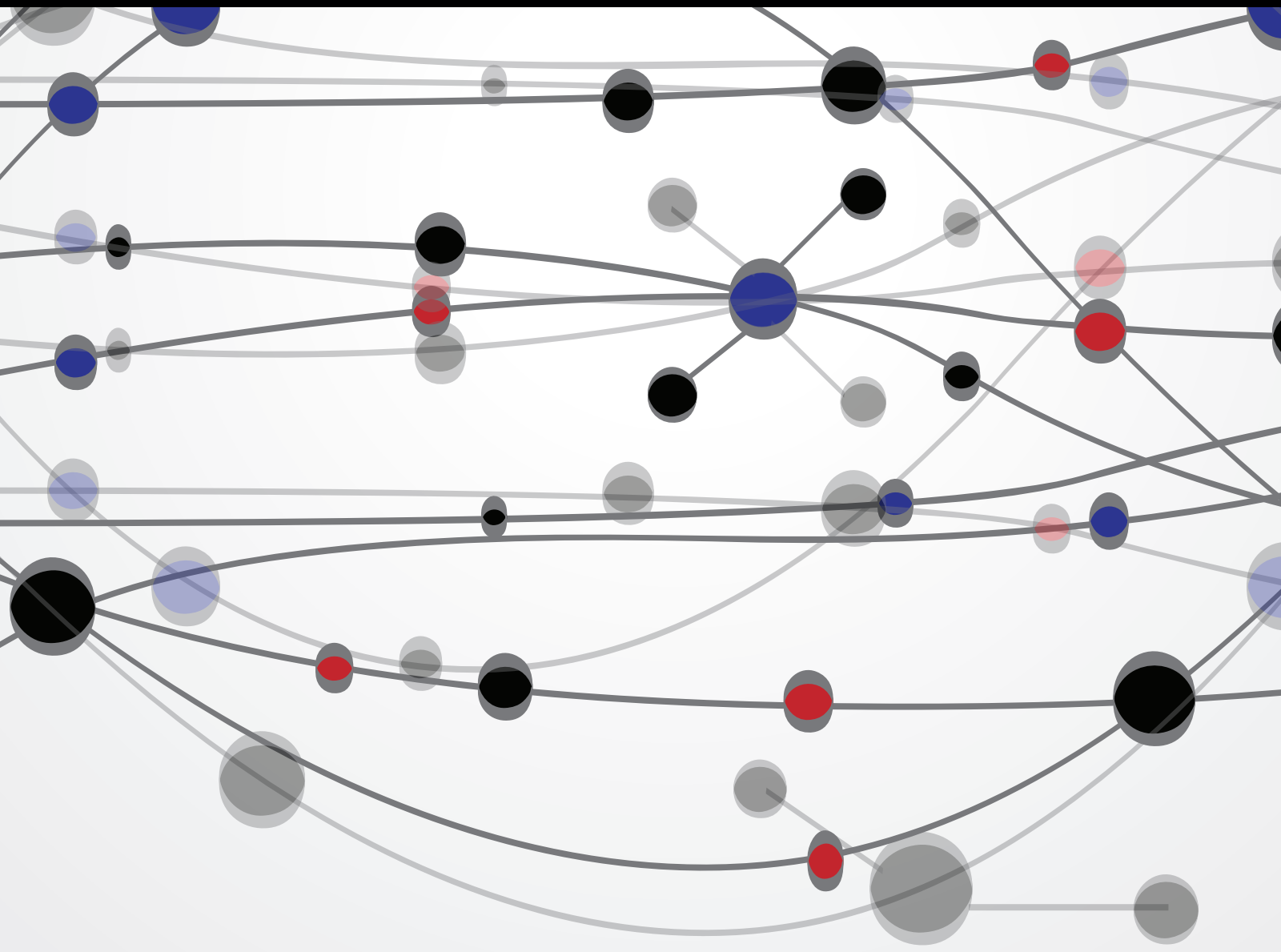
The Scientific World Journal

Recent Advancements in Soft Computing and its Application

Guest Editors: Albert Victoire, Anand Paul, SN Deepa, and Jaiji Wu



Recent Advancements in Soft Computing and its Application

The Scientific World Journal

Recent Advancements in Soft Computing and its Application

Guest Editors: Albert Victoire, Anand Paul, SN Deepa, and Jaiji Wu

Contents

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2015, Article ID 734957, 11 pages
http://dx.doi.org/10.1155/2015/734957

Research Article

Benchmarking RCGAu on the Noiseless BBOB Testbed

Babatunde A. Sawyerr,[1,2] Aderemi O. Adewumi,[1] and M. Montaz Ali[3]

[1] *School of Mathematics, Statistics and Computer Science, College of Agriculture, Engineering and Science, University of KwaZulu-Natal, Westville, South Africa*
[2] *Department of Computer Sciences, Faculty of Science, University of Lagos, Lagos, Nigeria*
[3] *School of Computational and Applied Mathematics, Faculty of Science and TCSE, Faculty of Engineering and Built Environment, University of the Witwatersrand, Johannesburg, South Africa*

Correspondence should be addressed to Aderemi O. Adewumi; adewumia@ukzn.ac.za

Received 19 July 2014; Accepted 9 November 2014

Academic Editor: Albert Victoire

RCGAu is a hybrid real-coded genetic algorithm with "*uniform random direction*" search mechanism. The *uniform random direction* search mechanism enhances the local search capability of RCGA. In this paper, RCGAu was tested on the BBOB-2013 noiseless testbed using restarts till a maximum number of function evaluations (#FEs) of $10^5 \times D$ are reached, where D is the dimension of the function search space. RCGAu was able to solve several test functions in the low search dimensions of 2 and 3 to the desired accuracy of 10^8. Although RCGAu found it difficult in getting a solution with the desired accuracy 10^8 for high conditioning and multimodal functions within the specified maximum #FEs, it was able to solve most of the test functions with dimensions up to 40 with lower precisions.

1. Introduction

The simple genetic algorithm (GA) introduced by Holland is a probabilistic algorithm based on the theory of natural selection by Charles Darwin. GA mimics the evolutionary process through the creation of variations in each generation and the survival of the fittest individuals through the blending of genetic traits. Individuals with genetic traits that increase their probability of survival will be given more opportunities to reproduce and their offspring will also profit from the heritable traits. Over the period of time these individuals will eventually dominate the population [1, 2].

GA consists of a set of potential solutions called chromosomes, a selection operator, a crossover operator, and a mutation operator. A chromosome is a string of zeros (0s) and ones (1s). It is a metaphor of the biological chromosome in living organisms. The zeros (0s) and ones (1s) are called genes. A gene is the transfer unit of heredity. It contains genetic traits or information that is passed on from a parent solution to its offspring. The selection operator selects solutions for mating based on the principle of "*survival of the fittest*." The crossover operator generates new solution pairs called children by combining the genetic materials of the selected parents. The mutation operator is an exploratory operator that is applied, with low probability, to the population of chromosomes to sustain diversity. Without the mutation operator, GAs can easily fall into premature convergence [1, 3].

The simple GA was designed to work on binary strings and it is directly applicable to pseudoboolean objective functions. However, most real life problems are represented as continuous parameter optimization problems. A decoding function was designed to map the solutions from binary space to the real-valued space. This decoding process can become prohibitively expensive for binary string GAs especially when the problem dimension increases [1, 3]. To tackle this problem real-coded genetic algorithms were introduced [4].

Real-coded genetic algorithms (RCGAs) use real-valued vectors to represent individual solutions. Surveys show that several variants of RCGAs have been proposed and used to solve a wide range of real life optimization problems. Some recent examples can be found in [1, 4–10].

Over the last three decades, researchers have continuously improved the performance of RCGAs through hybridization. RCGAs have been hybridized with other

(1) *Initialize* $P_{t=0}, P_t = \{x_{1,t}, x_{2,t}, \ldots, x_{N,t}\}$ *from* X
(2) $f(x_{i,t}) = evaluate(P_t), \{1 \leq i \leq N\}$
(3) *While not stopping condition, do Steps 4–12*
(4) *Calculate* $\sigma(f(P_t))$, *if* $\sigma(f(P_t)) \leq \varepsilon$ *do Step 5 else do Step 6*
(5) $\widehat{P}_t = perturb(P_t)$
(6) $\widehat{P}_t = tournamentSelection(P_t)$
(7) $C_t = blend\text{-}\alpha Crossover(\widehat{P}_t, p_c)$
(8) $M_t = non\text{-}uniformMutation(C_t, p_m)$
(9) $\Upsilon_t = ulsearch(M_t)$
(10) $f(x_{i,t}) = evaluate(\Upsilon_t)$
(11) $P_{t+1} = replace(P_t, \Upsilon_t)$
(12) $t = t + 1$
(13) *end while*

Algorithm 1: The RCGAu Algorithm.

optimizers such as Nelder-Mead algorithms [11], simplex method [12], quadratic approximation [13], and pattern search [14–16].

