilitate the encapsulation of *Metadata*, we created a collection of headers named *Metadata header* to contain these *Metadata* in order. *Metadatas header* mainly includes four fields: FEID, Res, portID, and Length. FEID identifies the arrival of packets or destination CE. Res is a reserved field. portID denotes the arrival of packets or destination port number. Length stands for field length. CE analyzes the *ForCES header* which is located in the *redirection* messages and draws out the *Metadata* information and then delivers it to CE. At this time, CE will convert the *Metadata* to virtual interface port (VIP) and attach *VIP header* to the IP packets. After doing this, CE will transmit the packets to VIP module for processing and then add an *IP tunnel header* before *VIP header*. Thus, IP tunnel + VIP + IP + DATA package format is formed. Finally the packets will be accessed and processed by *routing stack* in CE.

After the processing of request packets, *routing stack* generates specific response packets which also need to add specific *VIP header* in virtual network interface card. Then they will be sent to VIP through IP tunnel. When CE receives the packets, *Metadata header* will be added and they will be encapsulated into ForCES redirection message and sent to FE. Subsequently FE determines the outgoing physical network interface and sends the packets out according to the *Metadata header*.

What needs to be particularly pointed out is that *routing stack* described in the paper deals with two types of messages: *control and redirection* as illustrated in Figure 1. When *routing stack* needs to interact with the outer routers, *redirection*

channel will be used. While the routing table has been changed by *routing stack*, *control* channel will be used to synchronize routing table with FE.

## 3. The Synchronization from CE to FE

Routing table is made up of route entries and there are two kinds of routing tables existing in current routing system: integrated and distributed [17].

For edge router, a single table is usually used to store routing entries for the convenience of management, which is called integrated routing table. Each entry of the table includes *Destination*, *Mask*, *NHIP* (next hop IP), OID (outport ID), *Flag*, *MTU,* and *Metric*.

For core routers, as the number of route entries is too large (generally around 20000), enormous storage space will be wasted while using a single table. The reason is that in an integrated routing table, different destination networks may point to the same next hop. In that case, the amount of routing table can absolutely be compressed by reasonable designing. Thus distributed routing table is proposed. It maintains two subtables: *PT* (prefix table) and *NHT* (next hop table). Every entry of PT consists of fields of *Destination*, *Mask,* and *NHI* (next hop index). *NHT* contains *NHI*, *OID* (outport ID), *Flag*, *MTU,* and *NHIP* (next hop IP). The two tables correlate to each other by NHI. By using this type of design of routing table, the repetitive storage of information in a table may be greatly reduced.

Here a calculating model is set up to calculate the compression efficiency when an integrated routing table is converting into a distributed one. For integrated routing table we define every route entry as $RTEntry = \{Destination, Mask, NHIP, OID, Flag, Metric\}$. Let $Length_0$ denote the total bytes of all of the fields. Suppose that the number of route entry is $N$ and store each field's value of *RTEntry* into arrays $Destination[N]$, $Mask[N]$, $NHIP[N]$, $OID[N]$, and $Metric[N]$. Suppose, $K$ is the times when different values appear in $NHIP[N]$; then we can get a set $NHIP_{\text{val}} = \{NHIP_1, NHIP_2, \ldots, NHIP_K\}$ and any $\text{NHIP}_i$ in $\text{NNIP}_{\text{val}}$ is defined as follows:

$$NHIP_i = \{NHIP[i+j], NHIP[i+k], \ldots, NHIP[i+l]\} \tag{1}$$

subject to

$$1 \le i+j \le i+k \le i+j \le N,$$
$$NHIP[i+j] = NHIP[i+k] = \cdots = NHIP[i+l]. \tag{2}$$

Let $NHIPCount[i]$ be the number of items in set $NHIP_i$; we will get $\sum_{i=1}^{K} NHIPCount[i] = N$ and obviously $Size_0 = Length_0 * N$ can be used to denote the total bytes of an integrated routing table.

For distributed routing table we define entry of PTas $PTEntry = \{Destination, Mask, NHI\}$ and $Length_1$ as the total bytes of all of the fields in *PTEntry*; then the total bytes of PT can be calculated as

$$Size_1 = Length_1 * \sum_{i=1}^{K} NHIPCount[i]. \tag{3}$$

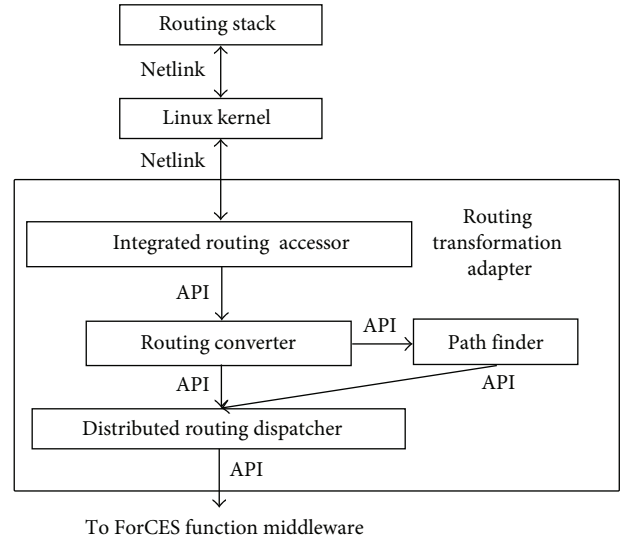



Figure 3: The structure of routing transformation adapter.

In distributed routing table, entry of NHT is defined as $NHTEntry = \{NHI, OID, Flag, MTU, NHIP\}$ and $Length_2$ denotes the total bytes of all of the fields in NHTEntry; then the total bytes of NHT are calculated as $Size_2 = Length_2 * K$.

So far we can deduce the compression efficiency which is denoted by $\eta$ when an integrated routing table converts into a distributed one. Equation (3) is the formula of $\eta$:

$$\eta = \frac{Size_0}{Size_1 + Size_2} = \frac{Length_0 * N}{Length_1 * \sum_{i=1}^{K} NHIPCount[i] + Length_2 * K}. \tag{4}$$

From (4) we can clearly see that the value of $\eta$ is determined by a function composed of three variables: $N$, $K$, and distribution of the *NHIP*.

The reality is that the routing stack in CE uses integrated routing table while FE uses the distributed. In order to make up the difference, *routing transformation adapter* is introduced in the paper and the software structure is depicted in Figure 3. The primary functions of *routing transformation adapter* are dedicated to transform integrated routing table to distributed one and dispatching it. In the former, integrated routing entries generated by the *routing stack are* located in Linux kernel; then *integrated routing accessor* in *routing transformation adapter* will communicate with Linux kernel bidirectionally through *Netlink* to read and write them. The integrated routing entries provided will be converted to distributed ones by *routing converter*. In our design for every distributed routing entry, the operating will be mapped to the modification of *PTEntry* and *NHTEntry*. Both *PTEntry* and *NHTEntry* are attributes of LFBs such as LPM (longest prefix match) and NextHop [4, 17]. In Figure 3, *path finder* calculates to get the ID of attribute in the format of ForCES path. While calculating, it is very difficult to locate the destination (i.e., FE) that the routes need to be sent to. For this problem, FE ID and Port ID should be used simultaneously

while numbering virtual interface. In this way, FE IDs can be extracted from the outgoing virtual interface-OID included in *NHTEntry*. Based on the new calculated ForCES paths, *distributed routing dispatcher* maps the configuration of *PTEntry* and *NHTEntry* in distributed routing table to the modification of LFB's attributes [2]. Figure 3 interacts with *ForCES functional middleware* in Figure 1; *distributed routing dispatcher* sometimes sends ForCES *control* message to FE to modify LFB's attributes via ForCES *function middleware*. Now we give the detailed example of adding a routing entry as follows.

(1) Searching for *NHIP* in *NHT* to get the corresponding *NHTEntry*. If it is found, return the *NHI* contained in *NHTEntry*. If not, allocate a new *NHI* and create an *NHTEntry*.

(2) Searching for the result of *Destination* and *Mask* to get the corresponding *PTEntry*. If it is found, end the procedure and return. If not, create a new *PTEntry* according to *Destination*, *Mask,* and *NHI*.

## 4. Evaluation

To evaluate the effect imposed by the mechanisms of redirection, we build a testbed shown in Figure 4. Both CE and FE are implemented by PC with Linux installed and configured with Intel core (TM) 2 Quad 2.8 GHz CPU and 2.96 GB RAM. Connect one port of FE to SMB1-2 port of SmartBits 600 which is manufactured by Spirent Network Ltd, USA. In SmartBits, TeraRouting simulates network topology and tracks the exchanging of OSPF link state via SMB1-2 port; OSPF link state packets going through the redirect channel will be encapsulated and decapsulated over and over, especially the routing scale increasing to a certain degree, extra cost of computation time will be unavoidably introduced into system. In our testing-scenario design, when TeraRouting changes the network topology, the network scale and number of route entry will change accordingly. Here we only have OSPF network LSAs [18] custom made to generate route entries. In OSPF specification every OSPF network LSA corresponds to a route entry and several LSAs will be encapsulated in one OSPF update packet entering the ForCES redirection channel. As a result, we can envisage that overhead of CPU of CE and FE will increase, and more bandwidth will be occupied by redirection process, and also the convergence time of OSPF with state "FULL" will be delayed. Here network scale mentioned in the behind is only by the number of route entries, by monitoring these factors, we capture the view of results from Figures 5 to 9.

Adding an additional redirection process between CE and FE adds overhead to the system. However, as a result of our design, the redirection mechanism does not add overhead to the data path of FE. To quantify this overhead, we measure the increased CPU utilization and FE's forwarding latency and bandwidth cross between CE and FE for network scale changing with and without the redirection.

Figure 5 shows the variation of overhead of CPU, obviously seeing that with redirection mechanism, CE and FE are more sensitive to network scale.

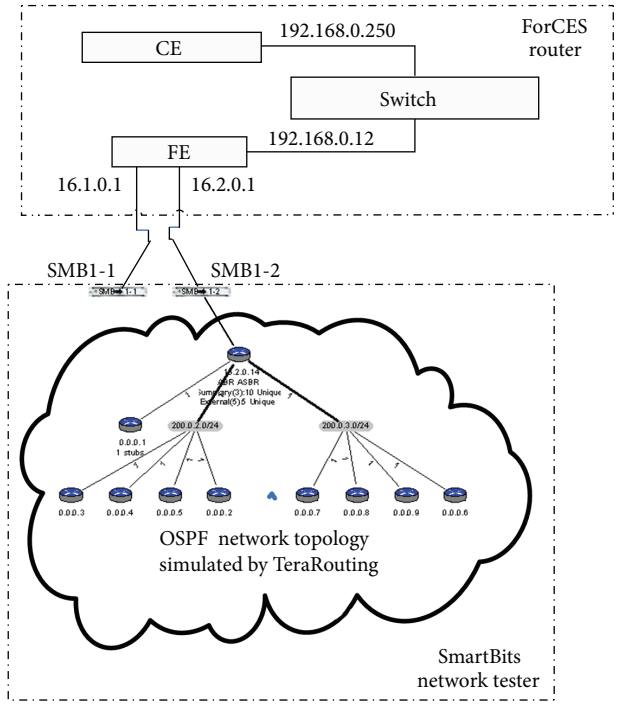



Figure 4: The construction of testbed.

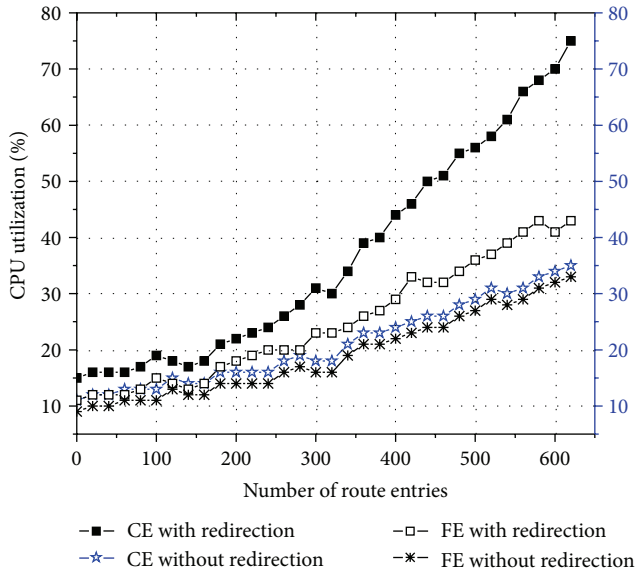



Figure 5: The view of overhead appended to CPU.

Our results (Figure 6) show that the redirection increases FE's forwarding latency, that is, the additional transfer time from SMB 1-1 to SMB 1-1, by about 16 ms on average. For latency sensitive applications, for example, web services in large data centers, 16 ms may be too much overhead.

All OSPF protocol messages including network LSA will be encapsulated as ForCES redirection message during redirection channel. However one determinative factor is the "*Length*" field in ForCES protocol message header [14]; it is a 16-bit-long integer, which means that the total ForCES

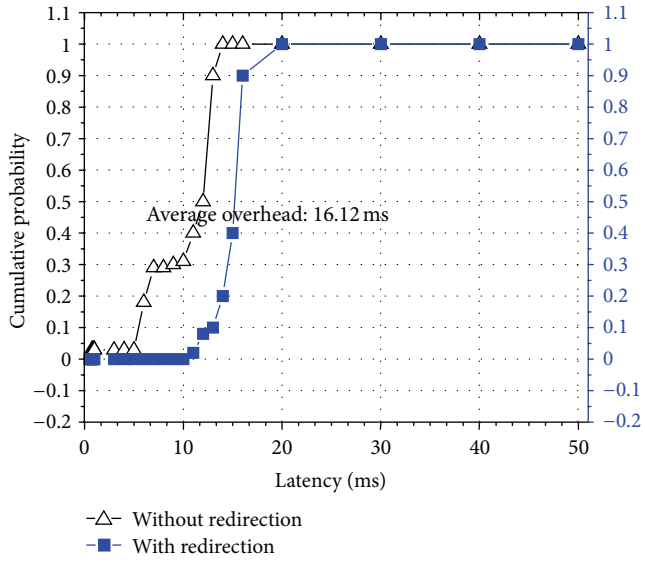



Figure 6: CDF of redirection overhead for FE's forwarding rate.

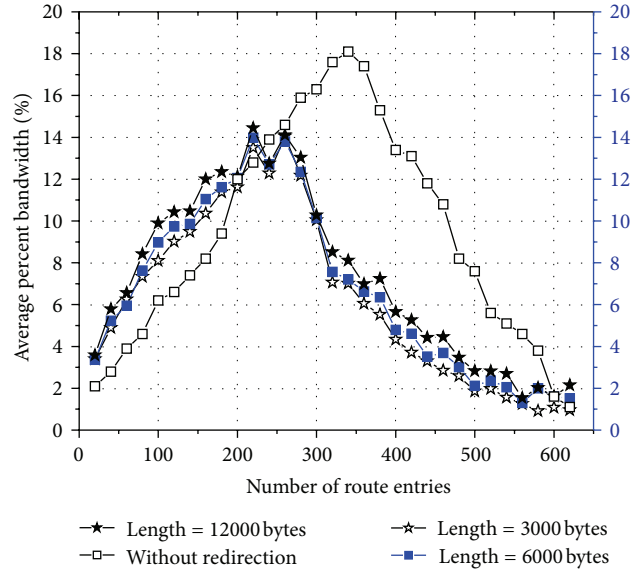



Figure 7: Effect of bandwidth occupation.

message length can reach the maximum, $2^{16}$. Obviously much more length can increase the transmission efficiency of redirection message, namely, the bandwidth cross between CE and FE, also to say that the ratio of bandwidth occupation is direct to *Length* to some extent. However when the network scale sharply goes upward, very large quantity of Network LSA appears and more exchanging time will be needed, finally the average bandwidth occupation will decrease. To validate the effect of bandwidth cross between CE and FE, the view of Figure 7 just reflects the analysis above.

Figure 8 shows the convergence time when redirection mechanism is used or not. It is obviously seeing that the latter needs much more convergence time. When the number of routing entries is no more than 300, the two curves are close to each others which indicates that the cost caused by redirection mechanism is trivial. However, when the scale of routing entries continues to increase, the influence of redirection gradually appears and the latter curve becomes more and more steep. In particular, when the number of routing entries reaches 640, the former's convergence time is 18 seconds while the latter's is up to about 50 seconds.

After the complete convergence of OSPF on CE, CE begins to synchronize routing entries with FE via routing transformation adapter module, and the routing synchronization time will be changed with the scale of testing routing entries. Figure 9 gives the variation of synchronization time. When the number of routing entries is no more than 500, the curve represents the character of linearity which is approximated to be a 40 degrees line and each increment of 20 routing entries brings about about 8 ms cost. However, with the continuous increment of testing routing entries, the linearity disappears gradually and the increment of synchronization time speeds up influenced by CPU, memory load, and so forth. When the scale reaches 620, synchronization time is 618 ms which is still tolerant and does not exert great influence on the overall performance.

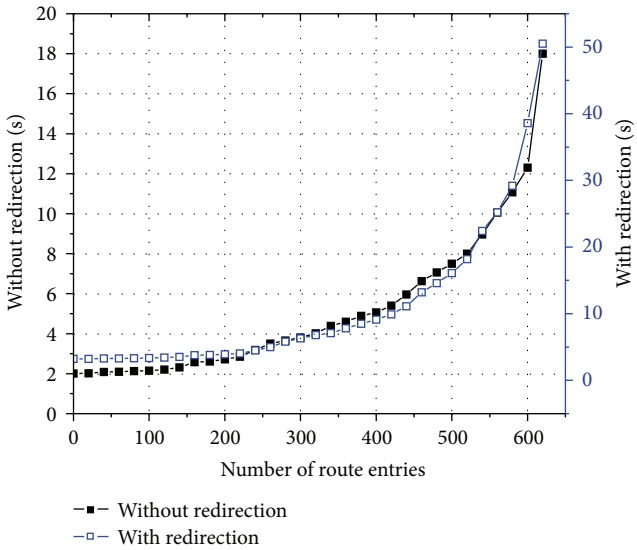



Figure 8: Convergence time of OSPF on CE.

## 5. Conclusion

We originally set out to answer two specific problems in ForCES architecture: (1) How can we make ForCES router exchanging routing protocol message (OSPF for example) with external in the manner of separation between forwarding and Control Element? (2) How can we realize the synchronization of routing from CE to multiple FEs?

We concluded earlier that if external merchant router wants to exchange OSPF packets with ForCES, then ForCES CE needs a way to make OSPF packets go across ForCES FE with no information lost. So we concluded that we should (1) give: IP tunnel-based redirection mechanism, a way to encapsulate OSPF packet as ForCES redirection message, (2) map physical ports in FE into virtual ports in CE,

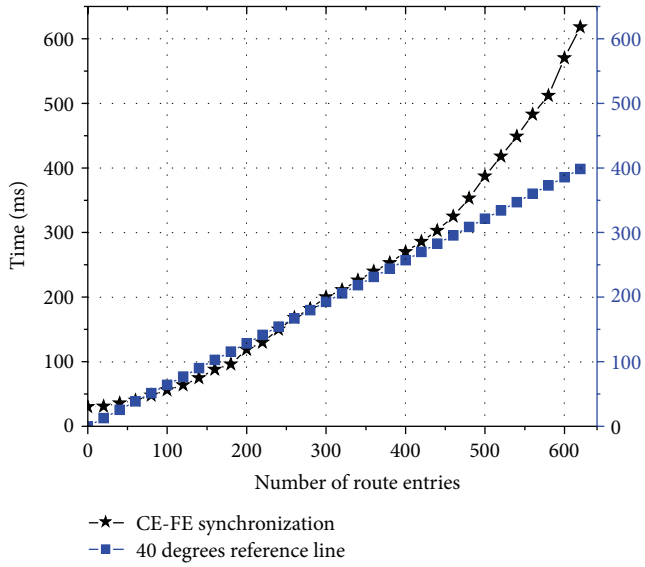



Figure 9: Synchronization time between CE and FE.

and (3) manage and number virtual ports in a special manner with FE ID included, this manner of number will be helpful to locate the outgoing FE during routing synchronization from CE to FE.

## Acknowledgments

This work was supported in part by a Grant from the National Basic Research Program of China (973 Program) (no. 2012CB315902) and the National Natural Science Foundation of China (nos. 61102074 and 61170215).

## References

[1] L. Yang, R. Dantu, and T. Anderson, "Forwarding and Control Element Separation (ForCES) Framework," http://datatracker.ietf.org/doc/rfc3746/.

[2] J. Halpern and J. H. Salim, "Forwarding and Control Element Separation (ForCES) Forwarding Element Model," http://tools.ietf.org/html/rfc5812.

[3] C. Kim, M. Caesar, and J. Rexford, "Seattle: a scalable ethernet architecture for large enterprises," *ACM Transactions on Computer Systems*, vol. 29, no. 1, article 1, 2011.

[4] L. Dong, B. Zhuge, and W. Wang, "Research on logical function blocks in ForCES-based routers," in *Proceedings of the 8th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD '07)*, pp. 172–177, Qingdao, China, August 2007.

[5] A. Ivan, C. Raul, and C. Walter, "Optical switch emulation in programmable software router testbed," *Photonic Network Communications*, vol. 25, no. 1, pp. 10–23, 2013.

[6] D. Drutskoy, E. Keller, and J. Rexford, "Scalable network virtualization in software-defined networks," *IEEE Internet Computing*, vol. 17, no. 2, pp. 20–27, 2013.

[7] Open Network Foundation, https://www.opennetworking.org/.

[8] Software-Defined Networking Research Group (SDNRG), http://irtf.org/sdnrg.

[9] S. Xing, C. Y. Zhang, and G. M. Jiang, "Research on OpenFlow-based SDN technologies," *Journal of Software*, vol. 24, no. 5, pp. 1078–1097, 2013.

[10] M. F. Bari, R. Boutaba, R. Esteves et al., "Data center network virtualization: a survey," *IEEE Communications Surveys and Tutorials*, vol. 15, no. 2, pp. 909–928, 2013.

[11] "Framework for DC Network Virtualization," http://datatracker.ietf.org/doc/draft-ietf-nvo3-framework/.

[12] W. Wang, L. Dong, B. Zhuge et al., "Design and implementation of an open programmable router compliant to IETF forces specifications," in *Proceedings of the 6th International Conference on Networking (ICN '07)*, p. 82, Martinique, France, April 2007.

[13] R. Shea and J. Liu, "Network interface virtualization: challenges and solutions," *IEEE Network*, vol. 26, no. 5, pp. 28–34, 2012.

[14] "ForCES Protocol Specification," http://datatracker.ietf.org/doc/rfc5810/.

[15] C. Li, S. Zhang, and W. Wang, "Network calculus based performance analysis of ForCES SCTP TML in congestion avoidance stage," *Information Technolodgy Journal*, vol. 12, no. 4, pp. 584–593, 2013.

[16] B. Xiao, W. Wang, and M. Gao, "The research and implementation of the layout for ForCES FE-Intra topology," *Journal of Convergence Information Technology*, vol. 7, no. 19, pp. 487–496, 2012.

[17] ForCES LFB Library, http://datatracker.ietf.org/doc/rfc6956/.

[18] J. Doyle and J. Carroll, *Routing TCP/IP*, vol. 1, Pearson Education Press, Upper Saddle River, NJ, USA, 2nd edition, 2007.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 927394, 11 pages
http://dx.doi.org/10.1155/2013/927394

*Research Article*

# High-Order Fuzzy Time Series Model Based on Generalized Fuzzy Logical Relationship

**Wangren Qiu,[1,2] Xiaodong Liu,[2] and Hailin Li[3]**

[1] *Department of Information Engineering, Jingdezhen Ceramic Institute, Jingdezhen 333001, China*
[2] *Research Center of Information and Control, Dalian University of Technology, Dalian 116024, China*
[3] *Institute of Systems Engineering, Dalian University of Technology, Dalian 116024, China*

Correspondence should be addressed to Wangren Qiu; wangrenqiu.cn@gmail.com

Received 22 July 2013; Accepted 17 August 2013

Academic Editor: Zhongmei Zhou



In view of techniques for constructing high-order fuzzy time series models, there are three methods which are based on advanced algorithms, computational methods, and grouping the fuzzy logical relationships, respectively. The last kind model has been widely applied and researched for the reason that it is easy to be understood by the decision makers. To improve the fuzzy time series forecasting model, this paper presents a novel high-order fuzzy time series models denoted as *GTS(M,N)* on the basis of generalized fuzzy logical relationships. Firstly, the paper introduces some concepts of the generalized fuzzy logical relationship and an operation for combining the generalized relationships. Then, the proposed model is implemented in forecasting enrollments of the University of Alabama. As an example of in-depth research, the proposed approach is also applied to forecast the close price of Shanghai Stock Exchange Composite Index. Finally, the effects of the number of orders and hierarchies of fuzzy logical relationships on the forecasting results are discussed.

## 1. Introduction

In the last two decades, fuzzy time series approach [1–3] has been widely used for its power of dealing with imprecise knowledge variables in decision making. Many studies have been made to propose new methods or improve forecasting accuracy for fuzzy time series forecasting. For simplifying the computational process, Chen [4] improved Song's methods and presented a simplified forecasting model in 1996. Since the lengths of intervals greatly affect forecasting accuracy in fuzzy time series, Yu and many others [5–10] adjusted the lengths of intervals by the distribution or the optimization technique. In view of higher accuracy of forecasting results, the weighted models concerned with the various recurrences and on chronological order had also been improved [11–15]. In addition, many models based on the conventional fuzzy time series were combined with novel algorithms or technologies. For example, Singh [16–18] proposed some methods to forecast the crop production based on computational method with different parameters. Lee et al. [19–22] presented several models based on the fuzzy time series, genetic algorithm, simulated annealing algorithm, and type-2 fuzzy set to forecast temperature and TAIFEX. Kuo [23–25] firstly introduced the particle swarm optimization (PSO) into the fuzzy time series models for forecasting TAIFEX. Song's [3] and Aladag's models [26, 27] gained more accurate forecasts by employing artificial neural network to determine fuzzy relationships.

Although the first-order fuzzy time series models have a simple structure, they are easy to encounter trouble on explaining more complex relationships. And the first-order models are not able to meet the demand of forecasting involved in multifactors or longterm time series. As compared with the alternative forecasting models, such as ARIMA, Hidden Markov, and ARCH models, there is still much room for higher forecasting accuracy in applying fuzzy time series models. For these reasons, Chen et al. [28–32] proposed some new methods which applied a high-order fuzzy time series model to forecast enrolments. Aladag et al. [9, 26] introduced a high-order model based on feed-forward neural network. Lee et al. [20, 33] also presented some high-order models based on two-factor and genetic-simulated

annealing techniques. Most of time series researchers [18, 22, 34–37] had showed their, respectively, interest in high-order fuzzy time series forecasting models.

In process of forecasting with fuzzy time series models, Fuzzy Logical Relationship (FLR) is one of the most critical factors that influence the forecasting accuracy. To obtain high forecasting accuracy, a lot of efforts have been put into mining the FLRs from fuzzy time series. In view of techniques for partitioning the universe of discourse and constructing the fuzzy logic relationships effectively, the above high-order models consist of three parts. The first one is mining the FLRs by applying some advanced algorithms or theories such as genetic algorithms, rough set, neural networks, type-2 fuzzy set, and simulated annealing algorithm [20, 22, 26, 27, 30, 32, 34, 35]. The second one is the class represented by Singh [16–18] whose models are on the basis of computational method with difference parameters. The last but not least one is the kind of models based on grouping the FLRs represented by [9, 28, 29, 31, 33, 36, 37]. In general, the first kind of hybrid models can get higher forecasting accuracy than the other two. However, the forecasting process of these algorithms is not easy to be understood. Unlike the fuzzy set theory, its procedure and forecasts are not understandable and accountable for most of decision makers. Although the second kind of models has been implemented on a real-life problem of crop production and rice production as well as enrolment forecasts, the models have little, if anything, to do with FLRs in the procedures of forecasting. The model obtains high forecasting accuracy by dividing the intervals to produce accurate localizations of the forecasting values. With regard to the third kind of models, the procedures of mining FLRs and forecasting principles are based solely on the FLRs sets. The forecasting procedure and principles are obvious and clear to fuzzy time series researchers and easy to be understood by the decision makers.

For these reasons, this paper proposes a high-order fuzzy time series model based on generalized fuzzy logical relationships [38]. The process of creating relationships' matrices and finding out the patterns of time series fluctuations is carried out on the basis of understandable fuzzy rules. Of the above three kinds of models, the proposed belongs to the third. There are three reasons for Hwang's [28] and Chen's models [29, 31] to be chosen as the counterparts for comparing the single-factor forecasting results with determinate length of interval. The first reason is that the models of Chen's [29] and Li's [36] are similar in finding the most appropriate forecasting principle with state-transition analysis and backtracking scheme. The second is that the models of Li et al. [36] and Lee et al. [33] aim at multifactor forecasting problems, and the last is because models [9, 37] are improved by finding an optimal interval length. As regards the experiment data sets, two data sets were used for the empirical analysis: the enrolments of the University of Alabama and the close price of Shanghai Stock Exchange Composite Index (SSECI). In view of the three criteria of evaluations, the root mean squared error, mean absolute error, and mean absolute percentage error, the proposed method gets more satisfactory forecasts than the counterparts.

The rest of this paper is organized as follows. In Section 2, we briefly review the concepts of fuzzy time series. In Section 3, a new model based on high-order generalized fuzzy logical relationships is implemented on the procedure of forecasting enrolments. In Section 4, we compare the average forecasting accuracy rates of the proposed method with the methods presented in [28, 29, 31]. The effects of parameters on forecasting accuracy are also discussed in this section. Conclusions and future works are given in Section 5.

## 2. Preliminaries

In view of making our exposition self-contained, this section reviews some definitions and the framework of fuzzy time series forecasting models. Followed with some related definitions of generalized fuzzy logical relationship, the framework [1–3] is summarized in this section.

*Definition 1* (see [39, 40]). A fuzzy set A of the universe of discourse $U$, $U = \{u_1, u_2, \ldots, u_n\}$, is defined as follows:

$$A = \frac{f_A(u_1)}{u_1} + \frac{f_A(u_2)}{u_2} + \cdots + \frac{f_A(u_n)}{u_n}, \tag{1}$$

where $f_A$ is the membership function of the fuzzy set $A$, $f_A : U \to [0, 1]$, $f_A(u_i)$ denotes the membership degrees in the fuzzy set $A$, $1 \le i \le n$.

*Definition 2.* Let $Y(t)$ $(t = \ldots, 0, 1, 2, \ldots)$ be the universe of discourse in which fuzzy sets $f_i(t)$ $(i = 1, 2, \ldots)$ are defined. Let $F(t)$ be a collection of $f_i(t)$. Then, $F(t)$ is called a fuzzy time series on $Y(t)$.

*Definition 3.* Let $F(t)$ be a fuzzy time series. If $F(t)$ is caused by $F(t, 1), \ldots, F(t, M)$, then the fuzzy logical relationship is represented by $F(t, M), F(t, M-1), \ldots, F(t, 1) \to F(t)$; and it is called the $M$th-order fuzzy time series forecasting model.

*Definition 4.* Let $F(t-1) = A_i$ and $F(t) = A_j$. The relationship between two consecutive observations, $F(t)$ and $F(t-1)$, is referred to as a fuzzy logical relationship (FLR) and denoted by $A_i \to A_j$, where $A_i$ is called the left-hand side (LHS) and $A_j$ the right-hand side (RHS) of the FLR.

*Definition 5.* Let $f_A(t-M) = (\mu_1(t-M), \mu_2(t-M), \ldots, \mu_n(t-M))$, $f_A(t) = (\mu_1(t), \mu_2(t), \ldots, \mu_n(t))$. If $\overline{\mu}_i^{t-M}$ and $\overline{\mu}_j^{t}$ are the maximum values of $(\mu_1(t), \mu_2(t), \ldots, \mu_n(t))$ and $(\mu_1(t), \mu_2(t), \ldots, \mu_n(t))$, respectively, then let $\overline{A}_i^{t-M} \to \overline{A}_j^{t}$ be called the $M$-order first principal fuzzy relationship, noted as $GTS(M, 1)$ (generalized fuzzy logical relationship). If $\overline{\mu}_i^{t-M}$ is the $N$th maximum value of $(\mu_1(t-M), \mu_2(t-M), \ldots, \mu_n(t-M))$, then $\overline{A}_i^{t} \to \overline{A}_j^{t+1}$ e called the $M$th-order $N$th-principal fuzzy logical relationship noted as $GTS(M, N)$.

From Definition 5, the fuzzy logical relationship is more general than that of conventional fuzzy time series model. In fact, the logical relationship is that of conventional models when $N = 1$, and the forecasting rules are obtained from grouping these relationships. We then named it generalized

fuzzy logical relationship. According to Definition 4, all fuzzy logical relationships in the training data set can be further grouped together into different fuzzy logical relationship groups according to the same left-hand sides of the fuzzy logical relationship. For given $M$ and $N$, the fuzzy logical relationships can be grouped into $M \times N$ matrices denoted as $R^{(k,l)}$ $(k = 1, 2, \ldots, M, l = 1, 2, \ldots, N)$ with the group method proposed by Lee et al. [13]. Here, $r^{(i,j)}$ $(i, j \in \{1, 2, \ldots, n\})$, the element of matrix $R^{(k,l)}$, is the number of fuzzy logical relationships $A_i^{(t-k,l)} \rightarrow A_j$.

Then, there are $N$ fuzzy logical relationships matrices for a given training data set. To forecast time series with these generalized fuzzy logical relationship matrices, we defined the intersection operation as follows.

*Definition 6.* Let $A_{t_j}^{(M,j)}$ represent the LHSs of FLRG in the $M$-order $j$th-principal fuzzy logical relationship at time $t$. Let $R^{(M,j)}(t_j, i)$ be the number of FLRG $A_{t_j} \rightarrow A_i$ in the $M$th-order $j$th-principal fuzzy logical relationship, $(j = 1, 2, \ldots, N; i, t_j = 1, 2, \ldots, n)$. To compute the logical relationships between $N$ FLRGs, the intersection operator $\wedge_N$ is defined as

$$\begin{aligned}\wedge_N &\left(A_{t_1}^{(M,1)}, \ldots, A_{t_j}^{(M,j)}, \ldots, A_{t_N}^{(M,N)}\right) \\ &= \left(\min_{1\le j\le N} R^{(M,j)}\left(t_j, 1\right), \ldots, \min_{1\le j\le N} R^{(M,j)}\left(t_j, i\right), \right. \\ &\qquad \left. \ldots, \min_{1\le j\le N} R^{(M,j)}\left(t_j, n\right)\right).\end{aligned} \tag{2}$$

Based on the above definitions, this paper presents a high-order fuzzy time series model in the following section.

## 3. Proposed Model

*3.1. Procedure of $GTS(M, N)$.* In this section, we present a new forecasting method based on high-order and generalized fuzzy logical relationships. Since the proposed model is related to the number of orders denoted by $M$ and hierarchies of principal fuzzy relationship denoted by $N$, we name the proposed model $GTS(M, N)$. In other words, $GTS(M, N)$ means an $M$-order fuzzy time series model based on $N$-principal fuzzy logical relationships.

*Step 1.* Define the universe of discourse and intervals for rules of extraction. The universe of discourse can be defined as $U =$ [starting, ending]. According to equal length of intervals, $U$ is partitioned into several intervals equally. For example, $U = \{u_1, u_2, \ldots, u_n\}$, $m_i$ is the midpoint of $u_i$ whose corresponding fuzzy set is $A_i$ $(i = 1, 2, \ldots, n)$.

*Step 2.* Define fuzzy sets based on the universe of discourse and fuzzify the historical data. The fuzzy set $A_i$ would be expressed as $A_i = (a_{i1}, a_{i2}, \ldots, a_{in})$, where $a_{ij} \in [0, 1]$, which indicates the membership degree of $u_j$ in $A_i$. The historical and observed data are fuzzified according to the definition of fuzzy sets. For example, a datum is fuzzified to $A_j$, when the maximal membership degree of the datum is the $j$th number. In other words, if $a_{ij} = \max\{a_{i1}, a_{i2}, \ldots, a_{in}\}$, then the data at time $t$ should be classified into the $j$th class. In this paper, the fuzzy sets are defined with triangular fuzzy function showed by formula (3).

Consider

$$\begin{aligned}
A_1 &= \frac{1}{u_1} + \frac{0.5}{u_2} + \frac{0}{u_3} + \cdots + \frac{0}{u_{n-1}} + \frac{0}{u_n}, \\
A_2 &= \frac{0.5}{u_1} + \frac{1}{u_2} + \frac{0.5}{u_3} + \frac{0}{u_4} + \cdots + \frac{0}{u_i} + \cdots + \frac{0}{u_{n-1}} + \frac{0}{u_n}, \\
&\vdots \\
A_i &= \frac{0}{u_1} + \cdots + \frac{0}{u_{i-2}} + \frac{0.5}{u_{i-1}} + \frac{1}{u_i} + \frac{0.5}{u_{i+1}} + \frac{0}{u_{i+2}} + \cdots + \frac{0}{u_n}, \\
&\vdots \\
A_n &= \frac{0}{u_1} + \frac{0}{u_2} + \frac{0}{u_3} + \cdots + \frac{0}{u_i} + \cdots + \frac{0}{u_{n-2}} + \frac{0.5}{u_{n-1}} + \frac{1}{u_n}.
\end{aligned} \tag{3}$$

The membership degree of the value $x_t$ at time $t$ in $A_i$ $(i = 1, 2, \ldots, n)$ is defined by formula (4). Consider

$$\begin{aligned}\mu_{A_i}&(x_t) \\ &= \begin{cases} 1, & \text{if } i = 1 \text{ and } x_t \le m_1, \\ 1, & \text{if } i = n \text{ and } x_t \ge m_n, \\ \max\left\{0, 1 - \dfrac{\left|x_i - m_i\right|}{2 \times l_{\text{in}}}\right\}, & \text{others,} \end{cases}\end{aligned} \tag{4}$$

where $x_t$ is the observed value at time $t$ and $l_{\text{in}}$ is the length of interval.

*Step 3.* Establish the fuzzy logical relationships based on the orders and hierarchies of principal fuzzy logical relationship. Given the sample data set and the definition of fuzzy sets, all fuzzy logical relationships between two consecutive data can be created. To forecast the time series, the fuzzy logical relationship matrix must be created in this step based on the fuzzy logical relationships.

Among many different methods, the method proposed by Lee in [13] is chosen in this paper. For example, the fuzzy logical relationships of a $GTS(M, N)$ model can be grouped into $M \times N$ relationship matrices denoted by $R^{(k,l)}$ $(k = 1, 2, \ldots, M; l = 1, 2, \ldots, N)$.

