

Research Article

Research on Community Competition and Adaptive Genetic Algorithm for Automatic Generation of Tang Poetry

Wujian Yang, Yining Cheng, Jie He, Wenqiong Hu, and Xiaojia Lin

Department of Information Management, Zhejiang University City College, Hangzhou, Zhejiang 310015, China

Correspondence should be addressed to Jie He; hej@zucc.edu.cn

Received 14 December 2015; Revised 8 March 2016; Accepted 21 March 2016

Academic Editor: Reza Jazar

Copyright © 2016 Wujian Yang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

As there are many researches about traditional Tang poetry, among which automatically generated Tang poetry has aroused great concern in recent years. This study presents a community-based competition and adaptive genetic algorithm for automatically generating Tang poetry. The improved algorithm with community-based competition that has been added aims to maintain the diversity of genes during evolution; meanwhile, the adaptation means that the probabilities of crossover and mutation are varied from the fitness values of the Tang poetry to prevent premature convergence and generate better poems more quickly. According to the analysis of experimental results, it has been found that the improved algorithm is superior to the conventional method.

1. Introduction

Tang poetry as one of the most precious cultural heritages in China for thousands of years has a great value to research. There are a large number of scholars researching on it in different ways [1]. For most modern people, composing a classic poem is still of great challenge [2]. In recent years, artificial intelligence plays an important role in our daily lives and it has become a hot topic in every field [3]. Computer has been regarded as an artist in our eyes, and, with the help of computer programs, autogenerating system can compose several beautiful Tang poems [4]. Thus, such system is of great value to be realized [5]. In particular, Dr. Zhou et al. have designed a system which can compose high quality TLSD (The Lyrics of the Song Dynasty) with evolution strategy [6]. However, the achievement of automatically generating Tang poetry is not accomplished. In order to write a suitable algorithm for generating Tang poetry, grasping the characteristics of Tang poetry is important thing to do first, when some factors are taken into account [7]. For instance, it has several styles, such as JUEJU (a poem of four lines) and LVSHI which has a strict tonal pattern and rhyme scheme.

Genetic algorithms (GA) employ a random search for locating the globally optimal solution [8, 9]. It has been researching and improving since GA was put forward by

Dr. Holland in 1975 [3]. Now, GA has formed well-established theoretical system after the continued research for nearly 40 years [10–12]. It is used in many fields, such as production scheduling, combinatorial optimization, automatic control, robotics, and image processing [13]. GA is the core technology of the artificial intelligence and, in the latest study, adaptive genetic algorithm (AGA) [14] has become the mainstream: the crossover and mutation probability are influenced by the fitness function; it can effectively prevent the problem of premature convergence.

Therefore, the application of GA in autogenerating system for Tang poetry is reasonable and feasible. However, there are problems still awaiting resolution. For example,

- (1) it takes too long to compose a poem;
- (2) the selection of the parameters mostly comes by experience rather than a lot of verification, such as crossover and mutation probability which are not flexible;
- (3) premature problem that a population for poetry converges too early results in being suboptimal.

It is easily trapped in the local optimum and premature convergence appeared. To solve these problems, this study has developed an Automatic Tang Poems Generating

System (ATPGS); the system ATPGS has the following two advantages: First, the community competition will classify the group when initial population was formed. It can not only effectively shorten the time but also avoid the problem of prematureness. Second, AGA will make crossover and mutation probabilities become more reasonable in this system. Furthermore, a coding scheme is proposed based on level and oblique tones, elitism algorithm, and roulette wheel selection.

Above all, this work represents progress in the field of autogenerating system of Chinese poetry. The rest of the paper is organized as follows: Section 2 shows details of related work; Section 3 describes the implementation details of the system and the optimization method for algorithm and presents the main framework of the algorithm; Section 4 analyzes the improved performance of the verification system; Section 5 provides the conclusions.

2. Related Work

Tang poetry is regarded as a pearl in the long history of China; there is no lack of formal researches on computer-assisted poetry's autogenerating system. The autogenerating system of Tang poetry is a complex task, as it involves several levels of language like phonetics, lexical choice, syntax, and semantics and usually requires a considerable amount of input knowledge [1, 2]. This study simply describes the process of autogenerating system of Tang poetry generation by genetic algorithms. The research has not detailed its branch study, such as thesaurus generation, semantic relatedness, and the study of poetry participle.