In this paper, a set of noiseless testbed from the black-box optimization benchmarking (BBOB) 2013 workshop is used to benchmark RCGAu, a hybrid real-coded genetic algorithm that consists of "*uniform random direction*" local search technique.

The RCGAu algorithm is presented in Section 2, Section 3 provides the CPU timing for the experiments, Section 4 presents the results and discussion, and finally Section 5 concludes the paper with some recommendations.

2. The RCGAu Algorithm

RCGAu is a hybrid RCGA with a simple derivative-free local search technique called "*uniform random direction*" local search method. The local search technique operates on all individuals after the mutation operator has been applied to the population of individuals.

The RCGAu used in this work is a modified version of the RCGAu used in [16, 17]. It consists of five major operators, namely, tournament selection, blend-α crossover, nonuniform mutation, uniform random direction local search method, and a stagnation alleviation mechanism. Algorithm 1 shows the RCGAu algorithm.

The notations used in this paper are defined as follows.

P_t denotes the population of individual solutions $x_{i,t}$ at time t, N is the size of P_t, $\sigma(f(P_t))$ represents the standard deviation of the fitness values $f(P_t)$ of all solutions $x_{i,t}, \in P_t$, $\widehat{P}_t$ is the mating pool containing the parent solutions, C_t is the population of offspring solutions obtained after applying crossover on the parents in $\widehat{P}_t$, p_c is the crossover probability, M_t is the resultant population of solutions after applying mutation on C_t, p_m is the mutation probability, and Υ_t is the population of solutions obtained after ulsearch has been applied to M_t, where ulsearch denotes the uniform random direction local search. Also, $\epsilon = 10^{-12}$, a very small positive value [18].

The evolutionary process in Algorithm 1 starts by initializing $P_{t=0}$ from the search space $X \subset \mathfrak{R}^n$. The domain of X is defined by specifying upper (u^j) and lower (l^j) limits of each jth component of x; that is, $l^j \leq x^j \leq u^j$ and $l^j, u^j \in \mathfrak{R}$, $j = 1, 2, \ldots, n$. Next, the fitness value $f(x_{i,t})$, $\forall x_{i,t} \in P_0$, is calculated and the population diversity of P_t is measured by calculating the standard deviation $\sigma(f(P_t))$ of $f(x_{i,t})$.

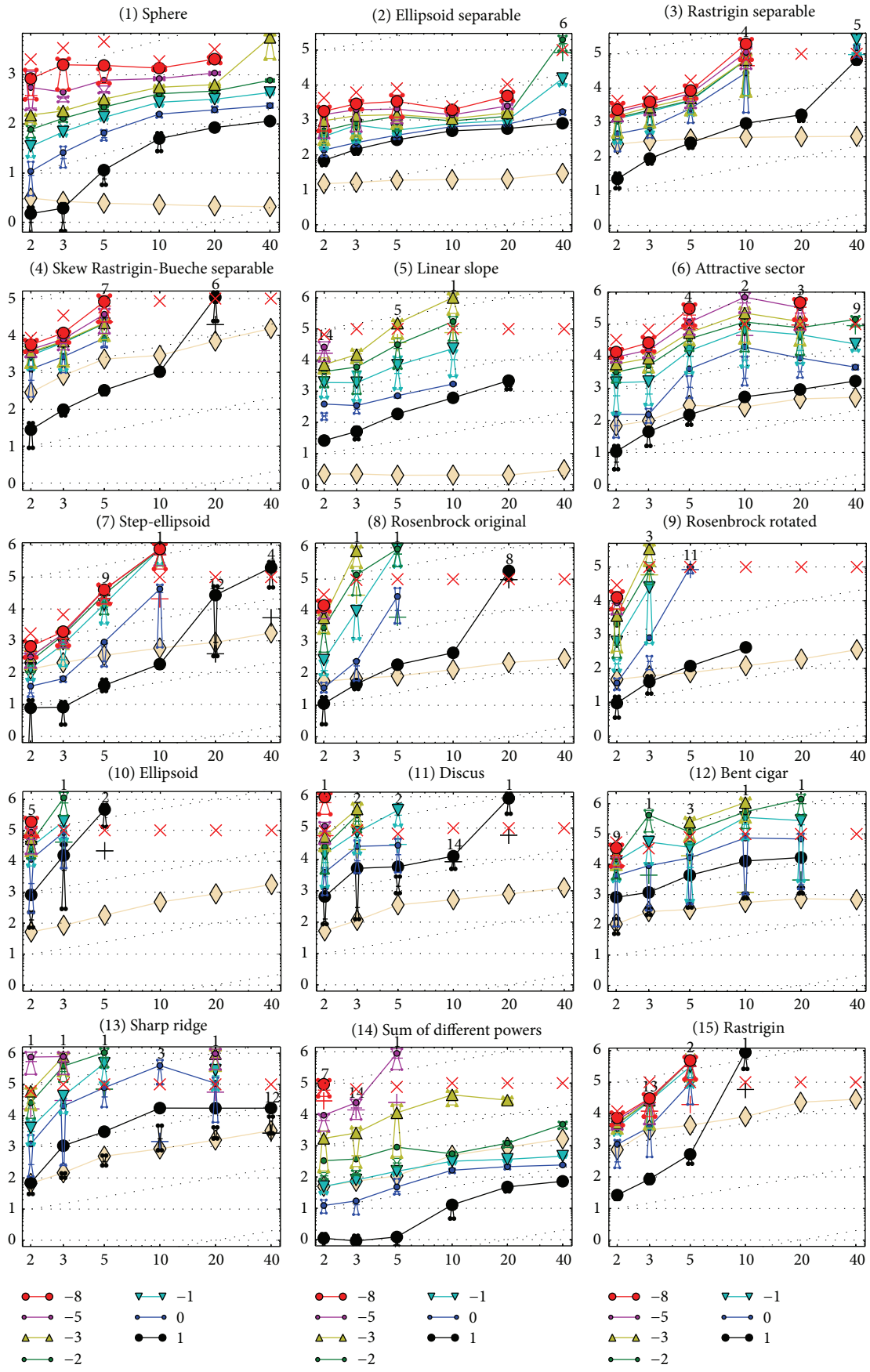
If $\sigma(f(P_t)) \leq \epsilon$ and the global optimum has not been found, then 90% of P_t is refreshed with newly generated solutions using the function perturb (P_t). P_t is refreshed by sorting the solutions according to their fitness values and preserving the top 10% of P_t. The remaining 90% of P_t are replaced with uniformly generated random values from the interval $[-4, 4]^D$ and the resultant population; $\widehat{P}_t = \{x_{1,t}, x_{2,t}, \ldots, x_{m,t}\}$ is created. m is the size of the mating pool $\widehat{P}_t$ and $m \leq N$. If, on the other hand, $\sigma(f(P_t)) > \epsilon$ then tournament selection is applied on P_t to create an equivalent mating pool $\widehat{P}_t$.

The tournament selection scheme works by selecting r number of solutions uniformly at random from P_t, where r is the tournament size and $r < N$. The selected r individuals are compared using their fitness values and the best individual is selected and assigned to $\widehat{P}_t$. This procedure is repeated m times to populate $\widehat{P}_t$.