*Step 4.* Forecasting model. Let $F(t-k) = (\mu_1(t-k), \mu_2(t-k), \ldots, \mu_n(t-k))$, and let $\mu_{t_l}(t-k)$ be the $l$th max membership degree, respectively. By Definition 6, we have the intersection fuzzy logical relationship $\wedge_N(A_{t_1}^{(k,1)}, \ldots, A_{t_l}^{(k,l)}, \ldots, A_{t_N}^{(k,N)})$,

Table 1: Membership degrees of enrollment with respect to fuzzy sets.

| Year | Value | $\mu_{A_1}$ | $\mu_{A_2}$ | $\mu_{A_3}$ | $\mu_{A_4}$ | $\mu_{A_5}$ | $\mu_{A_6}$ | $\mu_{A_7}$ | Fuzzified |
|---|---|---|---|---|---|---|---|---|---|
| 1971 | 13055 | 1.0000 | 0.2775 | 0 | 0 | 0 | 0 | 0 | $A_1; A_2$ |
| 1972 | 13563 | 0.9685 | 0.5315 | 0.0315 | 0 | 0 | 0 | 0 | $A_1; A_2$ |
| 1973 | 13867 | 0.8165 | 0.6835 | 0.1835 | 0 | 0 | 0 | 0 | $A_1; A_2$ |
| 1974 | 14696 | 0.4020 | 0.9020 | 0.5980 | 0.0980 | 0 | 0 | 0 | $A_2; A_3$ |
| 1975 | 15460 | 0.0200 | 0.5200 | 0.9800 | 0.4800 | 0 | 0 | 0 | $A_3; A_2$ |
| 1976 | 15311 | 0.0945 | 0.5945 | 0.9055 | 0.4055 | 0 | 0 | 0 | $A_3; A_2$ |
| 1977 | 15603 | 0 | 0.4485 | 0.9485 | 0.5515 | 0.0515 | 0 | 0 | $A_3; A_4$ |
| 1978 | 15861 | 0 | 0.3195 | 0.8195 | 0.6805 | 0.1805 | 0 | 0 | $A_3; A_4$ |
| 1979 | 16807 | 0 | 0 | 0.3465 | 0.8465 | 0.6535 | 0.1535 | 0 | $A_4; A_5$ |
| 1980 | 16919 | 0 | 0 | 0.2905 | 0.7905 | 0.7095 | 0.2095 | 0 | $A_4; A_5$ |
| 1981 | 16388 | 0 | 0.0560 | 0.5560 | 0.9440 | 0.4440 | 0 | 0 | $A_4; A_3$ |
| 1982 | 15433 | 0.0335 | 0.5335 | 0.9665 | 0.4665 | 0 | 0 | 0 | $A_3; A_2$ |
| 1983 | 15497 | 0.0015 | 0.5015 | 0.9985 | 0.4985 | 0 | 0 | 0 | $A_3; A_2$ |
| 1984 | 15145 | 0.1775 | 0.6775 | 0.8225 | 0.3225 | 0 | 0 | 0 | $A_3; A_2$ |
| 1985 | 15163 | 0.1685 | 0.6685 | 0.8315 | 0.3315 | 0 | 0 | 0 | $A_3; A_2$ |
| 1986 | 15984 | 0 | 0.2580 | 0.7580 | 0.7420 | 0.2420 | 0 | 0 | $A_3; A_4$ |
| 1987 | 16859 | 0 | 0 | 0.3205 | 0.8205 | 0.6795 | 0.1795 | 0 | $A_4; A_5$ |
| 1988 | 18150 | 0 | 0 | 0 | 0.1750 | 0.6750 | 0.8250 | 0.3250 | $A_6; A_5$ |
| 1989 | 18970 | 0 | 0 | 0 | 0 | 0.2650 | 0.7650 | 0.7350 | $A_6; A_7$ |
| 1990 | 19328 | 0 | 0 | 0 | 0 | 0.0860 | 0.5860 | 0.9140 | $A_7; A_6$ |
| 1991 | 19337 | 0 | 0 | 0 | 0 | 0.0815 | 0.5815 | 0.9185 | $A_7; A_6$ |
| 1992 | 18876 | 0 | 0 | 0 | 0 | 0.3120 | 0.8120 | 0.6880 | $A_6; A_7$ |

and then the $k$th forecasts $F^k(t)$ is conducted by formula (5). Consider

$$F^k(t) = \wedge_N \left(A_{t_1}^{(k,1)}, \ldots, A_{t_l}^{(k,l)}, \ldots, A_{t_N}^{(k,N)}\right) \circ \left(m_1, m_2, \ldots, m_n\right)^T, \quad (5)$$

where "$\circ$" is a composition operation for forecasting with the following principles:

(1) if the sum of $\wedge_N(A_{t_1}^{(k,1)}, \ldots, A_{t_l}^{(k,l)}, \ldots, A_{t_N}^{(k,N)})$ equals to 0, then the forecasted value is $m_{t_1}$, and the midpoint of the interval corresponding to $A_{t_1}$;

(2) otherwise, the forecasted value is the weighting aggregate of $m_{t_1}, \ldots, m_{t_N}$.

Then, for a given $M$, there are $M$ forecasts for time $t$. The conclusive forecasting value for time $t$ can be obtained by the following formula:

$$\overline{F}(t) = \sum_{k=1}^{M} F^k(t) * w_k, \quad (6)$$

where $w_k$ $(k = 1, 2, \ldots, M)$ is the adjustment parameter for the $k$th forecast; the parameter also can be obtained by minimizing the RMSE or other criteria of evaluations for the training data set.

*3.2. Computation of GTS$(M, N)$ on Forecasting Enrollments.* Since most of the conventional models have been presented for forecasting the historical enrolments of the University of Alabama, in this section, we present stepwise procedures of the proposed method for forecasting the time series data with $M = 3$ and $N = 2$. The historical enrolments of the University of Alabama from 1971 to 1992 are shown in Table 1.

*Step 1.* Partitioning the universe of discourse $U$ into seven intervals: $u_1, u_2, \ldots, u_7$, where $u_1 = [13000, 14000]$, $u_2 = [14000, 15000]$, $u_3 = [15000, 16000]$, $u_4 = [16000, 17000]$, $u_5 = [17000, 18000]$, $u_6 = [18000, 19000]$, and $u_7 = [19000, 20000]$; their midpoints are 13500, 14500, 15500, 16500, 17500, 18500, and 19500, respectively.

*Step 2.* Let $A_1$ = "not many," $A_2$ = "not too many," $A_3$ = "many," $A_4$ = "many many," $A_5$ = "very many," $A_6$ = "too many," and $A_7$ = "too many many." $A_1, A_2, \ldots, A_7$ are fuzzy sets corresponding to linguistic values for "enrolments."

With triangular-shaped membership function defined by formula (4), the fuzzy sets and all observations are defined by the last column of Table 1 by formula (3). Thus, the fuzzy logical relationships corresponding to given $M$ and $N$ are listed in Table 2.

*Step 3.* Divide the derived fuzzy logical relationships into groups based on the states of the enrollments of fuzzy logical relationships.

Table 2: Fuzzy logical relationships with $M = 3$ and $N = 2$.

| Year | fuzzified | $k = 1$ $l = 1; l = 2$ | $k = 2$ $l = 1; l = 2$ | $k = 3$ $l = 1; l = 2$ |
|---|---|---|---|---|
| 1971 | $A_1; A_2$ | | | |
| 1972 | $A_1; A_2$ | $A_1 \longrightarrow A_1; A_2 \longrightarrow A_1$ | | |
| 1973 | $A_1; A_2$ | $A_1 \longrightarrow A_1; A_2 \longrightarrow A_1$ | $A_1 \longrightarrow A_1; A_2 \longrightarrow A_1$ | |
| 1974 | $A_2; A_3$ | $A_1 \longrightarrow A_2; A_2 \longrightarrow A_2$ | $A_1 \longrightarrow A_2; A_2 \longrightarrow A_2$ | $A_1 \longrightarrow A_2; A_2 \longrightarrow A_2$ |
| 1975 | $A_3; A_2$ | $A_2 \longrightarrow A_3; A_3 \longrightarrow A_3$ | $A_1 \longrightarrow A_3; A_2 \longrightarrow A_3$ | $A_1 \longrightarrow A_3; A_2 \longrightarrow A_3$ |
| 1976 | $A_3; A_2$ | $A_3 \longrightarrow A_3; A_2 \longrightarrow A_3$ | $A_2 \longrightarrow A_3; A_3 \longrightarrow A_3$ | $A_1 \longrightarrow A_3; A_2 \longrightarrow A_3$ |
| 1977 | $A_3; A_4$ | $A_3 \longrightarrow A_3; A_2 \longrightarrow A_3$ | $A_3 \longrightarrow A_3; A_2 \longrightarrow A_3$ | $A_2 \longrightarrow A_3; A_3 \longrightarrow A_3$ |
| 1978 | $A_3; A_4$ | $A_3 \longrightarrow A_3; A_4 \longrightarrow A_3$ | $A_3 \longrightarrow A_3; A_2 \longrightarrow A_3$ | $A_3 \longrightarrow A_3; A_2 \longrightarrow A_3$ |
| 1979 | $A_4; A_5$ | $A_3 \longrightarrow A_4; A_4 \longrightarrow A_4$ | $A_3 \longrightarrow A_4; A_4 \longrightarrow A_4$ | $A_3 \longrightarrow A_4; A_2 \longrightarrow A_4$ |
| 1980 | $A_4; A_5$ | $A_4 \longrightarrow A_4; A_5 \longrightarrow A_4$ | $A_3 \longrightarrow A_4; A_4 \longrightarrow A_4$ | $A_3 \longrightarrow A_4; A_4 \longrightarrow A_4$ |
| 1981 | $A_4; A_3$ | $A_4 \longrightarrow A_4; A_5 \longrightarrow A_4$ | $A_4 \longrightarrow A_4; A_5 \longrightarrow A_4$ | $A_3 \longrightarrow A_4; A_4 \longrightarrow A_4$ |
| 1982 | $A_3; A_2$ | $A_4 \longrightarrow A_3; A_3 \longrightarrow A_3$ | $A_4 \longrightarrow A_3; A_5 \longrightarrow A_3$ | $A_4 \longrightarrow A_3; A_5 \longrightarrow A_3$ |
| 1983 | $A_3; A_2$ | $A_3 \longrightarrow A_3; A_2 \longrightarrow A_3$ | $A_4 \longrightarrow A_3; A_3 \longrightarrow A_3$ | $A_4 \longrightarrow A_3; A_5 \longrightarrow A_3$ |
| 1984 | $A_3; A_2$ | $A_3 \longrightarrow A_3; A_2 \longrightarrow A_3$ | $A_3 \longrightarrow A_3; A_2 \longrightarrow A_3$ | $A_4 \longrightarrow A_3; A_3 \longrightarrow A_3$ |
| 1985 | $A_3; A_2$ | $A_3 \longrightarrow A_3; A_2 \longrightarrow A_3$ | $A_3 \longrightarrow A_3; A_2 \longrightarrow A_3$ | $A_3 \longrightarrow A_3; A_2 \longrightarrow A_3$ |
| 1986 | $A_3; A_4$ | $A_3 \longrightarrow A_3; A_2 \longrightarrow A_3$ | $A_3 \longrightarrow A_3; A_2 \longrightarrow A_3$ | $A_3 \longrightarrow A_3; A_2 \longrightarrow A_3$ |
| 1987 | $A_4; A_5$ | $A_3 \longrightarrow A_4; A_4 \longrightarrow A_4$ | $A_3 \longrightarrow A_4; A_2 \longrightarrow A_4$ | $A_3 \longrightarrow A_4; A_2 \longrightarrow A_4$ |
| 1988 | $A_6; A_5$ | $A_4 \longrightarrow A_6; A_5 \longrightarrow A_6$ | $A_3 \longrightarrow A_6; A_4 \longrightarrow A_6$ | $A_3 \longrightarrow A_6; A_2 \longrightarrow A_6$ |
| 1989 | $A_6; A_7$ | $A_6 \longrightarrow A_6; A_5 \longrightarrow A_6$ | $A_4 \longrightarrow A_6; A_5 \longrightarrow A_6$ | $A_3 \longrightarrow A_6; A_4 \longrightarrow A_6$ |
| 1990 | $A_7; A_6$ | $A_6 \longrightarrow A_7; A_7 \longrightarrow A_7$ | $A_6 \longrightarrow A_7; A_5 \longrightarrow A_7$ | $A_4 \longrightarrow A_7; A_5 \longrightarrow A_7$ |
| 1991 | $A_7; A_6$ | $A_7 \longrightarrow A_7; A_6 \longrightarrow A_7$ | $A_6 \longrightarrow A_7; A_7 \longrightarrow A_7$ | $A_6 \longrightarrow A_7; A_5 \longrightarrow A_7$ |
| 1992 | $A_6; A_7$ | $A_7 \longrightarrow A_6; A_6 \longrightarrow A_6$ | $A_7 \longrightarrow A_6; A_6 \longrightarrow A_6$ | $A_6 \longrightarrow A_6; A_7 \longrightarrow A_6$ |

In this paper, we use the method of Lee [13] to construct the fuzzy logical relationships matrix. For example, let $k = 1$ and $l = 1$. The set of fuzzy logical relationships is listed as

$A_1 \to A_1, A_1 \to A_1, A_1 \to A_2, A_2 \to A_3, A_3 \to A_3, A_3 \to A_3, A_3 \to A_3, A_3 \to A_4, A_4 \to A_4,$

$A_4 \to A_4, A_4 \to A_3, A_3 \to A_3, A_3 \to A_3, A_3 \to A_3, A_3 \to A_3, A_3 \to A_4, A_4 \to A_6,$

$A_6 \to A_6, A_6 \to A_7, A_7 \to A_7, A_7 \to A_6,$ which are all of FLRGs in the third column of Table 2.

And they would be grouped and weighted by the recurrent fuzzy relationships as follows:

Group 1: $A_1 \to A_1$ with weight 2, $A_1 \to A_2$ with weight 1;

Group 2: $A_2 \to A_3$ with weight 1;

Group 3: $A_3 \to A_3$ with weight 7, $A_3 \to A_4$ with weight 2;

Group 4: $A_4 \to A_4$ with weight 2, $A_4 \to A_3$ with weight 1; $A_4 \to A_6$ with weight 1,

Group 5: $A_6 \to A_6$ with weight 1, $A_6 \to A_7$ with weight 1;

Group 6: $A_7 \to A_7$ with weight 1, $A_7 \to A_6$ with weight 1.

The fuzzy logical relationship matrix then is $R^{(1,1)}$. Based on the fourth, fifth, sixth, seventh, and eighth column of Table 2, $R^{(1,2)}$, $R^{(2,1)}$, $R^{(2,2)}$, $R^{(3,1)}$, and $R^{(3,2)}$ can also be obtained in the similar way. These six fuzzy logical relationship matrices have been listed as follows:

$$R^{(1,1)} = \begin{bmatrix} 2 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 7 & 2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 2 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix},$$

$$R^{(1,2)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 1 & 6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

$$R^{(2,1)} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5 & 3 & 0 & 1 & 0 \\ 0 & 0 & 2 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \quad (7)$$

$$R^{(2,2)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 6 & 1 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{8}$$

$$R^{(3,1)} = \begin{bmatrix} 0 & 1 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 4 & 0 & 2 & 0 \\ 0 & 0 & 3 & 2 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{9}$$

$$R^{(3,2)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 5 & 2 & 0 & 1 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 1 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

*Step 4.* Calculate the forecasts. To carry out the calculations by the proposed principles, we illustrate the forecasting process of enrolment of 1976 as follows. From Table 1, the fuzzy set of enrolment of 1975 is (0.02, 0.52, 0.98, 0.48, 0, 0, 0). The two maximum membership attributes are the third and the second. Then, $k = 1$, the third row of $R^{(1,1)}$ is (0, 0, 7, 2, 0, 0, 0) and the second row of $R^{(1,2)}$ is (2, 1, 6, 0, 0, 0, 0) from (7). According to Definition 6, we have $\wedge_2(A^{(1,1)}_{1975_1}, A^{(1,2)}_{1975_2}) = (0, 0, 1, 0, 0, 0, 0)$.

According to the second principle listed in Section 3.1, $F^1(1976)$ then is $m_3$, that is, 15500, which is the middle point of $A_3$.

When $k = 2$, the fuzzy set of 1974s is (0.402, 0.902, 0.598, 0.098, 0, 0, 0) shown in Table 1. The two maximum membership attributes are the second and the third. The second row of $R^{(2,1)}$ then is (0, 0, 1, 0, 0, 0, 0) and the third row of $R^{(2,2)}$ is (0, 0, 2, 0, 0, 0, 0) from (8). With Definition 6, we have $\wedge_2(A^{(2,1)}_{1975_1}, A^{(2,2)}_{1975_2}) = (0, 0, 1, 0, 0, 0, 0)$. By the first principle listed in Section 3.1, $F^2(1976)$ also is 15500.

In the similar way, from Table 1 and with $k = 3$, we can see that the fuzzy set of 1973s is (0.8165, 0.6835, 0.1835, 0, 0, 0, 0). The two maximum membership attributes are the first and second ones. The first row of $R^{(3,1)}$ is (0, 1, 2, 0, 0, 0, 0) and the second row of $R^{(3,2)}$ is (0, 1, 5, 2, 0, 0, 0) from (9). With Definition 6, we have $\wedge_2(A^{(3,1)}_{1975_1}, A^{(3,2)}_{1975_2}) = (0, 1, 2, 0, 0, 0, 0)$. With the second principle listed in Section 3.1 in mind, $F^3(1976)$ equals to $(1/3) \times 14500 + (2/3) \times 15500 = 15167$.

At last, the forecasted value of 1976, that is, $F(1976)$, is 15590 which equals to $0.9130 \times 15500 + 0.0790 \times 15500 + 0.0142 \times 1516$. Also $(w_1, w_2, w_3) = (0.9130,\ 0.0790,\ 0.0142)$ is the regress result of the enrolment data. Table 3 shows the actual and the forecasted enrolments of the data set.

## 4. Empirical Analysis

*4.1. Data Description.* To demonstrate the effectiveness of the proposed models, amounts of data are needed. Here, the enrolments and SSECI are used as the illustration data sets for the empirical analysis.

There are some causes for the two time series to be the subjects in our experiment. There are two causes for the choice of enrolment data at the University of Alabama. The first one is that most of the fuzzy time series studies have taken this well-known time series as their experiments. Thus, there are a lot of studies that can be used for our reference. The other is that it is simple and easy to display the process of the proposed model. Since time series models have been used to make predictions in the areas of stock price forecasting for many years, the daily SSECI covering the period from 1997 to 2006 is adopted for further experiment.

For the first data, the determined length of the seven intervals is 1000. All of the forecasting results, from one order to ten orders, are compared with those of the conventional fuzzy time series models based on fuzzy logical relationships.

To discuss the effects of parameters $M$ and $N$, the order number and hierarchies of fuzzy logical relationships, on the forecasting results, the ten-order (time-lag periods), from one order to ten orders, models are performed on the second data set with ten different lengths of intervals, that is, $30, 60, 90, \ldots, 300$. The forecasting results obtained by all of the high-order models are compared in terms of three evaluation criteria reviewed in following subsection.

*4.2. Criteria of Evaluation.* In statistics, the root mean squared error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE) are three typical ways to quantify the difference between values implied by an estimator and the actual values of the quantity being estimated. MSE is a risk function for measuring the average of the squares of the difference. For an unbiased estimator, the MSE is the variance, and the RMSE is the square root of the variance known as the standard error. Furthermore, RMSE is superior to MSE for the reason that its scale is the same as the forecasts. Thus, we take RMSE as the first representative of the size of an average error. As an average of the absolute percent errors, MAPE serves as a criterion for the comparisons of forecasting results in the paper. Some comparisons of accuracy in the forecasted values of our proposed models with other models are made on the basis of the three criteria.

*4.3. Performance Evaluation.* In Table 4, this study compares the RMSE, MAE, and MAPE forecasting value of the proposed method and the counterparts [28, 29, 31] on the enrolment experiment. Table 4 shows that the proposed method gets smaller RMSE, MAE, and MAPE than Hwang's model [28], and also smaller than Chen's model [31] of 2011 in most cases. Although there is a fly in the ointment, that is, the proposed model not always gets a higher average forecasting accuracy rate than the model [29] of 2002, these results are improving as the orders increase. In summary, the results suggest that the proposed model obtains better forecasts as

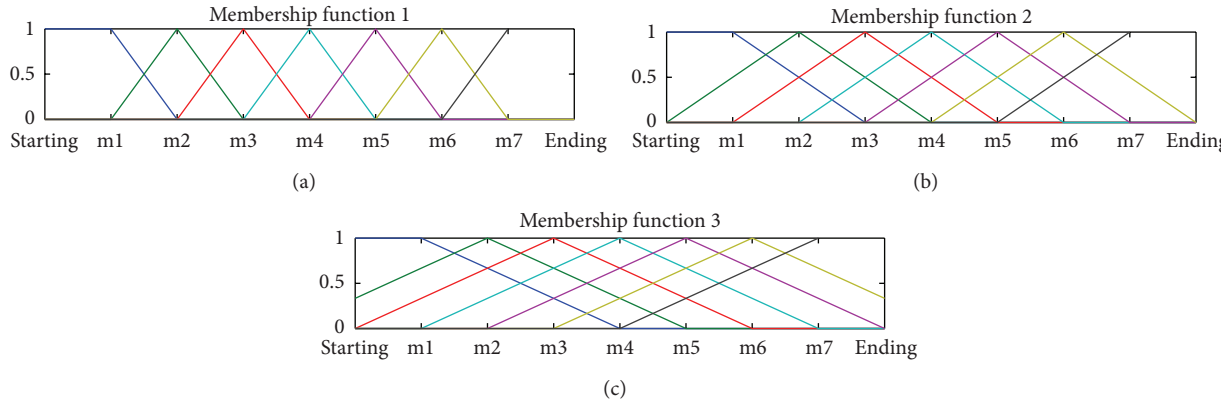



Figure 1: Three membership functions for forecasting SHI of 2003.

Table 3: Forecasts of enrollment year actual data.

| Year | Value | $M = 1$ | | $M = 2$ | | $M = 3$ | |
|---|---|---|---|---|---|---|---|
| | | $N = 1$ | $N = 2$ | $N = 1$ | $N = 2$ | $N = 1$ | $N = 2$ |
| 1971 | 13055 | — | — | — | — | — | — |
| 1972 | 13563 | 13901 | 13902 | — | — | — | — |
| 1973 | 13867 | 13901 | 13902 | 13980 | 13968 | — | — |
| 1974 | 14696 | 13901 | 13902 | 13980 | 13968 | 14062 | 13990 |
| 1975 | 15460 | 15576 | 15576 | 15491 | 15510 | 15533 | 15511 |
| 1976 | 15311 | 15799 | 15576 | 15792 | 15590 | 15760 | 15590 |
| 1977 | 15603 | 15799 | 15576 | 15858 | 15604 | 15812 | 15608 |
| 1978 | 15861 | 15799 | 16246 | 15858 | 16221 | 15915 | 16228 |
| 1979 | 16807 | 15799 | 16246 | 15858 | 16342 | 15915 | 16347 |
| 1980 | 16919 | 16832 | 17251 | 16790 | 17267 | 16821 | 17272 |
| 1981 | 16388 | 16832 | 17251 | 16823 | 17240 | 16832 | 17245 |
| 1982 | 15433 | 16832 | 15576 | 16823 | 15698 | 16821 | 15719 |
| 1983 | 15497 | 15799 | 15576 | 15891 | 15590 | 15915 | 15614 |
| 1984 | 15145 | 15799 | 15576 | 15858 | 15604 | 15904 | 15608 |
| 1985 | 15163 | 15799 | 15576 | 15858 | 15604 | 15915 | 15620 |
| 1986 | 15984 | 15799 | 15576 | 15858 | 15604 | 15915 | 15620 |
| 1987 | 16859 | 15799 | 16246 | 15858 | 16221 | 15915 | 16228 |
| 1988 | 18150 | 16832 | 17251 | 16790 | 17267 | 16821 | 17260 |
| 1989 | 18970 | 19093 | 18591 | 18863 | 18473 | 18816 | 18463 |
| 1990 | 19328 | 19093 | 19596 | 19161 | 19614 | 18900 | 19581 |
| 1991 | 19337 | 19093 | 19094 | 19161 | 19151 | 19132 | 19163 |
| 1992 | 18876 | 19093 | 19094 | 19061 | 19070 | 19101 | 19070 |
| MAE | | 473.365 | 372.0021 | 471.3957 | 375.308 | 501.1709 | 390.2802 |
| RMSE | | 625.4334 | 447.4897 | 631.6806 | 448.8908 | 643.2859 | 459.7683 |
| MAPE | | 0.0293 | 0.0227 | 0.0291 | 0.0227 | 0.0307 | 0.0236 |

the orders and hierarchies increase. Unlike this point, this is not the trend in the three counterparts.

Moreover, we also apply the proposed method to handle forecasting the close price of Shanghai stock index of 2003 with $l = 60$. The comparison of the three criteria is listed in Table 5. From this table, we can see that the proposed $GTS(M, N)$ gets the best forecasts of the three counterparts when $M > 5$ and $N > 2$ as well as its better performance than those of Hwang's and Chen's model [28, 29] in all cases, and the shortcoming shown in Table 4 is gone. On the whole, there is a "law" that forecasting errors will be reduced when $M$ or $N$ is increased in the proposed model. However, this trend is not evident for the three counterparts by Tables 4 and 5. In Table 4, the three evaluation criteria of Hwang's and Chen's models [28, 31] are decreasing while those of model [29] are increasing. In contrast, it is obvious that the three evaluation criteria of Hwang's and Chen's model are increasing while those of Chen's model [29] are decreasing from Table 5. The

Table 4: Comparison of forecasting enrollment.

| | Model | $M = 1$ | $M = 2$ | $M = 3$ | $M = 4$ | $M = 5$ | $M = 6$ | $M = 7$ | $M = 8$ | $M = 9$ | $M = 10$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| RMSE | $GTS(M, 1)$ | 625 | 632 | 643 | 641 | 658 | 575 | 562 | **289** | **203** | **191** |
| | $GTS(M, 2)$ | 447 | 449 | 460 | 436 | 448 | 456 | 437 | 413 | 391 | 284 |
| | $GTS(M, 3)$ | 447 | 449 | 460 | 436 | 448 | 456 | 406 | 382 | 365 | 225 |
| | $GTS(M, 4)$ | 447 | 449 | 460 | 434 | 445 | 453 | 423 | 415 | 397 | 253 |
| | $GTS(M, 5)$ | 447 | 449 | 460 | 434 | 445 | 453 | 423 | 415 | 397 | 253 |
| | [28] | 588 | 588 | 578 | 608 | 602 | 620 | 632 | 575 | 554 | 498 |
| | [29] | **292** | **298** | **294** | **299** | **307** | **313** | **323** | 320 | 321 | 311 |
| | [31] | 543 | 543 | 522 | 420 | 456 | 443 | 452 | 463 | 480 | 498 |
| MAE | $GTS(M, 1)$ | 473 | 471 | 501 | 484 | 501 | 366 | 411 | **212** | 326 | **140** |
| | $GTS(M, 2)$ | 372 | 375 | 390 | 365 | 383 | 394 | 406 | 364 | 292 | 218 |
| | $GTS(M, 3)$ | 372 | 375 | 390 | 365 | 383 | 393 | 374 | 336 | 333 | 176 |
| | $GTS(M, 4)$ | 372 | 375 | 390 | 367 | 384 | 393 | 402 | 386 | 333 | 195 |
| | $GTS(M, 5)$ | 372 | 375 | 390 | 367 | 384 | 393 | 402 | 386 | 333 | 195 |
| | [28] | 494 | 494 | 473 | 526 | 527 | 556 | 582 | 506 | 483 | 424 |
| | [29] | **252** | **262** | **256** | **259** | **272** | **277** | **289** | 284 | **282** | 271 |
| | [31] | 441 | 441 | 412 | 316 | 347 | 337 | 340 | 346 | 368 | 390 |
| MAPE | $GTS(M, 1)$ | 0.0293 | 0.0291 | 0.0307 | 0.0294 | 0.0305 | 0.0221 | 0.0246 | **0.0127** | **0.0103** | **0.0087** |
| | $GTS(M, 2)$ | 0.0227 | 0.0227 | 0.0236 | 0.0216 | 0.0227 | 0.0232 | 0.0239 | 0.0216 | 0.0195 | 0.0125 |
| | $GTS(M, 3)$ | 0.0227 | 0.0227 | 0.0236 | 0.0216 | 0.0227 | 0.0232 | 0.0220 | 0.0200 | 0.0176 | 0.0101 |
| | $GTS(M, 4)$ | 0.0227 | 0.0227 | 0.0236 | 0.0217 | 0.0227 | 0.0231 | 0.0236 | 0.0228 | 0.0198 | 0.0111 |
| | $GTS(M, 5)$ | 0.0227 | 0.0227 | 0.0236 | 0.0217 | 0.0227 | 0.0231 | 0.0236 | 0.0228 | 0.0198 | 0.0111 |
| | [28] | 0.0299 | 0.0299 | 0.0283 | 0.0317 | 0.0313 | 0.0329 | 0.0345 | 0.0299 | 0.0285 | 0.0253 |
| | [29] | **0.0153** | **0.0159** | **0.0153** | **0.0154** | **0.0162** | **0.0164** | **0.0170** | 0.0166 | 0.0165 | 0.0158 |
| | [31] | 0.0268 | 0.0268 | 0.0248 | 0.0188 | 0.0203 | 0.0195 | 0.0196 | 0.0198 | 0.0211 | 0.0223 |

The bold data means the minimum error of the models with the same order.

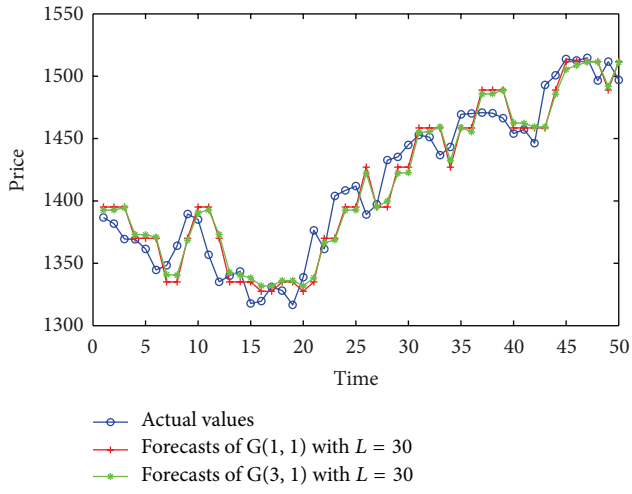



Figure 2: Actual values and forecasts of 2003 with the same $L$.

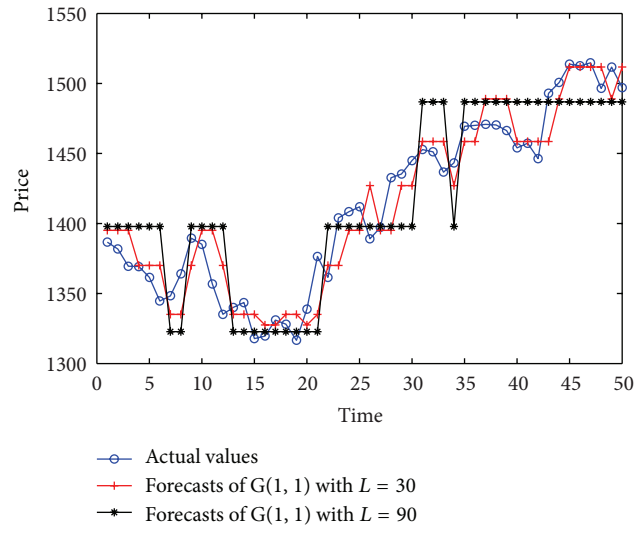



Figure 3: Actual values and forecasts of 2003 with different $L$.

use of different data sets seems to be the best reason to account for this contradiction. From these discussions, we get a conclusion that the proposed model's performance is more reasonable and robust than the three counterparts.

Tables 4 and 5 depict that $GTS(M, 3)$, $GTS(M, 4)$, and $GTS(M, 5)$ have the same results, although larger $M$ or $N$ is more easy to obtain the smaller forecasting errors. In fact, this trend is affected by the definitions of fuzzy sets and membership function. As an in-depth analysis, Figure 1 shows three triangular membership functions. The RMSEs of forecasts of 2003 by $GTS(M, N)$ with $L = 30$ and the three membership functions are listed in Table 6. The table tells us that the more intervals that are concerned in the membership

Table 5: Comparison of forecasting SSECI.

| | Model | $M = 1$ | $M = 2$ | $M = 3$ | $M = 4$ | $M = 5$ | $M = 6$ | $M = 7$ | $M = 8$ | $M = 9$ | $M = 10$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| RMSE | $GTS(M, 1)$ | 22.74 | 21.64 | 21.21 | 20.92 | 20.86 | 20.71 | 20.66 | 20.66 | 20.25 | 20.10 |
| | $GTS(M, 2)$ | 18.32 | 19.04 | 17.90 | 17.92 | 17.69 | 17.56 | 17.54 | 17.55 | 16.95 | 16.88 |
| | $GTS(M, 3)$ | 18.37 | 18.06 | 17.89 | 17.93 | **17.67** | **17.54** | **17.56** | **17.56** | **16.98** | **16.83** |
| | $GTS(M, 4)$ | 18.37 | 18.06 | 17.89 | 17.93 | **17.67** | **17.54** | **17.56** | **17.56** | **16.98** | **16.83** |
| | $GTS(M, 5)$ | 18.37 | 18.06 | 17.89 | 17.93 | **17.67** | **17.54** | **17.56** | **17.56** | **16.98** | **16.83** |
| | [28] | 18.41 | 18.41 | 19.42 | 19.43 | 19.51 | 19.39 | 20.05 | 19.71 | 20.00 | 20.26 |
| | [29] | 21.07 | 21.02 | 20.87 | 20.73 | 20.08 | 19.88 | 19.79 | 19.69 | 18.18 | 18.21 |
| | [31] | **17.56** | **17.56** | **17.81** | **17.78** | 18.38 | 18.31 | 18.67 | 19.25 | 19.47 | 19.60 |
| MAE | $GTS(M, 1)$ | 18.61 | 17.10 | 16.61 | 16.12 | 16.09 | 16.08 | 16.07 | 16.11 | 15.75 | 15.50 |
| | $GTS(M, 2)$ | 14.08 | 13.89 | 13.88 | 13.91 | 13.73 | 13.53 | **13.45** | 13.46 | 13.16 | 13.17 |
| | $GTS(M, 3)$ | 14.17 | 13.87 | 13.81 | 13.85 | 13.67 | 13.45 | **13.46** | **13.45** | **13.13** | **13.06** |
| | $GTS(M, 4)$ | 14.17 | 13.87 | 13.81 | 13.85 | 13.67 | 13.45 | **13.46** | **13.45** | **13.13** | **13.06** |
| | $GTS(M, 5)$ | 14.17 | 13.87 | 13.81 | 13.85 | 13.67 | 13.45 | **13.46** | **13.45** | **13.13** | **13.06** |
| | [28] | 13.92 | 13.92 | 14.78 | 15.02 | 15.22 | 15.18 | 15.41 | 15.38 | 15.62 | 15.90 |
| | [29] | 17.31 | 17.26 | 17.15 | 17.04 | 16.77 | 16.63 | 16.56 | 16.47 | 16.04 | 16.07 |
| | [31] | **12.81** | **12.81** | **13.29** | **13.06** | **13.57** | **13.38** | 14.19 | 14.37 | 14.42 | 15.51 |
| MAPE | $GTS(M, 1)$ | 0.0127 | 0.0117 | 0.0113 | 0.0110 | 0.0110 | 0.0110 | 0.0109 | 0.0110 | 0.0107 | 0.0105 |
| | $GTS(M, 2)$ | 0.0096 | 0.0095 | 0.0095 | 0.0095 | 0.0094 | 0.0092 | **0.0092** | 0.0092 | 0.0090 | 0.0090 |
| | $GTS(M, 3)$ | 0.0097 | 0.0095 | 0.0094 | 0.0094 | 0.0093 | 0.0092 | **0.0092** | **0.0091** | **0.0089** | **0.0089** |
| | $GTS(M, 4)$ | 0.0097 | 0.0095 | 0.0094 | 0.0094 | 0.0093 | 0.0092 | **0.0092** | **0.0091** | **0.0089** | **0.0089** |
| | $GTS(M, 5)$ | 0.0097 | 0.0095 | 0.0094 | 0.0094 | 0.0093 | 0.0092 | **0.0092** | **0.0091** | **0.0089** | **0.0089** |
| | [28] | 0.0095 | 0.0095 | 0.0101 | 0.0102 | 0.0104 | 0.0103 | 0.0105 | 0.0105 | 0.0106 | 0.0108 |
| | [29] | 0.0118 | 0.0118 | 0.0117 | 0.0116 | 0.0114 | 0.0113 | 0.0113 | 0.0112 | 0.0109 | 0.0109 |
| | [31] | **0.0087** | **0.0087** | **0.0091** | **0.0089** | **0.0092** | **0.0091** | 0.0097 | 0.0098 | 0.0098 | 0.0099 |

The bold data means the minimum error of the models with the same order.

Table 6: Effects of membership function.

| Model | $k = 3$ | | | $k = 6$ | | | $k = 9$ | | |
|---|---|---|---|---|---|---|---|---|---|
| | Fun 1 | Fun 2 | Fun 3 | Fun 1 | Fun 2 | Fun 3 | Fun 1 | Fun 2 | Fun 3 |
| $GTS(k, 1)$ | **17.5133** | 17.3290 | 17.3755 | **17.1512** | 16.9790 | 17.0653 | **16.4968** | 16.2923 | 16.4014 |
| $GTS(k, 2)$ | **15.8071** | 15.7476 | 15.9365 | **15.4880** | 15.4686 | 15.7182 | **14.7028** | 14.6918 | 14.7719 |
| $GTS(k, 3)$ | 15.8071 | **15.6926** | 15.8736 | 15.4880 | **15.3137** | 15.4631 | 14.7028 | **14.5116** | 14.4828 |
| $GTS(k, 4)$ | 15.8071 | **15.6902** | **15.8670** | 15.4880 | **15.3119** | **15.4813** | 14.7028 | **14.5218** | 14.4558 |
| $GTS(k, 5)$ | 15.8071 | 15.6902 | **15.9157** | 15.4880 | 15.3119 | **15.5422** | 14.7028 | 14.5218 | **14.5082** |
| $GTS(k, 6)$ | 15.8071 | 15.6902 | 15.9157 | 15.4880 | 15.3119 | 15.5424 | 14.7028 | 14.5218 | **14.5068** |
| $GTS(k, 7)$ | 15.8071 | 15.6902 | 15.9157 | 15.4880 | 15.3119 | 15.5424 | 14.7028 | 14.5218 | 14.5068 |
| $GTS(k, 8)$ | 15.8071 | 15.6902 | 15.9157 | 15.4880 | 15.3119 | 15.5424 | 14.7028 | 14.5218 | 14.5068 |

The bold data means the minimum error of the models with the same order.

function, the more different forecasting accuracy obtained. It is more obvious when $M$ is increasing. There is still another conclusion that the forecasts of the third membership function are not always better than those of the first function.

To further investigate the relation between the parameters $M$ and the length of interval, the proposed model has been applied to forecasting stock index close prices covering ten years from 1997 to 2006 with $M = 1, 2, \ldots, 50$ and $L = 30, 60, \ldots, 300$. Since it has been affirmed by Tables 4 and 5 that the characters of MAE, and MAPE are similar to those of RMSE, MAE and MAPE, we will only list the RMSE comparison of these experiments as follows. Some examples of actual values and forecasts of 2003 are depicted in Figures 2 and 3. Figure 2 shows us that $GTS(3, 1)$ has a better performance than $GTS(1, 1)$ in the same length of intervals, that is, $L = 30$. Figure 3 illustrates that $GTS(1, 1)$ gets the better forecasts when the lengths of intervals are small. Furthermore, some properties will be further described in Figures 4 and 5 which depict the mean forecasting errors of the ten years. Figure 4 shows the relation between the

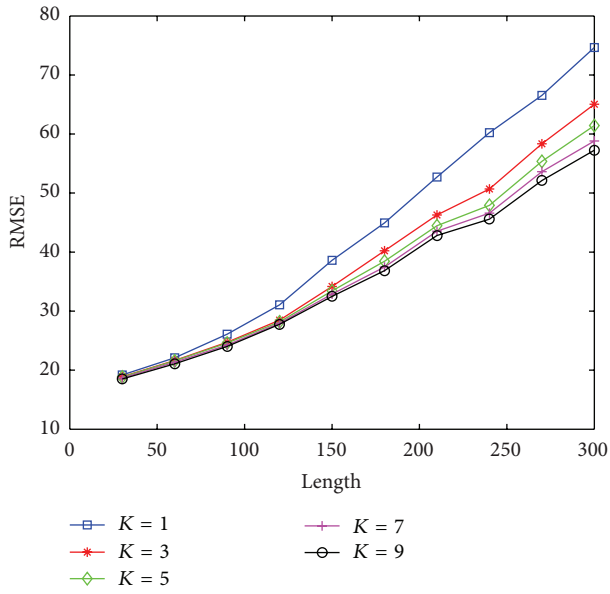



Figure 4: Comparison of RMSEs with different lengths of intervals.