2.1. About Semantic Relatedness. The study of words semantic relatedness can be viewed in various linguistic resources, such as Wikipedia or large scale text corpora for methods like Latent Semantic Analysis (LSA). Computing the semantic relatedness degree of words in Tang poetry is key to many applications such as searching, clustering, and automatically generation of poem. Aiming to increase computing efficiency and accuracy of semantic relatedness, the process of LSA was improved, which uses representation of words semantic by "words-by-poetry categories" instead of "words-by-poems." Meanwhile, the experiment which obtained segmentation words from more than 40000 poems is designed and the study computes relatedness by cosine value calculated from decomposed cooccurrence matrix with Singular Value Decomposition (SVD) method.

2.2. About Poetry Participle. In the poetry participle the ATPGS adopts the Nash Equilibrium Theorem which is of another area's research. The research results provide a more appropriate thesaurus, and when the composing of Tang poetry is finished, it will check the quality of Tang poetry automatically.

2.3. About Adaptive Genetic Algorithm. After research on a variety of adaptive genetic algorithm consideration, in this study the most suitable method for the characteristics of Tang poetry are selected [15, 16].

The main formulas are as follows:

$$pc = \begin{cases} pc_{\max} - \frac{(pc_{\max} - pc_{\min})(f - f_{\text{avg}})}{f_{\max} - f_{\text{avg}}}, & f \geq f_{\text{avg}} \\ pc_{\max}, & f < f_{\text{avg}} \end{cases} \quad (1)$$

The parameters are as follows: pc is the crossover probability; pc_{\max} is the maximum crossover probability; pc_{\min} is the minimum crossover probability; f_{\max} is the maximum fitness value; f_{avg} is the average fitness value; f is the fitness value for the parameter. Significant findings are as follows: when adaptive degree is greater than the average value, the greater the adaptive degree the smaller the probability of the corresponding crossover. When adaptive degree is less than the average value, it is with the greater crossover probability. Consider

$$pm = \begin{cases} pm_{\max} - \frac{(pm_{\max} - pm_{\min})(f_{\max} - f)}{f_{\max} - f_{\text{avg}}}, & f \geq f_{\text{avg}} \\ pm_{\max}, & f < f_{\text{avg}} \end{cases} \quad (2)$$

The parameters are as follows: pm is the mutation probability; pm_{\max} is the maximum mutation probability; pm_{\min} is the minimum mutation probability; f_{\max} is the maximum fitness value; f_{avg} is the average fitness value; f is the fitness value for the parameter. When it is greater than the average value, the greater adaptive degree is, the bigger the mutation probability is. When the adaptation is less than the average, the mutation probability is less.

3. Concrete Steps to Achieve

3.1. Coding Scheme. Choosing a suitable coding scheme in this system is very important. Currently, there is no quite accurate and efficient encoding scheme for the autogenerating of Tang poetry. Therefore, according to the previous literature of Professor Zhou, we proposed a coding scheme for the autogenerating system [17, 18], and, based on the patterns of level and oblique tones of Tang poetry, we convert it into the binary; "0" stands for level tones, "1" stands for oblique tones, "*" stands for "both are acceptable."

3.2. Generation of Initial Population. According to the preexisting verses in Tang poetry, the words are randomly chosen to generate sentences into standby database syntax. Main steps are the following:

- (1) Find related words according to the keywords to make a candidate space.
- (2) In accordance with the previous word in the verses to find other words whose correlation is less large, add them into the candidate space in turn. Repeat until the candidate space is large enough.
- (3) Add the words in their appropriate place randomly.

- (4) Find words which satisfied the rhyme in the candidate space and fill in the relevant position.
- (5) Repeat steps (3) and (4) to generate population groups for community competition. It is not only helpful to achieve global optimal, but also helpful to reduce the calculation time.