After the mating pool has been created, blend-α crossover is applied to a pair of parent solutions $(x_{i,t}, x_{k,t})$ if a randomly generated number τ drawn uniformly from the interval $[0, 1]$ is greater than the specified crossover probability threshold p_c. Blend-α crossover creates a pair of offspring $(c_{1,t}, c_{2,t})$ from the interval $[\min(x^j_{i,t}, x^j_{k,t}) - \alpha * d^j, \max(x^j_{i,t}, x^j_{k,t}) + \alpha * d^j]$ as follows:

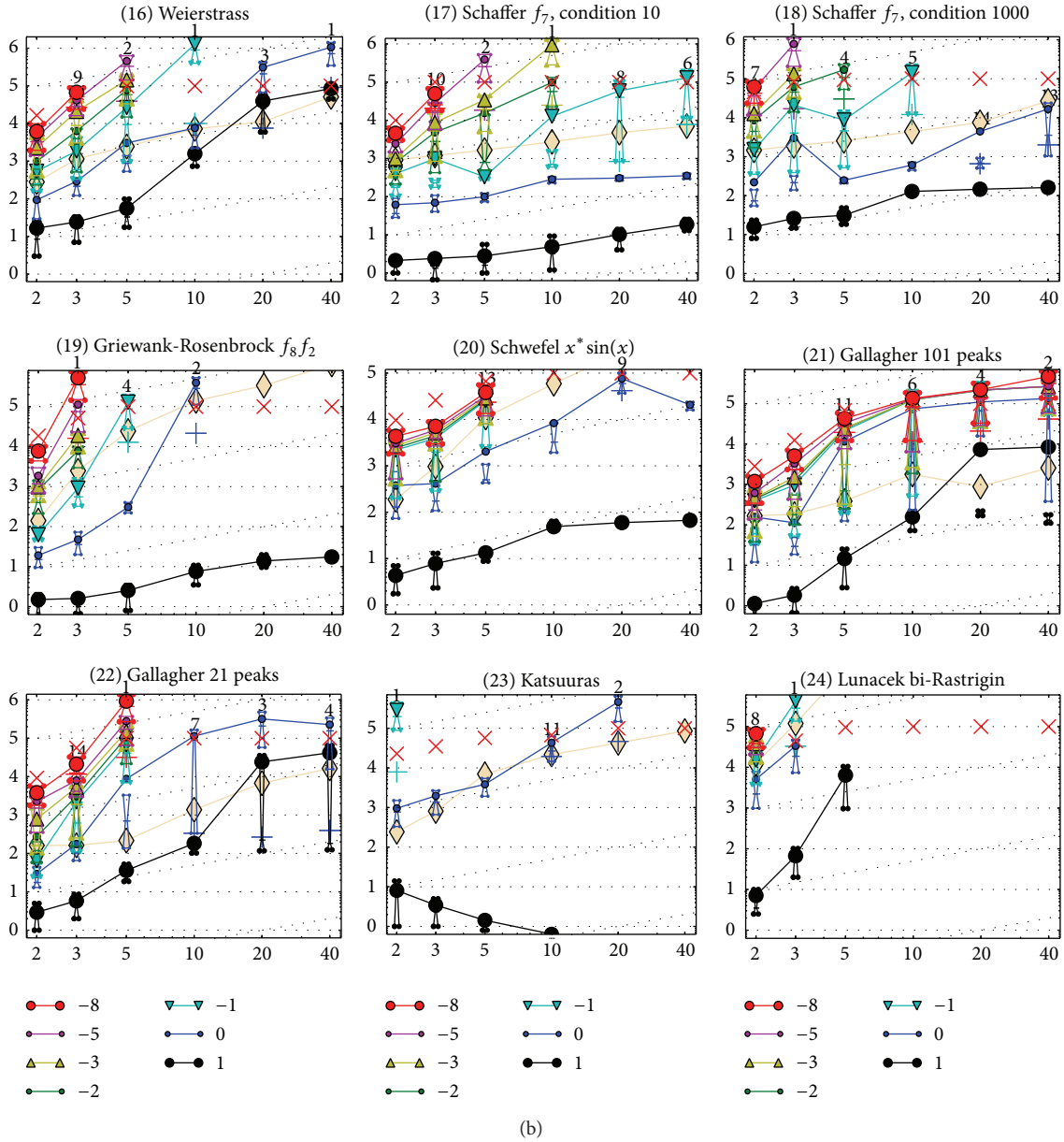
$$c^j_{1,t} = \left(\min\left(x^j_{i,t}, x^j_{k,t}\right) - \alpha * d^j, \max\left(x^j_{i,t}, x^j_{k,t}\right) + \alpha * d^j\right)$$
$$c^j_{2,t} = \left(\min\left(x^j_{i,t}, x^j_{k,t}\right) - \alpha * d^j, \max\left(x^j_{i,t}, x^j_{k,t}\right) + \alpha * d^j\right), \quad (1)$$

where $(1 \leq k \leq N)$, $\alpha = 0.3 + 0.2 \times z$, z is a uniform random number drawn from the interval $[0, 1]$, and $d^j = |x^j_{i,t} - x^j_{k,t}|$. The new pair $(c_{1,t}, c_{2,t})$ is then copied to the set C_t; otherwise the pair $(x_{i,t}, x_{k,t})$ is copied to C_t.



(a)

Figure 1: Continued.



(b)

FIGURE 1: Expected number of f-evaluations (ERT, lines) to reach $f_{\text{opt}} + \Delta f$; median number of f-evaluations (+) to reach the most difficult target that was reached not always but at least once; maximum number of f-evaluations in any trial (×); interquartile range with median (notched boxes) of simulated run lengths to reach $f_{\text{opt}} + \Delta f$; all values are divided by dimension and plotted as $\log_{10}$ values versus dimension. Also, $\Delta f = 10^{\{1,0,-1,-2,-3,-5,-8\}}$ are shown. Numbers above ERT-symbols (if appearing) indicate the number of trials reaching the respective targets. The light thick line with diamonds indicates the respective best results from BBOB-2009 for $\Delta f = 10^{-8}$. Horizontal lines mean linear scaling and slanted grid lines depict quadratic scaling.

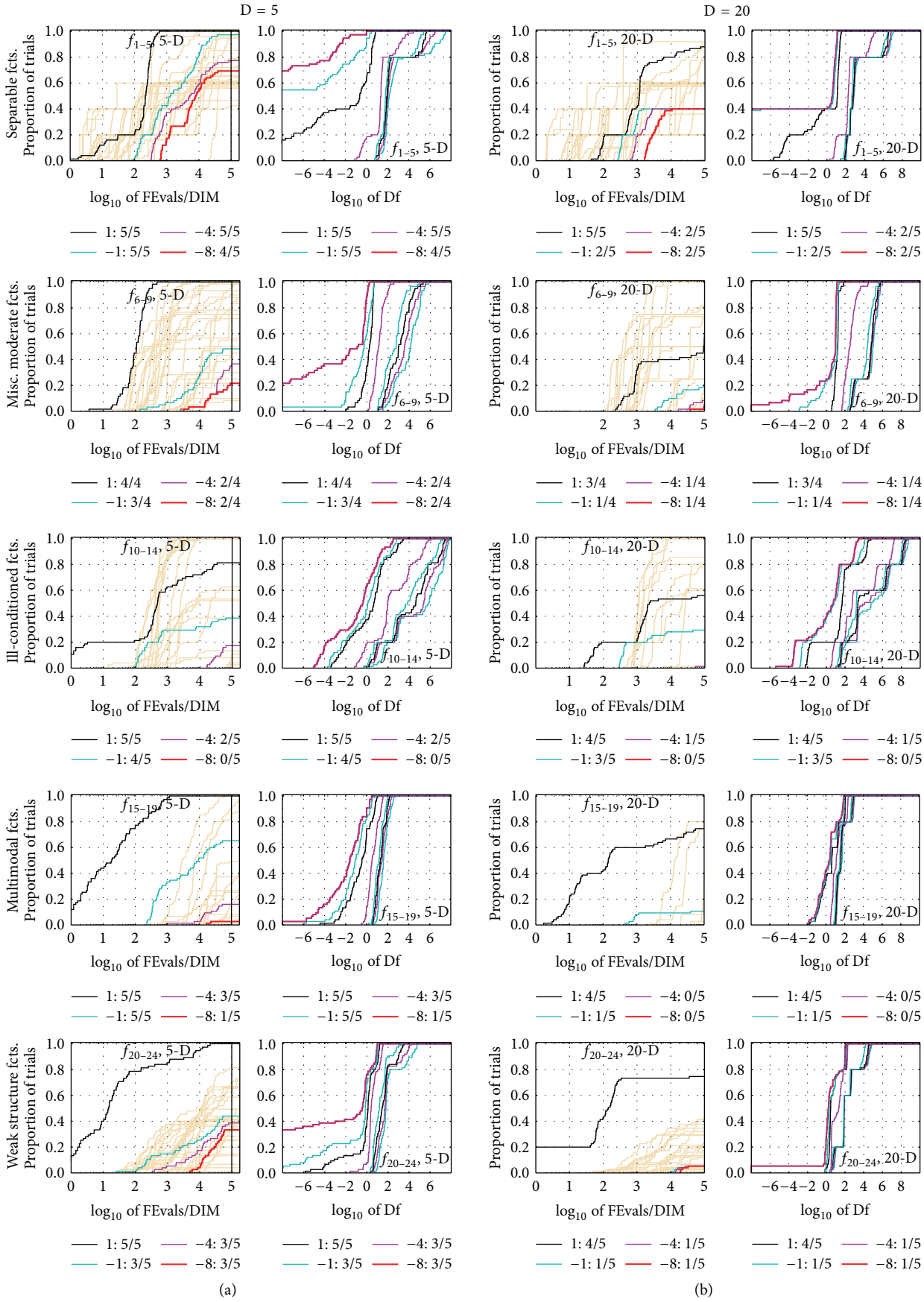
Then the nonuniform mutation [4] is applied to the components of each member of C_t with probability, p_m, as follows :

$$m_{i,t}^{j} = \begin{cases} c_{i,t}^{j} + \Delta\left(t, u^{j} - c_{i,t}^{j}\right) & \text{if } u \le 0.5, \\ c_{i,t}^{j} - \Delta\left(t, c_{i,t}^{j} - l^{j}\right) & \text{otherwise,} \end{cases} \tag{2}$$

where u is a uniformly distributed random number in the interval $[0, 1]$. u^j and l^j are the upper and lower boundaries of $x \in X$, respectively. The function $\Delta(t, u^j - c_{i,t}^j)$ given below takes a value in the interval $[0, y]$:

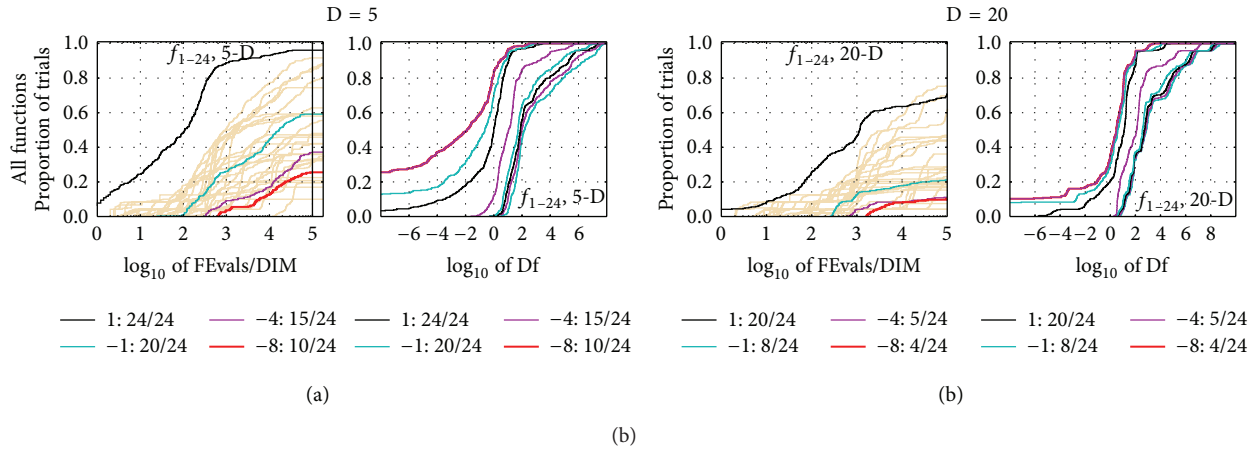
$$\Delta\left(t, y\right) = y\left(1 - r^{(1-(t/T))}\right)^{\beta}, \tag{3}$$

where r is a uniformly distributed random number in the interval $[0, 1]$, T is the maximum number of generations, and β is a parameter that determines the nonuniform strength of the mutation operator. The mutated individual $m_{i,t}$ is then copied to the set M_t; otherwise $c_{i,t}$ is copied to M_t.



(a)

Figure 2: Continued.



(b)

Figure 2: Empirical cumulative distribution functions (ECDF), plotting the fraction of trials with an outcome not larger than the respective values on the x-axis. Left subplots: ECDF of the number of function evaluations (FEvals) divided by search space dimension D, to fall below $f_{\text{opt}} + \Delta f$ with $\Delta f = 10^k$, where k is the first value in the legend. The thick red line represents the most difficult target value $f_{\text{opt}} + 10^{-8}$. Legends indicate for each target the number of functions that were solved in at least one trial within the displayed budget. Right subplots: ECDF of the best achieved Δf for running times of $0.5D, 1.2D, 3D, 10D, 100D, 1000D, \ldots$ function evaluations (from right to left cycling cyan-magenta-black...) and final Δf-value (red), where Δf and Df denote the difference to the optimal function value. Light brown lines in the background show ECDF for the most difficult target of all algorithms benchmarked during BBOB-2009.

Then ulsearch is applied on each solution $m_{i,t} \in M_t$ with the aim of performing local searches around the neighborhood of each solution. ulsearch works by randomly selecting a solution $m_{i,t} \in M_t$ and creating a trial point $y_{i,t}$ using

$$y_{i,t} = m_{i,t} + \Delta_t U, \tag{4}$$

where Δ_t is a step size parameter and $U = (U_1, U_2, \ldots, U_n)^T$ is a directional cosine with random components

$$U_j = \frac{R_j}{\left(R_1^2 + \cdots + R_n^2\right)^{1/2}}, \quad j = 1, 2, \ldots, n, \tag{5}$$

where $R_j \sim \text{Unif}([-1, 1])$. There are cases when the components of the trial point $y_{i,t} = (y_{i,t}^1, y_{i,t}^2, \ldots, y_{i,t}^n)$ generated by (4) fall outside the search space X during the search. In these cases, the components of $y_{i,t}$ are regenerated using

$$y_{i,t}^j = \begin{cases} m_{i,t}^j + \lambda\left(u^j - m_{i,t}^j\right), & \text{if } y_{i,t}^j > u^j \\ m_{i,t}^j + \lambda\left(m_{i,t}^j - l^j\right), & \text{if } y_{i,t}^j < l^j, \end{cases} \tag{6}$$

where $\lambda \sim \text{Unif}([0, 1])$ and $m_{i,t}^j$ is the corresponding component of the randomly selected solution $m_{i,t} \in M_t$.

The step size parameter, Δ_t, is initialized at time $t = 0$ according to [15, 16] by

$$\Delta_0 = \tau \times \max\left\{u^j - l^j \mid j = 1, 2, \ldots, n\right\}, \tag{7}$$

where $\tau \in [0, 1]$. The idea of using (7) to generate the initial step length is to accelerate the search by starting with a suitably large step size to quickly traverse the search space and as the search progresses the step size is adaptively adjusted at the end of each generation, t, by

$$\Delta_{t+1} = \frac{1}{K}\sum_{i=1}^{K} \gamma^i, \tag{8}$$

where K is the number of Euclidean distances $\{\gamma^1, \gamma^2, \ldots, \gamma^K\}$ between K nearest points to the mean $\overline{x}$ and $\overline{x}$ of a set of randomly selected distinct points $\Omega = \{x_1, x_2, \ldots, x_q\} \subset P_t$.

After the trial point $y_{i,t} \in \Upsilon$ has been created, it is evaluated and compared with $m_{i,t}$. If $y_{i,t} < m_{i,t}$, then $y_{i,t} \in \Upsilon$ is used to replace $m_{i,t} \in M_t$; otherwise the search direction is changed by changing the sign of the step length. The new step length is used to recalculate a new trial point. After a new trial point has been recalculated and evaluated, it is used to replace $m_{i,t} \in M_t$ with $y_{i,t}$, if $y_{i,t} < m_{i,t}$; otherwise $m_{i,t} \in M_t$ is retained.

At the end of ulsearch, P_t is updated with M_t to form P_{t+1} and elitism is used to replace the worst point in P_{t+1} with the best point in P_t because the generational model is the replacement strategy adopted in this work [19].

3. Experimental Procedure and Parameter Settings

The experimental setup was carried out according to [20] on the benchmark functions provided in [21, 22]. Two independent restart strategies were used to restart RCGAu whenever P_t stagnates or when the maximum number of generations is exceeded and f_{target} is not found. For each restart strategy, the experiment is reinitialized with an initial population P_0 which is uniformly and randomly sampled from the search space $[-4, 4]^D$ [6, 18].

Two stopping conditions used for the restart strategies are as follows.

(i) A test for stagnation is carried out to check if the best solution obtained so far did not vary by more than 10^{-12} during the last $(50 + 25 \times D)$ generations as in [6].