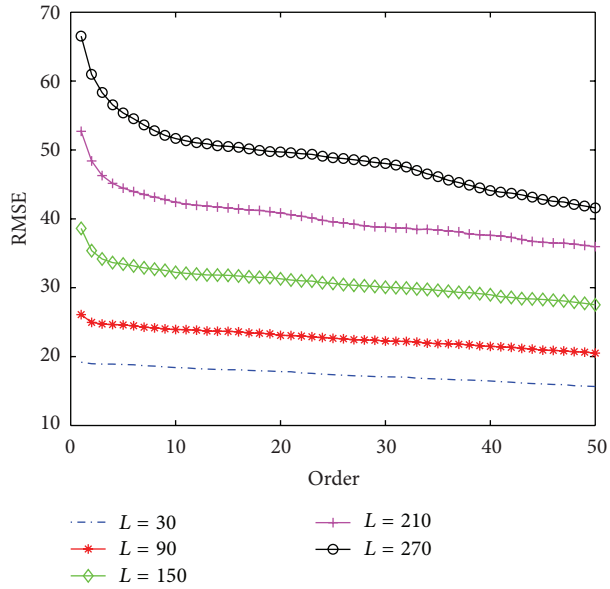



Figure 5: Comparison of RMSEs with different orders of the model.

RMSEs and lengths of intervals with different orders, and Figure 5 shows the relation between the RMSEs and orders with different lengths of intervals.

From Figure 4, it is clear that the longer the interval, the bigger the RMSE, and the higher order models are better than the lowers. This conclusion also was testified by Figure 5 which shows us another important message that the shorter length of intervals can result in robuster forecasts. Overall, these conclusions are important for the proposed mode to be applied on other data set or area.

## 5. Conclusion

After discussing the high-order fuzzy time series models and presenting the definition and operation for generalized fuzzy logical relationship, we have proposed a novel high-order fuzzy time series models based on the new relationship. The work is driven by the three main reasons. Firstly, it is urged to generalize the fuzzy logical relationship by the advanced fuzzy time series models. Secondly, it is to abstract the relationships matrices among time series and find out the patterns of time series fluctuations based on understandable fuzzy rules. The last one is to make the fuzzy time series model explain more complex relationships.

By using the enrolment of the University of Alabama and close price of Shanghai Stock Exchange Composite Index as data sets for evaluating the models, the experimental results give two conclusions: (1) the performance of $GTS(M, N)$ is more reasonable than the three conventional fuzzy time series models proposed earlier by Hwang et al. [28] and Chen and Chen [29, 31]; (2) the number of orders and principal fuzzy logical relationship affect the forecasting result slightly. The higher the order, the better the forecasting results, the more hierarchy principal fuzzy logical relationships, the less forecasts error but not infinite decreasing.

In the future research, some suggestions are provided to improve this paper. The relation between the principal fuzzy relationships and the conventional fuzzy relationships needs to be further discussed. For example, what's the effect brought by exchange the definition of membership function and the operations of principal fuzzy logical relationship? How great this kind of effect? Since the proposed model is on the basis of fuzzy logical relationship, a generalized fuzzy relationship, study work is worth devoting into improvement of the model hybridized with some advanced algorithms.

## Acknowledgments

This work was partially supported by the National Nature Science Foundation of China (nos. 31260273, 61261027), the Key Project of Chinese Ministry of Education (no. 210116), the Natural Science Foundation of Jiangxi Province, China (nos. 2010GQS0127, 20114BAB211013, 20122BAB211033, 20122BAB201044, 20122BAB2010), and the JiangXi Provincial Foundation for Leaders of Disciplines in Science (20113BCB22008).

## References

[1] Q. Song and B. S. Chissom, "Fuzzy time series and its models," *Fuzzy Sets and Systems*, vol. 54, no. 3, pp. 269–277, 1993.

[2] Q. Song and B. S. Chissom, "Forecasting enrollments with fuzzy time series. Part I," *Fuzzy Sets and Systems*, vol. 54, no. 1, pp. 1–9, 1993.

[3] Q. Song and B. S. Chissom, "Forecasting enrollments with fuzzy time series. Part II," *Fuzzy Sets and Systems*, vol. 62, no. 1, pp. 1–8, 1994.

[4] S. M. Chen, "Forecasting enrollments based on fuzzy time series," *Fuzzy Sets and Systems*, vol. 81, pp. 311–319, 1996.

[5] K. H. Huarng, "Effective lengths of intervals to improve forecasting in fuzzy time series," *Fuzzy Sets and Systems*, vol. 123, no. 3, pp. 387–394, 2001.

[6] K. H. Huarng and H. K. Yu, "A dynamic approach to adjusting lengths of intervals in fuzzy time series forecasting," *Intelligent Data Analysis*, vol. 8, no. 1, pp. 3–27, 2004.

[7] K. H. Huarng, "Ratio-based lengths of intervals to improve fuzzy time series forecasting," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 36, no. 2, pp. 328–340, 2006.

[8] Y. H. Leu, C. P. Lee, and Y. Z. Jou, "A distance-based fuzzy time series model for exchange rates forecasting," *Expert Systems with Applications*, vol. 36, no. 4, pp. 8107–8114, 2009.

[9] E. Egrioglu, Ç. H. Aladag, U. Yolcu, V. R. Uslu, and M. A. Basaran, "Finding an optimal interval length in high order fuzzy time series," *Expert Systems with Applications*, vol. 37, pp. 5052–5055, 2010.

[10] U. Yolcu, E. Egrioglu, V. R. Uslu, M. A. Basaran, and Ç. H. Aladag, "A new approach for determining the length of intervals for fuzzy time series," *Applied Soft Computing*, vol. 9, no. 2, pp. 647–651, 2009.

[11] H. K. Yu, "Weighted fuzzy time series model for TAIEX forecasting," *Physica A*, vol. 349, no. 3–4, pp. 609–624, 2005.

[12] C. H. Cheng, T. L. Chen, and C. H. Chiang, "Trend-weighted fuzzy timeseries model for TAIEX forecasting," *Lecture Notes in Computer Science*, vol. 4234, no. 3, pp. 469–477, 2006.

[13] M. H. Lee, E. Riswan, and I. Zuhaimy, "Modified weighted for enrollment forecasting based on fuzzy time series," *Matematika*, vol. 25, no. 1, pp. 67–78, 2009.

[14] W. R. Qiu, X. D. Liu, and H. L. Li, "A generalized method for forecasting based on fuzzy time series," *Expert Systems with Applications*, vol. 38, no. 8, pp. 10446–10453, 2011.

[15] W. R. Qiu, X. D. Liu, and L. D. Wang, "Forecasting Shanghai composite index based on fuzzy time series and improved C-fuzzy decision trees," *Expert Systems with Applications*, vol. 39, no. 9, pp. 7680–7689, 2012.

[16] S. R. Singh, "A robust method of forecasting based on fuzzy time series," *Applied Mathematics and Computation*, vol. 188, no. 1, pp. 472–484, 2007.

[17] S. R. Singh, "A simple method of forecasting based on fuzzy time series," *Applied Mathematics and Computation*, vol. 186, no. 1, pp. 330–339, 2007.

[18] S. R. Singh, "A computational method of forecasting based on high-order fuzzy time series," *Expert Systems with Applications*, vol. 36, pp. 10551–10559, 2009.

[19] L. W. Lee, L. H. Wang, and S. M. Chen, "Temperature prediction and TAIFEX forecasting based on fuzzy logical relationships and genetic algorithms," *Expert Systems with Applications*, vol. 33, pp. 539–550, 2007.

[20] L. W. Lee, L. H. Wang, and S. M. Chen, "Temperature prediction and TAIFEX forecasting based on high-order fuzzy logical relationships and genetic simulated annealing techniques," *Expert Systems with Applications*, vol. 34, pp. 328–336, 2008.

[21] K. H. Huarng and H. K. Yu, "A type 2 fuzzy time series model for stock index forecasting," *Physica A*, vol. 353, no. 1–4, pp. 445–462, 2005.

[22] L. Youdthachai, Y. J. Yang, and R. John, "High-order type-2 fuzzy time series," in *Proceedings of the International Conference on Soft Computing and Pattern Recognition (SoCPaR '10)*, pp. 363–368, Cergy Pontoise, France, 2010.

[23] I. H. Kuo, S. J. Horng, T. W. Kao, T. L. Lin, C. L. Lee, and Y. Pan, "An improved method for forecasting enrollments based on fuzzy time series and particle swarm optimization," *Expert Systems with Applications*, vol. 36, no. 3, part 2, pp. 6108–6117, 2009.

[24] I. H. Kuo, S. J. Horng, T. W. Kao et al., "An efficient flow-shop scheduling algorithm based on a hybrid particle swarm optimization model," *Expert Systems with Applications*, vol. 36, no. 3, part 2, pp. 7027–7032, 2009.

[25] I. H. Kuo, S. J. Horng, and Y. H. Chen, "Forecasting TAIFEX based on fuzzy time series and particle swarm optimization," *Expert Systems with Applications*, vol. 37, pp. 1494–1502, 2010.

[26] C. H. Aladag, E. Egrioglu, M. A. Basaran, U. Yolcu, and V. R. Uslu, "Forecasting in high order fuzzy times series by using neural networks to define fuzzy relations," *Expert Systems with Applications*, vol. 36, pp. 4228–4231, 2009.

[27] E. Egrioglu, Ç. H. Aladag, U. Yolcu, V. R. Uslu, and M. A. Basaran, "A new approach based on artificial neural networks for high order multivariate fuzzy time series," *Expert Systems with Applications*, vol. 36, pp. 10589–10594, 2009.

[28] J. R. Hwang, S. M. Chen, and C. H. Lee, "Handling forecasting problems using fuzzy time series," *Fuzzy Sets and Systems*, vol. 100, pp. 217–228, 1998.

[29] S. M. Chen, "Forecasting enrollements based on high-order fuzzy time series," *Cybernetics and Systems*, vol. 33, pp. 1–16, 2002.

[30] S. M. Chen and N. Y. Chung, "Forecasting enrollments using high-order fuzzy time series and genetic algorithms," *International Journal of Intelligent Systems*, vol. 21, no. 5, pp. 484–501, 2006.

[31] S. M. Chen and C. D. Chen, "Handling forecasting problems based on high-order fuzzy logical relationships," *Expert Systems with Applications*, vol. 38, pp. 3857–3864, 2011.

[32] C. H. Cheng, H. J. Teoh, and T. L. Chen, "High-order fuzzy time series based on rough set for forecasting TAIEX," in *Proceedings of the 6th International Conference on Machine Learning and Cybernetics*, vol. 3, pp. 1354–1358, Hong Kong, China, 2007.

[33] L. W. Lee, L. H. Wang, S. M. Chen, and Y. H. Leu, "Handling forecasting problems based on two-factors high-order fuzzy time series," *IEEE Transactions on Fuzzy Systems*, vol. 14, no. 3, pp. 468–477, 2006.

[34] J. I. Park, D. J. Lee, C. K. Song, and M. G. Chun, "TAIFEX and KOSPI 200 forecasting based on two-factors high-order fuzzy time series and particle swarm optimization," *Expert Systems with Applications*, vol. 37, pp. 959–967, 2010.

[35] H. J. Teoh, T. L. Chen, C. H. Cheng, and H. H. Chu, "A hybrid multi-order fuzzy time series for forecasting stock markets," *Expert Systems with Applications*, vol. 36, pp. 7888–7897, 2009.

[36] S. T. Li and Y. C. Cheng, "An enhanced deterministic fuzzy time series forecasting model," *Cybernetics and Systems*, vol. 40, no. 3, pp. 211–235, 2009.

[37] N. Y. Wang and S. M. Chen, "Temperature prediction and TAIFEX forecasting based on automatic clustering techniques and two-factors high-order fuzzy time series," *Expert Systems with Applications*, vol. 36, pp. 2143–2154, 2009.

[38] W. R. Qiu, X. D. Liu, and L. D. Wang, "Forecasting in time series based on generalized fuzzy logical relationship," *ICIC Express Letters A*, vol. 4, no. 5, pp. 1431–1438, 2010.

[39] L. A. Zadeh, "Fuzzy sets," *Information and Computation*, vol. 8, pp. 338–353, 1965.

[40] L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning—I," *Information Sciences*, vol. 8, pp. 199–249, 1975.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 270692, 6 pages
http://dx.doi.org/10.1155/2013/270692

*Research Article*

# Basic Unit Layer Rate Control Algorithm for H.264 Based on Human Visual System

**Xiao Chen[1,2] and Haiying Liu[2]**

[1] *Jiangsu Key Laboratory of Meteorological observation and Information Processing, Nanjing University of Information Science and Technology, Nanjing 210044, China*
[2] *School of Electronic and Information Engineering, Nanjing University of Information Science and Technology, Nanjing 210044, China*

Correspondence should be addressed to Xiao Chen; chenxiao@nuist.edu.cn

Received 26 May 2013; Accepted 31 July 2013

Academic Editor: Zhongmei Zhou



In the process of the video coding, special attention should be paid to the subjective quality of the image. In the JVT-G012 algorithm for H.264, the influence of the human visual characteristic in basic unit layer rate control was not taken into account. This paper takes the influence of the human visual characteristic into the full consideration and offers ways to improve the subjective quality of the image. The visual characteristic factor, which is constituted by the motion feature and edge feature, is used to reasonably allocate the target bits, and then its quantization parameter is adjusted by encoded frame information. The experimental results show that, in comparison to the original algorithm, the proposed algorithm can not only control the bit rate more accurately but also make the peak signal to noise ratio (PSNR) stable, so as to improve the stationarity of the video image. The subjective quality of the reconstructed video is more satisfying.

## 1. Introduction

With the rapid development and popularization of the internet, electronic devices gradually become an indispensable part of our daily life, such as online broadcasting, online advertising, e-commerce, VOD, distance education, telemedicine, real-time video conference, smart phones, 3D video [1], VCD, DVD, HDTV, and streaming of multimedia video [2]. However, the real-time data transmission and storage of multimedia data have become more difficult due to the limited communication bandwidth, especially in the video communication. Its high capacity of data has much difficulty in the process of transmission and storage. Thus with the limits of the bandwidth and the low storage capacity, the video coding aiming at using the least bits to represent image is very important.

The rate control plays an important role in the process of video coding. Since 70% of the information is obtained from eyes and the eyes is the final receiver of video information, it is vital important to take full advantage of the human visual characteristics to get a higher subjective quality of images.

According to the special structure of the human eyes, some scholars put forward some relevant algorithms [3–10]. An adaptive bit allocation method is presented in [3], based on the space and time perception functions. The work in [4, 5] presents a method based on the region of interest in the human eyes. A novel rate control algorithm is presented in [6], based on the visual perception characteristics. The work in [7] proposes a new digital video watermark method based on the human visual system (HVS). The work in [8] proposes a method based on the region of interest, aiming at distribution of the target bits. The work in [9] presents a video quality evaluation method based on the region of interest. The work in [10] presents an algorithm to distribute the target bits of the basic unit layer, which analyze the motion information and texture features.

Although the JVT-G012 is by now the most acceptable rate control algorithm, it still has shortcomings. The work in [11] proposes a joint rate-distortion optimization for the H.264 rate control algorithm with a novel distortion prediction equation, which avoids linear regression employed in other distortion predictors and can considerably speed up

rate estimation. Multiple quantization parameters determination algorithm based on the statistics of the deviation measure is proposed in [12], which can achieve accurately QP. The work in [13] proposes a rate control technique for H.264/AVC using subjective quality of video. The work in [14] presents a complexity coefficient to combine the target bits. This paper presents a reformative basic unit layer rate control algorithm based on the HVS. The HVS was not taken into account in the JVT-G012 algorithm. Since eyes are the final receiver of the video information, so it is vital important to take the HVS into account in the video coding process. The HVS is very sensitive to the brim part and motion part. However each pixel of JVT-G012 algorithm is treated equally in the basic unit layer. Though the work in [10] takes advantage of the HVS, it is not comprehensive. In this paper the visual characteristic factor is used to improve the rate control in the basic unit layer.

## 2. JVT-G012 Algorithm

The JVT-G012 basic unit layer rate control mainly consists of three steps. First, it predicts the MAD of the current basic unit in the current frame. Then, it computes the target bit for the current basic unit. Last, it calculates the quantization parameter of the current basic unit and performs RDO.

*2.1. Predict the MAD of the Current Basic Unit in the Current Frame.* Consider

$$\mathrm{MAD}_{\mathrm{cb}} = a_1 * \mathrm{MAD}_{\mathrm{pb}} + a_2, \tag{1}$$

where $\mathrm{MAD}_{\mathrm{cb}}$ is the predicted MAD of the current basic unit in the current frame. $\mathrm{MAD}_{\mathrm{pb}}$ is the actual MAD of the co-located basic unit in the previous frame. $a_1$ and $a_2$ are two coefficients of the predictive model, whose initial values are 1 and 0, respectively; after finishing encoding every basic unit, the coefficients $a_1$ and $a_2$ are updated.

*2.2. Compute the Target Bit for the Current Basic Unit.* The target bit for the current basic unit $\widetilde{b}_l$ is

$$\widetilde{b}_l = T_r \times \frac{\mathrm{MAD}_{\mathrm{cb}}^2}{\sum_{k=1}^{N_{\mathrm{unit}}} \mathrm{MAD}_{\mathrm{cb}}^2(k)}, \tag{2}$$

where $T_r$ and $N_{\mathrm{unit}}$ are the number of remaining bits for the all uncoded basic units in the current frame and the number of uncoded basic units, respectively. $\mathrm{MAD}_{\mathrm{cb}}$ is the predicted MAD of the current basic unit in (1).

*2.3. Compute the Quantization Parameter of the Current Basic Unit and Perform RDO.* Consider

$$\widetilde{b}_l = \mathrm{MAD}_{\mathrm{cb}} * \left(\frac{X_1}{Q} + \frac{X_2}{Q^2}\right), \tag{3}$$

where $Q$ is the quantization parameter of the current basic unit. $\widetilde{b}_l$ is the target bit of the current basic unit in (2). $\mathrm{MAD}_{\mathrm{cb}}$ is the predicted MAD of current basic unit in (1). $X_1$ and $X_2$ are the first-order and the second-order model parameters of the quadratic rate-distortion model, respectively. They are updated in the process of encoding.

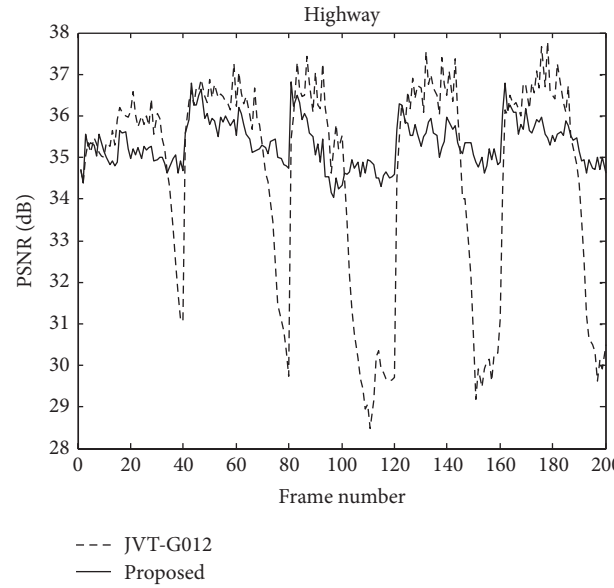



Figure 1: PSNR curves of the highway sequence comparison for the JVT-G012 and the proposed algorithms.

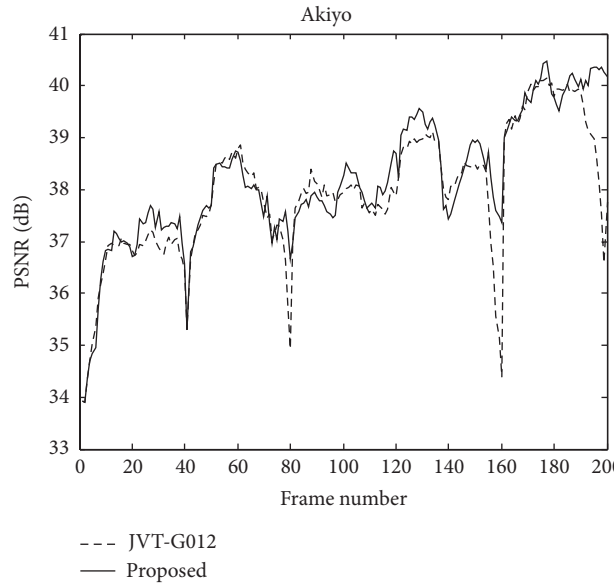



Figure 2: PSNR curves of the akiyo sequence comparison for the JVT-G012 and the proposed algorithms.

## 3. Improved Rate Control Algorithm

The reformative basic unit layer rate control algorithm mainly consists of two steps. First, it allocates the target bits based on the HVS. Then, it adjusts the quantization parameter and performs RDO.

*3.1. Compute the Target Bit of the Current Basic Unit Based on the HVS.* This paper takes the HVS perception mechanism into account because the human eyes are extremely sensitive to the edge part and the motion part of images. So the

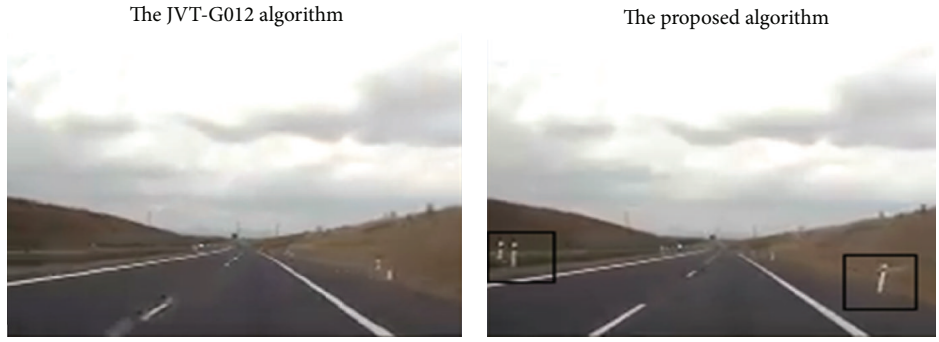



Figure 3: Comparison of the subjective quality in the highway sequence for the JVT-G012 and the proposed algorithms.

proposed algorithm assigns fewer bits to the unimportant region and assigns more bits to the region of interest in the process of allocating bits. It can achieve the goal of improving the overall video quality. This paper adjusts (2) with the visual characteristic factor $V(i, j)$:

$$\widetilde{b}_l = T_r \times V(i,j) \times \frac{\mathrm{MAD}_{\mathrm{cb}}^2}{\sum_{k=1}^{N_{\mathrm{unit}}} \mathrm{MAD}_{\mathrm{cb}}^2(k)}, \qquad V(i,j) = \mathrm{motion}(i,j) + \mathrm{edge}(i,j), \tag{4}$$

where the motion vision characteristics and the edge vision characteristics are denoted by motion($i$, $j$) and edge($i$, $j$), respectively.

*3.1.1. Motion Characteristics.* In the real scene, there exist two major motion scenes as following.

The whole scene changes a little while only parts of the objects move or change. At this time, human eyes are concerned much more about the moving and changing objects. That is, when $\mathrm{MV}_{\mathrm{avg}}$ is less than 4.5,

$$\mathrm{motion}(i,j) = \begin{cases} 1 & \mathrm{MV} < 1 \\ 2 - \dfrac{(6.5 - \mathrm{MV})}{8} & 2 < \mathrm{MV} \le 6.5 \\ 3 & \mathrm{MV} > 6.5. \end{cases} \tag{5}$$

When the whole scene moves fast, $\mathrm{MV}_{\mathrm{avg}}$ is more than 4.5, there are two subordinate situations in this case.

When there are many fast moving macroblocks, human eyes pay more attention to those objects that move little. This time MV is more than 2.5; motion($i$, $j$) can be expressed as

$$\mathrm{motion}(i,j) = \begin{cases} 1 + \dfrac{(6.5 - \mathrm{MV})}{10} & 2.5 < \mathrm{MV} \le 6.5 \\ 3 & \mathrm{MV} > 6.5. \end{cases} \tag{6}$$

When most of the objects move inconspicuously while only some of them move fast in the scene, human eyes pay more attention to the fast-moving part. This time, MV is less than 2.5; motion($i$, $j$) can be expressed as

$$\mathrm{motion}(i,j) = \begin{cases} 1 & \mathrm{MV} \le 1.5 \\ 2 - \dfrac{(3.5 - \mathrm{MV})}{10} & 1.5 < \mathrm{MV} < 2.5, \end{cases} \tag{7}$$

where the magnitude of motion vector for the $j$th basic unit in the $i$th frame and the magnitude of average motion vector in the remaining basic units of current frame are denoted by MV($i$, $j$) and $\mathrm{MV}_{\mathrm{avg}}$, respectively. One has $\mathrm{MV} = \sqrt{\mathrm{MV}_X^2 + \mathrm{MV}_Y^2}$, where $\mathrm{MV}_X$ represents the magnitude of the macroblock motion vector in horizontal direction and $\mathrm{MV}_Y$ represents the magnitude of the macroblock motion vector in vertical direction.

*3.1.2. Edge Characteristics.* This paper describes the edge characteristics of images with the variance because there is high variance in the edge area of images:

$$\mathrm{edge}(i,j) = \begin{cases} 1 - \dfrac{D(i,j)}{\max(D)} & D(i,j) < \mathrm{medium}(D) \\ \dfrac{D(i,j)}{\max(D)} & \mathrm{other}, \end{cases} \tag{8}$$

where $D(i, j)$ represents the variance for the $j$th basic unit in the $i$th frame. Medium($D$) and max($D$) represent the mid-value and the maximum value of the variance, respectively.

*3.2. Adjust the Quantization Parameter and Perform RDO.* To consider the feedback information of the encoded frames, this paper adapts the quantization parameter adjustment coefficient. This paper uses $\eta$ to adjust the quantization parameter, which is defined as the ratio of texture bits to the header bits:

$$QP_i(j) = \begin{cases} QP_i'(j) - 1 & 1.0 < \eta \le 2.0 \\ QP_i'(j) + 1 & 0.6 < \eta \le 1.0 \\ QP_i'(j) + 2 & \eta < 0.6 \\ Q_c & \mathrm{other}, \end{cases} \tag{9}$$

where $QP_i'(j)$ is the average value of quantization parameters for all basic units in the previous frame. $Q_c$ is the quantitative parameter in the JVT-G012 algorithm.

After the adjustment, the algorithm took into account the encoded frame information. The proposed algorithm achieves a good rate control. The algorithm performs RDO and updates the model parameters.

Table 1: Bit rate comparison for the proposed algorithm and the JVT-G012 algorithm.

| Test sequences | Output bit rate (kbps) | |
|---|---|---|
| | JVT-G012 | The proposed algorithm |
| Highway | 32.41 | 32.06 |
| Mother-daughter | 32.10 | 32.04 |
| Foreman | 32.10 | 32.06 |
| Claire | 32.15 | 32.06 |
| Hall | 32.14 | 32.07 |
| Silent | 32.09 | 32.06 |
| Akiyo | 32.08 | 32.04 |
| News | 32.04 | 32.04 |
| Carphone | 32.13 | 32.06 |

Table 2: PSNR comparison for the proposed algorithm and the JVT-G012 algorithm.

| Test sequences | PSNR (dB) | | |
|---|---|---|---|
| | JVT-G012 | The proposed algorithm | Gain |
| Highway | 34.60 | 35.34 | 0.74 |
| Mother-daughter | 34.79 | 34.90 | 0.11 |
| Foreman | 28.38 | 28.49 | 0.11 |
| Claire | 39.36 | 39.56 | 0.20 |
| Hall | 34.05 | 34.26 | 0.21 |
| Silent | 31.03 | 31.16 | 0.13 |
| Akiyo | 37.93 | 38.16 | 0.23 |
| News | 31.29 | 31.50 | 0.21 |
| Carphone | 31.56 | 31.70 | 0.14 |

Table 3: Bit rate comparison for the proposed algorithm and the Zheng algorithm.

| Test sequences | Output bit rate (kbps) | |
|---|---|---|
| | Zheng algorithm | The proposed algorithm |
| Highway | 31.94 | 32.06 |
| Mother-daughter | 31.95 | 32.04 |
| Foreman | 31.93 | 32.06 |
| Claire | 32.03 | 32.06 |
| Hall | 32.04 | 32.07 |
| Silent | 32.06 | 32.06 |
| Akiyo | 32.01 | 32.04 |
| News | 32.04 | 32.04 |
| Carphone | 32.04 | 32.06 |

Table 4: PSNR comparison for the proposed algorithm and the Zheng algorithm.

| Test sequences | PSNR (dB) | | |
|---|---|---|---|
| | Zheng algorithm | The proposed algorithm | Gain |
| Highway | 35.23 | 35.34 | 0.11 |
| Mother-daughter | 34.75 | 34.90 | 0.15 |
| Foreman | 28.35 | 28.49 | 0.14 |
| Claire | 39.49 | 39.56 | 0.07 |
| Hall | 34.00 | 34.26 | 0.26 |
| Silent | 31.07 | 31.16 | 0.09 |
| Akiyo | 38.04 | 38.16 | 0.12 |
| News | 31.40 | 31.50 | 0.10 |
| Carphone | 31.59 | 31.70 | 0.11 |

## 4. Simulation Results and Discussion

In order to validate the effectiveness of our algorithm in this paper, this paper has implemented the proposed rate control algorithm by the JM10.1 test model software. Also the proposed algorithm is compared with the JVT-G012 algorithm and the Zheng algorithm in [10]. The tested sequences are in QCIF4:2:0 formats: highway, mother-daughter, foreman, claire, hall, silent, akiyo, news, and carphone. In the experiments, all sequences are coded as the IPPP structure, the frame rate is set to 30 frames per second, the total number of frames is set to 200, the length of GOP is set to 40, and the target rate is 32 kbps.

Tables 1 and 2 show the comparison of the bit rate and the PSNR. As summarized in Tables 1 and 2, the proposed algorithm can control more accurately the bit rates than the JVT-G012 algorithm and obtain much better PSNR for the different video sequences. In particular for the test sequence highway, the proposed algorithm achieves a PSNR gain of 0.74 dB.

Tables 3 and 4 show the comparison of the bit rate and the PSNR. The proposed algorithm gets much better PSNR than the Zheng algorithm for different video sequences and also controls accurately the bit rate. The proposed algorithm can improve the average PSNR for all test sequences, while the Zheng algorithm can improve PSNR for a part of test sequences. For example, Zheng algorithm gets less PSNR than the JVT-G012 algorithm for test sequences: mother-daughter, foreman, and hall.

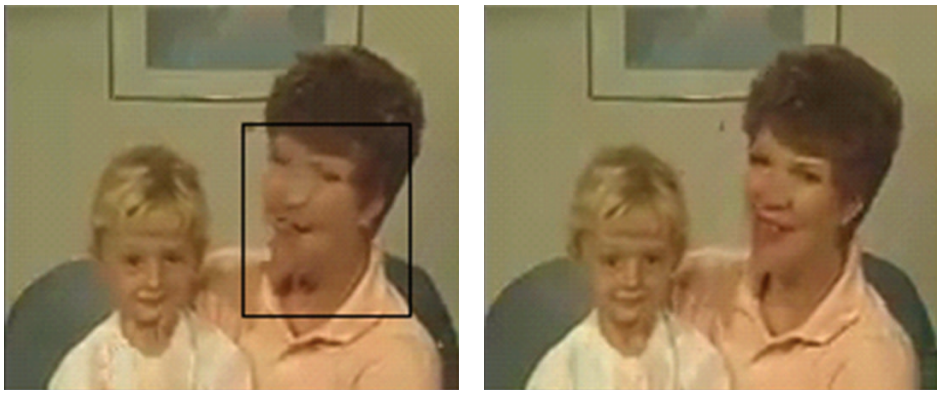
Figures 1 and 2 show that the PSNR curve is flatter than the one obtained from the JVT-G012 algorithm. The proposed algorithm suppresses the sharp drop of the PSNR and improves the stability of the picture quality.

Figure 3 shows obviously that the highway keeps the edge part of images to which human eyes are sensitive, while the one obtained from the JVT-G012 algorithm distorts and influences the subjective feeling of eyes.

Figures 4, 5, and 6 are the comparison of the subjective quality. For the test sequences claire, mother-daughter, and silent, their facial and body parts have a dramatic decline

(a) (b)

Figure 4: Comparison of the subjective quality in the claire sequence for the JVT-G012 and the proposed algorithms.



(a) (b)

Figure 5: Comparison of the subjective quality in the mother-daughter sequence for the JVT-G012 and the proposed algorithms.

(a) (b)

Figure 6: Comparison of the subjective quality in the silent sequence for the JVT-G012 and the proposed algorithms.

in image quality in the JVT-G012 algorithm. However, these features have better visual quality in the proposed algorithm.

Figure 7 is the comparison of subjective quality for the test sequence carphone. It is found that the sensitive regions of eyes in the image, such as the face, clothing, and edge parts, have become blurred in the JVT-G012. However, these features have better visual quality in the proposed algorithm.

## 5. Conclusions

This paper proposes a reformative basic unit layer rate control algorithm by using the visual characteristic factor and the adjusted quantization parameter. The proposed algorithm allocates bits based on the HVS in basic unit layer and adjusts the quantization parameter with texture bits and the header

(a) (b)

Figure 7: Comparison of the subjective quality in the carphone sequence for the JVT-G012 and the proposed algorithms.

bits of the current basic unit. The experimental results show that the proposed algorithm can control the bit rate more accurately and have a much better visual quality. Compared with the JVT-G012 algorithm, the PSNR can be improved by 0.2–0.7 dB. What is more, the PSNR has little fluctuation and the video image will become more stable. Compared with [10] algorithm, the PSNR for different test sequences has been improved obviously in this paper. The [10] algorithm has great effect on particular tests, while the proposed algorithm in this paper has universal applicability. In addition, the algorithm in this paper has great effect at low bit rates.

## Acknowledgments

This work was supported by the Qing Lan Project and the Priority Academic Program Development of Jiangsu Higher Education Institutions.

## References

[1] C. T. E. R. Hewage and M. G. Martini, "Reduced-reference quality assessment for 3D video compression and transmission," *IEEE Transactions on Consumer Electronics*, vol. 57, no. 3, pp. 1185–1193, 2011.

[2] Z. Chen, C. Lin, and X. Wei, "Enabling on-demand internet video streaming services to multi-terminal users in large scale," *IEEE Transactions on Consumer Electronics*, vol. 55, no. 4, pp. 1988–1996, 2009.

[3] M. Hrarti, H. Saadane, M. Larabi, A. Tamtaoui, and D. Aboutajdine, "A macroblock-based perceptually adaptive bit allocation for H264 rate control," in *Proceedings of the 5th International Symposium on I/V Communications and Mobile Networks (ISIVC '10)*, pp. 1–4, October 2010.

[4] M. Wang, T. Zhang, C. Liu, and S. Goto, "Region-of-interest based dynamical parameter allocation for H.264/AVC encoder," in *Proceedings of the Picture Coding Symposium (PCS '09)*, pp. 1–4, May 2009.

[5] Y. Liu, Z. G. Li, and Y. C. Soh, "Region-of-interest based resource allocation for conversational video communication of H.264/AVC," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 1, pp. 134–139, 2008.

[6] R. Ruolin, H. Ruimin, and L. Zhongming, "A novel rate control algorithm of video coding based on visual perceptual characteristic," in *Proceedings of the 6th International Conference on Computer Science and Education (ICCSE '11)*, pp. 843–846, August 2011.

[7] L. Liao, X. Zheng, Y. Zhao, and G. Liu, "A new digital video watermark algorithm based on the HVS," in *Proceedings of the International Conference on Internet Computing and Information Services (ICICIS '11)*, pp. 446–448, September 2011.

[8] Y. Liu, Z. G. Li, and Y. C. Soh, "Region-of-interest based resource allocation for conversational video communication of H.264/AVC," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 1, pp. 134–139, 2008.

[9] G. Q. Lu and J. L. Li, "Video quality evaluation method based on the visual region of interest," *Computer Engineering*, vol. 35, no. 10, pp. 217–219, 2011.

[10] Q. Zheng, M. Yu, Z. Peng, F. Shao, F. Li, and G. Jiang, "Human visual system-based rate control algorithm for H.264/AVC," *Guangdianzi Jiguang/Journal of Optoelectronics Laser*, vol. 22, no. 3, pp. 440–445, 2011.

[11] F. Chen and Y. Hsu, "Rate-distortion optimization of H.264/AVC rate control with novel distortion prediction equation," *IEEE Transactions on Consumer Electronics*, vol. 57, no. 3, pp. 1264–1270, 2011.

[12] J. Li and E. Abdel-Raheem, "Efficient rate control for H.264/AVC intra frame," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 2, pp. 1043–1048, 2010.

[13] S. L. P. Yasakethu, W. A. C. Fernando, S. Adedoyin, and A. Kondoz, "A rate control technique for off line H.264/AVC video coding using subjective quality of video," *IEEE Transactions on Consumer Electronics*, vol. 54, no. 3, pp. 1465–1472, 2008.

[14] X. Chen and F. Lu, "A reformative frame layer rate control algorithm for H.264," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 4, pp. 2806–2811, 2010.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 712028, 8 pages
http://dx.doi.org/10.1155/2013/712028

*Research Article*

# A Self-Learning Sensor Fault Detection Framework for Industry Monitoring IoT

**Yu Liu,[1] Yang Yang,[2] Xiaopeng Lv,[3] and Lifeng Wang[4]**

[1] *State Key Laboratory of Software Development Environment, Beihang University, Beijing 100191, China*
[2] *Information Center of Guangdong Power Grid Corporation, China Southern Power Grid, Guangzhou 510620, China*
[3] *Beijing Guotie Huachen Communication & Infomation Technology Co., Ltd., Beijing 10070, China*
[4] *Beijing Electronic Science and Technology Institute, Beijing 100070, China*

Correspondence should be addressed to Yu Liu; liuyu.paper@gmail.com

Received 8 June 2013; Accepted 3 August 2013

Academic Editor: Zhongmei Zhou



Many applications based on Internet of Things (IoT) technology have recently founded in industry monitoring area. Thousands of sensors with different types work together in an industry monitoring system. Sensors at different locations can generate streaming data, which can be analyzed in the data center. In this paper, we propose a framework for online sensor fault detection. We motivate our technique in the context of the problem of the data value fault detection and event detection. We use the Statistics Sliding Windows (SSW) to contain the recent sensor data and regress each window by Gaussian distribution. The regression result can be used to detect the data value fault. Devices on a production line may work in different workloads and the associate sensors will have different status. We divide the sensors into several status groups according to different part of production flow chat. In this way, the status of a sensor is associated with others in the same group. We fit the values in the Status Transform Window (STW) to get the slope and generate a group trend vector. By comparing the current trend vector with history ones, we can detect a rational or irrational event. In order to determine parameters for each status group we build a self-learning worker thread in our framework which can edit the corresponding parameter according to the user feedback. Group-based fault detection (GbFD) algorithm is proposed in this paper. We test the framework with a simulation dataset extracted from real data of an oil field. Test result shows that GbFD detects 95% sensor fault successfully.

## 1. Introduction

Internet of Things (IoT) has been paid more and more attention by the government, academe, and industry all over the world because of its great prospect [1–3]. In the IoT application field, intelligent industry is an important branch. A wired or wireless sensor network is the basic facility of the industry monitoring IoT. These networks comprising of thousands of inexpensive sensors can report their values to the data center. The aim of the monitoring system is to guarantee the process of production.