3.3. Fitness Function. The calculation of fitness function of genetic algorithm is very important, and it has influence on crossover probability and mutation probability, as well as the iteration stop condition of genetic algorithm [4]. This fitness function should try to meet the following conditions:

- (1) being single-valued, continuous, nonnegative, and maximized;
- (2) reasonability and consistency;
- (3) less calculation;
- (4) strong versatility.

The fitness function we set is mainly based on the parameters like level and oblique tones, antithetical parallelism, the prevalence of the words, and so on.

Level and oblique tones are as follows: they are the most basic requirements of judging a good poem. The system will exclude the verse that did not meet the requirement of it as far as the poem is generated. It is only one of the requirements of judging, but not the fitness function parameters.

Antithetical parallelism is as follows: for Tang poetry, JUEJU do not require antithetical parallelism generally, but LVSHI must use the antithetical parallelism. Therefore, it is a task for modern people to write Tang poetry. As antithetical parallelism can improve the mood of the poetry, if the system generates verses of antithetical parallelism, the parameter will be larger than other sentences without it.

The frequency of words is as follows: for a poem with the specific theme, some words appear with a high frequency. Then, if the Tang poetry which generated automatically has words that appear with a high frequency, then this poem will be relatively high fitness value. On the contrary, if the poem is composed of remote words, then the fitness value of this poem will be the low side.

Pattern matching is as follows: build a database of different kinds of verses that belong to specific themes. Each theme stores about 200–300 verses. The result of the pattern matching is related to the relevancy of the verse of the corresponding theme. If the frequency rate is over 70%, then it can be considered that the generated Tang poetry has a fitness value; otherwise, the fitness value will be reduced.

3.4. Selection Operator. Selection operator is also called the copy operation, for selecting the most adaptable individual by individual fitness function value from groups. Generally speaking, the selection will allow individuals with high fitness to reproduce more offspring, while the other individual will produce less offspring or even is eliminated [19]. We adopt elitism algorithm, roulette algorithm, and sons competitive in selection method. In the standard genetic algorithm, the next generation of each individual in the population is birth

from the crossover and mutation of individual in the mating pool. However, in real life, each group includes individuals from the older generation and the new generation. Moreover, all the adaptable new individuals will also make their parent individual genes eliminated, making the optimization process slower. This competition mechanism makes excellent individual preserved and helps to keep the superior genes and to improve the convergence speed. If the fitness of parents and their offspring is equal, then the offspring will have the priority to get into the next generation.

3.5. Crossover and Mutation Operator. In this paper we use the adaptive genetic algorithm. The crossover and mutation probability are changing according to the fitness. ATPGS uses the heuristic crossover to make the whole poem as a swap space to ensure the preservation of the superior individual and the inferior individuals will be eliminated [15]. The greater the genetic algorithm crossover, the faster the speed of the new individual generated. However, when the crossover probability is too large, the possibility that genetic model is destroyed is greater. It will lead to the individual structures with high quality destroyed soon. However, if the crossover probability is too small, it will make the search process slower and even stagnant. When the population of each individual's fitness tends to converge to local optimum, the crossover probability will increase [14, 19].

The mutation probability is based on the principles of genetic variation in the biogenetics to change the value of a gene or genes of an individual coding string according to a certain probability. If the mutation rate is too small, it is unlikely to produce a new individual structure; if the mutation probability value is too large, then the genetic algorithm will turn to a purely random search algorithm. When the fitness of each individual of the population tends to converge to local optimum, the mutation probability will increase, and when the group fitness scattered, the mutation probability will decrease [14, 19].

3.6. Community Competition. Community competition is essentially a way of parallel computing; this method can effectively solve the “premature convergence” issue which appears in the genetic algorithm. It will convert local optimum into global optimum, and, because of parallel computing, you can save a lot of computing time. Its main operation is to apply all the steps mentioned above to each community, so that 10 communities will compose a best poem of their own. After competing one more time, the poem of highest fitness value will survive.