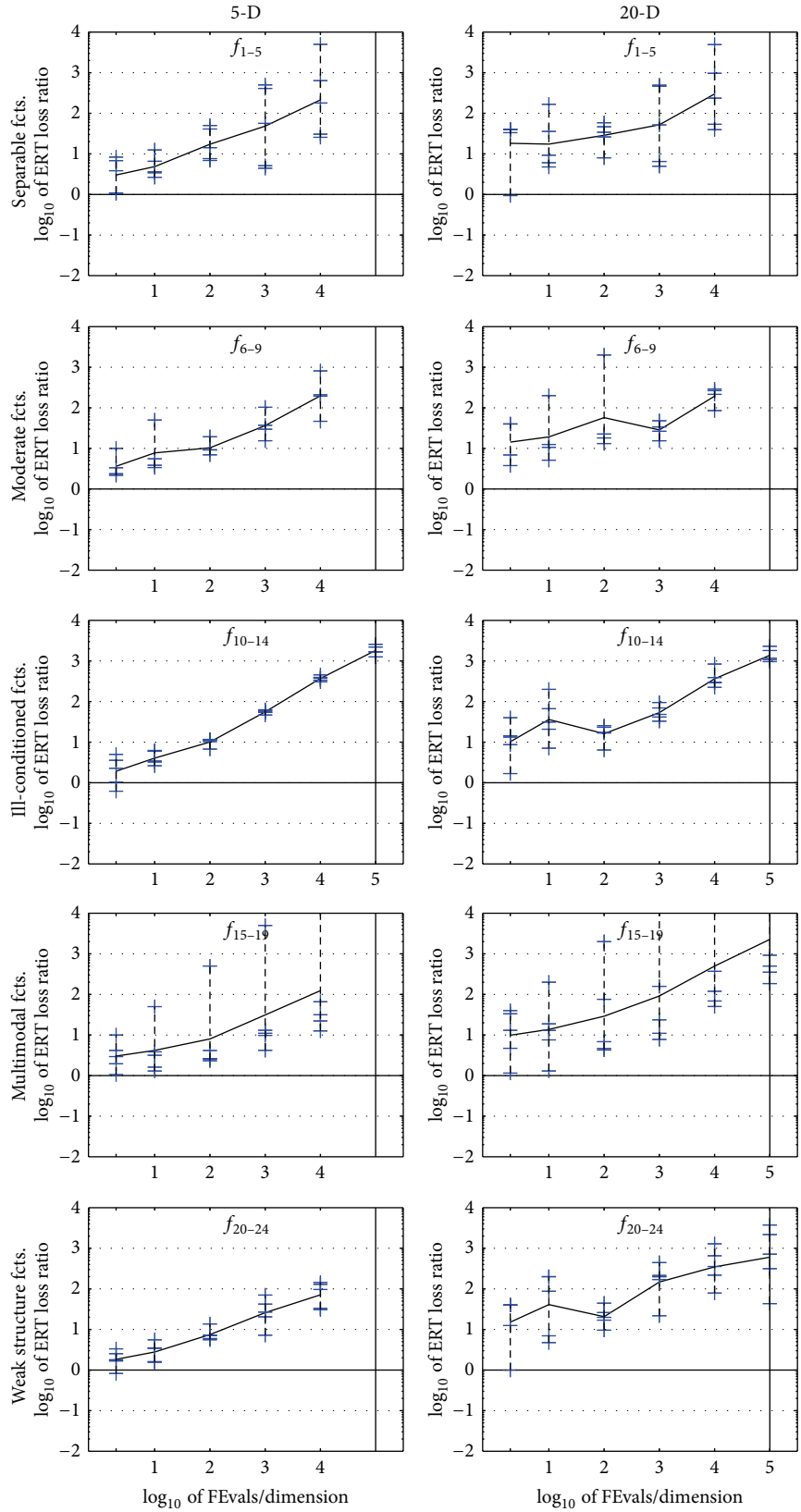


Figure 3: ERT loss ratios (see Table 2 for details). Each cross (+) represents a single function and the line is the geometric mean.

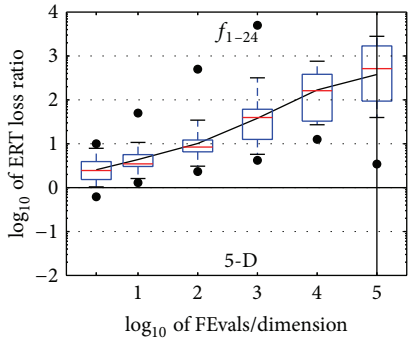
Table 1: Expected running time (ERT in number of function evaluations) divided by the best ERT measured during BBOB-2009. The ERT and, in braces, as dispersion measure, the half difference between 90 and 10 percentile of bootstrapped run lengths appear in the second row of each cell, the best ERT in the first. The different target Δf-values are shown in the top row. #succ is the number of trials that reached the (final) target $f_{\text{opt}} + 10^{-8}$.