Fault detection is an important process for industry monitoring IoT, but it is a difficult and complex task because there are many factors that influence data and could cause faults. And faults are application and sensor type dependent [4–6]. Sensors in an industry monitoring IoT have three features: (1) *Big*: thousands of sensors on different devices are working together, (2) *Multitypes*: many physical quantities are needed to determine the production status, (3) *Uncertainty*: different workload is needed according to the production plan and some devices need shut down for examination. So the values of correlative sensors will change between different levels.

From the data-centric view, we focus on the *Outliers*, *Stuck-at* faults and *Spikes* [7]. From the application and system view, we focus on the rational and irrational trend detections. A rational trend means the sensor value transforms from one level to another smoothly and it is caused by a rational reason, such as shut down a device. An irrational trend means value changed when something is wrong with a device. The typical two mistakes of the monitoring system are taking a rational trend for an *Outlier*, or ignoring an irrational trend while values are still in range.

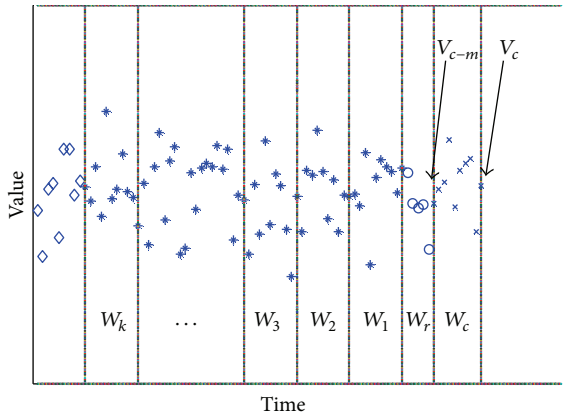



FIGURE 1: Estimation of the data distribution in the Statistics Sliding Windows (SSW) for one time instance.

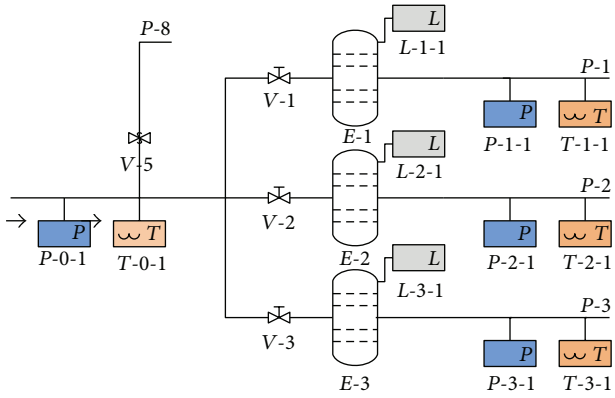



FIGURE 2: Typical flowchat of a gas processing plant.

In this paper, we propose a self-learning sensor fault detection framework for industry monitoring IoT. The data model design is described in Section 3. In Section 4, the framework and the core algorithm are discussed. A simulation experiment base on real data is shown in Section 5.

## 2. Related Work

Many researchers pay their attention to building a smart monitoring system. Bressan et al. [8] created a solid routing infrastructure through RPL. Castellani et al. [9] concentrate on the actual implementation of the communication technology and presented a lightweight implementation of an EXI library. Yuan et al. [10] present a parallel distributed structural health monitoring technology based on the wireless sensor network. An IoT communication framework for distributed worldwide health care applications is maintained in [11]. All these works are focused on the basic frameworks, protocols, and communication technologies of monitoring systems but discussed less on sensors management.

For modeling sensor network data, Guestrin et al. [12] propose a framework, for the nodes in the network to collaborate in order to fit a global function to each of their local measurements. This is a parametric approximation technique

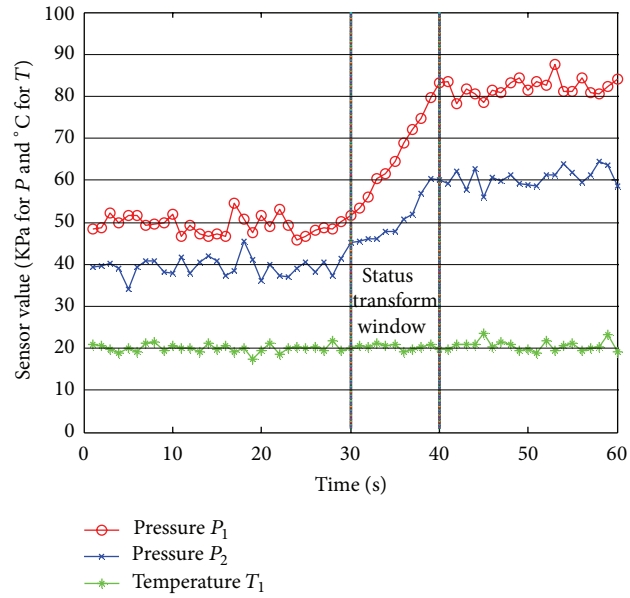



FIGURE 3: Status transformation process for two time instances.

and has more parameters then our approach. References [13, 14] study the problem of computing order statistics in a sensor network. There has also been work on predicting and caching the values generated by the sensors [15, 16], which can result in significant communication savings. But all these approaches are not fit our setting.

A similar approach for sensor fault detection in streaming data is described by Yamanishi et al. [17]. In contrast to our work, their method does not operate on sliding windows but rather on the entire history of the data values. Chan et al. [18] extend the study of algorithms for monitoring distributed data streams from whole data streams to a time-based sliding window, but their focus is on presenting a communication-efficient algorithm.

Ding et al. [19] combined trajectories of all nodes and the paramealgorithm which requires low computational overhead. The proposed algorithm compared its sensor reading with the median value of its neighbors' readings. Gao et al. [20] approach WSN fault detection problems using spatial correlation with the assumption of similar reading within cross range of neighbor nodes. Krishnamachari and Iyengar [21] tried to solve the faulty node detection problem by using localized event region and they assume that the system knows the location of sensor. In an industry monitoring sensor network, finding out a neighbor automatically is very hard. In our approach, we separate sensors into groups according to the production flow charts.

## 3. Data Model Design

This section focuses on the sensor data model design. We model sensors from the view of value for *Outliers*, *Stuck-at* faults, and *Spikes* detection and from the application view for event detection. The events we are interested in are rational trend and irrational trend.

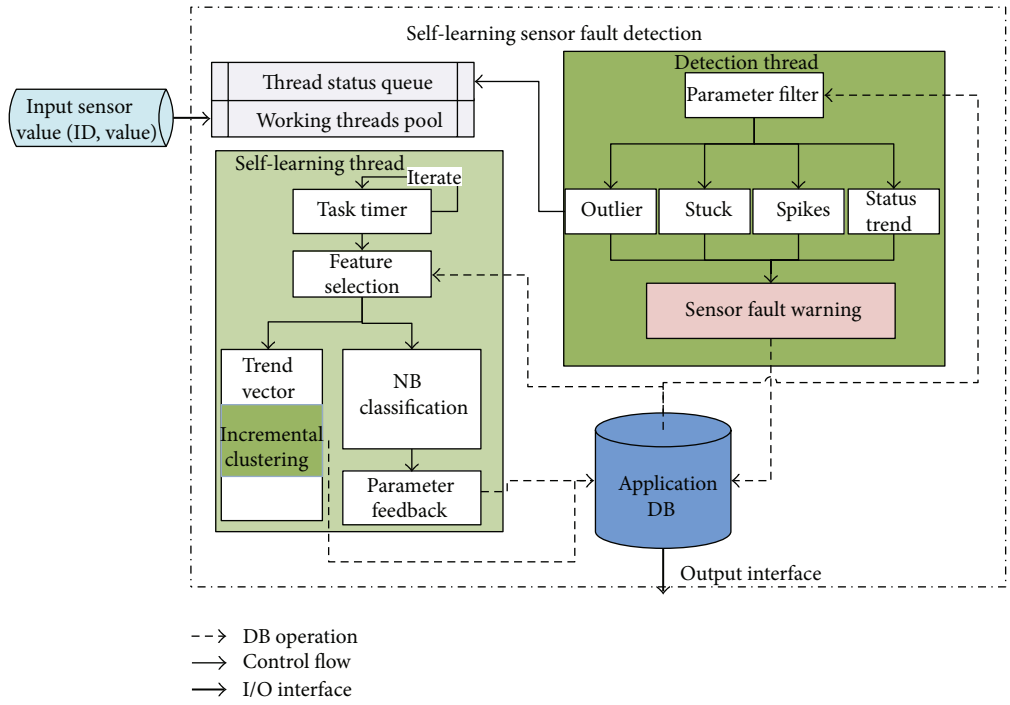



Figure 4: A self-learning sensor fault detection framework.

*3.1. Sensor Value.* For detection the *Outliers, Stuck-at* faults, and *Spikes*, we propose a statistics method. Figure 1 shows the mechanism of Statistics Sliding Windows (SSW). For sensor $s$, the current value $v_c$ and the previous $m$ values form the current windows $W_c = \{v_{c-m}, v_{c-m+1}, v_{c-m+2}, \ldots, v_c\}$. In the recent history, we can define $k$ windows with the same length and get the set $H_v = \{W_1, W_2, W_3, \ldots, W_k\}$. We estimate the values in each sliding window by Gaussian distribution, $W_i \rightarrow N(\mu_i, \sigma_i^2)$ (Formula (1)). If $\sigma_c^2 = 0$, a *Stuck-at* fault is detected. And when $\sigma_c^2$ is big enough, a *Spikes* fault may happened:

$$W_i \longrightarrow N\left(\mu, \sigma^2\right),$$
$$\mu = \frac{\sum_{k=1}^{m} v_k}{m}, \quad \sigma^2 = \frac{\sum_{k=1}^{m} \left(v_k - \mu\right)^2}{m}. \tag{1}$$

There is a buffer named $W_r$ in front of the current window $W_c$. With the new samples coming into $W_c$, the obsolete samples deserted by $W_c$ will become a member of $W_r$. When the size of $W_r$ reach $m$, the oldest window in $H_v$ will be discarded and the current $W_r$ will join $H_v$ as $W_1$. With SSW, large numbers of historical sensor values are regressed to $k$ pairs of Gaussian characteristics. In a real application, we need not hold all the recent values in the memory.

*3.2. Status Group.* In the industry monitoring IoT, the status of a production line may be uncertain. With the different manufacturing techniques and different workloads, some devices in the production line may be shut down. In this case, the whole production line is still working, but values of sensors monitoring the shutdown devices will run out of range (*Outliers*). At the same time, related devices may also change with the shutting down operation. In an industry monitoring IoT, the rational status transformation of sensors should be recognized and ignored.

Figure 2 shows a typical flow chat of a gas-processing plant. *P-1*, *P-2*, and *P-3* are three parallel subpipelines which are controlled by *V-1*, *V-2*, and *V-3*, respectively. Each subpipeline can be shut down independently and this operation will affect the value of *P-0-1*, a pressure sensor on the main input pipeline. In our approach, we deposit the sensors in the goal-processing plant into several status groups. Although the application background is important to the disposition method, we can follow some common rules as follows:

$$G = \{S_1, S_2, \ldots, S_n\}, \quad S_i \in \text{VALVE}_P,\ n \leq 10. \tag{2}$$

(1) Sensors on the opposite side of a valve are not in the same group. A valve is a typical controller in the industrial IoT. All the devices on the backward position can be shut down by the proper valve. The sensors on the different side may be in different status.

(2) If there are too many sensors which are controlled by one slave, they should be divided into different groups. In our approach, we will not put more than 10 sensors into one status group because the probable status space will expand acutely with the increasing sensor numbers. In this case, sensors can be grouped by their relative position with the most complex device, because the complex device may lead to status change most possibly.

(1) let $W^m$ be the statistics sliding windows size;
(2) let $\varepsilon_s$ be the outlier detection threshold for sensor $S$ and let $P$ be the global list to keep all of the $\varepsilon_s$;
(3) let $U$ and $V$ be the global arrays to keep the last 10 gaussian distribution characteristics for each sensor;
(4) let $W^n$ be the status transform windows size and $\theta_i/4$ be the trend vector merging threshold for group $G_i$, all the rational trend vectors of $G_i$ are stored in $R_i$;
(5) procedure **GFD ()**
(6) init $P$, loading $\varepsilon$ from Appilcation DB;
(7) init $U$ and $V$, loading the last 10 distribution characteristics from Application DB;
(8) create $C$ DetectionThread threads, $C = Num_{cpu} * 2$;
(9) start **SelfLearningThread**();
(10) do
(11) get a value set $v_i$ for sensors in a group $G_i$;
(12) find a idle DetectionThread;
(13) **DetectionThread**($v_i$);
(14) while not end;
(15) return;
(16) procedure **DetectionThread**($v_i$)
(17) if (**IsStuck**($v_i$) or **IsSpikes**($v_i$))
(18) return;
(19) if (**IsOutlier**($v_i$) and not **IsRatStatChange**($v_i$))
(20) return;
(21) **IsRatStatChange**($v_i$);
(22) return;
(23) procedure **IsStuck**($v_i$)
(24) get ${\sigma_{ij}}^2$ for sensor $j$ by $T_c$;
(25) if (${\sigma_{ij}}^2 = 0$)
(26) mark $s_{ij}$ as a Stuck;
(27) return;
(28) procedure **IsSpikes**($v_i$)
(29) get $\mu_{ij}$ and ${\sigma_{ij}}^2$ for sensor $j$ by $T_c$;
(30) if ($\mu_{ij}/\text{mean}(U(\mu_i)) < \xi$ and ${\sigma_{ij}}^2/\text{mean}(V({\sigma_i}^2)) > \tau$)
(31) mark $s_{ij}$ as a Spikes;
(32) return;
(33) procedure **IsOutlier**($v_i$)
(34) use $U$ and $V$ to caculate $\theta_{ij}$;
(35) if ($\theta_{ij} > \varepsilon_{ij}$) mark $s_{ij}$ as an Outlier;
(36) return;
(37) procedure **IsRatStatChange**($v_i$)
(38) use the last $n$ values of $s_i$ to fit the trend vector $S_i$;
(39) for each rational trend $r_i$ in $R_i$
(40) if (cosine($r_i, S_i$) < $\theta_i/4$)
(41) mark $S_i$ as a rational status change; $S_{ij} = \text{mean}(S_{ij}, r_{ij})$;
(42) mark the max unstable sensor $s_{ij}$ as a sensor falut;
(43) return;
(44) procudure **SelfLearningThread**()
(45) while (true) do:
(46) load the user feedback;
(47) for each miss alart
(48) if missed Outlier then $\varepsilon_{ij} = \varepsilon_{ij} \div \lambda$;
(49) if missed irrational status change then $\theta_i = \theta_i \times \lambda$;
(50) **IncClustering**($S_i$, *null*);
(51) for each false alart
(52) if false Outlier then $\varepsilon_{ij} = \varepsilon_{ij} \times \lambda$;
(53) if false irrational status change then $\theta_i = \theta_i \div \lambda$;
(54) **IncClustering**($S_i, r_{ij}$);
(55) sleep($T$); continues;

Algorithm 1: Continued.

```
(56) return;
(57) procedure IncClustering(S_i, r_ij)
(58)     if r_ij is not null then for each S_ij in S_i;
(59)       if there is cosine(r_i, S_i) < θ_i then S_ij = mean(S_ij, r_ij);
(60)       else add r_ij into S_i;
(61)    for each two S_ij, S_ik in S_i;
(62)         if (cosine(S_ij, S_ik) < θ_i)
(63)              S_ij = mean(S_ij, S_ik); remove S_ik;
(64) return;
```

Algorithm 1: A group-based fault detection Algorithm.

(3) The production line is normally divided into many units according to the geographical position. We can ignore the relationship between sensors in different sections.

For the 11 sensors shown in Figure 2, we divide them into four groups which are $G_0 = \{P_{0-1}, T_{0-1}\}$, $G_1 = \{P_{1-1}, L_{1-1}, T_{1-1}\}$, $G_2 = \{P_{2-1}, L_{2-1}, T_{2-1}\}$, and $G_3 = \{P_{3-1}, L_{3-1}, T_{3-1}\}$. $G_0$ represents the status of the main input pipeline. $G_1$, $G_2$, and $G_3$ are status groups which are derived from the three parallel subpipelines respectively.

*3.3. Status Model.* For a status group $G = \{S_1, S_2, \ldots, S_n\}$, all the sensors in $G$ may have different trends when the production status changed. Figure 3 shows a status transformation process. $P_1$, $P_2$, and $T_1$ are stable in the first 30 and the last 20 seconds. The interval from 30 s to 40 s is called the Status Transform Window (STW). $P_1$ and $P_2$ increase to a new level while $T_1$ keep the same trend. And the slopes of $P_1$ and $P_2$ are different. We use the trend vector which contains all the slopes of one sensors's group to represent the status transformation, that is, $S = \{k_1, k_2, \ldots, k_n\}$. The trend of a sensor can be found by fitting its values by a specific size of STW. Here, we use the least-square method [22] (Formula (3)) to fit the values and record the rational trend vector by the sensor index. We use the key idea of incremental clustering algorithm [23] to handle the trend vectors, get the cosine angle between the current trend and existing vectors, respectively. If the angle is big enough then mark it a new trend. Otherwise, a repeated trend is found and we only need to merge it with the closest vector. In Section 4, the specific clustering method will be given for details:

$$\widehat{Y} = \alpha + \beta\widehat{X},$$
$$\alpha = \frac{\sum Y}{n} - \frac{\beta \sum X}{n}, \quad \beta = \frac{n\sum XY - \sum X \sum Y}{n\sum X^2 - (\sum X)^2}. \tag{3}$$

## 4. Design of Architecture

In this section we propose the architecture for sensor fault detection in industry monitoring at first. Then, how to realize our algorithm is discussed.

Table 1: Simulation data description.

| Feature | Value |
|---|---|
| Sampling rate | 60 s |
| Sensor count | 5800 |
| Total samples | 751.68 million |
| Status groups | 1340 |
| *Outliers* | 437 |
| *Stuck-at* & *Spikes* | 163 |
| RT missed | 961 |
| IRT missed | 35 |

*4.1. A Self-Learning Framework.* The self-learning sensor fault detection architecture is shown in Figure 4. There are four modules in our approach.

(1) Application DB: all the parameters are stored in the Application DB, including the threshold $\varepsilon_s$ for sensor $s$ and the history statistics result $\mu_i$, $\sigma_i^2$. The grouping information is also serialized in this database. This DB is the interface for high layer application which can get the sensor fault prediction and input the user feedback.

(2) Detection Thread: it is a background service and contains the main detecting process. Since our approach is an online detection, a series of detection thread will be created and maintained by the working thread pool and a related status queue. When new data is coming, the working thread dispatcher a wakes up a pending thread to handle it.

(3) Self-Learning Thread: the self-learning thread uses the OS timer as a driver. The user feedback about the detection result will be rechecked by this thread to revise the trend vectors.

*4.2. The GbFD Algorithm.* Dividing the sensors into status group is the key idea in our approach. For group-based sensor fault detection, we propose the (group-based fault detection) GbFD algorithm.

The GbFD Algorithm 1 starts by initializing the global parameters (line 6-7); then it instantiates the two core processes *SelfLearningThread* and *DetectionThread*. The input

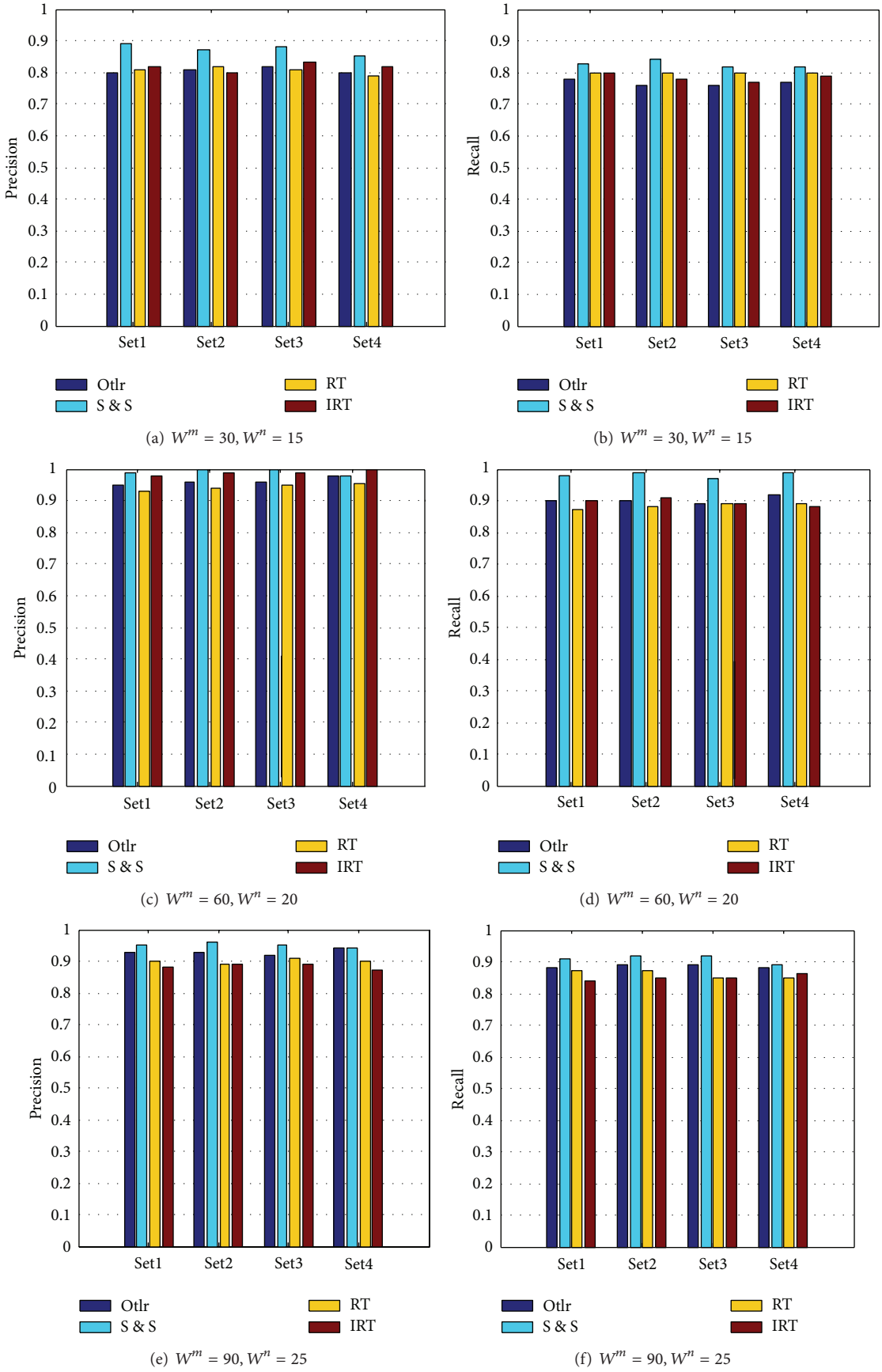



Figure 5: Four-folder cross-validation for precision (a, c, e) and recall (b, d, f) with different $W^m$ and $W^n$.

of GbFD is defined as a set that contains one time samples for a status group.

In the *DetectionTread*, we first check the *Stuck-at* faults and *Spikes* by calling the *IsStuck* and *IsSpikes* methods. And *IsRatStaChange* is called two times (line 19–21) to detect the sensor status transformation. When an *Outlier* was detected, GbFD should give the conclusion whether it is a rational operation such as shut down the device, and if no *Outlier* happened, we also need to check the abnormal status transformation. The first calling of *IsRatStaChange* is controlled by the *IsOutlier* method with an *AND* logic; no more algorithm complexity is increased by the twice calling. The *SelfLearningThread* works as a background service. Its responsibility is learning the user feedback to find out the missing detection and the false detection and adjusting the relational parameters. For a new rational trend vector, we will add it to the proper group rational trend history by the *IncClustering* method, which can recluster the grouped trend vectors according to the specified angle threshold.

The time complexity of the GbFD algorithm is dominated by *IsOutlier* and *IsRatStuChange* methods. For a sensor in group $G$, the detection computation takes $O((W^n + d) \times W^m)$, $W^n$ is the STW size, $d$ is the length of $G$, and $W^m$ is the SSW size. In a real application, the two windows sizes are steerable and $d$ is less than 20 for the most part. Moreover, a proper thread dispatch mechanism will guarantee GbFD to handle the real-time detection task. And, the self-learning process is more complex due to many iterative operations, but it is a background service and does not require real-time performance.

## 5. Experimental Evaluation

We built an application to evaluate our framework. This application was implemented in JAVA, about 2400 lines code and used Oracle as the Application DB.

*5.1. Data Preparation.* We use real data from an oil field in China. This oil field has 20 oil/gas treatment plants and all of the production equipments are monitored by sensors which can be mainly classified as temperature sensor, pressure sensor, and liquid level sensor. The sampling rate of the production IoT is 60 seconds. We obtained the sample data of 10,000 sensors between January 1st 8 PM to October 31th 8 AM, 2012. Since the production lines are relatively stable, we filtered the original data by two steps. Firstly, some production units that never make a mistake were eliminated. Secondly, for a plant, we discard some data in its stable period.

Table 1 shows features of our simulation dataset. We choose more than 751 million samples from 5800 sensors. According to the corresponding flow charts, we separate the sensors into 1340 groups. Each group has 4.33 sensors in average, and the maximum group contains 8 sensors. We analyze the user feedback history and find out four typical errors. *Outlier* means that a value runs out of range and it is caused by the sensor failure. We put *Stuck-at* faults and *Spikes* together. The *Rational Trend* (RT) missed fault means that sensors submit an exceptional value after a rational user operation, such as shuting down a device for examination. And the *Irrational Trend* (IRT) missed fault means that something wrong happened with the production line and the sensor values changed with it but still suitable with the threshold condition.

*5.2. Experiment.* We split the simulation data into four datasets according to the time sequence. Each time we use optional three data sets to train the GbFD algorithm and the remainder is used as the testing data.

We use three pairs SSW size and STW size, which are (30, 15), (60, 20), and (90, 25), to run the four-folder cross-validation test. The result is shown in Figure 5. When $W^m = 30$ and $W^n = 15$, each cross get a precision of about 80%. This result is not good enough. A satisfactory result is generated in the next test; $W^m = 60$ and $W^n = 20$ get a mean 95% accuracy. But with the increase of two windows size, the accuracy of GbFD will go to an opposite direction. This phenomenon indicates that the sizes of SSW and STW are strong correlating with our algorithm. Choosing the proper windows size can increase the detection sensitivity and the windows size (60, 20) is suitable with our simulation data.

## 6. Conclusions

We present a self-learning sensor fault detection framework in this paper. We propose a model which can represent the sensor value, sensor relationship, and sensor status transformation. GbFD algorithm is proposed to detect the sensor fault. And we use real data from an oil field for validation. Experimental results show that our system can detect 95% of data fault in the simulation data which contains 751.68 million samples from 5800 sensors.

We will continue validate our approach on other dataset to find out the proper statistical sliding window size and status transform windows size in different application contexts. Our goal is to build a sensor health management system for industry IoT that includes not only sensor fault detection, but also sensor lifecycle prediction and sensor inspection management.

## Acknowledgments

The authors would like to thank Jun Ma for helping them to dispose the simulation data. This research is supported by the National Natural Science Funds of China (61202238), the State Key Laboratory of Software Development Environment Funds (SKLSDE-2011ZX-08), and the Fundamental Research Funds for the Central Universities (2012RC0209).

## References

[1] Y. Liu and G. Zhou, "Key technologies and applications of internet of things," in *Proceedings of the 5th International Conference on Intelligent Computation Technology and Automation (ICICTA '12)*, pp. 197–200, Zhangjiajie, China, January 2012.

[2] W. Baoyun, "Review on Internet of Things," *Journal of Electronic Measurement and Instrument*, vol. 23, no. 12, pp. 1–7, 2009.

[3] S. Haller, S. Karnouskos, and C. Schroth, "The internet of things in an enterprise context," in *Future Internet*, vol. 5468 of *Lecture Notes in Computer Science*, pp. 14–28, Springer, Berlin, Germany, 2009.

[4] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–105, 2002.

[5] L. Paradis and Q. Han, "A survey of fault management in wireless sensor networks," *Journal of Network and Systems Management*, vol. 15, no. 2, pp. 171–190, 2007.

[6] Y. Zhang, N. Meratnia, and P. Havinga, "Outlier detection techniques for wireless sensor networks: a survey," *IEEE Communications Surveys and Tutorials*, vol. 12, no. 2, pp. 159–170, 2010.

[7] K. Ni, N. Ramanathan, M. N. H. Chehade et al., "Sensor network data fault types," *ACM Transactions on Sensor Networks*, vol. 5, no. 3, pp. 1–29, 2009.

[8] N. Bressan, L. Bazzaco, N. Bui, P. Casari, L. Vangelista, and M. Zorzi, "The deployment of a smart monitoring system using wireless sensor and actuator networks," *Proceedings of International Conference on Smart Grid Communications*, pp. 49–54, 2010.

[9] A. P. Castellani, M. Gheda, N. Bui, M. Rossi, and M. Zorzi, "Web services for the Internet of things through CoAP and EXI," in *Proceedings of the IEEE International Conference on Communications Workshops (ICC '11)*, pp. 1–6, Kyoto, Japan, June 2011.

[10] S. Yuan, X. Lai, X. Zhao, X. Xu, and L. Zhang, "Distributed structural health monitoring system based on smart wireless sensor and multi-agent technology," *Smart Materials and Structures*, vol. 15, no. 1, pp. 1–8, 2006.

[11] N. Bui and M. Zorzi, "Health care applications: a solution based on the Internet of Things," in *Proceedings of the 4th International Symposium on Applied Sciences in Biomedical and Communication Technologies (ISABEL '11)*, article 131, October 2011.

[12] C. Guestrin, P. Bodik, R. Thibaux, M. Paskin, and S. Madden, "Distributed regression: an efficient framework for modeling sensor network data," in *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN '04)*, pp. 1–10, Berkeley, Calif, USA, April 2004.

[13] J. Considine, M. Hadjieleftheriou, F. Li, J. Byers, and G. Kollios, "Robust approximate aggregation in sensor data management systems," *ACM Transactions on Database Systems*, vol. 34, no. 1, article 6, 2009.

[14] M. B. Greenwald and S. Khanna, "Power-conserving computation of order-statistics over sensor networks," in *Proceedings of the 23rd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS '04)*, pp. 275–285, Paris, France, June 2004.

[15] A. Jain, E. Y. Chang, and Y.-F. Wang, "Adaptive stream resource management using Kalman Filters," in *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD '04)*, pp. 11–22, Paris, France, June 2004.

[16] C. Olston, J. Jiang, and J. Widom, "Adaptive filters for continuous queries over distributed data streams," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 563–574, June 2003.

[17] K. Yamanishi, J.-I. Takeuchi, and G. Williams, "On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms," in *Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '01)*, pp. 320–324, August 2000.

[18] H.-L. Chan, T.-W. Lam, L.-K. Lee, and H.-F. Ting, "Continuous monitoring of distributed data streams over a time-based sliding window," *Algorithmica*, vol. 62, no. 3-4, pp. 1088–1111, 2012.

[19] M. Ding, D. Chen, K. Xing, and X. Cheng, "Localized fault-tolerant event boundary detection in sensor networks," in *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '05)*, pp. 902–913, March 2005.

[20] J.-L. Gao, Y.-J. Xu, and X.-W. Li, "Weighted-median based distributed fault detection for wireless sensor networks," *Journal of Software*, vol. 18, no. 5, pp. 1208–1217, 2007.

[21] B. Krishnamachari and S. Iyengar, "Distributed Bayesian algorithms for fault-tolerant event region detection in wireless sensor networks," *IEEE Transactions on Computers*, vol. 53, no. 3, pp. 241–250, 2004.

[22] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *Journal of the Society For Industrial & Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.

[23] X. Su, Y. Lan, R. Wan, and Y. Qin Y, "A fast incremental clustering algorithm," in *Proceedings of the International Symposium on Information Processing (ISIP '09)*, pp. 175–178, Huangshan, China, 2009.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 436368, 7 pages
http://dx.doi.org/10.1155/2013/436368

*Research Article*

# Classification Based on both Attribute Value Weight and Tuple Weight under the Cloud Computing

**Yifeng Zheng, Zaixiang Huang, and Tianzhong He**

*Department of Computer Science and Engineering, Minnan Normal University, Zhangzhou 363000, China*

Correspondence should be addressed to Yifeng Zheng; zhengyifengja@163.com

Received 17 July 2013; Accepted 3 September 2013

Academic Editor: Yuxin Mao



In recent years, more and more people pay attention to cloud computing. Users need to deal with magnanimity data in the cloud computing environment. Classification can predict the need of users from large data in the cloud computing environment. Some traditional classification methods frequently adopt the following two ways. One way is to remove instance after it is covered by a rule, another way is to decrease tuple weight of instance after it is covered by a rule. The quality of these traditional classifiers may be not high. As a result, they cannot achieve high classification accuracy in some data. In this paper, we present a new classification approach, called classification based on both attribute value weight and tuple weight (CATW). CATW is distinguished from some traditional classifiers in two aspects. First, CATW uses both attribute value weight and tuple weight. Second, CATW proposes a new measure to select best attribute values and generate high quality classification rule set. Our experimental results indicate that CATW can achieve higher classification accuracy than some traditional classifiers.

## 1. Introduction

Cloud computing has become a hot issue in recent years. With the rapid development of information technology and the popularity of cloud computing, it is necessary to mine useful information from magnanimity data [1–9]. Classification is one of the most important tasks in the data mining and the machine learning. Classification can predict the need of users from large data. First, it builds classification rules from training dataset. Second, it uses these rules to predict the class label of new instances.

The traditional classifiers [10–19] frequently adopt the following two ways. Some traditional classifiers remove instance after it is covered by a rule, such as FOIL [20] and ELEM2 [21]. Other traditional classifiers decrease tuple weight of instance after it is covered by a rule, such as PRM and CPAR [22]. Then, we introduce the feature of these classifiers. In the process of extracting rules, FOIL uses measure gain to select a best attribute value and generates one classification rule. It removes instance after it is covered by a rule. As a result, this method is ineffective. It generates a small rule set and cannot achieve high accuracy in some data. ELEM2 uses another measure to generate classification rules. It also removes instance after it is covered by a rule. ELEM2 considers the degree of relevance of an attribute-value pair and selects the most relevant pairs to generate rules. PRM modifies FOIL to achieve higher accuracy. PRM does not remove instance when it is covered by a rule. PRM gives the instance a tuple weight. Thus, PRM can insure that each instance is covered more than once. PRM selects only the best gain to generate rule. CPAR stands in the middle between exhaustive and greedy algorithms and combines the advantages of both. CPAR selects several best attribute values and builds several rules at one time. It does not remove instance immediately when it is covered by a rule. CPAR also uses tuple weight to guarantee that each instance can be covered more than once. These methods do not employ attribute value weight. They cannot get high quality classification rule set. As a result, they can not achieve high classification accuracy in some data.

In this paper, we propose a new algorithm, named classification based on both attribute value weight and tuple weight (CATW). CATW uses the both attribute value weight and tuple weight. Moreover, CATW uses a new measure to improve the quality of classification rule set. Our method has following advantages.

(1) After an instance is covered by a rule, instead of removing it, its weight is decreased by multiplying a factor. Thus, we can guarantee that each instance can be covered more than once.

(2) If we only use tuple weight, we cannot change the importance of an attribute-value pair in the dataset. Therefore, CATW uses attribute value weight to reduce the importance of attribute-value pair after the rule is generated. In this way, CATW can increase the chances of attaining other optimal attribute-value pairs. We can generate more high quality of rules.

(3) CATW presents a new measure to select the best attribute value. CATW uses two different measures: support and correlation confidence. If two different attribute-value pairs have same correlation confidence, CATW considers their support.

Experimental results indicate that: (1) if the instance is removed immediately after it is covered by a rule, the classifier generates a very small number of rules; (2) if the classifier is only using tuple weight, the quality of classification rule set is not good. (3) Since CATW uses both attribute value weight and tuple weight, it achieves high classification accuracy.

The outline of this paper is as follows. Section 2 presents the details of CATW and describes the process of rule generation in CATW. Section 3 discusses how to predict class label using the rules. The experimental results are presented in Section 4. Finally, we conclude the study in Section 5.

## 2. Rule Generation of CATW

The algorithm of CATW has three special points: the attribute value weight, the tuple weight, and the improved measure. First, we describe the method of how to use tuple weight. Second, we introduce the use of attribute weight. Third, we propose a new measure to generate high quality classification rule set. Finally, we show the whole process of how to generate rule set.

Let $T$ be a set of tuples. Each tuple $t$ has attributes $\{A_1, A_2, \ldots, A_m\}$. Suppose $C$ to be a set of class labels $\{C_1, C_2, \ldots, C_k\}$, where $k$ means the number of class label.

*Definition 1* (a literal). A literal $p$ is an attribute-value pair, which follows the pattern of $(A_i, v)$, where $A_i$ is an attribute and $v$ is a value of attribute $A_i$.

*Definition 2* (a classification rule). $X \rightarrow c$ is called a classification rule $r$, if $X$ consists of a conjunction of literals $p_1, p_2, \ldots, p_l$ with the form of $p_1 \wedge p_2 \wedge \cdots \wedge p_l$, where $c$ is a class label.

A tuple $t$ satisfies the antecedent of $r$ if and only if it has all literals in $r$. If $t$ satisfies the antecedent of $r$, $r$ predicts that $t$ has a class label $c$.

*2.1. The Tuple Weight.* In traditional classification, all rules are generated from the training database. If a tuple $t$ is covered by a rule $r$, they can not ensure that $r$ is the best rule for $t$. If $r$ is generated from the remaining dataset instead of the whole dataset [22], $r$ may not be the best rule. In order to improve the classification accuracy and increase the number of rules, some traditional classifiers use tuple weight. By depending on tuple weight, these classifiers can delay removing instance after it is covered by a rule. In our algorithm, after a tuple is covered by a rule, instead of removing it, its weight is decreased by multiplying a factor. We set a threshold for tuple weight. When the tuple weight of tuple $t$ is less than threshold, we remove the tuple $t$ from training data. CATW produces more rules. Each tuple can be covered by classification rules more than once.

In our approach, we can set an initial threshold and an end threshold. We can limit the number of rules which are generated according to actual situation. If we set a small end threshold, it generates a large number of rules. On the contrary, if we set a large end threshold, it generates a less number of rules. In our experiment, we set an initial threshold 1, a weight factor 0.75. Moreover, we set an end threshold. The end threshold is the third power of weight factor. We can make sure that each instance can be covered three times.

*2.2. The Attribute Value Weight.* Some traditional classifiers only use tuple weight. They do not change the importance of an attribute-value pair in the training data. After a rule is generated, these classifiers may select the duplicate attribute-value pair. Thus, they may miss some high quality rules which can be used to affect the classification accuracy. CATW uses attribute value weight to reduce the importance of attribute-value pair after the rule is generated. When the tuple is covered by a rule, our algorithm can reduce the importance of attribute-value pairs which are contained in it. In this way, we can increase the chances of attaining another optimal attribute-value pair.

*Example 3.* The following training dataset with two classes is shown in Table 1. Then, we demonstrate how to use attribute value weight.

Suppose $r$ = {OUTLOOK = rain $\wedge$ WINDY = TRUE $\rightarrow$ PLAY = no} to be just generated. Then, we set a weight factor 0.8, and set {PLAY = no} for positive examples. After a rule is generated, CATW uses weight factor to reduce the importance of all attribute values that are contained in antecedent of the rule in positive examples. The result is shown in Table 2.

The results of our experiment indicate that classification accuracy is influenced by attribute value weight. Compared with the classifiers which do not use attribute value weight, CATW can achieve higher classification accuracy in some data. Thus, the attribute value weight can be a help to improve the quality of classification rule.