3.7. Implementation Framework. See Figure 1.

4. Experimental Results

In this section we provide the experimental results of community competition and adaptive genetic algorithm. First, experiments are designed to find reasonable settings for the presented implementation framework in the previous sections by tuning parameters such as mutation probability,

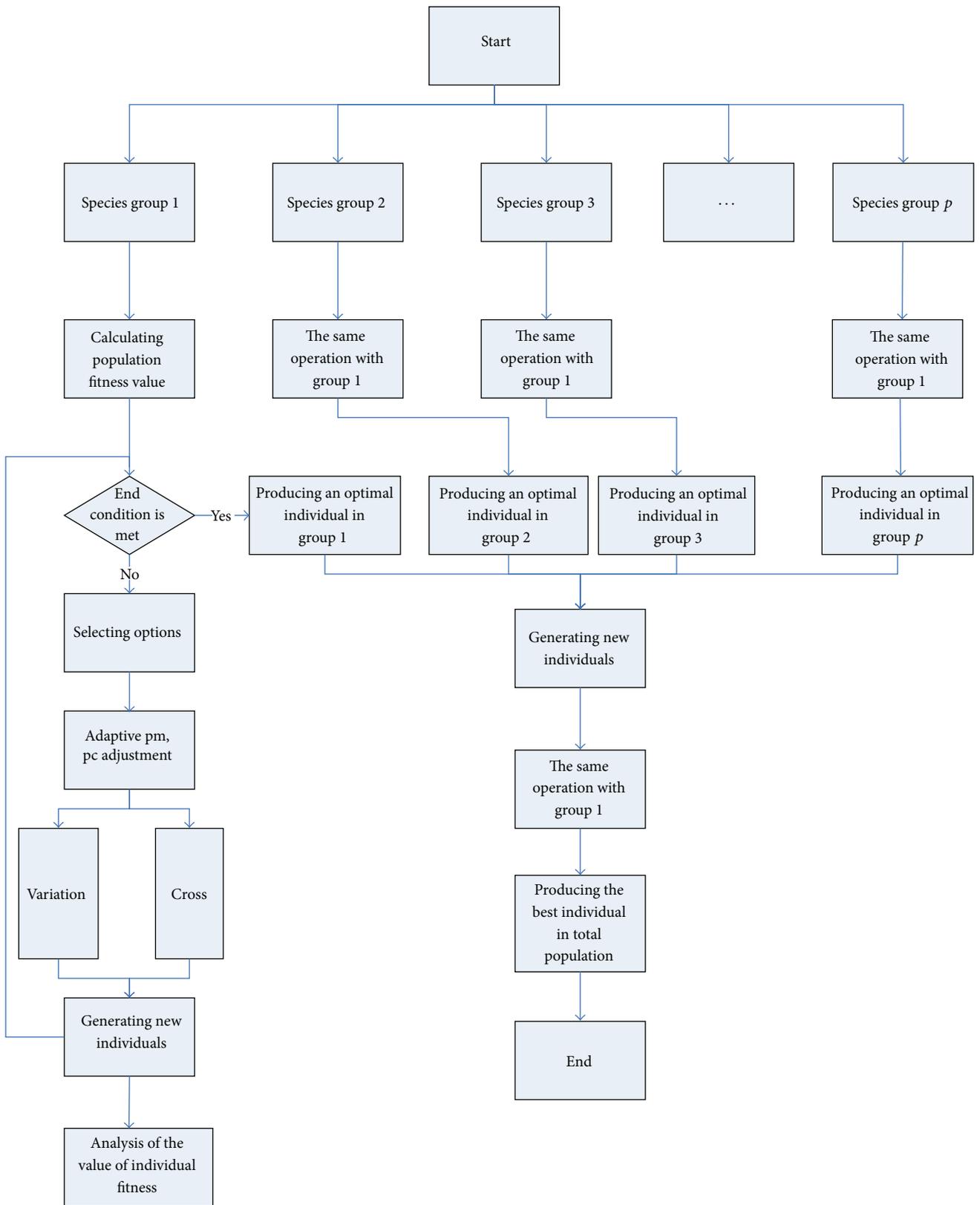


FIGURE 1: Framework of the improved genetic algorithm.

TABLE 1: Experiments based on crossover probability.