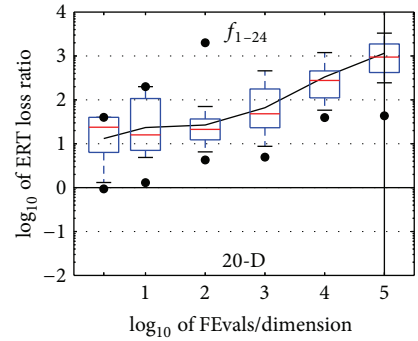
	5D								20D							
Δf	1e + 1	1e + 0	1e − 1	1e − 2	1e − 3	1e − 5	1e − 7	#succ	Δf	1e + 1	1e + 0	1e − 1	1e − 2	1e − 3	1e − 5	1e − 7	#succ
$\mathbf{f}_1$	11	12	12	12	12	12	12	15/15	$\mathbf{f}_1$	43	43	43	43	43	43	43	15/15
	5.2 (6)	27 (15)	56 (15)	91 (22)	131 (29)	319 (193)	580 (800)	15/15		39 (7)	90 (17)	148 (19)	218 (28)	294 (36)	506 (112)	787 (173)	15/15
$\mathbf{f}_2$	83	87	88	89	90	92	94	15/15	$\mathbf{f}_2$	385	386	387	388	390	391	393	15/15
	16 (3)	21 (6)	29 (10)	71 (118)	79 (118)	112 (180)	155 (176)	15/15		29 (5)	39 (8)	50 (8)	66 (17)	81 (30)	128 (41)	189 (78)	15/15
$\mathbf{f}_3$	716	1622	1637	1642	1646	1650	1654	15/15	$\mathbf{f}_3$	5066	7626	7635	7637	7643	7646	7651	15/15
	1.8 (1)	7.8 (9)	13 (15)	13 (15)	16 (16)	21 (20)	24 (19)	15/15		6.6 (6)	∞	∞	∞	∞	∞	∞*1.1e6*	0/15
$\mathbf{f}_4$	809	1633	1688	1758	1817	1886	1903	15/15	$\mathbf{f}_4$	4722	7628	7666	7686	7700	7758	1.4e5	9/15
	2.0 (0.9)	26 (26)	62 (83)	60 (79)	60 (63)	99 (109)	129 (134)	7/15		459 (473)	∞	∞	∞	∞	∞	∞*1.1e6*	0/15
$\mathbf{f}_5$	10	10	10	10	10	10	10	15/15	$\mathbf{f}_5$	41	41	41	41	41	41	41	15/15
	93 (20)	359 (112)	3404 (3910)	15683 (18516)	76807 (77456)	∞	∞*2.7e5*	0/15		1075 (273)	∞	∞	∞	∞	∞	∞*1.2e6*	0/15
$\mathbf{f}_6$	114	214	281	404	580	1038	1332	15/15	$\mathbf{f}_6$	1296	2343	3413	4255	5220	6728	8409	15/15
	6.5 (6)	95 (217)	263 (443)	432 (351)	470 (459)	552 (489)	749 (685)	4/15		14 (3)	75 (28)	277 (325)	365 (470)	465 (474)	971 (1056)	1110 (1193)	3/15
$\mathbf{f}_7$	24	324	1171	1451	1572	1572	1597	15/15	$\mathbf{f}_7$	1351	4274	9503	16523	16524	16524	16969	15/15
	8.4 (6)	14 (22)	58 (89)	98 (114)	126 (139)	126 (133)	125 (134)	9/15		402 (507)	∞	∞	∞	∞	∞	∞*1.0e6*	0/15
$\mathbf{f}_8$	73	273	336	372	391	410	422	15/15	$\mathbf{f}_8$	2039	3871	4040	4148	4219	4371	4484	15/15
	13 (4)	517 (732)	13503 (15418)	12258 (13199)	∞	∞	∞*2.8e5*	0/15		1796 (1474)	∞	∞	∞	∞	∞	∞*2.0e6*	0/15
$\mathbf{f}_9$	35	127	214	263	300	335	369	15/15	$\mathbf{f}_9$	1716	3102	3277	3379	3455	3594	3727	15/15
	17 (7)	3920 (3354)	∞	∞	∞	∞	∞*5.0e5*	0/15		∞	∞	∞	∞	∞	∞	∞*2.0e6*	0/15
$\mathbf{f}_{10}$	349	500	574	607	626	829	880	15/15	$\mathbf{f}_{10}$	7413	8661	10735	13641	14920	17073	17476	15/15
	6725 (7986)	∞	∞	∞	∞	∞	∞*3.1e5*	0/15		∞	∞	∞	∞	∞	∞	∞*2.0e6*	0/15
$\mathbf{f}_{11}$	143	202	763	977	1177	1467	1673	15/15	$\mathbf{f}_{11}$	1002	2228	6278	8586	9762	12285	14831	15/15
	205 (236)	698 (864)	2442 (2575)	∞	∞	∞	∞*2.6e5*	0/15		17715 (19470)	∞	∞	∞	∞	∞	∞*1.1e6*	0/15
$\mathbf{f}_{12}$	108	268	371	413	461	1303	1494	15/15	$\mathbf{f}_{12}$	1042	1938	2740	3156	4140	12407	13827	15/15
	201 (576)	303 (585)	490 (704)	1392 (1453)	2674 (2913)	∞	∞*2.7e5*	0/15		320 (961)	706 (1035)	2024 (2557)	8893 (9507)	∞	∞	∞*2.0e6*	0/15
$\mathbf{f}_{13}$	132	195	250	319	1310	1752	2255	15/15	$\mathbf{f}_{13}$	652	2021	2751	3507	18749	24455	30201	15/15
	115 (216)	2009 (2341)	9467 (10306)	16356 (17378)	∞	∞	∞*3.3e5*	0/15		529 (907)	1072 (1477)	2052 (2407)	5467 (5911)	1026 (1134)	792 (852)	∞*1.2e6*	0/15
$\mathbf{f}_{14}$	10	41	58	90	139	251	476	15/15	$\mathbf{f}_{14}$	75	239	304	451	932	1648	15661	15/15
	0.62 (0.6)	5.9 (5)	13 (5)	50 (23)	406 (473)	17736 (19499)	∞*3.1e5*	0/15		13 (6)	18 (3)	24 (5)	53 (29)	612 (196)	∞	∞*2.0e6*	0/15
$\mathbf{f}_{15}$	511	9310	19369	19743	20073	20769	21359	14/15	$\mathbf{f}_{15}$	30378	1.5e5	3.1e5	3.2e5	3.2e5	4.5e5	4.6e5	15/15
	5.1 (4)	50 (60)	81 (91)	120 (151)	118 (142)	115 (142)	114 (130)	2/15		∞	∞	∞	∞	∞	∞	∞*1.2e6*	0/15
$\mathbf{f}_{16}$	120	612	2662	10163	10449	11644	12095	15/15	$\mathbf{f}_{16}$	1384	27265	77015	1.4e5	1.9e5	2.0e5	2.2e5	15/15
	2.3 (2)	25 (7)	50 (79)	38 (49)	69 (80)	196 (217)	∞*2.9e5*	0/15		569 (695)	227 (268)	∞	∞	∞	∞	∞*1.3e6*	0/15
$\mathbf{f}_{17}$	5.2	215	899	2861	3669	6351	7934	15/15	$\mathbf{f}_{17}$	63	1030	4005	12242	30677	56288	80472	15/15
	2.7 (3)	2.3 (0.8)	1.8 (0.6)	27 (33)	47 (57)	308 (353)	∞*2.6e5*	0/15		3.2 (3)	6.0 (2)	294 (397)	∞	∞	∞	∞*1.1e6*	0/15
$\mathbf{f}_{18}$	103	378	3968	8451	9280	10905	12469	15/15	$\mathbf{f}_{18}$	621	3972	19561	28555	67569	1.3e5	1.5e5	15/15
	1.5 (1)	3.3 (0.8)	11 (11)	101 (106)	∞	∞	∞*2.6e5*	0/15		4.7 (2)	22 (3)	∞	∞	∞	∞	∞*1.1e6*	0/15
$\mathbf{f}_{19}$	1	1	242	1.0e5	1.2e5	1.2e5	1.2e5	15/15	$\mathbf{f}_{19}$	1	1	3.4e5	4.7e6	6.2e6	6.7e6	6.7e6	15/15
	13 (15)	1532 (999)	2682 (3069)	∞	∞	∞	∞*1.8e5*	0/15		277 (139)	∞	∞	∞	∞	∞	∞*1.1e6*	0/15
$\mathbf{f}_{20}$	16	851	38111	51362	54470	54861	55313	14/15	$\mathbf{f}_{20}$	82	46150	3.1e6	5.5e6	5.5e6	5.6e6	5.6e6	14/15
	4.2 (3)	12 (21)	3.4 (3)	2.7 (3)	2.7 (3)	3.0 (3)	3.4 (3)	13/15		15 (3)	33 (33)	∞	∞	∞	∞	∞*1.2e6*	0/15

Table 1: Continued.

	5D								20D							
Δf	$1e+1$	$1e+0$	$1e-1$	$1e-2$	$1e-3$	$1e-5$	$1e-7$	#succ	Δf	$1e+1$	$1e+0$	$1e-1$	$1e-2$	$1e-3$	$1e-5$	$1e-7$	#succ
$\mathbf{f_{21}}$	41	1157	1674	1692	1705	1729	1757	14/15	$\mathbf{f_{21}}$	561	6541	14103	14318	14643	15567	17589	15/15
	1.8 (2)	50 (60)	64 (106)	71 (101)	72 (100)	96 (121)	109 (116)	11/15		261 (4)	338 (435)	304 (365)	300 (362)	295 (344)	280 (317)	250 (301)	4/15
$\mathbf{f_{22}}$	71	386	938	980	1008	1040	1068	14/15	$\mathbf{f_{22}}$	467	5580	23491	24163	24948	26847	1.3*e*5	12/15
	2.5 (2)	113 (258)	364 (426)	545 (684)	754 (889)	1337 (1548)	4297 (4718)	1/15		1036 (2143)	1139 (1385)	∞	∞	∞	∞	∞*1.2e6*	0/15
$\mathbf{f_{23}}$	3.0	518	14249	27890	31654	33030	34256	15/15	$\mathbf{f_{23}}$	3.2	1614	67457	3.7*e*5	4.9*e*5	8.1*e*5	8.4*e*5	15/15
	2.4 (1)	37 (42)	∞	∞	∞	∞	∞*1.7e5*	0/15		1.7 (2)	5701 (6238)	∞	∞	∞	∞	∞*1.1e6*	0/15
$\mathbf{f_{24}}$	1622	2.2*e*5	6.4*e*6	9.6*e*6	9.6*e*6	1.3*e*7	1.3*e*7	3/15	$\mathbf{f_{24}}$	1.3*e*6	7.5*e*6	5.2*e*7	5.2*e*7	5.2*e*7	5.2*e*7	5.2*e*7	3/15
	20 (27)	∞	∞	∞	∞	∞	∞*1.8e5*	0/15		∞	∞	∞	∞	∞	∞	∞*1.1e6*	0/15

Table 2: ERT loss ratio versus the budget (both in number of f-evaluations divided by dimension). The target value f_t for a given budget FEvals is the best target f-value reached within the budget by the given algorithm. Shown is the ERT of the given algorithm divided by best ERT seen in GECCO-BBOB-2009 for the target f_t, or, if the best algorithm reached a better target within the budget, the budget divided by the best ERT. Line: geometric mean. Box-Whisker error bar: 25–75%-ile with median (box), 10–90%-ile (caps), and minimum and maximum ERT loss ratio (points). The vertical line gives the maximal number of function evaluations in a single trial in this function subset. See also Figure 3 for results on each function subgroup.





#FEs/D	f_1–f_{24} in 5-D, maxFE/D = 100018					
	best	10%	25%	**med**	75%	90%
2	0.62	1.0	1.4	2.5	4.0	8.5
10	1.3	1.6	2.9	3.5	5.8	16
100	2.3	2.6	6.4	8.4	13	42
1*e*3	4.2	5.0	12	40	62	4.2*e*2
1*e*4	13	25	32	1.6*e*2	3.9*e*2	1.2*e*3
1*e*5	3.4	35	83	5.1*e*2	1.7*e*3	3.0*e*3
1*e*6	3.4	35	1.4*e*2	8.6*e*2	1.2*e*4	2.2*e*4
RL_{US}/D	3*e*4	4*e*4	5*e*4	5*e*4	7*e*4	1*e*5
#FEs/D	f_1–f_{24} in 20-D, maxFE/D = 100004					
	best	10%	25%	**med**	75%	90%
2	0.94	1.1	5.8	24	40	40
10	1.3	4.8	7.0	16	1.3*e*2	2.0*e*2
100	4.3	6.2	11	21	39	2.7*e*2
1*e*3	5.0	7.6	23	48	1.8*e*2	4.7*e*2
1*e*4	39	53	1.0*e*2	2.8*e*2	5.2*e*2	1.7*e*3
1*e*5	43	2.3*e*2	4.0*e*2	9.4*e*2	1.9*e*3	8.3*e*3
1*e*6	2.4*e*2	4.2*e*2	9.9*e*2	3.9*e*3	1.1*e*4	6.8*e*4
RL_{US}/D	4*e*4	5*e*4	5*e*4	6*e*4	1*e*5	1*e*5

(ii) A test is carried out to check if the maximum number of generations is satisfied and f_{target} is not found.