*2.3. The Measure of CATW.* Some classifiers use FOIL gain to select literal. FOIL gain is used to measure the information gained from adding literal $p$ to the current rule. Let us suppose that $|P|$ means the number of positive examples which satisfies the antecedent of the current rule $r$ and $|N|$

Table 1: The training dataset.

| | Outlook | Temperature | Humidity | Windy | Play |
|---|---|---|---|---|---|
| | Sunny | Hot | >75 | False | No |
| AW | 1 | 1 | 1 | 1 | |
| | Sunny | Hot | >75 | True | No |
| AW | 1 | 1 | 1 | 1 | |
| | Overcast | Hot | >75 | False | Yes |
| AW | 1 | 1 | 1 | 1 | |
| | Rain | Mild | >75 | False | Yes |
| AW | 1 | 1 | 1 | 1 | |
| | Rain | Cool | >75 | False | Yes |
| AW | 1 | 1 | 1 | 1 | |
| | Rain | Cool | ≤75 | True | No |
| AW | 1 | 1 | 1 | 1 | |
| | Sunny | Mild | ≤75 | True | Yes |
| AW | 1 | 1 | 1 | 1 | |

Table 2: Attribute value weight in positive examples.

| | Outlook | Temperature | Humidity | Windy | Play |
|---|---|---|---|---|---|
| | Sunny | Hot | >75 | False | No |
| AW | 1 | 1 | 1 | 1 | |
| | Sunny | Hot | >75 | True | No |
| AW | 1 | 1 | 1 | 0.8 | |
| | Overcast | Hot | >75 | False | Yes |
| AW | 1 | 1 | 1 | 1 | |
| | Rain | Mild | >75 | False | Yes |
| AW | 1 | 1 | 1 | 1 | |
| | Rain | Cool | >75 | False | Yes |
| AW | 1 | 1 | 1 | 1 | |
| | Rain | Cool | ≤75 | True | No |
| AW | 0.8 | 1 | 1 | 0.8 | |
| | Sunny | Mild | ≤75 | True | Yes |
| AW | 1 | 1 | 1 | 1 | |

Table 3: Characteristics of UCI datasets.

| Dataset | No. of instances | No. of attributes | No. of class |
|---|---|---|---|
| Auto | 205 | 25 | 7 |
| Cleve | 303 | 13 | 2 |
| Glass | 214 | 9 | 7 |
| Heart | 270 | 13 | 2 |
| Hepati | 155 | 19 | 2 |
| Horse | 368 | 22 | 2 |
| Iono | 351 | 34 | 2 |
| Iris | 150 | 4 | 3 |
| Labor | 57 | 16 | 2 |
| Lymph | 148 | 18 | 4 |
| Wine | 178 | 13 | 3 |
| Zoo | 101 | 16 | 7 |

means the number of negative examples which satisfy the antecedent of the current rule $r$. After literal $p$ is added to $r$, $|P^*|$ means the number of positive examples which satisfy the antecedent of the new rule, and $|N^*|$ means the number of negative examples which satisfy the antecedent of the new rule [22]. The FOIL gain of $p$ is defined as:

$$\text{gain}(p) = |P^*| \left( \log \left( \frac{|P^*|}{(|P^*| + |N^*|)} \right) - \log \left( \frac{|P|}{(|P| + |N|)} \right) \right). \tag{1}$$

In our experiment, we employ two different improved measures.

*2.3.1. Improved FOIL Measure.* In our experiment, $|P|$ means total tuple weight of positive examples which satisfy the antecedent of current rule $r$. $|N|$ means total tuple weight of negative examples which satisfy the antecedent of current rule $r$. After literal $p$ is added to $r$, $|P^*|$ means total attribute value weight of literal $p$ in positive examples, and $|N^*|$ means total attribute value weight of literal $p$ in negative examples. Therefore, CATW uses both tuple weight and attribute value weight when it measures literal $p$. We call this measure an improved FOIL measure.

*2.3.2. Improved Correlation Measure.* In traditional FOIL gain, $|P^*|$ has a huge influence to select a best attribute value. For example, if $\log(|P^*|/(|P^*| + |N^*|)) - \log(|P|/(|P| + |N|))$ is too small and $|P^*|$ is too large, the result of gian($p$) is not the best for rule. We use two different measures: support and correlation confidence. We divide the traditional FOIL measure in two parts.

(1) PART I: $|P^*|$.

(2) PART II: $\log(|P^*|/(|P^*|+|N^*|)) - \log(|P|/(|P|+|N|))$.

When we select literal $p$, a global order of literal $p$ is composed. Given two literal $p_1$ and $p_2$, $p_1$ is better than $p_2$, denoted as $p_1 > p_2$.

$p_1 > p_2$ if and only if (1) PART II ($p_1$) > PART II ($p_2$) or (2) PART II ($p_1$) = PART II ($p_2$) and PART I ($p_1$) > PART I ($p_2$). We call this measure an improved correlation measure.

*2.4. Algorithm of CATW.* In this part, we will introduce our algorithm in detail. The CATW algorithm is presented in Algorithm 1.

## 3. Classification of CATW

Before making any prediction, we use the Laplace expected error estimate [23] to evaluate the quality of rules. It is defined as follows:

$$\text{Laplace accuracy} = \frac{(n_c + 1)}{(n_{\text{tot}} + k)}, \tag{2}$$

where $k$ is the number of classes and $n_{\text{tot}}$ is the total number of examples satisfying the antecedent of rule, among which $n_c$ examples belong to $c$.

Table 4: The accuracy of CATW with improved FOIL gain measure.

| | FOIL | CMAR | CPAR | TW | AW(0.8)/TW | AW(0.5)/TW |
|---|---|---|---|---|---|---|
| Auto | 0.776 | 0.781 | 0.82 | 0.7984 | 0.7934 | 0.7883 |
| Cleve | 0.7423 | 0.822 | 0.815 | 0.7695 | 0.7907 | 0.8014 |
| Glass | 0.7156 | 0.701 | 0.744 | 0.7385 | 0.7481 | 0.7481 |
| Heart | 0.8148 | 0.822 | 0.826 | 0.8214 | 0.8095 | 0.8095 |
| Hepati | 0.78 | 0.805 | 0.794 | 0.8444 | 0.8579 | 0.8705 |
| Horse | 0.7124 | 0.826 | 0.842 | 0.7856 | 0.7915 | 0.8032 |
| Iono | 0.889 | 0.915 | 0.926 | 0.9109 | 0.9293 | 0.9263 |
| Iris | 0.9533 | 0.94 | 0.947 | 0.9583 | 0.9583 | 0.9583 |
| Labor | 0.7567 | 0.897 | 0.847 | 0.8148 | 0.9206 | 0.9365 |
| Lymph | 0.7424 | 0.831 | 0.823 | 0.8157 | 0.8380 | 0.8454 |
| Wine | 0.9379 | 0.95 | 0.955 | 0.9526 | 0.9708 | 0.9708 |
| Zoo | 0.9409 | 0.971 | 0.951 | 0.9503 | 0.9310 | 0.9310 |
| Average | 0.8134 | 0.8551 | 0.8575 | 0.8467 | 0.8616 | 0.8658 |

Table 5: The accuracy of CATW with improved correlation measure.

| | FOIL | CMAR | CPAR | TW | AW(0.8)/TW | AW(0.5)/TW |
|---|---|---|---|---|---|---|
| Auto | 0.776 | 0.781 | 0.82 | 0.7927 | 0.8054 | 0.7773 |
| Cleve | 0.7423 | 0.822 | 0.815 | 0.7941 | 0.7945 | 0.8227 |
| Glass | 0.7156 | 0.701 | 0.744 | 0.7385 | 0.7385 | 0.7433 |
| Heart | 0.8148 | 0.822 | 0.826 | 0.8056 | 0.8333 | 0.8294 |
| Hepati | 0.78 | 0.805 | 0.794 | 0.8305 | 0.8370 | 0.8644 |
| Horse | 0.7124 | 0.826 | 0.842 | 0.7236 | 0.7942 | 0.7915 |
| Iono | 0.889 | 0.915 | 0.926 | 0.9137 | 0.9352 | 0.9322 |
| Iris | 0.9533 | 0.94 | 0.947 | 0.9583 | 0.9583 | 0.9583 |
| Labor | 0.7567 | 0.897 | 0.847 | 0.918 | 0.9180 | 0.9365 |
| Lymph | 0.7424 | 0.831 | 0.823 | 0.7676 | 0.8310 | 0.8380 |
| Wine | 0.9379 | 0.95 | 0.955 | 0.9646 | 0.9646 | 0.9532 |
| Zoo | 0.9409 | 0.971 | 0.951 | 0.9495 | 0.9697 | 0.9596 |
| Average | 0.8134 | 0.8551 | 0.8575 | 0.8464 | 0.865 | 0.8672 |

When using rules to predict the class-label of unknown instance, we use several rules which are matched by the instance. If all the rules have the same consequent of rule, we assign that label to the instance. If all the best rules have several classes, we calculate the average Laplace accuracy of each class. Then, we select the class label with the highest average value and assign it to the instance.

## 4. Experimental Results

All experiments are performed on 12 different datasets from the UCI data collection. All datasets were conducted using stratified tenfold cross-validation. In cross-validation, the data set is divided into 10 blocks. Each block is held out once. The classifier is trained on the remaining 9 blocks. The character of each dataset is shown in Table 3. We perform our experiments on a 2.2 GHz PC with 2 G memory, running Microsoft Windows XP.

In Tables 4 and 5, Column 1 shows the accuracy of FOIL. Column 2 shows the accuracy of CMAR. Column 3 shows the accuracy of CPAR. Column 4 shows the accuracy of CATW without attribute value weight, set tuple weight 0.75. Column 5 shows the accuracy of CATW, set attribute value weight 0.8 and tuple weight 0.75. Column 6 shows the accuracy of CATW, set attribute value weight 0.5 and tuple weight 0.75.

In Table 4, we use the measure which is an improved FOIL measure. Figure 1 gives the accuracy of FOIL, CMAR, CPAR, and CATW based on Table 4. CATW uses both attribute value weight and tuple weight and employs the improved FOIL measure. From Figure 1 and Table 4, we can see that CATW can achieve higher accuracy than FOIL, CMAR, and CPAR.

In Table 5, we use the measure which is an improved correlation measure. Figure 2 gives the accuracy of FOIL, CMAR, CPAR, and CATW based on Table 5. CATW uses both attribute value weight and tuple weight and employs the improved correlation measure. From Figure 2 and Table 5, we can see that CATW with improved correlation measure can also achieve higher accuracy than FOIL, CMAR, and CPAR.

By comparison, the accuracy of CATW with the improved correlation measure is higher than the accuracy of CATW with the improved FOIL measure. From Tables 4 and 5, we can see that it is necessary to use the improved correlation measure.

Table 6: Comparing different attribute value weights with improved FOIL gain measure.

| | AW(0.8)/TW | AW(0.75)/TW | AW(0.67)/TW | AW(0.5)/TW | AW(0.33)/TW |
|---|---|---|---|---|---|
| Auto | 0.7934 | 0.7881 | 0.7730 | 0.7883 | 0.8035 |
| Cleve | 0.7907 | 0.7946 | 0.7874 | 0.8014 | 0.7836 |
| Glass | 0.7481 | 0.7481 | 0.7530 | 0.7481 | 0.7433 |
| Heart | 0.8095 | 0.8135 | 0.8056 | 0.8095 | 0.8054 |
| Hepati | 0.8579 | 0.8640 | 0.8709 | 0.8705 | 0.8239 |
| Horse | 0.7915 | 0.7735 | 0.7884 | 0.8032 | 0.7971 |
| Iono | 0.9293 | 0.9230 | 0.9294 | 0.9263 | 0.9263 |
| Iris | 0.9583 | 0.9583 | 0.9583 | 0.9583 | 0.9583 |
| Labor | 0.9206 | 0.9206 | 0.9048 | 0.9365 | 0.9048 |
| Lymph | 0.8380 | 0.8181 | 0.8449 | 0.8454 | 0.8523 |
| Wine | 0.9708 | 0.9766 | 0.9708 | 0.9708 | 0.9766 |
| Zoo | 0.9310 | 0.9310 | 0.9310 | 0.9310 | 0.9209 |
| Average | 0.8616 | 0.8591 | 0.8598 | 0.8658 | 0.858 |

Table 7: Comparing different attribute value weights with improved correlation measure.

| | AW(0.8)/TW | AW(0.75)/TW | AW(0.67)/TW | AW(0.5)/TW | AW(0.33)/TW |
|---|---|---|---|---|---|
| Auto | 0.8054 | 0.7876 | 0.7720 | 0.7773 | 0.7670 |
| Cleve | 0.7945 | 0.8085 | 0.8087 | 0.8227 | 0.8264 |
| Glass | 0.7385 | 0.7283 | 0.7431 | 0.7433 | 0.7431 |
| Heart | 0.8333 | 0.8254 | 0.8294 | 0.8294 | 0.8056 |
| Hepati | 0.8370 | 0.8574 | 0.8439 | 0.8644 | 0.8513 |
| Horse | 0.7942 | 0.7913 | 0.7915 | 0.7915 | 0.7738 |
| Iono | 0.9352 | 0.9261 | 0.9322 | 0.9322 | 0.9353 |
| Iris | 0.9583 | 0.9583 | 0.9583 | 0.9583 | 0.9583 |
| Labor | 0.9180 | 0.9180 | 0.9339 | 0.9365 | 0.9180 |
| Lymph | 0.8310 | 0.8028 | 0.8241 | 0.8380 | 0.8245 |
| Wine | 0.9646 | 0.9529 | 0.9412 | 0.9532 | 0.9587 |
| Zoo | 0.9697 | 0.9596 | 0.9596 | 0.9596 | 0.9495 |
| Average | 0.865 | 0.8597 | 0.8615 | 0.8672 | 0.8593 |

Table 6 displays the accuracy of different attribute value weights in CATW. In Table 6, CATW employs the improved FOIL measure. Table 7 displays the accuracy of different attribute value weights in CATW. In Table 7, CATW employs the improved correlation measure. The results of the two tables indicate that (1) the accuracy of improved correlation measure is higher than the accuracy of improved FOIL measure and (2) different value of attribute value weight has different influence on the accuracy of classification.

Through all the above results of our experiment, we can conclude that (1) it is necessary to use attribute value weight and tuple weight; (2) it is necessary to use improved correlation measure; (3) different value of attribute value weight has different influence on the accuracy of classification.

## 5. Conclusions and Future Work

With the rapid development of information technology and the popularity of cloud computing, it is necessary to mine useful information from magnanimity data. Some traditional classification methods frequently adopt the following two ways. One way is that it does not use tuple weight to remove instance after it is covered by a rule. Another way is that it only gives tuple weight of instance after it is covered by a rule. As result, they cannot achieve high classification accuracy in some data. In this paper, we present a novel approach CATW. First, CATW uses both attribute value weight and tuple weight. Second, CATW proposes a new measure which is the improved correlation measure. CATW employs the improved correlation measure to select best attribute values and generate high quality classification rule set. The results of our experiment indicate that CATW can generate a reasonable number of classification rules. In addition, CATW can achieve high classification accuracy. Our experiment shows that different value of attribute value weight has different influence on the accuracy of classification. At present, we cannot find the regular change in selecting an optimal attribute value weight. In future research, we will focus on it. We also focus on another research. We will combine distributed data mining with cloud computing platform in order to improve the efficiency of CATW.

```
Input: Training set D = P ∪ N (P and N are the sets of all positive and negative example, respectively)
Output: A set of rules for predicting class labels for examples
Procedure CATW
   attributeWeight ← α
   tupleWeight ← β
   tupleThreshold ← λ
   rules ← null
   while |P| > 0
      N′ ← N, P′ ← P
      r ← null
      while |N| > 0 and r.length < max_rule_length
         find the best attribute value av use the improved correlation measure combine tuple weight with attribute weight
         add av to r
         remove from P′ all examples not satisfying r
         remove from N′ all examples not satisfying r
      end
      add r to rules
      for each attribute at that is included in antecedent of r in P
         at.weight ← attributeWeight ∗ at.weight
      end
      for each example t in P satisfying r's body
         t.weight ← tupleWeight ∗ t.weight
         if t.weight < tupleThreshold then remove t from P
      end
   end
return rules
```

Algorithm 1: Classification based on both attribute value weight and tuple weight (CATW).

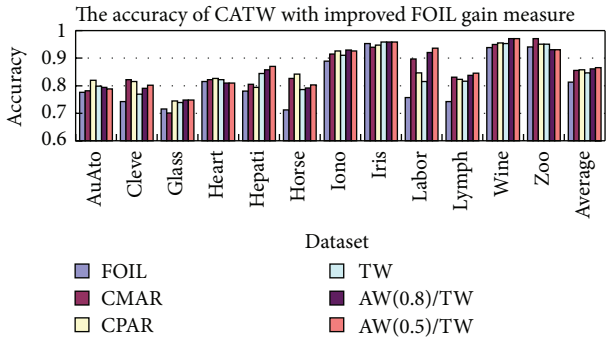



Figure 1: The accuracy of CATW with improved FOIL gain measure.

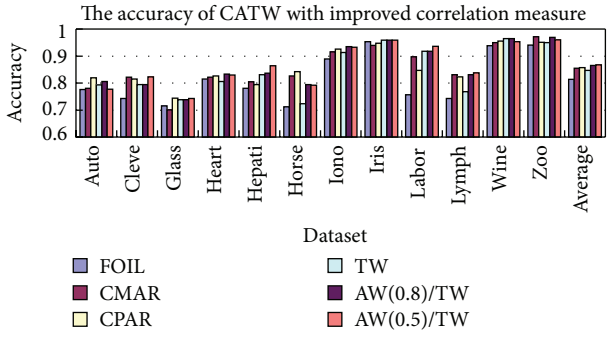



Figure 2: The accuracy of CATW with improved correlation measure.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## Acknowledgment

This work is funded by the China NFS Program (no. 61170129), and by the Fujian province NSF Program (no. 2013J01259).

## References

[1] K. Lal and N. C. Mahanti, "A novel data mining algorithm for semantic web based data cloud," *International Journal of Computer Science and Security*, vol. 4, no. 2, pp. 160–175, 2010.

[2] S. Adapa, M. Kalyan Srinivas, and A. V. R. K. Harsha Vardhan Varma, "A study on cloud computing data mining," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 1, no. 5, pp. 1232–1237, 2013.

[3] Z. Qureshi, J. Bansal, and S. Bansal, "A survey on association rule mining in cloud computing," *International Journal of Emerging Technology and Advanced Engineering*, vol. 3, no. 4, pp. 318–321, 2013.

[4] J. Ding and S. Yang, "Classification rules mining model with genetic algorithm in cloud computing," *International Journal of Computer Applications*, vol. 48, no. 18, pp. 24–32, 2012.

[5] L. Hu, Z. Zhang, F. Wang, and K. Zhao, "Optimization of the deployment of temperature nodes based on linear programing

in the internet of things," *Tsinghua Science and Technology*, vol. 18, no. 3, pp. 250–258, 2013.

[6] S. Gond, A. Patil, and V. B. Nikam, "A survey on parallelization of data mining techniques," *International Journal of Engineering Research and Applications*, vol. 3, no. 4, pp. 520–526, 2013.

[7] N. Mishra, S. Sharma, and A. Pandey, "High performance cloud data mining algorithm and data mining in clouds," *IOSR Journal of Computer Engineering*, vol. 8, no. 4, pp. 54–61, 2013.

[8] A. Pareek and M. Gupta, "Review of data mining techniques in cloud computing database," *International Journal of Advanced Computer Research*, vol. 2, no. 2, pp. 52–55, 2012.

[9] R.-Ş. Petre, "Data mining in cloud computing," *Database Systems Journal*, vol. 3, no. 3, pp. 67–71, 2012.

[10] Y. Jiao, "Research of an improved apriori algorithm in data mining association rules," in *Proceedings of the IEEE International Conference on Information Theory and Information Security (ICITIS '11)*, November 2011.

[11] F. Thabtah, P. Cowling, and Y. Peng, "MCAR: multi-class classification based on association rule," in *Proceedings of the 3rd ACS/IEEE International Conference on Computer Systems and Applications*, pp. 127–133, January 2005.

[12] G. Dong, X. Zhang, L. Wong, and J. Li, "CAEP: classification by aggregating emerging patterns," *Discovery Science*, vol. 1721, pp. 30–42, 1999.

[13] J. Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation," in *Proceedings of the ACM SIGMOD international Conference on Management of Data (SIGMOD '00)*, pp. 1–12, 2000.

[14] W. Li, J. Han, and J. Pei, "CMAR: accurate and efficient classification based on multiple class-association rules," in *Proceedings of the 1st IEEE International Conference on Data Mining (ICDM '01)*, pp. 369–376, San Jose, Calif, USA, November 2001.

[15] F. A. Thabtah and P. I. Cowling, "A greedy classification algorithm based on association rule," *Applied Soft Computing Journal*, vol. 7, no. 3, pp. 1102–1111, 2007.

[16] B. Liu, W. Hsu, and Y. Ma, "Integerating classification and association rule mining," in *Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD '98)*, pp. 80–86, New York, NY, USA, August 1998.

[17] X. Wang, Z. Zhou, and G. Pan, "CMER: classification based on multiple excellent rules," *Journal of Theoretical and Applied Information Technology*, vol. 48, pp. 661–665, 2013.

[18] G. Chen, H. Liu, L. Yu, Q. Wei, and X. Zhang, "A new approach to classification based on association rule mining," *Journal of Decision Support Systems*, vol. 42, no. 2, pp. 674–689, 2006.

[19] P. Leng and F. Coenen, "The effect of threshold values on association rule based classification accuracy," *Journal of Data and Knowledge Engineering*, vol. 60, no. 2, pp. 345–360, 2007.

[20] J. Ross Quinlan and R. Mike Cameron-Jones, "FOIL: a midtern report," in *Proceedings of the European Conference Machine Learning*, pp. 3–20, Vienna, Austria, 1993.

[21] A. An, "Learning classification rules from data," *Computers & Mathematics with Applications*, vol. 45, no. 4-5, pp. 737–748, 2003.

[22] X. Yin and J. Han, "CPAR: classification based on predictive association rules," in *Proceedings of the Society for Industrial and Applied Mathematics (SIAM) International Conference on Data Mining*, May 2003.

[23] P. Clark and R. Boswell, "Rule induction with CN2: somerecent improvements," in *Proceedings of European Working Session on Learning (EWSL '91)*, pp. 151–163, Porto, Portugal, March 1991.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 765027, 6 pages
http://dx.doi.org/10.1155/2013/765027

*Research Article*

# A Parameter Matching Method of the Parallel Hydraulic Hybrid Excavator Optimized with Genetic Algorithm

**Xiaoliang Lai and Cheng Guan**

*Mechanical Design, Zhejiang University, Hangzhou 310027, China*

Correspondence should be addressed to Xiaoliang Lai; eric.laixl@gmail.com

Received 22 May 2013; Accepted 31 August 2013

Academic Editor: Mahmoud Abdel-Aty



This paper proposed a parameter matching method based on the energy storage unit, the accumulator in the parallel hydraulic hybrid excavator (PHHE). The working condition, system structure, and control strategy of the excavator were all considered. It took the 20-ton series PHHE as the example and displayed the parameter matching course of the main components: engine, accumulator, and hydraulic secondary regulatory pump. Their installed powers were reduced after the matching course. Furthermore, the parameters of the PHHE system were optimized with the genetic algorithm to get the most suitable values for system initialization. By analyzing the simulation results, it could be concluded that the parameter matching method had an impressive improvement of the energy saving under the same working condition and brought obscure influence to the mechanism dynamics.

## 1. Introduction

Due to the global energy crisis, the energy saving became the popular research aspect. Huge fuel consumption of hydraulic excavator, the most useful machine in the construction projects, attracted researcher's interest. Even though the hydraulic excavator had used the higher-efficiency diesel engine, it had a lot of problems on the fuel consumption efficiency for a long time until the hybrid technology occurred, because the excavator had complex working condition and sharply changed load. The hybrid technology was applied in the vehicle field firstly and then was introduced into the excavator field. The topic research aspect was focused on the parallel hybrid electric excavator in the recent years whose storage unit was Ni-H battery or superelectric capacity. The Ni-H battery was cheap in price but large in volume and its life greatly depended on the battery management system. On the other hand, the superelectric capacity had good characteristics but was expensive in price [1, 2]. Besides, the excavator had such more fluctuating load than the vehicle system that electric scheme would not be able to compensate the difference timely. As an exploration, the hydraulic scheme was proposed to overcome the shortcomings of the electric scheme mentioned above [3–5]. The hydraulic scheme used the accumulator to replace the energy storage unit in the electric scheme. It had the advantage of high power density and easy combination with the excavator system. In addition, with the potential energy from recovery subsystem, the fuel consumption would be better due to the reduction of the energy conversation process [6–8].

Meanwhile, the change of the components in the hydraulic hybrid excavator needed a new round of choosing on their patterns and parameters. A suitable parameter matching could maximize the efficiency, reduce the volume of every component, and cut down the system's mass and price. Papers [9–11] presented their parameter matching method the electric hybrid excavator and got an obvious improvement in the energy saving compared with the unmatched pattern. But their match courses only take the restrictions as a concern to get the approximate parameters and they were designed especially for the electric scheme. The parameters were often an experience value and had a big adjustable range.

Based on the reason mentioned before, this paper proposed a parameter matching method for the parallel hydraulic hybrid excavator (PHHE). It was based on the structure, load, and the control strategy of the excavator. And a genetic

algorithm was used for optimization to acquire more precise and suitable parameters of the components in the PHHE system.

## 2. Description of the Hydraulic Hybrid System Excavator

*2.1. The Principle of the Hydraulic Hybrid System.* The hydraulic hybrid system contained two power sources. One was the chemical energy generated by burning fuel in the diesel engine. The other one was the hydraulic energy generated by recycling the potential energy from gravitation. The first part was converted from mechanical energy to hydraulic energy, which was the surplus value of the engine output power. The second part was converted from gravitational potential energy to hydraulic energy, which was wasted during excavator working originally. The second part was a special form of the first part in some sense. However, the second part conversation process brought no extra load to the engine. Moreover, the conversation process would adjust the load curve, stabilize the engine speed, and improve the fuel consumption efficiency, because the process would supply energy in high load and absorb energy in low load. In this paper, the hydraulic energy only contained the boom gravitation because of its most typical and largest recoverable amount.

*2.2. The Structure of the PHHE System.* The structure of the hydraulic hybrid excavator system mentioned in this paper was shown in Figure 1. The multiple manifolds were the flow distribution device of the original hydraulic system. The main pump of the excavator's hydraulic system, the energy conversation unit, and the engine was connected coaxially which made up the parallel hybrid power style. The energy conversation unit, the energy storage unit and the energy controller constituted the assistant power subsystem. Here, the energy conversation unit meant the hydraulic secondary regulation component and the energy storage unit meant the hydraulic accumulator. When the energy conversation unit worked as a pump, it absorbed the surplus power of the engine and charged the energy storage unit. When the energy conversation unit worked as a motor, it supplied the lacking power of the engine and released the energy storage unit. In this system, the boom gravitational potential energy was directly charged into energy storage unit unlike the chemical energy. The chemical energy coming from the engine must be transformed by the energy conversation unit. But when the energy in the storage unit was reused, it would through the energy conversation unit only. The energy transmitted after the engine output was in the form of hydraulic fluid. The arrows in Figure 1 indicated the allowed flow direction in the hydraulic system.

## 3. The Modeling and Parameter Matching Method of PHHE

*3.1. The Modeling of the Components.* The diesel engine was the main power source of the whole PHHE system. Modeling

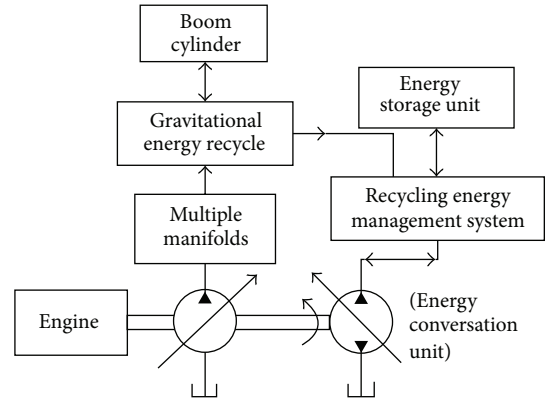



Figure 1: The structure of the hydraulic hybrid system.

the engine internal burning course was too complex and difficult. We simplify modeling the engine based on Newton's first law as (1)

$$T_l = T_e - J_e\dot{\omega} - C_e\omega, \tag{1}$$

where $T_l$ was the output torque of the engine, $T_e$ was the torque that forces on the axial, $J_e$ was equivalent engine inertia, $C_e$ was equivalent viscous damping, and $\omega$ was the engine angular velocity.

In this system, the energy storage unit meant the accumulator. We considered that the energy absorbing/releasing courses were both under the heat insulation. As a result, the accumulator was modeled as (2)

$$\Delta V = V_a \cdot \left(1 - \left(\frac{P_a}{P}\right)^{1/n}\right). \tag{2}$$

And the change of the energy stored in the accumulator could be calculated as (3)

$$F = \frac{P_a V_a}{n-1} \cdot \left[\left(\frac{P}{P_a}\right)^{(1-n)/n} - 1\right], \tag{3}$$

where $P_a$ was the accumulator precharge pressure, $P$ was the accumulator real-time pressure, $V_a$ was the accumulator volume, $\Delta V$ was the difference between the precharge volume and the real-time volume, and $n$ was the gas polytrophic index which was selected as 1.4.

In this system, the energy conversation meant the secondary regulatory pump which could work as pump or motor by adjusting the swash plate in different quadrant. The flow could be expressed as (4)

$$Q_m = \frac{\pi}{4} d^2 z n D \tan\beta, \tag{4}$$

where $Q_m$ was the flow of the secondary regulatory pump, $d$ was the diameter of the plunger, $z$ was the quantity of the plunger, $n$ was the speed of the axial, $D$ was the circle diameter of the plunger distribution, and $\beta$ was the angle of the swash plate. And the torque was expressed as (5)

$$T_m = \frac{\Delta P \cdot Q_m}{2\pi n}, \tag{5}$$

where $T_m$ was the theory torque of the secondary regulatory pump and $\Delta P$ was the difference between the inlet and the outlet pressures of the pump. Because the secondary regulatory pump could work as two states, the efficiency should be modeled separately. Equation (6) represented the efficiency under pump state and (7) represented the efficiency under motor state:

$$\eta_p = \frac{Q_m - \Delta Q}{Q_m} \tag{6}$$

$$\eta_m = \frac{Q_m}{Q_m + \Delta Q}, \tag{7}$$

where $\eta_p$ and $\eta_m$ were the efficiency rates and $\Delta Q$ was the difference between the inlet and outlet of the secondary regulatory pump.

In this system, the energy consumption unit meant the hydraulic cylinder. It could be described as (8)

$$M\ddot{x} + B\dot{x} + F(t) = p_1 A_1 - p_2 A_2, \tag{8}$$

where $x$ was the displacement of the piston, $M$ was the equivalent mass on the piston rod, $B$ was the cylinder viscous damping coefficient, $F(t)$ was the force on the cylinder, $p_1$ was the pressure of the chamber without rod, $A_1$ was the area of the chamber without rod, $p_2$ was the pressure of the chamber with rod, and $A_2$ was the area of the chamber with rod.

*3.2. The Parameter Matching Method.* Due to the low energy density and high power density of the PHHE system, the storage volume and precharge pressure of the accumulator became the most important parameters. They decided the working ability limit of the assistant subsystem. At the same time, the parameters of the accumulator, the secondary regulatory pump, the engine, and the other new components should satisfy different, sometimes even opposite, requirements. But the principles of parameters matching should be fixed as below:

(1) the working ability of the excavator should not decrease after the parameters matching,

(2) every component should work in; high-efficient area as much time as possible.

The parameter matching was an optimal course for the specific system. The load, the structure, and the control strategy of the PHHE system would bring influences to the optimal course.

*3.2.1. The Working Condition of the Excavator.* The load curve of a 20$t$ excavator was shown in Figure 2. Obviously, the load had a periodicity while the excavator was operated through mining, upgrading, turning, and discharging. On the contrary, it sharply changed between different motions.

The aim of the parameter matching was lower the fuel consumption and reduce the dimension in the case of guaranteeing the normal operation under the complex working condition. It could be described as (9)

$$\min : \text{fuel}\left(T_e, n, ge\right), V\left(V_{\text{acc}}, V_e, V_m\right), \tag{9}$$

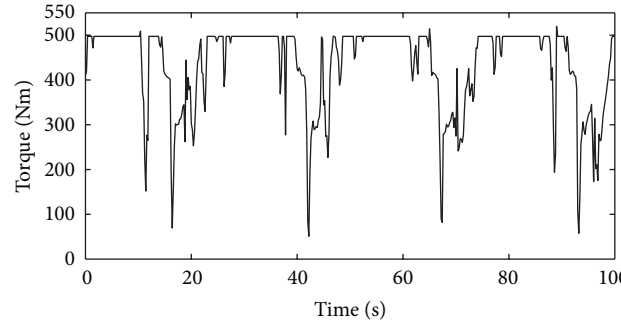



Figure 2: The load curve of the 20$t$ excavator.

where $n$ was the engine speed, $ge$ was the fuel consumption rate, $V_{\text{acc}}$ was the accumulator dimensions, $V_e$ was the engine dimensions, and $V_m$ was the motor dimensions.

*3.2.2. The Constraints of the Parameter Match.* In this paper, the parallel hybrid system must meet the dynamic characteristics in order to maintain the original excavator work characteristics. The power was the most important requirement which should be satisfied firstly as (10)

$$P_e + P_m = P_p \geq P_l, \tag{10}$$

where $P_e$ was the engine power, $P_m$ was the secondary regulatory pump power, $P_p$ was the main pump power, and $P_l$ was the load power. Since the engine, the main pump, and the secondary regulatory pump were connected coaxially, the torque constraint was got as (11)

$$T_e + T_m \geq T_p, \tag{11}$$

where $T_m$ was the torque of the secondary regulatory pump and $T_p$ was torque of the main pump. The accumulator played multiple roles in this system. When it was charged, its pressure was regarded as a resistant source; otherwise it was a power source. In particular when accumulator absorbed the boom gravitational potential energy, the parameters of the accumulator determined the speed and finish time of the motion of the boom directly. In order to maintain the motion, (12) and (8) must be considered when the boom gravitational potential energy is recovered by the accumulator:

$$x = \int_0^{t_i'} \dot{x} \cdot dt. \tag{12}$$

The time of the excavator motions should meet

$$\left|t_i' - t_i\right| \leq \varepsilon_i, \tag{13}$$

where $t_i$ was the original time of the boom motion and $\varepsilon_i$ was the allowed value of the time difference defined by the operator.

*3.2.3. The Key of the Parameter Match.* In PHHE system, the energy storage unit accumulator became the key component. Its parameters should be determined earlier than the assistant

power source. It was the most obvious difference of the parameter matching process between the PHHE system and the hybrid electric excavator system.

From (8), (5), it was known that when the accumulator worked as a resistance source absorbing the boom gravitational potential energy, the higher the accumulator internal pressure $P_a$ was, the shorter the braking time was. At the same time, $P_a$ also had an effect on the pressure of the chamber with the rod. The pressure should avoid reaching the value of the system overflowing which would bring extra energy loss. When the accumulator worked in the hybrid subsystem, the accumulator internal pressure $P_a$ became the input pressure of the secondary regulatory pump. The higher pressure might reduce the limit displacement of the secondary regulatory pump to create the same requirement torque value.

So it could be seen that whenever the accumulator worked in potential energy recovering system or hybrid subsystem, the high pressure was recommended.

From (2), (3), and (8) it was known that the energy storage capacity and precharge pressure have a parabolic relationship. Improper precharge pressure would weaken the energy storage capacity of the key component accumulator which determined the ability of the PHHE system.

## 4. The Realization of Parameter Matching

This paper practiced a parameter matching process for a 20-ton series excavator by considering the matching principle and constraints listed before.

*4.1. The Choice of the Primary Power Source.* Although the accumulator was the key component of the PHHE system, the engine was still the main power source which should be chosen firstly to satisfy the average power requirement. From the data of Figure 2, the average power requirement of a 20-ton excavator that operated under heavy working condition could be calculated as (14)

$$\overline{P}_l = \frac{1}{T}\int P(t)\,dt = 95.8\,\text{kw}, \tag{14}$$

where $P_l$ was the average requirement power and $T$ was the total time. Considering the safety factor, the engine rated power and rated speed were selected as 118 kw and 2200 r/m.

*4.2. The Choice of the Key Energy Storage Device.* The accumulator had different styles such as piston style, bladder style, spring style, and gravity style that suited variety of situations. In PHHE system, the bladder accumulator was selected because of its quick response, big specific volume, and easy precharging pressure set. Further, the inert nitrogen, the gas in the bladder, had more security features in the high temperature and vibration environment. The precharge pressure of the accumulator determined the lowest output pressure and the maximum energy storage. Meanwhile, the pressure affected the characteristics of the recycle system absorbing the potential energy. According to the area ratio of the cylinder, the maximum pressure of the accumulator should not exceed 25 MPa to avoid the pressure of the chamber with rod exceed the system overflow value and the minimum pressure just need satisfying the constraint equations.

*4.3. The Choice of the Energy Conversation Unit.* When the accumulator had been chosen, the assistant secondary regulatory pump was selected almost because the lowest pressure of the accumulator determined the lowest rate displacement of the secondary regulatory pump. From (4), (5), the secondary regulatory pump had to make the displacement big enough to ensure that the torque difference was compensated when the accumulator pressure gets lower. So, the displacement of secondary regulatory pump was calculated as 123.77 $\text{cm}^3$ under the extreme condition. By searching in the product list, we selected the most suitable product, whose displacement was 140 $\text{cm}^3$ and maximum rotation speed was 3250 r/m.

## 5. The Optimization with GA

From the last section, we had determined several important parameters which were constrained by the technical requirements. But to acquire the best balance between energy saving effect and economy, there were many parameters that should be chosen carefully such as the bladder accumulator's volume, precharging pressure, and engine's target speed.

Because the hybrid excavator's working cycles were nonlinear and impossible to describe by some specific formulas, the only things we could know were the parameters we set and the result we measured. By considering these adverse conditions, after comparing the popular optimized algorithms, we chose the genetic algorithm to optimize the accumulator's volume and the engine target speed since the method did not need a precise certain equation for a whole complex system [12–14].

The genetic algorithm objective function was showed as

$$F = f_1\alpha + f_2(1-\alpha), \tag{15}$$

$$f_1 = \frac{G_h}{G_0}, \tag{16}$$

$$f_2 = \left(\frac{P_h}{P_0}\right)^{\varphi}, \tag{17}$$

where $\alpha$ was the weight coefficient, $G_h$, $G_0$ were the fuel consumption ratios of the hybrid system and the normal system, and $P_h$, $P_0$ were the prices of the hybrid system and the normal system. We selected 120 as the number of the generation, 20 as the number of the population size, 0.75 as the number of the crossover rate, and 0.01 as the number of the mutation rate.

## 6. The Analysis of the Simulation Result

We built the simulation model in the Matlab software by considering the mathematical equations above and used the double working point control strategy to examine the effect of the parameter matching method proposed in this paper [15].

The strategy was showed in Figure 3. After the excavator starting initialization, the engine would be controlled at the

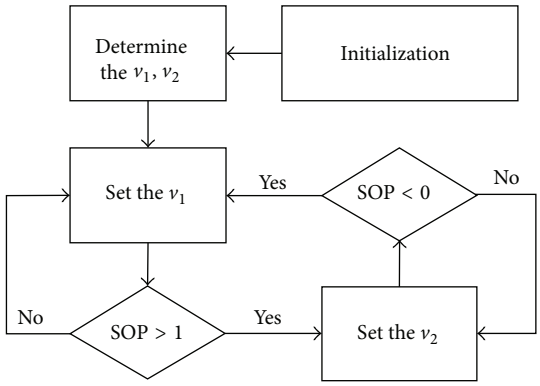



Figure 3: Double working points control strategy.