Crossover probability	Score	Time
0.5	1830	33.0007029
0.6	1867	33.07072548
0.7	1999	33.64421489
0.8	1946	33.15097804
0.9	1909	33.20562783
1	1520	33.01007189

crossover probability, and elitist probability [20, 21]. Elitist probability is adopted to allow the best organism from the current generation to carry over into the next generation which can guarantee that the solution quality obtained by the GA will not decrease from one generation to the next. Second, experiments are designed to demonstrate how the value of evolution time can impact the results. The experiments set the crossover probability and the mutation probability to be constant, and the value of evolution time is changeable. For the purpose of accuracy of report, we set different constraint condition. The experimental results show that when the evolution time is more than 40, the composing time exceeds 60 s. Considering it is unfriendly to spend more than 60 s producing a poem, so evolution time more than 40 is no longer taken into account in the experiments. Finally, we compare the effectiveness of the four genetic algorithms including noncommunity competition and nonadaptive genetic algorithm (NCNA), noncompetition competition and adaptive genetic algorithm (NCA), community competition and nonadaptive genetic algorithm (CNA), and community competition and adaptive genetic algorithm (CA).

4.1. Tuning Crossover Probability and Elitist Probability to Find Reasonable Settings. It is worth tuning parameters such as the mutation probability, crossover probability, and elitist probability to find reasonable settings.

To compare with experimental data of crossover probability based on the community competition and adaptive genetic algorithm, we set the probability of elitist to 0.01 and evolution times to 20 and change the crossover probability. Statistics are shown in Table 1 in a certain range.

Through the reorganization of data analysis, we can get more intuitive icons. Under the same conditions, when the value of crossover probability is 0.7, 0.8, and 0.9, the generations of the poems are likely to reach the high score.

Based on Table 1, 0.7, 0.8, and 0.9 are chosen as crossover probabilities to find whether the elitist probability will produce a changeable result. From the comparison of Table 2, these figures are changing with some clear trend. With more careful comparisons, when the elitist probability is 0.01 and evolution time is 40, the scores are almost always higher than others. Results indicate that 0.01 is the best elitist probability to compose a high quality poem when evolution has enough times.

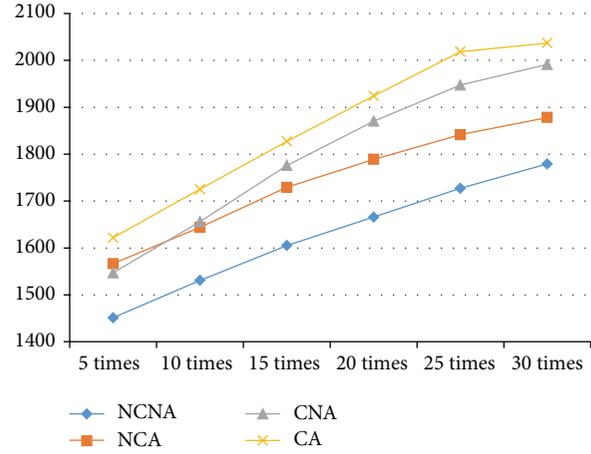


FIGURE 2: The average score of different evolution types.

4.2. Influence of Different Evolution Times. We find that the times of evolution can impact the result, so we set the crossover probability and the mutation probability to be constant to demonstrate the changing value of different evolution times. For the purpose of accuracy of report, we set different constraint condition. The data of score are shown in Table 3.

These data show that the quality of the generated poetry will get better with the increase of the evolution times. In this system, score over 2000 points can be accepted as a high quality poem. Then we can conclude that when the value of evolutionary time is about 30, the convergence of genetic algorithm will be more reasonable.

4.3. Comparison of Different Evolution Types. In this part, we will compare the effectiveness of the four evolution types. When the mutation and crossover probability are in the same circumstances, we need to discuss the connection between evolution times and the quality of autogenerating poems. The scores of the poems are listed as follows.

Table 4 shows that the quality of the poems generated by community competition and adaptive genetic algorithm is better than other three genetic algorithms.

Figure 2 plots the concise result of different evolution types by average score of three kinds of parameters. CA genetic algorithm can not only generate good poem quickly but also improve poem quality constantly with the increasing evolution times. It can be also seen from Figure 2 that the adaptive method plays an important role in the early stage, but with the increase in the number of evolution times, the role is becoming more and more limited.