The parameters used for RCGAu on all functions are

(i) population size = $\min(100, 10 \times D)$, where D is the problem dimension;

(ii) maximum number of evaluations #FEs = $10^5 \times D$;

(iii) tournament size $r = 3$;

(iv) crossover rate $p_c = 0.8$;

(v) mutation rate $p_m = 0.15$;

(vi) nonuniformity factor for the mutation $\beta = 15$;

(vii) elitism $E = 1$;

(viii) crafting effort CrE = 0 [20].

4. CPU Timing Experiment

The CPU timing experiment was conducted for RCGAu using the same independent restart strategies on the function f_8 for a duration of 30 seconds on an AMD Turion (tm) II Ultra Dual-Core Mobile *M*620 CPU processor, running at 2.50 GHz under a 32-bit Microsoft Windows 7 Professional service pack 1 with 2.75 GB RAM usable and Matlab 7.10 (*R*2010*a*).

The time per function evaluation was 2.5, 2.6, 2.9, 3.0, 3.2, and 3.5 times 10^{-4} seconds for RCGAu in dimensions 2, 3, 5,10, 20, and 40, respectively.

5. Results

The results of the empirical experiments conducted on RCGAu according to [20] on the benchmark functions given in [21, 22] are presented in Figures 1, 2, and 3 and in Tables 1 and 2.

Figure 1 shows the performance of RCGAu on all the noiseless problems with the dimensions 2, 3, 5, 10, 20, and 40. RCGAu was able to solve many test functions in the low search dimensions of 2 and 3 to the desired accuracy of 10^8. It is able to solve most test functions with dimensions up to 40 at lowest precision of 10^1.

Although RCGAu found it difficult in getting a solution with the desired accuracy 10^8 for high conditioning and multimodal functions within the specified maximum #FEs it was able to solve f_{21} with dimensions up to 40, f_1 and f_2 with dimensions up to 20, f_3 and f_7 with dimensions up to 10, and f_4, f_6, f_{15}, f_{20}, and f_{22} with dimensions up to 5.

In Figure 2, the left subplot graphically illustrates the empirical cumulative distribution function (ECDF) of the number of function evaluations divided by the dimension of the search space, while the right subplot shows the ECDF of the best achieved Δf. This figure graphically shows the performance of RCGAu in terms of function evaluation.

Table 1 presents the performance of RCGAu in terms of the expected running time (ERT). This measure estimates the run time of RCGAu by using the number of function evaluations divided by the best ERT measured during BBOB 2009 workshop. This benchmark shows that RCGAu needs some improvement in terms of performance.

6. Conclusion

The performance of RCGAu on the suite of noiseless black-box optimization testbed has been average on a number of problems but it has excelled in solving functions f_1, f_2, f_3, f_7, and f_{21}. Studies have currently been carried out to find out why RCGAs do not efficiently solve highly conditioned problems. Further modifications to RCGAs are needed to exploit the full strength of evolutionary processes.

Conflict of Interests

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

References

[1] T. Back, *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*, Oxford University press, New York, NY, USA, 1996.

[2] J. H. Holland, *Adaptation in Natural and Artificial Systems, An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*, MIT Press, Cambridge, Mass, USA, 1975.

[3] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, Mass, USA, 1989.

[4] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, Springer, Berlin, Germany, 1996.

[5] M. M. Ali and A. Törn, "Population set-based global optimization algorithms: some modifications and numerical studies," *Computers & Operations Research*, vol. 31, no. 10, pp. 1703–1725, 2004.

[6] Y.-C. Chuang and C.-T. Chen, "Black-box optimization benchmarking for noiseless function testbed using a direction-based RCGA," in *Proceedings of the 14th International Conference on Genetic and Evolutionary Computation (GECCO '12)*, pp. 167–174, Philadelphia, PA, USA, July 2012.

[7] K. Deep and M. Thakur, "A new crossover operator for real coded genetic algorithms," *Applied Mathematics and Computation*, vol. 188, no. 1, pp. 895–911, 2007.

[8] K. Deep and M. Thakur, "A new mutation operator for real coded genetic algorithms," *Applied Mathematics and Computation*, vol. 193, no. 1, pp. 211–230, 2007.

[9] K. Deep and M. Thakur, "A real coded multi parent genetic algorithms for function optimization," *Applied Mathematics and Computation*, vol. 1, no. 2, pp. 67–83, 2008.

[10] P. Kaelo and M. M. Ali, "Integrated crossover rules in real coded genetic algorithms," *European Journal of Operational Research*, vol. 176, no. 1, pp. 60–76, 2007.

[11] R. Chelouah and P. Siarry, "Genetic and Nelder-Mead algorithms hybridized for a more accurate global optimization of continuous multiminima functions," *European Journal of Operational Research*, vol. 148, no. 2, pp. 335–348, 2003.

[12] J. Andre, P. Siarry, and T. Dognon, "An improvement of the standard genetic algorithm fighting premature convergence in continuous optimization," *Advances in Engineering Software*, vol. 32, no. 1, pp. 49–60, 2001.

[13] K. Deep and K. N. Das, "Quadratic approximation based hybrid genetic algorithm for function optimization," *Applied Mathematics and Computation*, vol. 203, no. 1, pp. 86–98, 2008.

[14] W. E. Hart, *Adaptive global optimization with local search [Ph.D. thesis]*, University of California, San Diego, Calif, USA, 1994.

[15] B. A. Sawyerr, M. M. Ali, and A. O. Adewumi, "A comparative study of some real-coded genetic algorithms for unconstrained global optimization," *Optimization Methods & Software*, vol. 26, no. 6, pp. 945–970, 2011.

[16] B. A. Sawyerr, *Hybrid real coded genetic algorithms with pattern search and projection [Ph.D. thesis]*, University of Lagos, Lagos, Nigeria, 2010.

[17] B. A. Sawyerr, A. O. Adewumi, and M. M. Ali, "Real-coded genetic algorithm with uniform random local search," *Applied Mathematics and Computation*, vol. 228, pp. 589–597, 2014.

[18] B. A. Sawyerr, A. O. Adewumi, and M. M. Ali, "Benchmarking projection-based real coded genetic algorithm on BBOB-2013 noiseless function testbed," in *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation (GECCO '13)*, pp. 1193–1200, Amsterdam, The Netherlands, July 2013.

[19] K. A. DeJong, *An analysis of the behavior of a class of genetic adaptive systems [Ph.D. thesis]*, University of Michgan, Ann Arbor, Mich, USA, 1975.

[20] N. Hansen, A. Auger, S. Finck, and R. Ros, "Real-parameter black-box optimization benchmarking 2012: experimental setup," Tech. Rep., INRIA, 2012.

[21] S. Finck, N. Hansen, R. Ros, and A. Auger, "Real-parameter black-box optimization benchmarking 2009: presentation of the noiseless functions," Tech. Rep. 2009/20, Research Center PPE, 2009.

[22] N. Hansen, S. Finck, R. Ros, and A. Auger, "Real-parameter black-box optimization benchmarking 2009: noiseless functions definitions," Tech. Rep. RR-6829, INRIA, 2009.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2015, Article ID 704587, 14 pages
http://dx.doi.org/10.1155/2015/704587

Research Article

From Determinism and Probability to Chaos: Chaotic Evolution towards Philosophy and Methodology of Chaotic Optimization

Yan Pei

Computer Science Division, The University of Aizu, Tsuruga, Ikki-machi, Aizu-Wakamatsu, Fukushima 965-8580, Japan

Correspondence should be addressed to Yan Pei; peiyan@u-aizu.ac.jp

Received 28 July 2014; Revised 11 October 2014; Accepted 12 November 2014

Academic Editor: Albert Victoire

We present and discuss philosophy and methodology of chaotic evolution that is theoretically supported by chaos theory. We introduce four chaotic systems, that is, logistic map, tent map, Gaussian map, and Hénon map, in a well-designed chaotic evolution algorithm framework to implement several chaotic evolution (CE) algorithms. By comparing our previous proposed CE algorithm with logistic map and two canonical differential evolution (DE) algorithms, we analyse and discuss optimization performance of CE algorithm. An investigation on the relationship between optimization capability of CE algorithm and distribution characteristic of chaotic system is conducted and analysed. From evaluation result, we find that distribution of chaotic system is an essential factor to influence optimization performance of CE algorithm. We propose a new interactive EC (IEC) algorithm, interactive chaotic evolution (ICE) that replaces fitness function with a real human in CE algorithm framework. There is a paired comparison-based mechanism behind CE search scheme in nature. A simulation experimental evaluation is conducted with a pseudo-IEC user to evaluate our proposed ICE algorithm. The evaluation result indicates that ICE algorithm can obtain a significant better performance than or the same performance as interactive DE. Some open topics on CE, ICE, fusion of these optimization techniques, algorithmic notation, and others are presented and discussed.