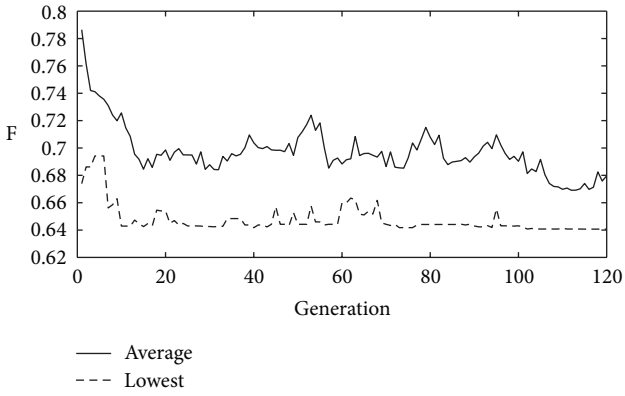



Figure 4: The Average and lowest values in every generation.

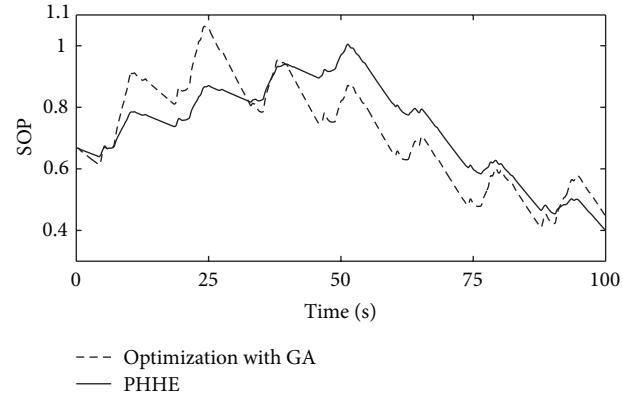



Figure 5: The SOP comparison between hybrid and optimized excavator.

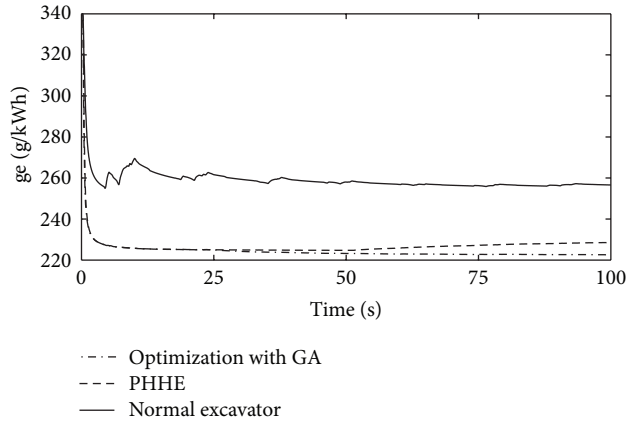



Figure 6: The *ge* comparison among the normal, hybrid, and optimized excavator.

two selected aim speed points $v_1$ and $v_2$ depending on the value of SOP, which indicated the energy storage state of the accumulator.

By optimizing with GA method, the function value mentioned in (9), (15) was showed in Figure 4. There were two lines in the figure, the average line which was the average function value of every generation and the lowest line which was the best function value in every generation. After 100th generation, the average and lowest values both approached the best values during the optimization course. In the 120th generation, the average vale was 0.6797 and the lowest value was 0.6407. We chose the parameters of the accumulator volume (50 L) and the engine second target speed (2035 rpm) when the 120th lowest function value occurred to analyze the parameter matching method.

Figures 5 and 6 presented the work state and the effect of the parameter matching method, respectively. From Figure 5, we could find that the accumulator plays a role of absorbing surplus energy and providing lacking energy during the work of the excavator. And it illustrated that the PHHE system was following the double working point control strategy showed in Figure 3. The Figure 6 compared the average *ge* value which indicated the fuel consumption directly among the normal excavator, PHHE system, and the PHHE system optimized with the GA method. The latter two had an obviously better *ge* than the normal excavator and the last one had a little better ge than the PHHE system without GA. The accuracy stabilized average *ge* values were 256.7, 228.6, and 222.6 (g/kwh). Compared to the limit optimal value 215 (g/kwh), the *ge* value of the PHHE system with GA was pretty good. The result illustrated the parameters matching course was useful and necessary.

Because the optimization course did not take the fuel consumption as the only object, the inconspicuous improvement in the energy saving effect was reasonable.

Besides, the motions of the mechanical equipment were tracked to check the influence brought by the pressure replacement in potential energy recovery under the heavy working condition as in Figure 7.

After the boom falling down process, the final location error was almost none and the time error was about

$$\text{error}_{\text{bt}} = \frac{11.33 - 11.02}{11.02 - 8.85} \times 100\% = 14.29\%, \tag{18}$$

where $\text{error}_{\text{bt}}$ was the final time error of the rod displacement between the normal and the hybrid systems. The time error was a little obvious but still acceptable. Additional only the motion was under the same control signal. The operator could

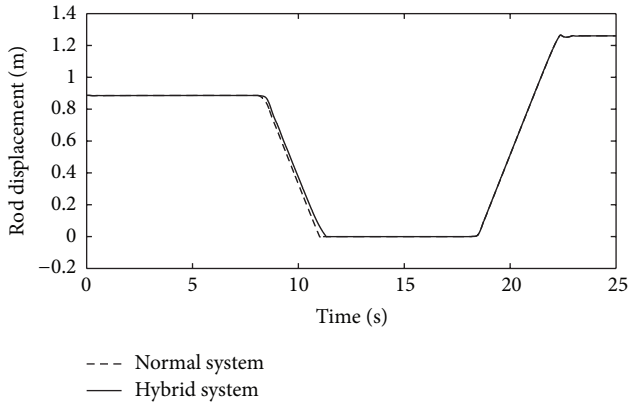



Figure 7: The rod displacement of the boom.

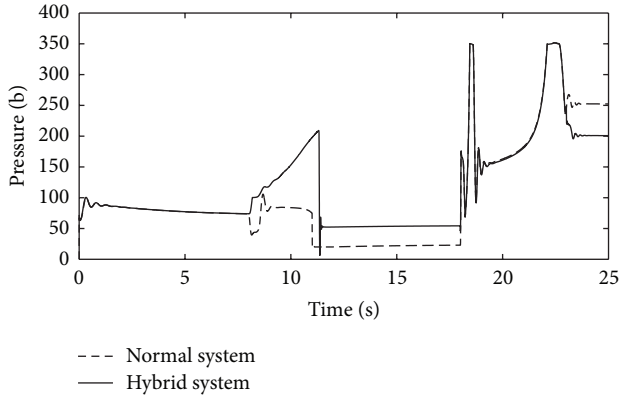



Figure 8: The pressure of the chamber without rod.

adjust the operation style to satisfy the action requirements easily in real work condition.

Figure 8, respectively, showed the pressure difference in hydraulic pipeline of the cylinder which proved that the energy recovery system only worked during the boom falling down.

## 7. Conclusion

The parameter matching method proposed in this paper took the storage unit as the key component. During the matching course, the energy saving effect, working ability, and the economy were all considered. The method was especially for the hydraulic hybrid excavator and useful. From the result of the simulation, we could conclude that: (1) the parameter matching method brought an obvious saving effect in fuel consumption (nearly 15%); (2) the motion of the excavator was less affected; (3) the GA optimization made the parameters more accurate for the parallel hydraulic hybrid system.

## Acknowledgment

This research is sponsored by the National High-Tech R&D Program of China (Grant no. 2010AA044401).

## References

[1] D. Wang, C. Guan, S. Pan, M. Zhang, and X. Lin, "Performance analysis of hydraulic excavator powertrain hybridization," *Automation in Construction*, vol. 18, no. 3, pp. 249–257, 2009.

[2] D. Jiuyu, Y. Shihua, and W. Chao, "The application and development of hydraulic hybrid powertrain of vehicle," *Machine Tool & Hydraulics*, vol. 37, no. 2, pp. 181–184, 2009.

[3] T. Lin, Q. Wang, B. Hu, and W. Gong, "Research on the energy regeneration systems for hybrid hydraulic excavators," *Automation in Construction*, vol. 19, no. 8, pp. 1016–1026, 2010.

[4] P. Matheson and J. Stecki, "Development and simulation of a hydraulic-hybrid powertrain for use in commercial heavy vehicles," SAE Technical Paper 2003-01-3370, New York, NY, USA, 2003.

[5] M. Kokkolaras, Z. Mourelatos, and L. Louca, "Design under uncertainty and assessment of performance reliability of a dual-use medium truck with hydraulic-hybrid powertrain and fuel cell auxiliary power unit," SAE Technical Paper 2005-01-1396, New York, NY, USA, 2005.

[6] D. Xin, Z. Chenning, and L. Xincheng, "Energy recovery system simulation and research of hybrid hydraulic excavator," *Journal of Beijing Technology and Business University*, vol. 28, no. 1, pp. 43–48, 2010.

[7] I. Y. Jong, K. K. Ahn, and Q. T. Dinh, "A study on an energy saving electro-hydraulic excavator," in *Proceedings of the ICROS-SICE International Joint Conference*, pp. 3825–3830, Fukuoka,Japan, August 2009.

[8] M. Zhang, Q. Wang, and C. Guan, "Simulation research of parallel hydraulic hybrid excavator," *China Mechanical Engineering*, vol. 21, no. 16, pp. 1932–1935, 2010.

[9] Q. Xiao, Q. Wang, and Y. Zhang, "Control strategies of power system in hybrid hydraulic excavator," *Automation in Construction*, vol. 17, no. 4, pp. 361–367, 2008.

[10] X. Lin, C. Guan, S. Pan, and D. Wang, "Parameters matching method for parallel hybrid hydraulic excavators," *Journal of Agricultural Machinery*, vol. 40, no. 6, pp. 28–32, 2009.

[11] H. Yuanjun, Y. Chengliang, and Z. Jianwu, "Parameter matching of parallel hybrid electric city-bus powertrain system," *Journal of Shanghai Jiaotong University*, vol. 41, no. 2, pp. 272–277, 2007.

[12] S. Hui, "Multi-objective optimization for hydraulic hybrid vehicle based on adaptive simulated annealing genetic algorithm," *Engineering Applications of Artificial Intelligence*, vol. 23, no. 1, pp. 27–33, 2010.

[13] A. Amirjanov, "The development of a changing range genetic algorithm," *Computer Methods in Applied Mechanics and Engineering*, vol. 195, no. 19-22, pp. 2495–2508, 2006.

[14] X. Wang, A. C. Yu, and W. Chen, "Optimal matching on driving system of hydraulic hybrid vehicle," in *Proceedings of the International Conference on Advanced in Control Engineering and Information Science (CEIS '11)*, C. Ran and G. Yang, Eds., vol. 15, pp. 5294–5298, Elsevier, 2011.

[15] X. Lin, S.-X. Pan, and D.-Y. Wang, "Dynamic simulation and optimal control strategy for a parallel hybrid hydraulic excavator," *Journal of Zhejiang University A*, vol. 9, no. 5, pp. 624–632, 2008.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 323750, 8 pages
http://dx.doi.org/10.1155/2013/323750

*Research Article*

# A Multidimensional and Multimembership Clustering Method for Social Networks and Its Application in Customer Relationship Management

**Peixin Zhao,[1] Cun-Quan Zhang,[2] Di Wan,[3] and Xin Zhang[4]**

[1] *School of Management, Shandong University, Jinan, Shandong 250100, China*
[2] *Department of Mathematics, West Virginia University, Morgantown, WV 26506, USA*
[3] *Department of Physics and Astronomy, University of Victoria, Victoria, BC, Canada V8W 2Y2*
[4] *Foundation Department, Shandong College of Electronic Technology, Jinan, Shandong 250200, China*

Correspondence should be addressed to Peixin Zhao; pxzhao@126.com

Received 15 July 2013; Accepted 7 August 2013

Academic Editor: Yoshinori Hayafuji



Community detection in social networks plays an important role in cluster analysis. Many traditional techniques for one-dimensional problems have been proven inadequate for high-dimensional or mixed type datasets due to the data sparseness and attribute redundancy. In this paper we propose a graph-based clustering method for multidimensional datasets. This novel method has two distinguished features: nonbinary hierarchical tree and the multi-membership clusters. The nonbinary hierarchical tree clearly highlights meaningful clusters, while the multimembership feature may provide more useful service strategies. Experimental results on the customer relationship management confirm the effectiveness of the new method.

## 1. Introduction

A social network is a set of people or groups each of which has connections of some kind to some or all of the others. Although the general concept of social networks seems simple, the underlying structure of a network implies a set of characteristics which are typical to all complex systems. Social network plays an extremely important role in many systems and processes and has been intensively studied over the past few years in order to understand both local phenomena, such as clique formation and their dynamics, and network-wide processes, for example, flow of data in computer networks [1], energy flow in food webs [2], customer relation management [3–6], and so forth. Modern information and communication technology has offered new interaction modes between individuals, like mobile phone communications and online interactions. Such new social exchanges can be accurately monitored for very large systems, including millions of individuals, representing a huge opportunity for the study of social science.

Clustering analysis is a data mining technique developed for the purpose of identifying groups of entities that are similar to each other with respect to certain similarity measures. Many different clustering methods have been proposed and used in a variety of fields. Jain [7] broadly divided these methods into two groups: hierarchical clustering and partitioned clustering. Hierarchical clustering is the grouping of objects of interest according to their similarity into a hierarchy, with different levels reflecting the degree of inter-object resemblance. The most well-known hierarchical methods are singlelink and completelink. In singlelink hierarchical methods, the two clusters whose two closest members have the smallest distance are merged in each step; in completelink cases, the two clusters whose merger has the smallest diameter are merged in each step. Compared to hierarchical clustering methods, partitioned clustering methods find all the clusters simultaneously as a partition of the data lK-means, which is widely used for the ease of implementation, simplicity, and efficiency where a certain data point cannot be simultaneously included in

more than one cluster [8]. Based on the difference of their capabilities, applicability, and computational requirements, clustering methods can be categorized into several different approaches: partitioning, hierarchical, density-based, grid-based, and model-based. No particular clustering method has been shown to be superior to all its competitors in all aspects [9].

In recent years, community detection based on clustering has become a growing research field partly as a result of the increasing availability of a huge number of networks in the real world. The most intuitive and common definition of community structure is that such network seems to have communities in them: subsets of vertices within which vertex-vertex connections are dense, but between which connections are relatively sparse. Yang and Luo [10] show that community structure has close relationship with some functionality such as robustness and fast diffusion. It is an important network property and is able to reveal many hidden features of the given network [11]. The detection and analysis of communities in social networks have played an important role in the mining of different kinds of networks, including the World Wide Web [12, 13], communication networks [14], and biological networks [15].

Most traditional community detection algorithms based on clustering are limited to handling one-dimensional datasets [16, 17]. However, the datasets to be mined in real life often contain millions of objects described by many various types of attributes or variables. For example, in customer relation management, a customer can be depicted by multidimensional data or mixed type data such as gender, age, income, education level, and so forth. In such cases, data mining operations and methods are required to be scalable as well as capable of dealing datasets' complex structures and dimensions. Previous researches were mainly focused on the representation of a set of items with a single attribute, which is apparently unsuitable for the scenarios described above: (i) a single attribute can not accurately represent all the dimensions of items; (ii) clustering according to a single attribute often fails to capture the inherent dependency among multiple attributes and leads to meaningless cluster.

Under such considerations, in this paper we firstly introduce two pretreatment methods for multi-dimensional and mixed type data, followed by a new clustering approach for community detection in social networks. In this approach, individuals and their relationships are denoted by weighted graphs, and the graph density we defined gives a better quantity depict of the overall correlation among individuals in a community, so that a reasonable clustering output can be presented. In particular, our method produces "trees" of simple hierarchy and allows for fuzzy (overlapping) clusters, which distinguishes it from other methods. In order to verify the utility/effectiveness of our method, we did a (preliminary) evaluation against a mobile customer segmentation use case. The numerical output of which shows supporting evidence for further (improvement) application.

The rest of the paper is organized as follows. In Section 2 we summarize the related works of community detections in social networks. In Section 3, we introduce the details of the novel clustering approach for multiattribute data sets. As an application in customer relationship management, this approach is used to analyze mobile customer segmentation problem in Section 4. Finally, a summary and conclusions are given in Section 5.

## 2. Related Works

The detection for communities has brought about significant advances to the understanding of many real-world complex networks. Plenty of detection algorithms and techniques have been proposed drawing on methods and principles from many different areas, including physics, artificial intelligence, graph theory, and even electrical circuits [11]. The spectral bisection methods [18] and the Kernighan-Lin [19] algorithm are early solutions to this problem in computer society. The spectral approach bisects graph iteratively, which is unsuitable to general networks. For the Kernighan-Lin algorithm, it requires a priori knowledge about the sizes of the initial divisions. In 2002, Girvan and Newman [20] proposed a divisive hierarchical clustering algorithm referred to as GN, which can generate optimizion of the division of a network by iteratively cutting the edge with the greatest betweenness value. However, a disadvantage of GN is that its time complexity is $O(m^2n)$ on a network of $n$ nodes and $m$ edges or $O(m^3)$ on a sparse network; then Newman [21] proposed a faster algorithm, referred to as NM, with time complexity $O(n^2)$ or $O((m+n)n)$ on a sparse network. A lot of works have been done to improve GN and NM; for example, Radicchi et al. [22] proposed a similar algorithm with GN by using the edge-clustering coefficient as a new metric with a smaller time complexity $O(m^2)$; Clauset et al. [23] have also proposed a fast clustering algorithm with $O(n \log^2 n)$ time complexity on sparse graph. Especially in 2007, Ou and Zhang [24] proposed a new clustering method with the feature of hierarchical tree and overlapping clusters, the complexity of this method is $O(hn^2 \log n)$ where $h$ denotes the height of the hierarchical structure. This method was, respectively, used to cluster extremist web pages [25] and some classic social networks [26] with single weighted edges.

Random walk has also been successfully used in finding network communities [27, 28]. The idea of this method is that the walk tends to be trapped in dense parts of a network corresponding to communities. Pons and Latapy [27] proposed a measure of similarity between vertices based on random walks which has several important advantages: it captures well the community structure in a network, it can be computed efficiently, and it can be used in an agglomerative algorithm to compute efficiently the community structure of a network. The algorithm called Walktrap runs in time $O(mn^2)$ and space $O(n^2)$ in the worst case and in time $O(n^2 \log n)$ and space $O(n^2)$ in most real-world cases; Hu et al. [29] proposed a method for the identification of community structure based on a signaling process of complex networks. Each node is taken as the initial signal source to excite the whole network one time, and the source node is associated with an $n$-dimensional vector which records the effects of the signaling process. By this process, the topological relationship of nodes on the network could be transferred into a geometrical

structure of vectors in $n$-dimensional Euclidean space. Then the best partition of groups is determined by F statistics, and the final community structure is given by the K-means clustering method.

Spectral clustering techniques have seen an explosive development and proliferation over the past few years [30–32]. Previous work indicated that a robust approach to community detection is the maximization of the benefit function known as "modularity" over possible divisions of a network, but Newman and Girvan [30] showed that the maximization process can be written in terms of the modularity matrix, which plays a role in community detection similar to that played by the graph Laplacian in graph partitioning calculations, and the time complexity of this algorithm is $O(n^2)$. They also proposed an objective function for graph clustering called the $Q$ function, which allows for automatic selection of the number of clusters, and then higher values of the $Q$ function were proven to correlate well with good graph clustering. White and Smyth [31] showed how the $Q$ function can be reformulated as a spectral relaxation problem and proposed two new spectral clustering algorithms that seek to maximize $Q$. Capocci et al. [32] developed some spectral-based algorithm to reveal the structure of a complex network, which could be blurred by the bias artificially overimposed by the iterative bisection constraint. Such a method should be able to conjugate the power of spectral analysis to the caution needed to reveal an underlying structure when there is no clear cut partitioning, as is often the case in real networks.

Lots of other community detection algorithms have also been proposed in the recent literatures. For example, Wu and Huberman [33] proposed a method which partitions a network into two communities, where the network is viewed as an electric circuit, and a battery is attached to two random nodes that are supposed to be within two communities. Shi et al. [11] proposed a new genetic algorithm for community detection, using the fundamental measure criterion modularity $Q$ as the fitness function. A special locus-based adjacency encoding scheme is applied to represent the community partition; Shi et al. [34] proposed a novel method based on particle swarm optimization to detect community structures by optimizing network modularity.

## 3. Multidimensional and Multimembership Clustering Method for Social Networks

*3.1. Similarity of Multidimensional Data.* Traditional distance functions include Euclidean distance, Chebyshev distance, Manhattan distance, Mahalanobis distance, Weighted Minkowski distance, and Cosine distance. Among these distance functions, Mahalanobis distance is based on correlations between variables by which different patterns can be identified and analyzed. It gauges similarity of an unknown sample set to a known one. It differs from Euclidean distance in that it takes into account the correlations of the data set and is scale-invariant. In other words, it is a multivariate effect size.

All these distance functions have their own advantages and disadvantages in practical applications. Some research results shows that Euclidean distance has better performance in vector models, while some other numerical examples in high dimensional spaces show that the farthest and nearest distance are almost equal, although Euclidean distance is used to measure the similarity between data points. That is in high-dimensional data, traditional similarity measures as used in conventional clustering algorithms are usually not meaningful. This problem and related phenomena require adaptations of clustering approaches to the nature of high-dimensional data. This area of research has been a highly active one in recent years. Common approaches are known as, for example, subspace clustering, projected clustering, pattern-based clustering, or correlation clustering. Subspace clustering is the task of detecting all clusters in all subspaces, which means that a point might be a member of multiple clusters, each existing in a different subspace. Subspaces can either be axis parallel or affine. Projected clustering seeks to assign each point to a unique cluster, but clusters may exist in different subspaces. The general approach is to use a special distance function together with a regular clustering algorithm. Correlation clustering provides a method for clustering a set of objects into the optimum number of clusters without specifying that number in advance.

In 2011, A new function "Close()" is presented based on the improvement of traditional algorithm to compensate their inadequacy for high-dimensional space [35]. Let

$$X = (x_1, x_2, \ldots, x_n), \quad Y = (y_1, y_2, \ldots, y_n) \tag{1}$$

denote two points in $n$-dimensional space. The function "Close()" is defined as

$$\text{Close}(X, Y) = \frac{\sum_{i=1}^{n} e^{-|x_i - y_i|}}{n}. \tag{2}$$

It depicts the similarity degree between two data points and has the following properties.

(a) The minimum value of the function is 0, which means that the similarity degree between $X$ and $Y$ is smallest since the difference comes closest to infinity in each dimension.

(b) The maximum value of the function is 1, which means that the similarity degree between $X$ and $Y$ is largest since they come closest to coinciding in each dimension.

Similar to the weighted operator in traditional distance functions, the close function can be corrected as

$$\text{Close}(X, Y) = \frac{\sum_{i=1}^{n} \omega_i e^{-|x_i - y_i|}}{n}, \tag{3}$$

where $\omega_i \in [0, 1]$ denotes the importance degree of data in the $i$th dimension. Advantages of the new function are obvious in high-dimensional similarity measurement according to the comparison in [35]. Quantitative analysis also proved that this function can avoid the effects of noise and the curse of high-dimension.

*3.2. Similarity of Mixed Type Data.* For clustering multiattributes datasets, we first introduce a method for the measurement of similarity between items as follows [36]. The multiattribute datasets can be separated into two parts: the pure numeric datasets and pure categorical datasets. Some existing efficiency clustering methods designed for these two types of data sets are employed to produce corresponding clusters. For the similarity matrix, we define $S(i, j)$ as the number of times the given sample pair $x_i$ and $x_j$ has co-occurred in a cluster [37].

Consider

$$S(i,j) = S\left(x_i, x_j\right) = \frac{1}{H}\sum_{k=1}^{H}\delta\left(\pi_k\left(x_i\right), \pi_k\left(x_j\right)\right), \qquad \delta(a,b) \equiv \begin{cases} 1, & a = b, \\ 0, & a \neq b, \end{cases} \tag{4}$$

where $H$ denotes the number of the clustering. $\pi_k(x_i)$ and $\pi_k(x_j)$ denote the cluster label of items $x_i$ and $x_j$, respectively. Then for the pure numerical datasets, the similarity can be defined as

$$S_1(i,j) = \frac{n}{N} = \frac{\sum_{i=1}^{N} C(i,j)}{N}, \tag{5}$$

where $N$ is the number of clustering and $n$ is the number of times the pattern pair $(x_i, x_j)$ is assigned to the same cluster among the $N$ clustering. If $(x_i, x_j)$ is assigned to the same cluster, $C(i, j) = 1$, otherwise $C(i, j) = 0$.

For the pure categorical datasets, the similarity can be defined as

$$S_2(i,j) = \frac{n}{m} = \frac{\sum_{i=1}^{m} C(i,j)}{m}, \tag{6}$$

where $m$ denotes the number of attributes. Then the similarity of multiattribute datasets can be denoted by

$$S = S_1 + \alpha S_2, \tag{7}$$

where $\alpha$ is a user-defined parameter. If $\alpha > 1$, the categorical datasets is more important than the numerical datasets; if $\alpha < 1$, numerical datasets is more important. $S_1(i, j)$ and $S_2(i, j)$ can also be used as two-dimensional (or multidimensional) datasets to represent the similarities between items $x_i$ and $x_j$.

*3.3. Multidimensional and Multimembership Clustering Method for Social Networks.* A graph or network is one of the most commonly used models to represent real-valued relationships of a set of input items. Since many traditional techniques for one-dimensional problems have been proven inadequate for high-dimensional or mixed type datasets due to the data sparseness and attribute redundancy, the graph-based clustering method for single dimensional datasets proposed in [24–26] can be extended as follows to directly cluster multidimensional datasets.

Let $G = (V, E)$ be a graph with the vertex set $V$ and associated with $r$ weights:

$$\omega_k : E(G) \longmapsto [0,1], \quad k = 1, 2, \ldots, r. \tag{8}$$

For a subgraph $C(|V(C)| > 1)$ of $G$, we define the $k$th density of $C$ by

$$d_k(C) = \frac{2\sum_{e\in E(C)}\omega_k(e)}{|V(C)|\left(|V(C)| - 1\right)}. \tag{9}$$

In single weighted graph $C$, if $\omega(e) = 1$ and $d(C) = 1$ for every edge $e$ in $C$, the subgraph $C$ induces a clique. For a multiweighted graph $(G; \omega_1, \omega_2, \ldots, \omega_r)$, a subgraph $C$ is called a $\Delta$-quasiclique if $d_k(C) \geq \Delta$ for some positive real number $\Delta$ and for every $k \in \{1, 2, \ldots, r\}$ ($r$ is the number of weights on the edge).

Clustering is a process that detects all dense subgraphs in $G$ and constructs a hierarchically nested system to illustrate their inclusion relation.

A heuristic process is applied here for finding all quasicliques with density of various levels. The core of the algorithm is deciding whether or not to add a vertex to an already selected dense subgraph $C$. For a vertex $v \notin V(C)$, we define the contribution of $v$ to $C$ by

$$c_k(v, C) = \frac{\sum_{u\in V(C)}\omega_k(uv)}{|V(C)|}. \tag{10}$$

A vertex $v$ is added into $C$ if $c_k(v, C) > \alpha d(C)$ where $\alpha$ is a user specified parameter.

In short, the main steps of our algorithm can be described as shown in Algorithm 1.

Trace the process of each vertex, and obtain the hierarchic tree.

Our detailed community detection algorithm that can find $\Delta$-quasicliques in $G$ with various levels of $\Delta$ is as follows. A hierarchically nested system is constructed to illustrate their inclusion relation.

*Step 0.* $l \leftarrow 1$ where $l$ is the indicator of the levels in the hierarchical system:

$$M_0 \longleftarrow \gamma \max\left\{\omega_k(e) : \forall e \in E(G), \forall k\right\}, \tag{11}$$

where $\gamma$ $(0 < \gamma < 1)$ is a user specified parameter ($\gamma$ is a cut-off threshold).

*Step 1* (the initial step). Let $F$ be the set of all edges $e$ of $G$ with

$$\min\left\{\omega_k(e) : k = 1, 2, \ldots, r\right\} \geq M_0. \tag{12}$$

Let $m = |F|$. Sort the edges of the set $F$ as a sequence $S = e_1, \ldots, e_m$ such that

$$\sum_{k=1}^{r}\omega_k(e_1) \geq \sum_{k=1}^{r}\omega_k(e_2) \geq \cdots \geq \sum_{k=1}^{r}\omega_k(e_m), \tag{13}$$

$\mu \leftarrow 1$, $p \leftarrow 0$, and $L_l \leftarrow \emptyset$ where $L_l$ is the community sets in the $l$th hierarchical level.

*Step 2* (One has starting a new search).

$$p \longleftarrow p + 1, \qquad C_p \longleftarrow V\left(e_\mu\right). \qquad L_l \longleftarrow L_l \cup \left\{C_p\right\}. \tag{14}$$

**Input**: A graph $G = (V; \omega_1, \omega_2, \ldots, \omega_r)$ is a multi-weighted graph with $\omega_k$: $E(G) \mapsto [0, 1]$.
**Output**: Meaningful community sets in $G$.

**Algorithm**: Detect $\Delta$-quasi-cliques in $G$ with various levels of $\Delta$, and construct a hierarchically nested system to illustrate their inclusion relation.

While $E(G) \neq \emptyset$
begin
determine the value of $M_0$
*Decompose*$(G, M_0)$
$E_0 = \{e \in E(G)$: $\omega_k(e) \geq M_0, k = 1, 2, \ldots, r\}$
for each edge in $E_0$ in decreasing order of weights, if the two vertexes of edge are not in any community, create a new empty community $C$ Choose $v$ in the rest vertex sets that have maximum contribution to $C$ and add $v$ in it.
*Merging* $(G)$
Merge two communities according to their common vertexes;
Contract each community to a vertex and redefine the weight of the corresponding edges.
Store the resulted graph to $G$.
End.

Algorithm 1

*Step 3* (*growing*)

*Substep 3.1.* $U \leftarrow V(G) - V(C_p)$; if $U = \emptyset$, go to Step 4; otherwise continue.

$(*)$ Pick $v \in U$ such that $\Pi_{k=1}^{r} c_k(v, C_p)$ is a maximum.

If, for every $k$,

$$c_k\left(v, C_p\right) \geq \alpha_n d_k\left(C_p\right), \tag{15}$$

where $n = |V(C_p)|$ and $\alpha_n = 1 - (1/2)\lambda(n + t)$ with $\lambda \geq 1$, $t \geq 1$ as user specified parameters, then $C_p \leftarrow C_p \cup \{v\}$, and go back to Substep 3.1.

If Inequality (15) is not satisfied, then

$$U \longleftarrow U - \{v\}. \tag{16}$$

If $U \neq \emptyset$, repeat $(*)$. If $U = \emptyset$, go to Substep 3.2.

*Substep 3.2.* $\mu \leftarrow \mu + 1$. If $\mu > m$ go to Step 4; otherwise continue.

*Substep 3.3.* Suppose $e_\mu = xy$. If at least one of $x, y \notin \cup_{i=1}^{p-1} V(C_i)$, then go to Step 2; otherwise go to Substep 3.2.

*Step 4* (*merging*).

*Substep 4.1.* List all members of $L_l$ as a sequence $C_1, \ldots, C_s$ such that

$$\left|V\left(C_1\right)\right| \geq \left|V\left(C_2\right)\right| \geq \cdots \geq \left|V\left(C_s\right)\right|, \tag{17}$$

where $s = |L_l|$. $h \leftarrow 2$, $j \leftarrow 1$.

*Substep 4.2.* If

$$\left|C_j \cap C_h\right| > \beta \min\left(\left|C_j\right|, \left|C_h\right|\right), \tag{18}$$

(where $0 < \beta < 1$ is a user specified parameter), then $C_{s+1} \leftarrow C_j \cup C_h$, and the sequence $L_l$ is rearranged as follows:
$C_1, \ldots, C_{s-1} \leftarrow$ deleting $C_j$, $C_h$ from $C_1, \ldots, C_{s+1}$.
$s \leftarrow s - 1$, $h \leftarrow \max\{h - 2, 1\}$, and go to Substep 4.4.

*Substep 4.3.* $j \leftarrow j + 1$. If $j < h$, go to Substep 4.2.

*Substep 4.4.* $h \leftarrow h + 1$, $j \leftarrow 1$. If $h \leq s$, go to Substep 4.2.

*Step 5.* Contract each $C_p \in L_l$ as a vertex:

$$V(G) \longleftarrow \left[V(G) - \bigcup_{p=1}^{s} V\left(C_p\right)\right] \cup \{C_1, \ldots, C_s\},$$

$$\omega_k(uv) \longleftarrow \omega_k\left(C_i, C_j\right) = \frac{\sum_{e \in E_{i,j}} \omega_k(e)}{E_{i,j}}, \quad k = 1, 2, \ldots, r. \tag{19}$$

The vertex $u$ is obtained by contracting $C_i$ and $v$ is obtained by contracting $C_j$ where $E_{ij}$ is the set of crossing edges which is defined as

$$E_{i,j} = \left\{xy : x \in C_i, y \in C_j, x = y\right\}. \tag{20}$$

For

$$q \in V(G) - \{C_1, \ldots, C_s\}, \tag{21}$$

define $\omega_k(q, C_i) = \omega_k(\{q\}, C_i)$. Other cases are defined similarly.

If $|V(G)| \geq 2$, then go to Step 6; otherwise go to End.

*Step 6.* One has

$$\begin{gathered} l \longleftarrow l + 1, \qquad L_l \longleftarrow \emptyset, \\ \omega_0 \longleftarrow \gamma \max\{\omega(e) : \forall e \in E(G), \forall k\}, \end{gathered} \tag{22}$$

where $\gamma$ $(0 < \gamma < 1)$ is a user specified parameter, and go to Step 1 (to start a new search in a higher level of the hierarchical system).

*End.*

Trace the movement of each vertex and generate the hierarchic tree.

Table 1: Some information of 3000 mobile customers.

| Customer number | Local call fee (Yuan) | Long distance call fee (Yuan) | Roaming fee (Yuan) | Text message and WAP fee (Yuan) | Package type |
|---|---|---|---|---|---|
| 1 | 55.3 | 13.7 | 120.6 | 14.2 | D |
| 2 | 132 | 44.8 | 36.2 | 5.6 | B |
| 3 | 47.1 | 233.6 | 79.4 | 6.2 | B |
| 4 | 173 | 19.3 | 87.5 | 19.3 | C |
| 5 | 23.7 | 80.5 | 21 | 9 | A |
| 6 | 62.3 | 62.9 | 77.8 | 10.6 | E |
| 7 | 242.5 | 21.8 | 23.5 | 24.2 | A |
| 8 | 166.2 | 34.5 | 8 | 19.5 | C |
| ... | ... | ... | ... | ... | ... |
| 3000 | 77.6 | 67 | 21.2 | 24.7 | D |

If the input data is an unweighted graph $G$, the adjacency information is used for establishing the similarity matrix of $G$. Let $A = (a_{ij})$ be the adjacency matrix of $G$ where

$$a_{ij} = \begin{cases} 1, & \text{there is an edge between } i \text{ and } j \\ 0, & \text{otherwise} \end{cases} \quad (23)$$

and the inner product of the $i$th and the $j$th row of $A$ is used to describe the similarity between nodes $i$ and $j$ and stored as $G(i, j)$ in the similarity matrix $G$.

## 4. Simulation Examples

In order to validate the feasibility of the proposed novel approach to cluster multi-dimensional data sets, we randomly took 3000 customers' consumption lists of August 2012 from Shandong Mobile Corporation and use our new approach to divide these customers into distinguishing clusters according to 4 evaluation indices: local call fee, long distance call fee, roaming fee and text message and WAP fee. The original data of 3000 customers are listed in Table 1.

We have applied our approach to this problem, and the results of segmentation and their average consumption are listed in Table 2 and Figure 1.

As we can see from the clustering result, the long distance fee of group 1 has a high proportion of their total expenses; Groups 3 and 4 have high roaming fees; Group 8 has lower cost in each index; Groups 2, 3, and 4 have higher text message and WAP fees. Mobile corporations can initiate corresponding policies according to the clustering results. For example, for the customers in Groups 2, 3, and 4, mobile corporation should provide them with some discount text message package; for the customers in Groups 3, 4, and 6, some discount package of roaming will also help to increase customer loyalty and stability.

On the other hand, we noticed that the sum of the last column of Table 2 is larger than 3000. This is because our method allows multimembership clustering; thus some customers can belong to more than one group. For instance, Groups 8 and 1 are low value customer and high value customer respectively, and some special policies should be recommended for the 39 customers, who belong to either Group 1 or 8 to help them become loyal higher value customers.

Table 2: The customer segmentation of mobile network.

| Cluster number | Average local call fee (Yuan) | Average long distance call fee (Yuan) | Average roaming fee (Yuan) | Average text message and WAP fee (Yuan) | Number of customer |
|---|---|---|---|---|---|
| 1 | 156.9 | 172.8 | 39.8 | 58.5 | 121 |
| 2 | 299.1 | 43.2 | 38.7 | 46.9 | 64 |
| 3 | 42.6 | 32.9 | 174.7 | 36.2 | 168 |
| 4 | 212.8 | 103.3 | 574.3 | 39.7 | 13 |
| 5 | 187.9 | 871.5 | 35.3 | 28.7 | 9 |
| 6 | 162.1 | 262.3 | 354.8 | 21.2 | 12 |
| 7 | 43.0 | 25.8 | 13.7 | 21.2 | 2077 |
| 8 | 19.2 | 7.5 | 4.8 | 13.5 | 792 |

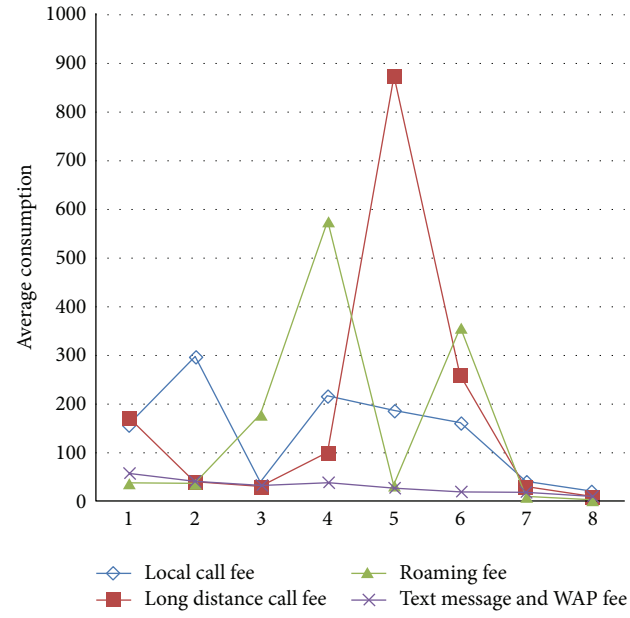



Figure 1: Average consumption list of 8 Groups.

## 5. Conclusions

In this paper, a graph-based new clustering method for multi-dimensional datasets is proposed. Due to the inherent sparsity of data points, most existing clustering algorithms do not work efficiently for multi-dimensional datasets, and it is not feasible to find interesting clusters in the original full space of all dimensions. These researches were mainly focused on the representation of a set of items with a single attribute, which cannot accurately represent all the attributes and capture the inherent dependency among multiple attributes. The new clustering method we proposed in this paper overcomes

this problem by directly clustering items according to the multidimensional information. Since it does not need data preprocessing, this new method may significantly improves clustering efficiency. It also has two-distinguished features: nonbinary hierarchical tree and multimembership clusters. The application in customer relationship management has proved the efficiency and feasibility of the new clustering method.

## Conflict of Interests

Peixin Zhao, Cun-Quan Zhang, Di Wan, and Xin Zhang certify that there is no actual or potential conflict of interests in relation to this paper.

## Acknowledgments

The first author is partially supported by the China Postdoctoral Science Foundation funded Project (2011M501149), the Humanity and Social Science Foundation of Ministry of Education of China (12YJCZH303), the Special Fund Project for Postdoctoral Innovation of Shandong Province (201103061), the Informationization Research Project of Shandong Province (2013EI153), and Independent Innovation Foundation of Shandong University, IIFSDU (IFW12109). The second is author partially supported by an NSA Grant H98230-12-1-0233 and an NSF Grant DMS-1264800.