5. Conclusions and Future Work

Poetry composition is a difficult task in the field of language generation. The highlight in ATPGS is the community competition, adaptive crossover, and mutation operation. However, with the utilizing of abundant lexical resources, making autogenerating system of Tang poetry comes true.

TABLE 2: Experiments based on elitist probability.

Crossover probability	Elitist probability	Evolution for 20 times		Evolution for 30 times		Evolution for 40 times	
		Score	Time	Score	Time	Score	Time
0.7	0.01	1874	33.644	2074	49.215	2189	64.669
	0.02	1997	33.551	2172	49.024	2288	63.690
	0.03	1956	33.523	2028	48.440	2225	64.114
0.8	0.01	1946	33.160	2071	48.198	2207	63.627
	0.02	1964	33.539	2069	48.942	2174	62.704
	0.03	1991	33.273	2150	48.013	2153	63.113
0.9	0.01	1861	33.206	1974	48.280	2051	64.430
	0.02	1869	32.956	1980	47.669	2051	63.488
	0.03	1858	33.054	1971	47.288	2043	62.281

TABLE 3: Experiments based on different evolution times.

Evolution times	Mutation probability/crossover probability/elitist probability (0.29/0.7/0.01)		Mutation probability/crossover probability/elitist probability (0.19/0.8/0.01)		Mutation probability/crossover probability/elitist probability (0.09/0.9/0.01)	
	Score	Time	Score	Time	Score	Time
10	1704	18.41302945	1735	18.20988195	1705	18.31040418
15	1786	25.9637337	1820	27.04685529	1771	25.64028699
20	1874	33.64421489	1946	33.15097804	1861	33.20562783
25	1922	41.42040713	2049	40.7430073	1924	40.72415348
30	2074	49.21586105	2071	48.19742672	1974	48.27986591
35	2122	56.38273411	2135	56.45745017	2015	55.53458099
40	2189	64.66857841	2207	63.62565136	2047	63.429246

TABLE 4: Experiments based on mutation probability and elitist probability.

Evolution types	Evolution times	Mutation probability/crossover probability/elitist probability (0.1/0.85/0.05)	Mutation probability/crossover probability/elitist probability (0.19/0.8/0.01)	Mutation probability/crossover probability/elitist probability (0.18/0.8/0.02)
Noncommunity competition and nonadaptive (NCNA)	5	1442	1451	1461
	10	1509	1521	1563
	15	1576	1603	1637
	20	1659	1668	1671
	25	1708	1749	1725
	30	1739	1811	1788
noncommunity competition and adaptive (NCA)	5	1561	1572	1567
	10	1637	1646	1649
	15	1695	1749	1744
	20	1750	1812	1805
	25	1799	1867	1859
	30	1831	1905	1899
Community competition and nonadaptive (CNA)	5	1544	1548	1549
	10	1646	1660	1662
	15	1768	1778	1782
	20	1835	1886	1890
	25	1910	1969	1963
	30	1934	2021	2019
Community competition and adaptive (CA)	5	1609	1626	1631
	10	1699	1735	1742
	15	1796	1829	1857
	20	1862	1946	1964
	25	1954	2044	2053
	30	1979	2071	2069

Poems composed by our system are with high quality no matter in rhyme or semantics. Furthermore, there will be the antithesis of the sentence which is much better than the online website of Tang poetry's generating. We can conclude that community-based and self-adaptive genetic algorithm to compete in the autogenerating system of Tang Poem has a better effect.

In the future, researches would focus on quantities style evaluation criteria so that computers can automatically label poem to their distinctive bold-and-unrestrained or graceful-and-restrained values. We plan to incorporate more criteria into the constrained summarization framework, which should consider aspects like the structure, novelty, and semantics of poems.

Competing Interests

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

Acknowledgments

The authors would like to thank the scholars who ever inspired and helped us.

References

- [1] R. Yan, H. Jiang, M. Lapata, S.-D. Lin, X. Lv, and X. Li, "I, poet: automatic Chinese poetry composition through a generative summarization framework under constrained optimization," in *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI '13)*, pp. 2197–2203, AAAI Press, August 2013.
- [2] F. J. Lo, Y. P. Lee, and W. C. Tsao, "The format auto-checking and database indexing teaching system of Chinese poetry and lyrics," *Journal of Chinese Information Processing*, vol. 13, no. 1, pp. 35–42, 1999 (Chinese).
- [3] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, University of Michigan Press, 1975.
- [4] H. G. Oliveira, "PoeTryMe: a versatile platform for poetry generation," *Computational Creativity, Concept Invention, and General Intelligence*, vol. 1, article 21, 2012.
- [5] R. W. Bailey, "Computer-assisted poetry: the writing machine is for everybody," in *Computers in the Humanities*, Mitchell JL, Ed., pp. 283–295, Edinburgh University Press, Edinburgh, Scotland, 1974.
- [6] J. S. Su, C. L. Zhou, and Y. H. Li, "The establishment of the annotated corpus of Song dynasty poetry based on the statistical word extraction and rules and forms," *Journal of Chinese Information Processing*, vol. 21, no. 2, pp. 52–57, 2007.
- [7] J. F. Hu, *The lexicon meaning analysis-based computer aided research work of Chinese ancient poems [Ph.D. thesis]*, Peking University, Beijing, China, 2001 (Chinese).
- [8] D. Whitley, "A genetic algorithm tutorial," *Statistics and Computing*, vol. 4, no. 2, pp. 65–85, 1994.
- [9] E. Onieva, J. E. Naranjo, V. Milanés, J. Alonso, R. García, and J. Pérez, "Automatic lateral control for unmanned vehicles via genetic algorithms," *Applied Soft Computing Journal*, vol. 11, no. 1, pp. 1303–1309, 2011.
- [10] T. V. Mathew, *Genetic Algorithm*, Indian Institute of Technology Bombay, Mumbai, India, 2012.
- [11] M. Mitchell, *An Introduction to Genetic Algorithms (Complex Adaptive Systems)*, 1998.
- [12] H. Mühlenbein, "Parallel genetic algorithms in combinatorial optimization," in *Computer Science and Operations Research*, Pergamon Press, New York, NY, USA, 1992.
- [13] M. D. Vose, *The Simple Genetic Algorithm: Foundations and Theory*, MIT Press, Boston, Mass, USA, 1998.
- [14] J. Liu, Z. Yu, and D. Ma, "An adaptive fuzzy min-max neural network classifier based on principle component analysis and adaptive genetic algorithm," *Mathematical Problems in Engineering*, vol. 2012, Article ID 483535, 21 pages, 2012.
- [15] C. Pan, S. Cao, K. Zheng, and Z. He, "Optimizing appropriate two probable Value method by adaptive genetic algorithm," in *Proceedings of the International Conference on Computational Problem-Solving (ICCP '12)*, pp. 171–173, IEEE, Leshan, China, October 2012.
- [16] K. Deb and H. G. Beyer, "Self-adaptive genetic algorithms with simulated binary crossover," *Evolutionary Computation*, vol. 9, no. 2, pp. 197–221, 2001.
- [17] L. Jiang and M. Zhou, "Generating Chinese couplets using a statistical MT approach," in *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pp. 377–384, Association for Computational Linguistics, 2008.
- [18] Y. B. Liu, S. W. Yu, and Q. S. Sun, "The implementation of a computer-aided environment for ancient poetry researches," *Journal of Chinese Information Processing*, vol. 11, no. 1, pp. 27–35, 1996 (Chinese).
- [19] T. T. Liang, P. K. Deng, and B. J. Lin, "Research of improved adaptive genetic algorithm used in evolvable hardware," *Computer Engineering and Design*, vol. 33, no. 2, pp. 711–717, 2012.
- [20] S. Baluja and R. Caruana, "Removing the genetics from the standard genetic algorithm," in *Machine Learning: Proceedings of the Twelfth International Conference on Machine Learning, Tahoe City, California, July 9–12, 1995*, pp. 38–46, Morgan Kaufmann Publishers, 1995.
- [21] Q. Long, C. Wu, X. Wang, L. Jiang, and J. Li, "A multiobjective genetic algorithm based on a discrete selection procedure," *Mathematical Problems in Engineering*, vol. 2015, Article ID 349781, 17 pages, 2015.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