## References

[1] X. Jin, C. M. K. Cheung, M. K. O. Lee, and H. Chen, "How to keep members using the information in a computer-supported social network," *Computers in Human Behavior*, vol. 25, no. 5, pp. 1172–1181, 2009.

[2] A. Bodini, "The qualitative analysis of community food webs: implications for wildlife management and conservation," *Journal of Environmental Management*, vol. 41, no. 1, pp. 49–65, 1994.

[3] P. C. Verhoef and K. N. Lemon, "Successful customer value management: key lessons and emerging trends," *European Management Journal*, vol. 31, no. 1, pp. 1–15, 2013.

[4] C. Kiss and M. Bichler, "Identification of influencers—measuring influence in customer networks," *Decision Support Systems*, vol. 46, no. 1, pp. 233–253, 2008.

[5] E. S. Bernardes and G. A. Zsidisin, "An examination of strategic supply management benefits and performance implications," *Journal of Purchasing and Supply Management*, vol. 14, no. 4, pp. 209–219, 2008.

[6] D. Li, W. Dai, and W. Tseng, "A two-stage clustering method to analyze customer characteristics to build discriminative customer management: a case of textile manufacturing business," *Expert Systems with Applications*, vol. 38, no. 6, pp. 7186–7191, 2011.

[7] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651–666, 2010.

[8] S. Guha, A. Meyerson, N. Mishra, R. Motwani, and L. O'Callaghan, "Clustering data streams: theory and practice," *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, no. 3, pp. 515–528, 2003.

[9] F. Cao, "A weighting $K$-modes algorithm for subspace clustering of categorical data," *Neurocomputing*, vol. 108, pp. 23–30, 2012.

[10] S. Yang and S. Luo, "A local quantitative measure for community detection in networks," *International Journal of Intelligent Engineering Informatics*, vol. 1, no. 1, pp. 38–52, 2010.

[11] C. Shi, Y. Wang, B. Wu, and C. Zhong, "A new genetic algorithm for community detection," in *Complex Sciences, Part II*, vol. 5 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pp. 1298–1309, 2009.

[12] M. Hoerdt and U. Louis, "Completeness of the internet core topology collected by a fast mapping software," in *Proceedings of the 11th International Conference on Software, Telecommunications and Computer Networks*, pp. 257–261, 2003.

[13] A. Broder, P. Kumar, F. Maghoul et al., "Graph structure in the web," in *Proceedings of the 9th International Conference on the World Wide Web*, pp. 15–19, 2003.

[14] J. Scott, *Social Network Analysis: A Handbook*, Sage, 2000.

[15] D. A. Fell and A. Wagner, "The small world of metabolism," *Nature Biotechnology*, vol. 18, no. 11, pp. 1121–1122, 2000.

[16] T. Chen, N. L. Zhang, T. Liu, K. M. Poon, and Y. Wang, "Model-based multidimensional clustering of categorical data," *Artificial Intelligence*, vol. 176, pp. 2246–2269, 2012.

[17] L. Poon, N. L. Zhang, T. Liu, and A. H. Liu, "Model-based clustering of high-dimensional data: variable selection versus facet determination," *International Journal of Approximate Reasoning*, vol. 54, no. 1, pp. 196–215, 2012.

[18] A. Pothen, H. D. Simon, and K.-P. Liou, "Partitioning sparse matrices with eigenvectors of graphs," *SIAM Journal on Matrix Analysis and Applications*, vol. 11, no. 3, pp. 430–452, 1990, Sparse matrices (Gleneden Beach, OR, 1989).

[19] B. W. Kernighan and S. Lin, "A efficient heuristic procedure for partitioning graphs," *Bell System Technical Journal*, vol. 49, pp. 291–307, 1970.

[20] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99, no. 12, pp. 7821–7826, 2002.

[21] M. E. J. Newman, "Fast algorithm for detecting community structure in networks," *Physical Review E*, vol. 69, no. 6, Article ID 066133, pp. 1–66133, 2004.

[22] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Paris, "Defining and identifying communities in networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, no. 9, pp. 2658–2663, 2004.

[23] A. Clauset, M. E. J. Newman, and C. Moore, "Finding community structure in very large networks," *Physical Review E*, vol. 70, no. 6, Article ID 066111, 6 pages, 2004.

[24] Y. Ou and C.-Q. Zhang, "A new multimembership clustering method," *Journal of Industrial and Management Optimization*, vol. 3, no. 4, pp. 619–624, 2007.

[25] X. Qi, K. Christensen, R. Duval et al., "A hierarchical algorithm for clustering extremist web pages," in *Proceedings of the International Conference on Advances in Social Network Analysis and Mining (ASONAM '10)*, pp. 458–463, August 2010.

[26] P. Zhao and C. Zhang, "A new clustering method and its application in social networks," *Pattern Recognition Letters*, vol. 32, no. 15, pp. 2109–2118, 2011.

[27] P. Pons and M. Latapy, "Computing communities in large networks using random walks," *Journal of Graph Algorithms and Applications*, vol. 10, no. 2, pp. 191–218, 2006.

[28] S. V. Dongen, *Graph clustering by flow simulation [Ph.D. dissertation]*, University of Utrecht, 2000.

[29] Y. Hu, M. Li, P. Zhang, Y. Fan, and Z. Di, "Community detection by signaling on complex networks," *Physical Review E*, vol. 78, no. 1, Article ID 016115, 2008.

[30] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical Review E*, vol. 69, no. 2, Article ID 026113, 2004.

[31] S. White and P. Smyth, "A spectral clustering approach to finding communities in graphs," in *Proceedings of SIAM International Conference on Data Mining*, pp. 76–84, 2005.

[32] A. Capocci, V. D. P. Servedio, G. Caldarelli, and F. Colaiori, "Detecting communities in large networks," *Physica A*, vol. 352, no. 2-4, pp. 669–676, 2005.

[33] F. Wu and B. A. Huberman, "Finding communities in linear time: a physics approach," *European Physical Journal B*, vol. 38, no. 2, pp. 331–338, 2004.

[34] Z. Shi, Y. Liu, and J. Liang, "PSO-based community detection in complex networks," in *Proceedings of the 2nd International Symposium on Knowledge Acquisition and Modeling (KAM '09)*, pp. 114–119, December 2009.

[35] C. Shao, W. Lou, and L. Yan, "Optimization of algorithm of similarity measurement in high dimensional data," *Computer Technology and Development*, vol. 20, no. 2, pp. 1–4, 2011.

[36] H. Luo and H. Wei, "Clustering algorithm for mixed data based on clustering ensemble technique," *Computer Science*, vol. 37, no. 11, pp. 234–238, 2010.

[37] A. Fred, "Finding consistent clusters in data partitions," in *Multiple Classifier Systems*, vol. 2096 of *Lecture Notes in Computer Science*, pp. 309–318, 2001.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 878542, 10 pages
http://dx.doi.org/10.1155/2013/878542

*Research Article*

# A Decentralized Virtual Machine Migration Approach of Data Centers for Cloud Computing

**Xiaoying Wang, Xiaojing Liu, Lihua Fan, and Xuhan Jia**

*Department of Computer Technology and Applications, Qinghai University, Xining, Qinghai Province 810016, China*

Correspondence should be addressed to Xiaoying Wang; xiaoyingwang.paper@gmail.com

Received 3 June 2013; Accepted 19 July 2013

Academic Editor: Yuxin Mao



As cloud computing offers services to lots of users worldwide, pervasive applications from customers are hosted by large-scale data centers. Upon such platforms, virtualization technology is employed to multiplex the underlying physical resources. Since the incoming loads of different application vary significantly, it is important and critical to manage the placement and resource allocation schemes of the virtual machines (VMs) in order to guarantee the quality of services. In this paper, we propose a decentralized virtual machine migration approach inside the data centers for cloud computing environments. The system models and power models are defined and described first. Then, we present the key steps of the decentralized mechanism, including the establishment of load vectors, load information collection, VM selection, and destination determination. A two-threshold decentralized migration algorithm is implemented to further save the energy consumption as well as keeping the quality of services. By examining the effect of our approach by performance evaluation experiments, the thresholds and other factors are analyzed and discussed. The results illustrate that the proposed approach can efficiently balance the loads across different physical nodes and also can lead to less power consumption of the entire system holistically.

## 1. Introduction

Recently, as a newly emerged technology, cloud computing [1] becomes a new paradigm for dynamic provisioning of various services. It provides a way to deliver the infrastructure, platform, and software as services available to consumers in a pay-as-you-go manner [2]. Such typical commercial service providers include *Amazon*, *Google*, and *Microsoft*. In cloud computing environments, large-scale data centers [3] are usually the essential computing infrastructure, which are comprised of plenty of physical nodes with multiple virtual machines running upon them.

The virtualization technology enables a novel model such that customized virtual environments could be created upon the physical infrastructure [4]. The use of virtualization techniques provides great flexibility with the capability to consolidate multiple virtual machines on a single physical node [5]. In this way, the resource capacity allocated to different virtual machines could be resized, and virtual machines could also be migrated [6] across different physical nodes on demand to achieve various purposes. Recently, most modern virtualization technologies products have realized the notion of live or seamless migration of virtual machines that involve extremely short downtimes ranging from tens of milliseconds to a second [7]. Thus, the migration of virtual machines has emerged as a promising technique to be utilized by resource management algorithms to rapidly solve the problems in virtualized data centers.

To fully utilize the underlying cloud resources, the provider has to ensure that the services can be flexibly delivered to meet various consumer requirements which are usually specified by (service level agreements) SLAs [8], while keeping the consumers isolated from the underlying physical infrastructure. However, since the workloads of the different applications or services fluctuate a lot as time elapses, it is a challenge to adaptively manage the resource allocation and make appropriate decisions. Thus, the potential benefits of migrating virtual machines provide the opportunity to address the issues of high performance concern of the service provider.

On the other hand, as green computing [9] gains a lot of attention due to significant energy consumption of large-scale

data centers, the focus of the researcher is gradually shifted from optimizing the pure performance to optimizing the energy efficiency while maintaining high quality of services. Consequently, the energy costs become an important part of the (total cost of ownership) TCO, which needs to be suppressed from the point of the providers' views. Under such consideration, the proper migration of virtual machines could also lead to more consolidation and thus release some underutilized nodes, in order to further save energy costs.

A number of approaches addressing the issues of resource management for cloud computing have been proposed [10–17]. Many of them are based on centralized architectures, which are known to be not very scalable and might suffer from fault-tolerant issues. For example, the crash of the centralized resource arbiter will disable all of the later possible adaptive resource management actions, leading the whole system into a static state. Under the demand of autonomy, a truly decentralized solution is preferable, which brings improved scalability and naturally fault tolerant.

Given this analysis above, the main objective of our work is to design a decentralized virtual migration approach for data centers in cloud computing environment. The aim of the approach is to dynamically adjust the resource allocation amount by migration and reduce possible energy wastes at the same time. The main contributions of this paper include the following: (1) the definition of system models and power models for the cloud computing infrastructure discussed in this paper; (2) the design and development of the autonomic and decentralized mechanisms for dynamic virtual machine management to satisfy service quality requirements and reduce energy consumption as much as possible; (3) comprehensive performance evaluation results which illustrate the effect and efficiency of the proposed approach, from the aspects of SLA violation, power consumption, load-balancing effects, and so on.

The rest of this paper is then organized as follows: in Section 2 some related work in this area are presented and discussed; the system models of the target data center we studied is described in Section 3; in Section 4 we propose the decentralized virtual machine management approach; performance evaluation results are illustrated in Section 5; finally, in Section 6, we conclude the paper with some final remarks and future directions of this work.

## 2. Related Work

*2.1. Virtualization-Based Resource Management for Cloud.* As the employment of virtualization facilitates the fine-grained resource allocation in cloud environment, a number of researchers have made efforts on the study of virtualization-based resource management for cloud. Iqbal et al. [18] implemented a prototype that actively monitors the response time of each VM and adaptively scales up the application to satisfy the SLA promise. Maniymaran and Maheswaran [19] present a centralized heuristic algorithm to solve the VM creation and location problem, using a local search technique. Campegiani [20] proposed a genetic algorithm to find the optimal allocation of virtual machines in a multitier distributed environment. Almeida et al. [21] modeled each VM in the system as an M/G/1 open queue and applied Markov's Inequality to estimate the SLA violation possibility. Also, in our recent work [22], we have exploited model-free methodologies to adaptively manage the resources and energy consumption in virtualized environments.

However, from an architectural point of view, the resource manager in the above research usually lies on a central node as a single module, which is vulnerable to potential failures. Hence, we turn to exploit decentralized management approaches which could be a possible solution to address the availability issues of the central control unit.

*2.2. Virtual Machine Migration.* Since most major virtualization platforms support live migration within a local area network (LAN), some work has been done to study the migration mechanism and strategies of virtual machine migration inside the data center. Abdul-Rahman et al. [7] have surveyed relevant work in the area of migration-based resource manager for virtualized environments and also discussed several types of management algorithms. Liu et al. [23] have investigated the performance and energy cost for live VM migrations from both theory and practice. Choi et al. [24] have presented a framework that autonomously finds the VM migration thresholds at run time, using the history resource utilization. Park et al. [25] proposed an automated strategy for virtual machine migration in a self-managing virtualized environment, as well as an optimization model based on linear programming.

In contrast, in our work we employ a two-threshold VM migration strategy to balance the loads across different physical nodes in the data center, in order to meet the varying demands of user applications.

*2.3. Power-Aware Resource Allocation Approaches.* Besides performance, energy consumption becomes another critical design parameter in modern data center, and enterprise environments, because it directly impacts both the power deployment and operational cost. Hence, plenty of work has focused on energy-aware and power-aware resource allocation approaches. Krioukov et al. [26] presented an energy agile cluster that is power proportional and exposes slack. Zhu et al. [27] proposed several power-aware storage cache management algorithms and also an online power-aware cache replacement algorithm. Petrucci et al. [28] presented a dynamic configuration support for specifying and deploying power management policies in a platform running multiple application services. Chen et al. [29] designed a server provisioning and load dispatching algorithm which aimed to save energy without sacrificing user experiences.

Similarly, as designing the decentralized VM migration scheme, our work also considers power consumption as an important metric too. Specifically, by migrating out VMs, underutilized physical nodes could be released to save more energy.

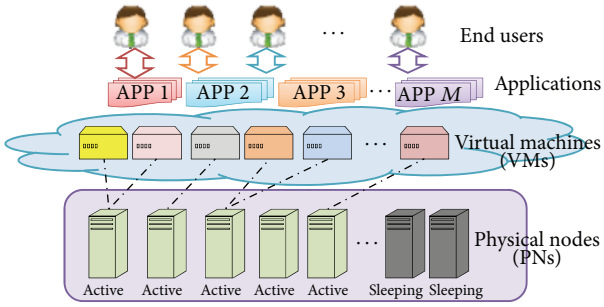



Figure 1: Data center architecture.

## 3. System Models

In this section, the basic architecture of the target system is described in detail as well as other model definitions. Then, the decentralized mechanism will be introduced.

*3.1. Large-Scale Data Centers for Cloud.* The target cloud environment we discuss in this paper is mainly a large-scale data center, which is usually comprised of a great number of physical nodes (PNs). The high-level system architecture is shown in Figure 1. Upon the physical infrastructure, virtual machines (VMs) are widely used to host many third-party applications. Multiple VMs can be dynamically started or stopped on a physical node according to incoming workloads, sharing the resources from the same physical devices. These VMs can run applications based on different operating system environments on a single physical node, providing usability and flexibility for cloud end users. At the upper level, end users obtain services from applications deployed in multiple virtual machines, which are residing on the underlying physical infrastructure.

Since the incoming workload varies significantly, the resource demands of each VM will fluctuate a lot too. To consolidate these workloads and release some underutilized resources, VM can be dynamically migrated across different physical nodes. In this way, some physical nodes could be turned off or into sleep mode in order to save extra energy. In Figure 1, we use "active" and "sleeping" to describe two types of node states, which are represented in light green and grey colors, respectively.

Besides, in the following problem discussion, we assume there are $N$ physical nodes in the data center and $K$ virtual machines running upon these nodes.

*3.2. Power Model.* Here, we present the power consumption model used in this paper. Since CPU usually consumes much more energy than the other parts of the computer, hereafter we focus on managing the power consumption and usage of CPU resources.

Most modern CPUs support Dynamic Voltage and Frequency Scaling (DVFS) techniques to dynamically change its own frequency to reduce energy wastes. Hence, we consider that the CPU utilization is typically proportional to the workload intensity, and the power consumption of a physical node is mainly impacted by its current CPU utilization. However, an idle physical node even with 0% utilization could still consume a plenty of power. Let $\alpha$ be the fraction of power consume by an idle node compared to a full utilized node and $\theta$ the current CPU utilization of the node. Then, we use the power model defined as follows to compute the power consumption of PN $i$:

$$P_i = \alpha \cdot P_i^{\mathrm{MAX}} + (1 - \alpha) \cdot \theta \cdot P_i^{\mathrm{MAX}}, \quad (1)$$

where $P_i^{\mathrm{MAX}}$ is the power consumption of PN $i$ when it is fully utilized (i.e., it reaches 100% of CPU utilization).

*3.3. Decentralized Mechanism.* To manage the resources inside a large-scale data center, a central resource manager is usually designed and implemented to adjust the systemwide resources and make appropriate decisions. However, such centralized manner is vulnerable facing single-point failure, which might lead to an unmanaged status of the whole system. Here, we propose a decentralized mechanism to address such issues and provide guarantees for availability of the VM management actions in various cases.

As shown in Figure 2, each active node will send a load index of itself to some other nodes during each control interval. At the same time, it will receive some load indexes from other active nodes. The sending targets are randomly picked at each interval and will possibly change in the next interval. Each node will add the load information it received into its own load vector. Then, the average length of the load vectors over all nodes will be equal to the number of load indexed sending times, which is denoted as $\eta$ in this paper.

In this way, the nodes are sending and receiving information to each other in a decentralized manner, without a central manager on the upper level. Such exchange will not be impacted even if some of the nodes fail to run or crack due to some unpredicted reason. On the other hand, the network flow would also be distributed and dispatched among different nodes in this case, rather than concentrated to a common node which receives all the information.

## 4. Decentralized Virtual Machine Migration Approach

In this section, the decentralized VM migration approach and resource management scheme will be presented, including the details of how to select the target VM and how to determine the destination node.

*4.1. Load Vector Establishment.* In order to make VM migration decisions, load information have to be collected first on each physical node. According to the decentralized working mechanism, each physical node maintains a load vector to receive load indexes from other peer nodes. Here, we use a tuple to represent the load index LI, defined as follows:

$$\mathrm{LI} = \langle src, dest, util \rangle, \quad (2)$$

where *src* indicates where the load index information comes from, *dest* indicates the ID of the target physical node that is

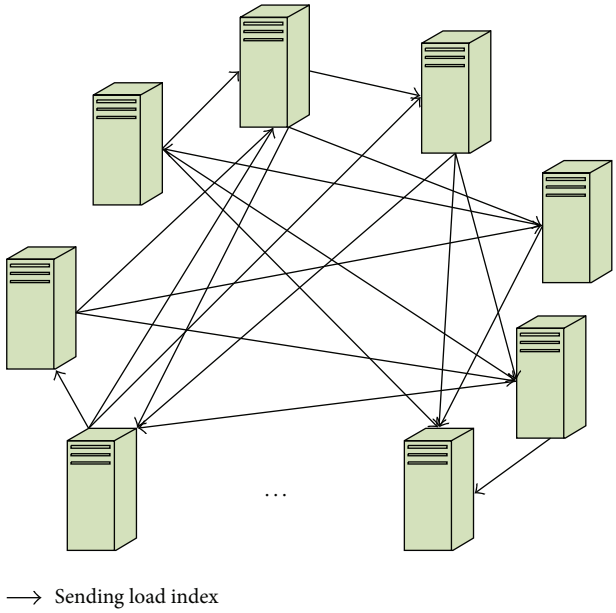



Figure 2: Decentralized load index exchange.

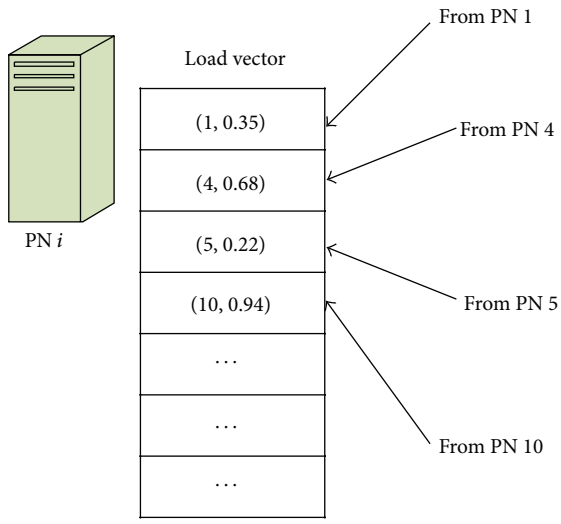



Figure 3: Load vector establishment.

going to receive this load index, and *util* denotes the current CPU utilization of the source node (as *src* indicates).

As shown in Figure 3, PN *i* will receive load indexes from other PNs and add them into the load vector of itself, which could be implemented by an array or a queue. Each element of the load vector contains information about the ID of the source node and its current CPU utilization. After all the load indexes in the current control interval have been received, the load vector of PN *i* could be established, which provides necessary directions for later migration decisions.

*4.2. Selecting a VM to Migrate.* Since the VMs are hosting different applications with varying workloads, the CPU utilization of the physical nodes will change a lot over time. When the CPU utilization is too high, exceeding a certain level, some VMs should be migrated to other PNs to release some resource capacity. On the other hand, if the CPU utilization is too low, below a certain level, we regard the node as "underutilized", and the VMs on it should also be migrated out to clear the load of this node. In this case, the blank node with no jobs could be turned into sleeping state so that more energy could be saved.

Hence, we introduce a double-threshold VM migration strategy, with two predefined threshold values called "lower threshold (LT)" and "upper threshold", respectively. The aim of setting the upper threshold is to preserve extra CPU capacity for unpredicted workload rises and to prevent SLA violations as much as possible. The objective of setting the lower threshold is trying to switch more physical nodes which are not fully utilized into sleep mode, leading to much less energy consumption than idling.

The pseudocode for the algorithm is presented in Algorithm 1, which describes how to select a VM to migrate on PN *i* which is overloaded. Lines 1~2 are the initialization of getting the current utilization of PN *i*. Lines 3~32 are loop to select VMs one by one to migrate out until the current utilization becomes less than the predefined upper threshold.

When the current utilization of PN *i* is higher than the upper threshold, there are three scenarios as shown in Figure 4. In case (a), we can find a VM that if it is migrated out, the resulted utilization of PN *i* will be reduced to a value lower than the upper threshold and higher than the lower threshold. In this case, we only need to choose this VM to migrate and end the selecting procedure, as shown in lines 6~16 of Algorithm 1. Otherwise, if we did not find any VM meeting the requirements in case (a), the reason is perhaps that all VMs utilize relatively little amount of CPU capacity, as shown in case (b). In this case, several VMs will be migrated out until the resulted utilization drops below the upper threshold, as shown in lines 17~29. The last scenario is that we cannot find any VM in case (a) and (b). That is probably because some VM occupies too much CPU capacity so that if it is selected, the utilization of PN *i* will fall down below the lower threshold, as shown in case (c). In this case, the VM will not be selected, since the migration of this VM will also cause overutilization on the target node. The corresponding code is as shown in line 30. At last, line 31 is a function to find a destination for the previously selected VM, which will be elaborated in the next section.

If the utilization of a physical node is below the lower threshold, we regard it as "underutilized". Then, all the VMs residing on the PN will be selected to migrate. The detailed algorithm is omitted here due to space constraint.

*4.3. Decide the Destination.* After some VM is selected to be migrated, the next necessary step is to find a migrating destination for it. Here, we use a function called *findDest*(*vmToMig*) to conduct the destination decision, which will find a proper physical node for *vmToMig* according to the *BestFit* strategy. The pseudo-code of the detailed algorithm is shown as Algorithm 2.

```
(1)  double util=this.utilization; //get the current utilization of PN i
(2)  VirtualMachine vmToMig=null;
(3)  while (util > UT)
(4)  {
(5)     vmToMig=null;
(6)     for each (vm in this.VMlist)
(7)     {
(8)        if (vm.toBeMigrated) continue; //skip the VMs that have been marked
(9)        double vmutil=calcutilz(vm); //get the CPU utilization of vm
(10)       if (vmutil > util-UT && vmutil <util-LT)
(11)       {   //case (a)
(12)          util = util – vmutil;
(13)          vmToMig=vm;
(14)          break; //find a VM to migrate
(15)       }
(16)    }
(17)    if (vmToMig==null)
(18)    {
(19)       for each(vm in this.VMlist)
(20)       {
(21)          if (vm.toBeMigrated) continue;
(22)          double vmutil=calcutilz(vm);
(23)          if (vmutil <util-LT)
(24)          {   //case (b)
(25)             util = util – vmutil;
(26)             vmToMig=vm;
(27)             break; //find a VM to migrate
(28)          }
(29)       }
(30)    if (vmToMig==null) break; //case (c)
(31)    findDest(vmToMig); //find a destination for the current VM to migrate to
(32) }
```

Algorithm 1: Selecting a VM to migrate on PN $i$ which is overloaded.

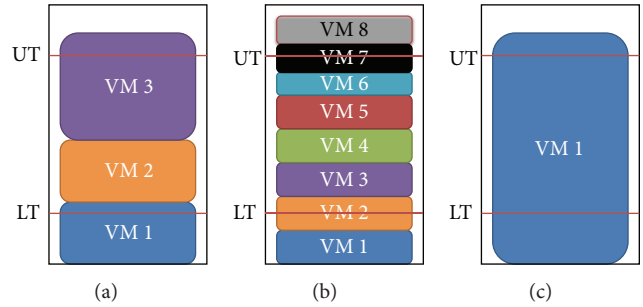



Figure 4: Some typical scenarios on overutilized nodes.

First, the utilization of the selected VM has to be computed as shown in line 1. Then, the PN traverses all the load indexes in its own load vector and try to find a target so that if the VM is migrated there, the resulted utilization will be between the lower threshold and the upper threshold. Among such nodes, the algorithm is prone to choose the one with minimum utilization currently, as shown in lines 2~13.

As shown in lines 18~24, if there is no such proper node meeting the above requirements, we continue to find a target with little resource utilization that its future utilization will still be below the lower threshold even with an additional VM.

Furthermore, if there is still not any target found during the last two rounds, it means that there are no suitable nodes which can accept an extra VM (maybe every active node is too over-utilized). Then, the source node of *vmToMig* will attempt to require a sleeping node to wake up and receive the VM to be migrated, as shown in lines 25~30. If it fails to require any sleeping node successfully, the whole procedure

```
(1)  calculate the utilization of vmToMig as vmutil
(2)  double minUtil=1.0;
(3)  int bestTarget=−1;
(4)  for each (li in LoadVector)
(5)  {   if (vmutil + li.util >=LT
(6)          && vmutil + li.util <= UT)
(7)      {   if (li.util < minUtil)
(8)          {
(9)            minUtil=li.util;
(10)           bestTarget=li.PNid;
(11)         }
(12)     }
(13) }
(14) if (bestTarget >0) //find target successfully
(15) {   vmToMig.dest=bestTarget;
(16)     return 0;
(17) }
(18) for each (li in LoadVector)
(19) {
(20)     if (vmutil + li.util <=LT)
(21)     {   vmToMig.dest=bestTarget;
(22)         return 0;
(23)     }
(24) }
(25) int getSleepNodeID=requireSleepNode();
(26) if (getSleepNodeID < 0) return −1;
(27) else
(28) {   vmToMig.dest=getSleepNodeID;
(29)     return 0;
(30) }
(31) return −1;
```

Algorithm 2: Function *findDest*(vmToMig).

of finding destination will be terminated, and *vmToMig* will not be migrated but still remain on its original physical node.

## 5. Performance Evaluation

In this section, we conduct a series of performance evaluation experiments of the decentralized VM migration approach proposed in Section 4. Since it is extremely difficult to conduct repeatable large-scale experiments on real-world infrastructure, we chose simulation methods to evaluate the performance of the proposed approach. We used C#.NET to develop an event-driven simulation environment, which could simulate the workload variation, application behaviors, task completion status, and energy consumption. Simulation parameter settings are first described and then the results will be illustrated later.

*5.1. Parameter Settings.* We have simulated a data center comprising 50 homogeneous physical nodes. Each node is modeled to have a CPU capacity of 750 MIPS. Power consumption is defined according to the model presented in Section 3.2, where $P_i^{\mathrm{MAX}}$ is set to 259 W according to the SPECpower benchmark [30], and $\alpha$ is set to 50%. Then, a physical node consumes 129.5 W with 0% CPU utilization and consumes 259 W with 100% CPU utilization. Upon the underlying physical infrastructure, there are 150 heterogeneous VMs hosting different kinds of applications. The workload and CPU demand of each VM varies as time elapsed. The initial CPU demand, minimum and maximum demand, and the variation amount of the VMs are set randomly according to a uniformly distributed variable, which simulated the independent fluctuation of different types of applications.

Furthermore, during the experiments, VM migration decisions have to be made across a constant control interval time, which is set to 60 seconds. Besides, we set the delay from sending the load information to receiving the information as 1 s, the VM migration time is set to 5 s, and the wakeup time of a sleeping physical node is set to 15 s. The simulation time of the entire experiment is 1440 minutes in total, which simulates a whole day effect of system running.

Notably, the length of the load vector of each physical node is set to 10, if not specified explicitly.

*5.2. Threshold Analysis.* In this subsection, we intend to examine the performance of the decentralized VM migration approach when setting different lower and upper utilization thresholds.

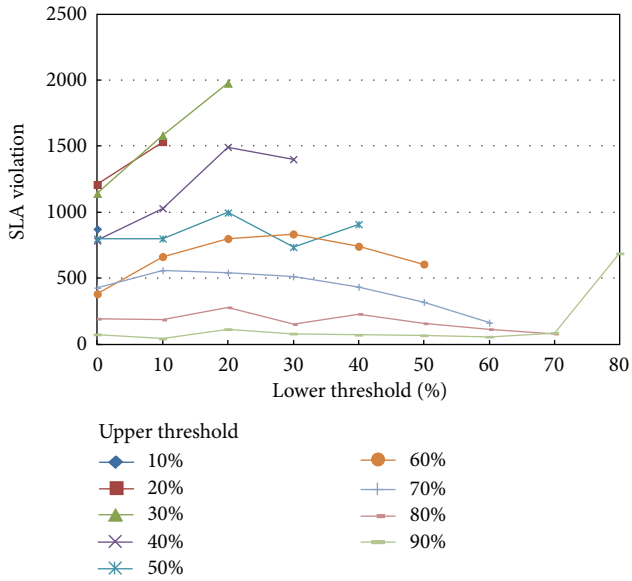



Figure 5: SLA violation analysis with different lower thresholds and upper thresholds.

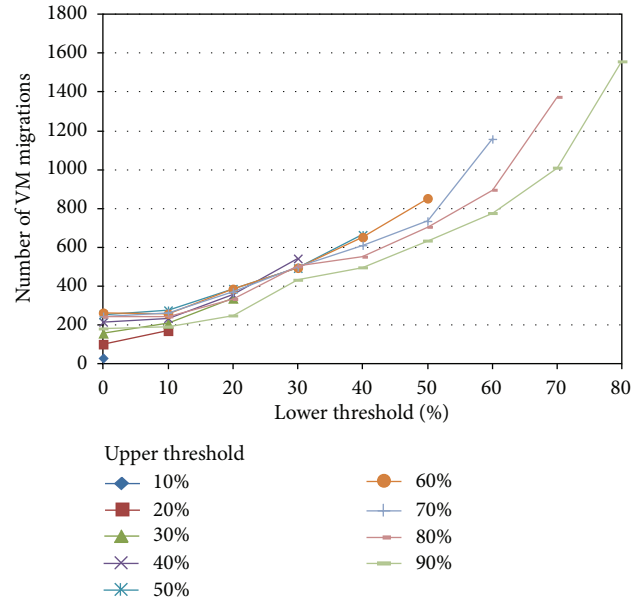



Figure 6: Number of VM migrations analysis with different lower thresholds and upper thresholds.

First, the average SLA violation amount is recorded and illustrated in Figure 5. As shown, we can see that increasing the upper threshold beyond 40% helps to reduce SLA violation remarkably. However, the impact of the lower utilization threshold is not regularly noticeable. The best result occurs at LT = 10% and UT = 90%, and we will use this setting for the following experiments if not specified explicitly.

Furthermore, we also investigate the number of VM migration times with different lower and upper utilization thresholds, and the results are illustrated in Figure 6. It can be observed that as the lower threshold increases, the number of VM migration times rises obviously. The reason is that as the lower threshold increases, more physical nodes will judge themselves as "underutilized" and more VM migrations will be triggered to eliminate resource waste. On the other hand, with the same lower threshold, higher upper threshold leads to the reduction of migration times. This is because the that as the upper threshold increases, the physical nodes are allowed to hold more VM demands, and then fewer migrations will be triggered due to overutilization.

When we jointly consider the results of Figures 5 and 6 together, it can be seen that although most results are relatively good at the aspect of SLA violation when UT = 90%, higher LT will trigger more VM migration times, which incurs heavier overhead to the whole system. Synthetically, we regard the combination of LT = 10% and UT = 90% as a possible appropriate choice for later experiments.

*5.3. Load Balancing Effect.* Here, we intend to examine the load balancing effect of the proposed decentralized approach. The experiments are repeated using three different strategies as follows.

(1) *Static*: In the initialization stage, the virtual machines are allocated onto the physical nodes as long as the utilization of each node does not reach 100%. Then, the VM placement scheme will not change during the system execution procedure.

(2) *RR*: In the initialization stage, the virtual machines are allocated onto the physical nodes one by one in a round robin manner, in order to balance the load among multiple physical nodes. VM migration is not supported in this strategy.

(3) *DVM*: As described in Section 4, load information is collected in a decentralized manner between different node pairs. Virtual machines will be dynamically migrated according to the load distribution among the physical nodes.

In this group of experiments, we compared the standard deviation across all of the 50 physical nodes. The results are shown in Figure 7. As it can be observed, the *Static* strategy leads to large deviation value since the VMs are distributed in an unbalancing way. When using *RR* strategy, although the deviation is small during the first several rounds due to the evenly distributed VMs in the initialization stage, the load becomes remarkably unbalanced in later time periods. In comparison, by dynamically migrating VMs among different physical nodes using *DVM* strategy, the resource utilization values are kept relatively more balanced, leading to the least deviation among all nodes.

*5.4. Energy Consumption.* In this subsection, we focus on the energy consumption of our approach compared to *RR* strategy. The number of physical nodes and VMs are set to 50 and 100, respectively, in order to simulate a relatively lighter workload scenario.

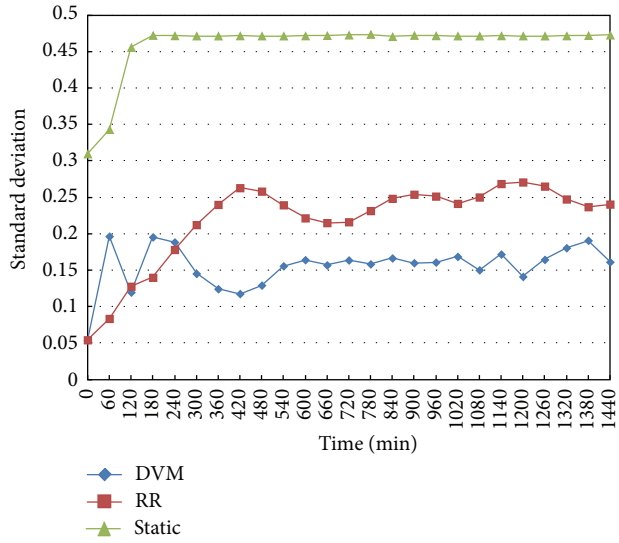



Figure 7: Standard deviation of all physical nodes with three different strategies.

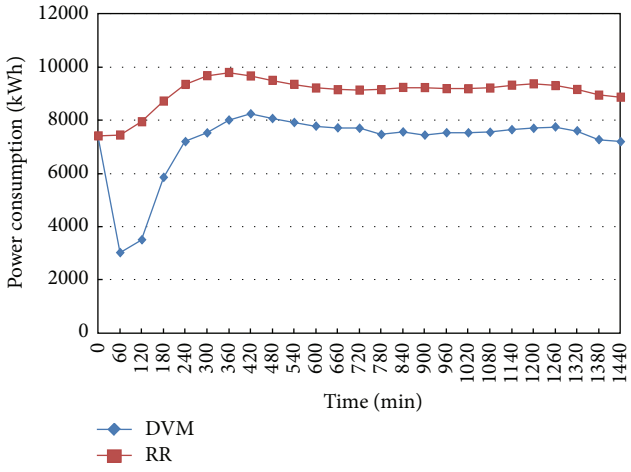



Figure 8: Power consumption comparison of DVM and RR strategies (no SLA violation).

The experimental results are shown in Figure 8. It is notable that the *DVM* strategy achieves less power consumption, leading to more than 20% energy savings. The reason is that our approach considers migrating VMs from the physical nodes whose utilization is under the predetermined lower threshold. In this way, the underutilized physical node could be released and be turned into sleeping status, which incurs much less power consumption than the idling state.

*5.5. Impact of Load Vector Length.* At last, we attempt to investigate the impact of the load vector length on the performance of our approach. The length of the load vector determines how many load indexes will be transferred during the system execution. Longer load vector may provide more information for the current physical node, but will also bring more overheads at the same time. We repeated several experiments with the same lower threshold and upper threshold but different length of the load vector, and the results are illustrated in Figure 9.

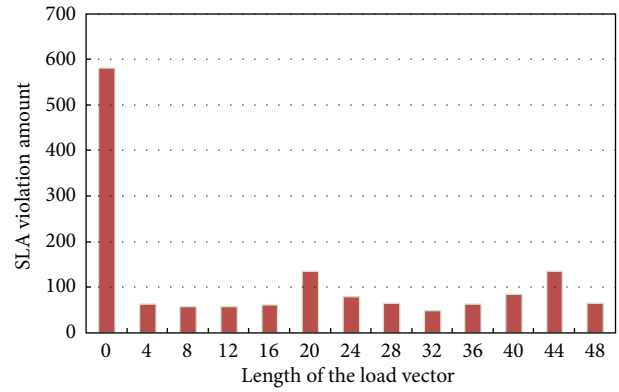



Figure 9: Power consumption comparison of DVM and RR strategies (no SLA violation).

It is notable that a large value of the load vector length will not always lead to better performance, even though for each node it gets more information from the other nodes. The reason is that a node will choose the most light-loaded node from its load vector as the destination target. However, a possible scenario is that multiple nodes choose the same node as the target but do not know that situation from each other. As a result, a light-loaded load might be selected as the target for many times, which makes it overloaded in the next interval and leads to much SLA violation. In other words, the performance is not proportional with the increase of the load vector length due to the decentralized mechanism.

From another point of view, shorter load vector could also achieve better performance which benefits from incomplete messages among different node pairs. Besides, the smaller value of the length could also reduce the network overhead for sending and receiving load indexes. Thus, we found 8 to 12 is an appropriate value for the load vector length in a data center comprised of 50 physical nodes.

## 6. Conclusions and Future Work

In this paper, we have proposed a decentralized resource management approach for data centers which use virtual machines to host many third-party applications. The system models are defined and described in detail. Then, we present the design of the decentralized VM migration approach, which considers both load balancing and saving of energy costs by turning some underutilized nodes into sleeping state. The VM migration decisions are made according to the two thresholds predetermined for the system, and several load indexes of one node will be sent to another several nodes randomly chosen according to the load vector length. Performance evaluation results of the simulation experiments illustrate that our approach can achieve better load balancing effect and less power consumption than other strategies. Besides, we also examine and discuss the impact of some key

factors in our approach on the final performance. The benefit of the decentralized approach is to eliminate the fatal problem of single-point failure, which helps improve the availability of the entire system.

As part of ongoing work, we plan to incorporate the proposed methods into our realistic cloud environment and examine its effect and efficiency when putting into real-world usage. Also, we are considering combining the centralized management and decentralized management approach together to further utilized their advantages in different aspects.

## Acknowledgments

This paper is granted by the National Natural Science Foundation of China (no. 61363019) and the Tsinghua-Tencent Joint Laboratory for Internet Innovation Technology (no. 2011-1).

## References

[1] B. Hayes, "Cloud computing," *Communications of the ACM*, vol. 51, no. 7, pp. 9–11, 2008.

[2] M. Armbrust, A. Fox, R. Griffith et al., "Above the clouds: a berkeley view of cloud computing," Tech. Rep. UCB/EECS-2009-28, EECS Department, University of California, Berkeley, 2009.

[3] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility," *Future Generation Computer Systems*, vol. 25, no. 6, pp. 599–616, 2009.

[4] M. Stillwell, D. Schanzenbach, F. Vivien, and H. Casanova, "Resource allocation using virtual cluster," in *Proceedings of the 9th IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID '09)*, pp. 260–267, May 2009.

[5] X. Wang, D. Lan, G. Wang et al., "Appliance-based autonomic provisioning framework for virtualized outsourcing data center," in *Proceedings of the 4th International Conference on Autonomic Computing (ICAC '07)*, p. 29, Fla, USA, June 2007.

[6] M. Rosenblum and T. Garfinkel, "Virtual machine monitors: current technology and future trends," *Computer*, vol. 38, no. 5, pp. 39–47, 2005.

[7] O. Abdul-Rahman, M. Munetomo, and K. Akama, "Live migration-based resource managers for virtualized environments: a survey," in *Proceedings of the 1st International Conference on Cloud Computing, GRIDs, and Virtualization (CLOUD COMPUTING '10)*, pp. 32–40, 2010.

[8] P. Patel, A. Ranabahu, and A. Sheth, "Service level agreement in cloud computing," in *Proceedings of the Cloud Workshops at OOPSLA*, pp. 1–10, 2009.

[9] W.-C. Feng, X. Feng, and R. Ge, "Green supercomputing comes of age," *IT Professional*, vol. 10, no. 1, pp. 17–23, 2008.

[10] X. Wang, Z. Du, Y. Chen, and S. Li, "Virtualization-based autonomic resource management for multi-tier Web applications in shared data center," *Journal of Systems and Software*, vol. 81, no. 9, pp. 1591–1608, 2008.

[11] X. Wang, Y. Xue, L. Fan, R. Wang, and Z. Du, "Research on adaptive QoS-aware resource reservation management in cloud service environments," in *Proceedings of the IEEE Asia-Pacific Services Computing Conference (APSCC '11)*, pp. 147–152, December 2011.

[12] X. Wang, Z. Du, Y. Chen et al., "An autonomic provisioning framework for outsourcing data center based on virtual appliances," *Journal of Networks Software Tools and Applications*, vol. 11, no. 3, pp. 229–245, 2008.

[13] X. Wang, H. Xie, R. Wang, Z. Du, and L. Jin, "Design and implementation of adaptive resource co-allocation approaches for cloud service environments," in *Proceedings of the 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE '10)*, pp. V2484–V2488, August 2010.

[14] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing," *Future Generation Computer Systems*, vol. 28, no. 5, pp. 755–768, 2012.

[15] H. N. Van, F. D. Tran, and J.-M. Menaud, "Autonomic virtual resource management for service hosting platforms," in *Proceedings of the ICSE Workshop on Software Engineering Challenges of Cloud Computing (CLOUD '09)*, pp. 1–8, May 2009.

[16] R. Urgaonkar, U. C. Kozat, K. Igarashi, and M. J. Neely, "Dynamic resource allocation and power management in virtualized data centers," in *Proceedings of the 12th IEEE/IFIP Network Operations and Management Symposium (NOMS '10)*, pp. 479–486, April 2010.

[17] H. N. Van, F. D. Tran, and J.-M. Menaud, "SLA-aware virtual resource management for cloud infrastructures," in *Proceedings of the IEEE 9th International Conference on Computer and Information Technology (CIT '09)*, pp. 357–362, October 2009.

[18] W. Iqbal, M. Dailey, and D. Carrera, "SLA-driven adaptive resource management for Web applications on a heterogeneous compute cloud," *Lecture Notes in Computer Science*, vol. 5931, pp. 243–253, 2009.

[19] B. Maniymaran and M. Maheswaran, "Virtual clusters: a dynamic resource coallocation strategy for computing utilities," in *Proceedings of the 16th IASTED International Conference on Parallel and Distributed Computing and Systems*, pp. 53–58, November 2004.

[20] P. Campegiani, "A genetic algorithm to solve the virtual machines resources allocation problem in multi-tier distributed systems," in *Proceedings of the 2nd International Workshop On Virtualization Performances: Analysis, Characterization and Tools (VPACT '09)*, 2009.

[21] J. Almeida, V. Almeida, D. Ardagna, C. Francalanci, and M. Trubian, "Resource management in the autonomic service-oriented architecture," in *Proceedings of the 3rd International Conference on Autonomic Computing (ICAC '06)*, pp. 84–92, June 2006.

[22] X. Wang, Z. Du, and Y. Chen, "An adaptive model-free resource and power management approach for multi-tier cloud environments," *Journal of Systems and Software*, vol. 85, no. 5, pp. 1135–1146, 2012.

[23] H. Liu, C.-Z. Xu, H. Jin, J. Gong, and X. Liao, "Performance and energy modeling for live migration of virtual machines," in *Proceedings of the 20th ACM International Symposium on High-Performance Parallel and Distributed Computing (HPDC '11)*, pp. 171–181, June 2011.

[24] H. W. Choi, H. Kwak, A. Sohn, and K. Chung, "Autonomous learning for efficient resource utilization of dynamic VM migration," in *Proceedings of the 22nd ACM International Conference on Supercomputing (ICS '08)*, pp. 185–194, June 2008.

[25] J.-G. Park, J.-M. Kim, H. Choi, and Y.-C. Woo, "Virtual machine migration in self-managing virtualized server environments," in *Proceedings of the 11th International Conference on Advanced*

*Communication Technology (ICACT '09)*, pp. 2077–2083, February 2009.

[26] A. Krioukov, S. Alspaugh, P. Mohan, S. Dawson-Haggerty, D. E. Culler, and R. H. Katz, "Design and evaluation of an energy agile computing cluster," Tech. Rep. UCB/EECS-2012-13, EECS Department, University of California, Berkeley, 2012.

[27] Q. Zhu, F. M. David, C. F. Devaraj, Z. Li, Y. Zhou, and P. Cao, "Reducing energy consumption of disk storage using power-aware cache management," in *Proceedings of the 10th International Symposium on High Performance Computer Architecture*, pp. 118–129, February 2004.

[28] V. Petrucci, O. Loques, B. Niteroi, and D. Mossé, "Dynamic configuration support for power-aware virtualized server clusters," in *Proceedings of the WiP Session of the 21th Euromicro Conference on Real-Time Systems*, Dublin, Ireland, 2009.

[29] G. Chen, W. He, J. Liu et al., "Energy-aware server provisioning and load dispatching for connection-intensive internet services," in *Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation*, pp. 337–350, 2008.

[30] K.-D. Lange, "Identifying shades of green: the SPECpower benchmarks," *Computer*, vol. 42, no. 3, pp. 95–97, 2009.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 387083, 7 pages
http://dx.doi.org/10.1155/2013/387083

*Research Article*

# A QoS-Satisfied Prediction Model for Cloud-Service Composition Based on a Hidden Markov Model

**Qingtao Wu, Mingchuan Zhang, Ruijuan Zheng, Ying Lou, and Wangyang Wei**

*College of Information Engineering, Henan University of Science and Technology, Luoyang 471023, China*

Correspondence should be addressed to Qingtao Wu; wuqingtao.cn@hotmail.com

Received 31 May 2013; Accepted 4 July 2013

Academic Editor: Yuxin Mao



Various significant issues in cloud computing, such as service provision, service matching, and service assessment, have attracted researchers' attention recently. Quality of service (QoS) plays an increasingly important role in the provision of cloud-based services, by aiming for the seamless and dynamic integration of cloud-service components. In this paper, we focus on QoS-satisfied predictions about the composition of cloud-service components and present a QoS-satisfied prediction model based on a hidden Markov model. In providing a cloud-based service for a user, if the user's QoS cannot be satisfied by a single cloud-service component, component composition should be considered, where its QoS-satisfied capability needs to be proactively predicted to be able to guarantee the user's QoS. We discuss the proposed model in detail and prove some aspects of the model. Simulation results show that our model can achieve high prediction accuracies.

## 1. Introduction

Cloud computing is a term used to refer to the use of widespread, shared computing resources. It is an alternative to having local servers handle computing applications. Cloud computing groups together a large number of computing servers and other resources, often offering their combined capacity on an on-demand, pay-per-cycle basis. The end users of a cloud computing network usually have no idea where the servers are physically located—they just open their applications and start working [1–3].

In general, the service resources in cloud computing will include hardware resources (e.g., processors, storage, and networking) and software resources (e.g., web servers, databases, message queuing systems, and monitoring systems). Cloud service types can be abstracted into three layers, namely software as a service (SaaS), platform as a service (PaaS), and infrastructure as a service (IaaS) [4]. Hardware and software resources form the basis for delivering IaaS and PaaS. The SaaS layer at the top focuses on application services by making use of services provided by the lower layers. PaaS/SaaS services are often developed and provided by third-party service providers who are different from the IaaS provider.

Therefore, matching the cloud-service components to the users' quality of service (QoS) is very important because user experience is a principal reason for promoting the development of cloud computing. For each user request, the provider should select an appropriate composition of cloud-service components to serve the user if there is no single cloud-service component that satisfies the user's QoS perfectly. This will require predictions to be made about the QoS satisfaction for compositions of cloud-service components.

Wang et al. [5] propose a composition method for selecting cloud-based web services from candidate services, changing a single service into a more powerful composite service, which uses the Skyline operator and Particle Swarm Optimization. Javadi et al. [6] investigate cloud-computing resource provision to extend the computing capacity of local clusters in the presence of failures. Their three steps in resource provision include resource brokering, dispatch sequences, and scheduling. Benouaret et al. [7] present an approach to composing Data Web services automatically while taking into account user preferences, which is based on fuzzy sets and top-k optimization. Liu et al. [8] introduce a scheme to address the particularities of manufacturing-resource service composition and optimization, where the user's QoS is considered. Zhang et al. [4] present an investigation of an intelligent decision-support system for selecting cloud-based infrastructure services to choose the best mix

of service offerings from an abundance of possibilities. Although those papers discuss the composition of cloud-service components, QoS-satisfied predictions are not considered.

Huang et al. [9] introduce a method for addressing the problem of composing a sequence of service components for QoS-guaranteed service provision in a virtualization-based cloud-computing environment. Huang et al. [10] designs a suboptimal resource-allocation system in a cloud-computing environment, and a corresponding prediction mechanism is realized by using support-vector regression to estimate the resource utilization. Di and Wang [11] propose a fully distributed, VM-multiplexing resource-allocation scheme to manage decentralized resources. Jiang et al. [12] propose a new method for cloud-capacity planning, with the goals of utilizing the physical resources fully and providing an integrated system with intelligent cloud-capacity prediction.

The Markov model (MM), particularly the hidden Markov model (HMM), has been shown to be a good technique for solving prediction problems. Zhang and Pathirana [13] present an adaptive HMM for identifying underlying path losses. Choi et al. [14] present a sparsely correlated HMM for assessing multiple genomic datasets. Botev et al. [15] present a versatile Monte Carlo method for estimating multidimensional integrals, with application to rare-event probability estimation. Xie et al. [16] present a new structurally discrete approach to predicting network traffic called the nested hidden semi-Markov model, which includes a nested latent semi-Markov chain and one observable discrete stochastic process. However, further research into QoS-satisfied capability should be considered.

This paper focuses on QoS-satisfied predictions for compositions of cloud-service components, based on the HMM and our previous work [17–19]. The remainder of this paper is organized as follows. In Section 2, the basic model used in later discussion is presented. In Section 3, we propose a QoS-satisfied prediction model and explain it in detail. In Section 4, the simulation and analysis are discussed. Finally, we conclude the paper in Section 5.

## 2. Basic Model

The MM is a statistical model with a wide area of application. Many other Markov models are derived fundamentally from MM, including the HMM and the semi-Markov model. The HMM is used extensively for performance modeling and performance-prediction analysis, where the HMM can predict the future state of a target system based on its current state. In reality, because the relationship between the observed time and the observed state is not one to one, a group of probability distributions for two stochastic processes are involved, called the HMM.

In an HMM, the states are not observable, but when we visit a state, an observation is recorded that is a probabilistic function of the state. We assume a discrete observation in each state from the set

$$\{v_1, v_2, \ldots, v_M\} : b_j(m) \triangleq \Pr\left(O_t = v_m \mid q_t = S_j\right), \quad (1)$$

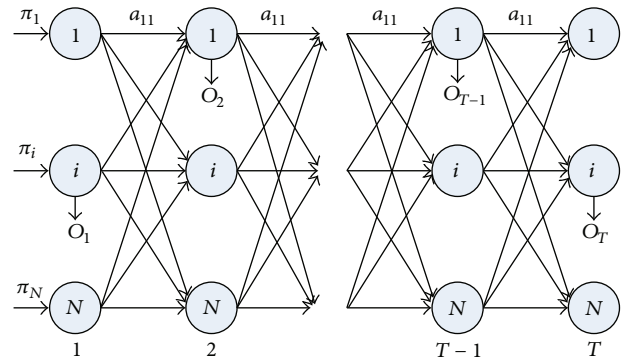



Figure 1: An example of an observable model for an HMM.

where $b_j(m)$ is the *observation* or *emission probability*. We also assume a homogeneous model for which the probabilities do not depend on $t$. The values thus observed constitute an observation sequence $O$. The state sequence $Q$ is not observed directly (being "hidden"), but it should be possible to infer it from the observation sequence $O$. An example of an observable model for an HMM is shown in Figure 1, where $M = N$ and $b_j(m) = 1$ if $j = m$ and $b_j(m) = 0$ otherwise. To summarize and formalize, an HMM has the following elements:

(1) $N$: the number of states in the model

$$S = \{S_1, S_2, \ldots, S_N\} \quad (2)$$

(2) $M$: the number of distinct observation symbols in the *alphabet*

$$V = \{v_1, v_2, \ldots, v_M\}. \quad (3)$$

(3) State transition probabilities:

$$A = \left[a_{ij}\right], \quad \text{where } a_{ij} = \Pr\left(q_{t+1} = S_j \mid q_t = S_i\right). \quad (4)$$

(4) Observation probabilities:

$$B = \left[b_j(m)\right], \quad \text{where } b_j(m) = \Pr\left(O_t = v_m \mid q_t = S_j\right). \quad (5)$$

(5) Initial state probabilities:

$$\Pi = \left[\pi_i\right], \quad \text{where } \pi_i = \Pr\left(q_1 = S_i\right). \quad (6)$$

$N$ and $M$ are implicitly defined by the other parameters, leaving $\lambda = (A, B, \Pi)$ as the parameter set for an HMM. Given $\lambda$, the model can be used to generate an arbitrary number of observation sequences of arbitrary length, but we are usually interested in the other direction, namely, that of estimating the parameters of the model given a training set of sequences.

## 3. Proposed QoS-Satisfied Prediction Model for Cloud-Service Composition

In this section, we propose a prediction model for cloud-service composition based on an HMM. We first present an overview of cloud-service composition and then give details of the prediction model.

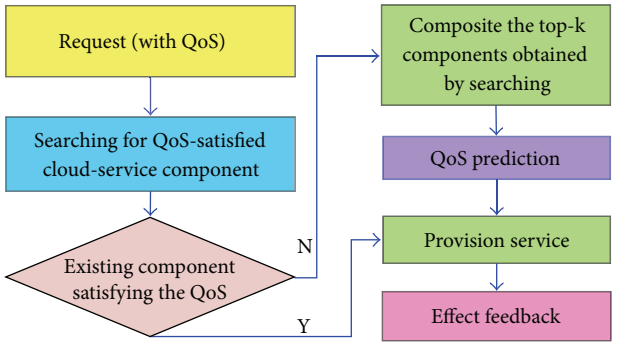



Figure 2: General flowchart for cloud-service provision.

*3.1. An Overview of Cloud-Service Composition and Matching.* The key roles played in a cloud environment are those of the service user and the service provider. The cloud-service user needs anytime, anywhere QoS-satisfied and low-cost services that are flexible and easy to use. The important hurdles to users adopting cloud services involve security, availability, and reliability. We should therefore assess the QoS-satisfied capability for each service provision generated by matching cloud-service components with the users' QoS. Sometimes, if a single cloud-service component does not satisfy the users' QoS, multiple cloud-service components should be composed to provide a more complex service. General flowcharts for cloud services and for a cloud-service component-matching model are shown in Figures 2 and 3, respectively.

*3.2. QoS-Satisfied Prediction Model.* We propose a QoS-satisfied prediction model based on an HMM to predict whether a composition of cloud-service components can satisfy the user's QoS. Using the basic HMM model, we assume that the QoS capability of cloud-service components is a state set $\{v_1, v_2, \ldots, v_M\} : b_j(m) \triangleq \Pr(O_t = v_m \mid q_t = S_j)$, where $b_j(m)$ is the observation probability of obtaining $v_m$ ($m = 1, 2, \ldots, M$) when the composition state is $S_j$ and $O$ is a sequence of obtained $v_m$ values. To reduce the complexity of calculation for $\Pr(O \mid \lambda)$, we define a forward variable

$$\alpha_t(i) \triangleq \Pr(O_1, O_2, \ldots, O_t, q_t = S_i \mid \lambda), \quad 1 \le i \le N. \tag{7}$$

The three stages of its recursion are shown in Figure 4(a) and described as follows.

(1) Initialization:

$$\begin{aligned} \alpha_1(i) &\triangleq \Pr(O_1, q_1 = S_i \mid \lambda) \\ &= \Pr(O_1 \mid q_1 = S_i, \lambda) \Pr(q_1 = S_i \mid \lambda) \\ &= \pi_i b_i(O_1). \end{aligned} \tag{8}$$

(2) Recursion:

$$\begin{aligned} \alpha_{t+1}(j) &\triangleq \Pr(O_1, \ldots O_{t+1}, q_{t+1} = S_j \mid \lambda) \\ &= \left[\sum_{i=1}^{N} \alpha_t(i)\, a_{ij}\right] b_j(O_{t+1}). \end{aligned} \tag{9}$$

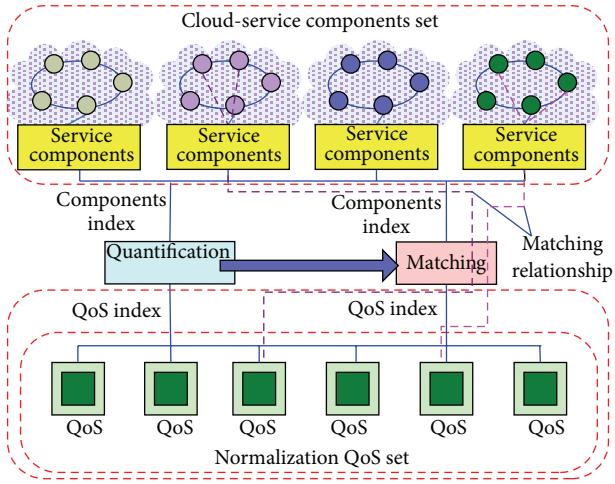



Figure 3: Cloud-service component-matching model.

*Proof*

$$\begin{aligned} \alpha_{t+1}(j) &\triangleq \Pr(O_1, \ldots, O_{t+1}, q_{t+1} = S_j \mid \lambda) \\ &= \Pr(O_1, \ldots, O_{t+1} \mid q_{t+1} = S_j, \lambda) \Pr(q_{t+1} = S_j \mid \lambda) \\ &= \Pr(O_1, \ldots, O_t \mid q_{t+1} = S_j, \lambda) \\ &\quad \cdot \Pr(O_{t+1} \mid q_{t+1} = S_j, \lambda) \Pr(q_{t+1} = S_j \mid \lambda) \\ &= \Pr(O_1, \ldots, O_t, q_{t+1} = S_j \mid \lambda) \\ &\quad \cdot \Pr(O_{t+1} \mid q_{t+1} = S_j, \lambda) \\ &= \Pr(O_{t+1} \mid q_{t+1} = S_j, \lambda) \\ &\quad \cdot \sum_{i=1}^{N} \Pr(O_1, \ldots, O_t, q_t = S_i, q_{t+1} = S_j \mid \lambda) \\ &= \Pr(O_{t+1} \mid q_{t+1} = S_j, \lambda) \\ &\quad \cdot \sum_{i=1}^{N} \Pr(O_1, \ldots, O_t, q_{t+1} = S_j \mid q_t = S_i, \lambda) \\ &\quad \cdot \Pr(q_t = S_i \mid \lambda) \\ &= \Pr(O_{t+1} \mid q_{t+1} = S_j, \lambda) \\ &\quad \cdot \sum_{i=1}^{N} \Pr(O_1, \ldots, O_t, q_t = S_i \mid \lambda) \\ &\quad \cdot \Pr(q_{t+1} = S_j \mid q_t = S_i, \lambda) \\ &= \left[\sum_{i=1}^{N} \alpha_t(i)\, a_{ij}\right] b_j(O_{t+1}). \end{aligned} \tag{10}$$

□

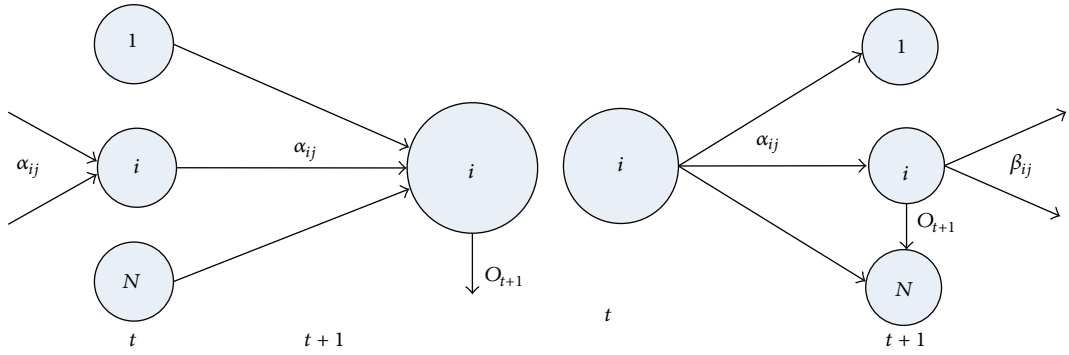



Figure 4: Forward-backward procedure: (a) computation of $\alpha_t(j)$; (b) computation of $\beta_t(j)$.

(3) End:

$$\Pr(O \mid \lambda) = \sum_{i=1}^{N} \Pr(O, q_T = S_i \mid \lambda) = \sum_{i=1}^{N} \alpha_T(i). \tag{11}$$

Similarly, we define a backward variable

$$\beta_t(i) \triangleq \Pr(O_{t+1}, \ldots, O_T \mid q_t = S_i, \lambda). \tag{12}$$

The three stages of its recursion are shown in Figure 4(b) and described as follows.

(1) Initialization:

$$\beta_T(i) = 1. \tag{13}$$

(2) Recursion:

$$\begin{aligned} \beta_t(i) &\triangleq \Pr(O_{t+1}, \ldots, O_T \mid q_t = S_i, \lambda) \\ &= \sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j). \end{aligned} \tag{14}$$

*Proof*

$$\begin{aligned} \beta_t(i) &\triangleq \Pr(O_{t+1}, \ldots, O_T \mid q_t = S_i, \lambda) \\ &= \sum_{j=1}^{N} \Pr(O_{t+1}, \ldots, O_T, q_{t+1} = S_j \mid q_t = S_i, \lambda) \\ &= \sum_{j=1}^{N} \Pr(O_{t+1}, \ldots, O_T \mid q_{t+1} = S_j, q_t = S_i, \lambda) \\ &\quad \cdot \Pr(q_{t+1} = S_j \mid q_t = S_i, \lambda) \\ &= \sum_{j=1}^{N} \Pr(O_{t+1} \mid q_{t+1} = S_j, q_t = S_i, \lambda) \\ &\quad \cdot \Pr(O_{t+2}, \ldots, O_T \mid q_{t+1} = S_j, q_t = S_i, \lambda) \\ &\quad \cdot \Pr(q_{t+1} = S_j \mid q_t = S_i, \lambda) \\ &= \sum_{j=1}^{N} \Pr(O_{t+1} \mid q_{t+1} = S_j, \lambda) \\ &\quad \cdot \Pr(O_{t+2}, \ldots, O_T \mid q_{t+1} = S_j, \lambda) \\ &\quad \cdot \Pr(q_{t+1} = S_j \mid q_t = S_i, \lambda) \\ &= \sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j). \end{aligned} \tag{15}$$

□

(3) End:

$$\Pr(O \mid \lambda) = \sum_{i=1}^{N} \beta_1(i). \tag{16}$$

From these forward and backward variables, we have

$$\Pr(O \mid \lambda) = \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j), \quad 1 \le t \le T-1. \tag{17}$$

We therefore obtain $\lambda^* = \arg\max_{\lambda} \Pr(O \mid \lambda)$, which is a functional extremum problem. Because the length of the training sequence is limited, there is no optimal method for estimating $\lambda$. Therefore, recursion is used to achieve a maximum local value for $\Pr(O \mid \lambda)$, thereby obtaining the model parameter set $\lambda = (A, B, \Pi)$.

We define $\xi_t(i, j)$ as the probability of being in $S_i$ at time $t$ and in $S_j$ at time $t+1$, given the whole observation sequence $O$ and the model parameter set $\lambda$. Then

$$\begin{aligned}\xi_t(i,j) &\triangleq \Pr\left(q_t = S_i, q_{t+1} = S_j \mid O, \lambda\right) \\ &= \frac{\alpha_t(i)\, b_j(O_{t+1})\, \beta_{t+1}(j)\, a_{ij}}{\sum_{k=1}^{N}\sum_{l=1}^{N} \alpha_t(k)\, a_{kl} b_l(O_{t+1})\, \beta_{t+1}(l)}.\end{aligned} \tag{18}$$

*Proof*

$$\begin{aligned}\xi_t(i,j) &\triangleq \Pr\left(q_t = S_i, q_{t+1} = S_j \mid O, \lambda\right) \\ &= \Pr\left(O \mid q_t = S_i, q_{t+1} = S_j, \lambda\right) \\ &\quad \cdot \frac{\Pr\left(q_t = S_i, q_{t+1} = S_j \mid \lambda\right)}{\Pr(O \mid \lambda)} \\ &= \Pr\left(O \mid q_t = S_i, q_{t+1} = S_j, \lambda\right) \\ &\quad \cdot \Pr\left(q_{t+1} = S_j \mid q_t = S_i, \lambda\right) \\ &\quad \cdot \frac{\Pr(q_t \mid \lambda)}{\Pr(O \mid \lambda)} \\ &= \Pr(O_1, \ldots, O_t \mid q_t = S_i, \lambda) \\ &\quad \cdot \Pr\left(O_{t+1} \mid q_{t+1} = S_j, \lambda\right) \\ &\quad \cdot \Pr\left(O_{t+2}, \ldots, O_T \mid q_{t+1} = S_j, \lambda\right) \\ &\quad \cdot \frac{a_{ij} \Pr\left(q_t = S_i \mid \lambda\right)}{\Pr(O \mid \lambda)} \\ &= \Pr(O_1, \ldots, O_t, q_t = S_i \mid \lambda) \Pr\left(O_{t+1} \mid q_{t+1} = S_j, \lambda\right) \\ &\quad \cdot \frac{\Pr\left(O_{t+2}, \ldots, O_T \mid q_{t+1} = S_j, \lambda\right) a_{ij}}{\Pr(O \mid \lambda)} \\ &= \frac{\alpha_t(i)\, b_j(O_{t+1})\, \beta_{t+1}(j)\, a_{ij}}{\Pr(O \mid \lambda)} \\ &= \frac{\alpha_t(i)\, b_j(O_{t+1})\, \beta_{t+1}(j)\, a_{ij}}{\sum_{k=1}^{N}\sum_{l=1}^{N} \Pr\left(q_t = S_k, q_{t+1} = S_l, O \mid \lambda\right)} \\ &= \frac{\alpha_t(i)\, b_j(O_{t+1})\, \beta_{t+1}(j)\, a_{ij}}{\sum_{k=1}^{N}\sum_{l=1}^{N} \alpha_t(k)\, a_{kl} b_l(O_{t+1})\, \beta_{t+1}(l)}.\end{aligned} \tag{19}$$

Thus, the probability of being in $s_i$ at time $t$ is therefore given by

$$\gamma_t(i) = \Pr(O, q_t = S_i \mid \lambda) = \sum_{j=1}^{N} \xi_t(i,j), \tag{20}$$

where $\sum_{t=1}^{T-1} \xi_t(i, j)$ denotes the mathematical expectation of transition from state $S_i$. Therefore, the probability of transition from $S_i$ to $S_j$ is given by

$$\widehat{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}. \tag{21}$$

In state $S_j$, the probability of observing $v_m$ is given by

$$\widehat{b}_j(m) = \frac{\sum_{t=1}^{T} \gamma_t(j)\, 1_{(O_t = v_m)}}{\sum_{t=1}^{T} \gamma_t(j)}. \tag{22}$$

We therefore obtain a new model $\widehat{\lambda} = (\widehat{A}, \widehat{B}, \widehat{\Pi})$, where $\widehat{A} = [\widehat{a}_{ij}]$, $\widehat{B} = [b_j(m)]$, $\widehat{\Pi} = [\widehat{\pi}_i]$, and $\widehat{\pi}_i = \gamma_1(i)$. We can prove $\Pr(O \mid \widehat{\lambda}) \geq \Pr(O \mid \lambda)$. The training process is repeated, and the model parameters are adjusted gradually until $\Pr(O \mid \widehat{\lambda})$ converges. Finally, we obtain the prediction model $\widehat{\lambda}$, which is used to assess the QoS-satisfied capability for compositions of cloud-service components. □

## 4. Stimulation Experiment and Analysis

To check the feasibility of the proposed QoS-satisfied prediction model, we constructed a simulation experiment based on a cloud-computing experimental platform at our university. The simulation system included more than 100 cloud-service components, such as adaptable components and servers, and the open-source Eucalyptus system was adopted.

Using our simulation system, we designed three cloud storage services whose QoS were different to check the prediction model. First, we trained the prediction model by adjusting the model parameters gradually, using a machine learning algorithm based on a support vector machine (SVM). In the SVM, the penalty factor $c$, the kernel function parameter $g$, and the boundary range $p$ were set to 1024, 1024, and 0.0097, respectively. After obtaining the model parameter set $\lambda$, we achieved the prediction model $\widehat{\lambda}$. We then made predictions for the three cloud storage services by using the prediction model and acquiring their simulated results. The simulation experiment was run 10 times to produce average results. The QoS-satisfied capabilities for the prediction values and observation values of the three cloud storage services ($A$, $B$, and $C$) are shown in Figures 5 and 6, respectively. The comparison between actual and observed values for service $B$ is shown in Table 1, where the selected observation times from 74 onwards show the acquisition of a stable observed effect.

Note that the error is small (no more than 0.01%). This is because there is a stable network environment, without any other factors' effects in our simulated system. In reality, a larger error would be expected. However, it should be able to satisfy the user's QoS requirements in most cases.

## 5. Conclusions

Cloud computing refers to the use of shared computing resources and is an alternative to having local servers handle

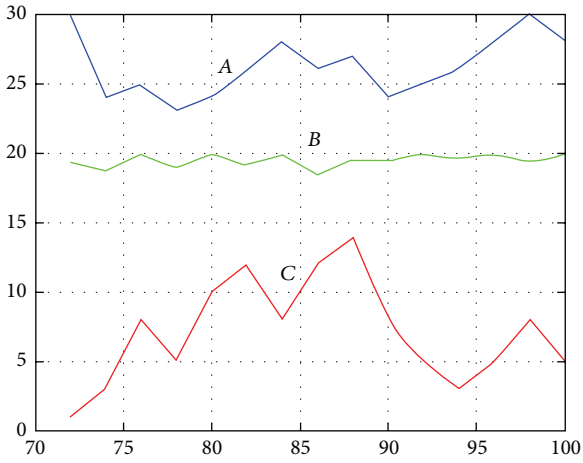



Figure 5: Prediction values.

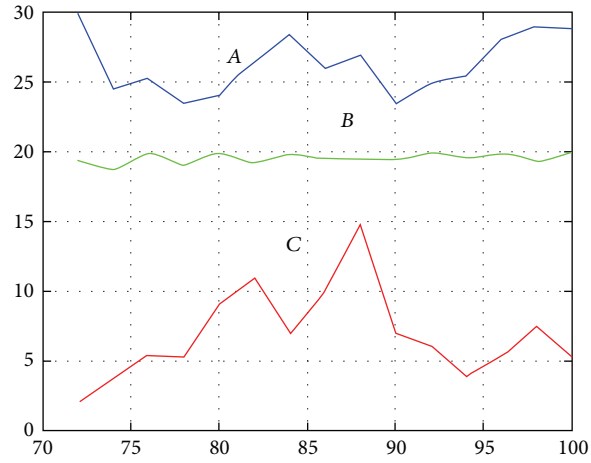



Figure 6: Observation values.

Table 1: Comparison of prediction results.

| Time | Actual value | Observation value | Error (%) |
|---|---|---|---|
| 74 | 19.4 | 19.3995 | 0.003 |
| 76 | 18.7 | 18.7011 | 0.006 |
| 78 | 20 | 19.9994 | 0.003 |
| 80 | 19 | 19.001 | 0.005 |
| 82 | 20 | 19.9993 | 0.004 |
| 84 | 19.2 | 19.2008 | 0.004 |
| 86 | 19.9 | 19.8991 | 0.005 |
| 88 | 18.5 | 18.5011 | 0.006 |
| 90 | 19.5 | 19.4991 | 0.005 |
| 92 | 19.4 | 19.3986 | 0.007 |
| 94 | 20 | 19.999 | 0.005 |
| 96 | 19.5 | 19.4993 | 0.004 |
| 98 | 20 | 19.9991 | 0.005 |
| 100 | 19.3 | 19.2987 | 0.007 |

user applications. A match between cloud-service components and users' QoS is therefore very important if user experience is the basis for promoting the development of cloud computing. If no single cloud service satisfies the user's QoS perfectly, a composition of multiple cloud-service components should be considered, which should also include predicting the QoS satisfaction of the composite service. This paper presents a QoS-satisfied prediction model based on an HMM to assess the QoS-satisfied capability for compositions of components. We discuss the proposed model in detail and prove aspects of the model. Simulation results show that our model can achieve high prediction accuracies. In future work, we will introduce heterogeneous cloud services into the system and develop the prediction model to make it a better fit with real applications.

## Acknowledgments

The authors would like to thank the reviewers for their detailed reviews and constructive comments, which have helped improve the quality of this paper. This work was supported in part by National Natural Science Foundation of China (NSFC) under Grant no. 61003035 and no. U1204614, in part by the Plan for Scientific Innovation Talent of Henan Province under Grant no. 124100510006, and in part by the Science and Technology Development Programs of Henan Province under Grant no. 112102210187, in part by the Youth Foundation of Henan University of Science and Technology under Grant no. 2011QN51.

## References

[1] D. Nurmi, R. Wolski, C. Grzegorczyk et al., "The eucalyptus open-source cloud-computing system," in *Proceedings of the 9th IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID '09)*, pp. 124–131, Shanghai, China, May 2009.

[2] M. Armbrust, A. Fox, R. Griffith et al., "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[3] L. Wang, R. Ranjan, J. Chen, and B. Benatallah, *Cloud Computing: Methodology, Systems, and Applications*, CRC Press, 2011.

[4] M. Zhang, R. Ranjan, A. Haller, D. Georgakopoulos, and P. Strazdins, "Investigating decision support techniques for automating Cloud service selection," in *Proceedings of the 4th International Conference on Cloud Computing Technology and Science*, pp. 759–764, Taipei, China, December 2012.

[5] S. Wang, Q. Sun, H. Zou, and F. Yang, "Particle swarm optimization with skyline operator for fast cloud-based web service composition," *Mobile Networks and Applications*, vol. 18, no. 1, pp. 116–121, 2013.

[6] B. Javadi, P. Thulasiraman, and R. Buyya, "Enhancing performance of failure-prone clusters by adaptive provisioning of cloud resources," *The Journal of Supercomputing*, vol. 63, no. 2, pp. 467–489, 2013.

[7] K. Benouaret, D. Benslimane, A. Hadjali, and M. Barhamgi, "Top-k web service compositions using fuzzy dominance relationship," in *Proceedings of the IEEE International Conference on Services Computing (SCC '11)*, pp. 144–151, Washington, England, July 2011.

[8] W.-N. Liu, B. Liu, and D.-H. Sun, "A conceptual framework for dynamic manufacturing resource service composition and

optimization in service-oriented networked manufacturing," in *Proceedings of the International Conference on Cloud and Service Computing (CSC '11)*, pp. 118–125, Hong Kong, China, December 2011.

[9] J. Huang, Y. Liu, R. Yu, Q. Duan, and Y. Tanaka, "Modeling and algorithms for QoS-aware service composition in virtualization-based cloud computing," *IEICE TRANSACTIONS on Communications*, vol. 96, no. 1, pp. 10–19, 2013.

[10] C. J. Huang, C. T. Guan, H. M. Chen et al., "An adaptive resource management scheme in cloud computing," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 1, pp. 382–389, 2013.

[11] S. Di and C. Wang, "Dynamically optimizing multi-attribute execution efficiency on self-organizing cloud," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 3, pp. 464–478, 2013.

[12] Y. Jiang, C. Perng, T. Li, and R. Chang, "Self-adaptive cloud capacity planning," in *Proceedings of the IEEE Ninth International Conference on Services Computing*, pp. 73–80, Honolulu, Hi, USA, June 2012.

[13] H. Zhang and P. Pathirana, "Uplink power control via adaptive hidden Markov model based pathloss estimation," *IEEE Transactions on Mobile Computing*, vol. 12, no. 4, pp. 657–665, 2013.

[14] H. Choi, D. Fermin, A. I. Nesvizhskii, D. Ghosh, and Z. S. Qin, "Sparsely correlated hidden Markov models with application to genome-wide location studies," *Bioinformatics*, vol. 29, no. 5, pp. 533–541, 2013.

[15] Z. I. Botev, P. L'Ecuyer, and B. Tuffin, "Markov chain importance sampling with applications to rare event probability estimation," *Statistics and Computing*, vol. 23, no. 2, pp. 271–285, 2013.

[16] Y. Xie, J. Hu, S. Tang, and X. Huang, "A forward-backward algorithm for nested hidden semi-Markov model and application to network traffic," *The Computer Journal*, vol. 56, no. 2, pp. 229–238, 2013.

[17] R. Zheng, M. Zhang, Q. Wu, S. Sun, and J. Pu, "Analysis and application of bio-inspired multi-net security model," *International Journal of Information Security*, vol. 9, no. 1, pp. 1–17, 2010.

[18] R. Zheng, Q. Wu, M. Zhang, G. Li, J. Pu, and H. Wang, "A self-optimization mechanism of system service performance based on autonomic computing," *Computer Research and Development*, vol. 48, no. 9, pp. 1676–1684, 2011.

[19] M.-C. Zhang, Q.-T. Wu, R.-J. Zheng, W.-Y. Wei, and G.-F. Li, "Research on grade optimization self-tuning method for system dependability based on autonomic computing," *Journal of Computers*, vol. 7, no. 2, pp. 333–340, 2012.