1. Introduction

Philosophy of determinism comes from the development of classic mechanic that was originally established and studied by Isaac Newton, Pierre-Simon Laplace, Gottfried Wilhelm Leibniz, and so forth. Strict determinism indicates that causality can be expressed and implemented by mathematical calculation and logical reasoning. As Pierre-Simon Laplace said, "we may regard the present state of the universe as the effect of its past and the cause of its future" [1]. In the philosophy of determinism, everything is deterministic and predictable. However, the discovery of probability breaks dominated position of determinism in scientific philosophy. Particularly, the proposal of law of larger numbers extends the recognition scale of science, which explains the relationship between probability and frequency. The two philosophies and methodologies, determinism and probability, dominate researches in science until the discovery of chaos, another philosophy and methodology which can present and explain the natural world. In optimization field, deterministic and stochastic optimization algorithms are theoretically supported by philosophy and methodology of determinism and probability. However, because some fundamental works of chaos theory are not completed yet, research and development of chaotic optimization algorithm are still rarely mentioned and studied in optimization field. This paper tries to present a limit work on chaotic optimization algorithm with evolution concept.

Most of evolutionary computation (EC) algorithms are inspired from natural phenomena, such as genetic algorithm that mimics the process of natural selection and survival of the fittest. EC can be involved in continuous and combinational optimization problems, and its algorithm has a metaheuristic or stochastic characteristic in their search mechanisms [2]. As EC algorithms have been developed further and deeply, not only biological phenomena but also physical and mathematical phenomena and mechanisms are introduced into EC area to implement new EC algorithms. This research subject is one of computational paradigm studies in natural computing area and enriches research

context of computational intelligence. In a viewpoint of EC search scheme, its algorithms encompass two components in their search mechanisms. One is a search methodology by a variety of implementations, and the other is an iterative process to simulate evolution. Most of EC algorithms do not require the optimization problem to have some specific characteristics, and few of them utilize any mathematical properties or mechanisms to ensure convergence of the algorithms. If we introduce mathematical optimization property or mechanism that simulates the natural phenomena, such as chaotic ergodicity, into an optimization iterative process, we may implement new EC algorithms that partially ensure their convergence. The hint and philosophy behind this motivation benefit are to improve the global convergence characteristic of the algorithm.

The study on chaos theory comes from the real problems of physics, ecology, and mathematics since three-body problem was studied by Poincare et al. [3, 4]. It is developed originally to describe the system behaviour that cannot be described either by deterministic system or by stochastic system, which enriches our knowledge on unpredictable property of natural system. That means the system cannot be normalized by a set of differential equations or probability density function. Chaos and chaotic system have many characteristics for implementing an evolutionary search prototype and framework and improving EC algorithms. Nonlinearity, ergodicity, and sensitiveness are major explicit properties of chaos and chaotic system. The ergodicity of chaotic system can support more diversity to enhance exploration and exploitation capability of EC algorithms. It can approach to any desired point in search space with arbitrary accuracy and movement track. The sensitiveness of initial condition of chaotic system can lead to different search paths and escape from local optima for enhancing search performance of EC algorithms. The motion of perturbation around chaotic attractor can be simulated as an optimization search scheme and framework by EC.

There are typically three applications in which chaos is used in optimization area, that is, a local search method, a parameter tuning technique, and the new EC algorithm inspiration resource. Some chaotic systems are introduced into conventional EC algorithm frameworks to make population diversity in local search and to tune the parameters of algorithm [5]. The ergodicity property and behaviour are modelled as a search prototype to make a new EC algorithm, such as chaotic evolution optimization framework [6]. Chaotic Krill Herd algorithm was proposed by fusing Krill Herd algorithm and chaos theory [7], and some of its improved versions were proposed and studied [8, 9].

Chaotic evolution (CE) is a population-based algorithm framework that simulates chaotic motion behaviour in a search space [6]. Because of the ergodicity of chaotic system, chaotic motion can visit any point with arbitrary accuracy. CE algorithms take into account ergodicity property in its search mechanism to implement a new optimization scheme. There are three parameters as algorithm setting in its framework, a chaotic system and its parameter(s), a direction rate to guide the percentage of search direction, and a crossover rate. The chaotic system supports a basic simulation parameter, chaotic parameter, to implement the search function. Parameter of the chaotic system is simply set at the value that can lead to a chaotic output of the system. The direction rate decides search direction of each individual in CE algorithm, which is usually set at a random value. The crossover operation reserves extensibility of CE algorithm for further study, which can be set at 100%. It means that the mutant vector of CE algorithm can be directly presented as the chaotic vector of CE algorithm. For an empirical study, it is not a necessary parameter in CE optimization framework [6]. However, we cannot deny its effectiveness in theory from our current best knowledge.

This paper extends the work of [6] by introducing new chaotic systems into the CE algorithm framework and conducts a comparison study with DE algorithms. We use 25 benchmark functions with 10- and 30-dimensional setting to evaluate our new designed CE algorithms. We also establish a new interactive EC (IEC) algorithm by using CE optimization framework and evaluate its performance by using a pseudo-IEC user, that is, a Gaussian mixture model. Some statistical tests, such as Wilcoxon sign-ranked test, Friedman test, and Bonferroni-Dunn's test, are applied to evaluate significance and ranking of the algorithms in the related experiments and discussions. This work does not pursue obtaining a final winner algorithm from comparisons but to discover and discuss the fundamental aspect of CE from viewpoint of chaotic optimization and optimization mechanism of CE algorithm with different chaotic systems behind the evaluation results. Some study subjects are analysed and discussed taking into account evaluation results and statistical tests. These topics are the relationship between distribution characteristic of chaotic system and optimization performance, algorithm ranking, interactive CE algorithm, disadvantages and improvement approaches of CE algorithm, a notation system of CE, and so forth. These research subjects present the originality and primary contribution of this paper.

Following this introductory section, an overview of the CE algorithm framework and chaotic systems used in this paper are reported in Section 2. We present several new CE algorithms in this paper, which demonstrates scalability of the chaotic algorithm framework. Interactive CE (ICE) algorithm framework is introduced in Section 2 as well. In Section 3, experimental evaluations of chaotic evolution are performed and some statistical tests are applied. We discuss our proposed CE algorithms and obtained evaluation results. In Section 4, we analyse our proposed new IEC algorithm. We discuss some issues based on ICE experimental simulation using a pseudo-IEC user, which is implemented by a Gaussian mixture model. The topic on algorithmic notation of CE is defined for its further development and study. Finally, we conclude the whole work, and some open topics, further opportunities, and future works are discussed in Section 5.

2. Chaotic Evolution Algorithm Framework, Interactive Chaotic Evolution, and Chaotic System

2.1. A Brief Review of Chaotic Evolution Algorithm. Deterministic system and stochastic system are two views in

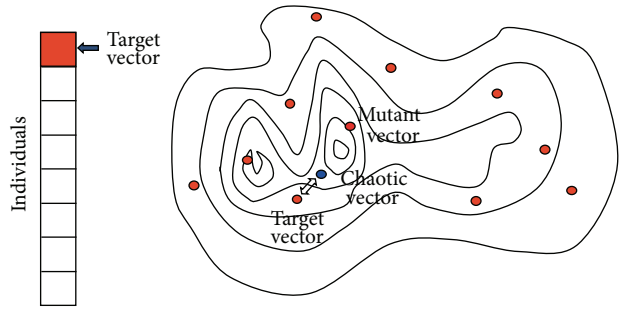


Figure 1: Chaotic evolution algorithm framework; there are three special vectors in chaotic evolution algorithm, a target vector that is an individual in the population, a mutant vector that simulates the chaotic motion from a chaotic system, and a chaotic vector that is made by conducting crossover operation on a target vector and a mutation vector.

which we understand and describe the nature in philosophy and in science. They lead to two corresponding algorithm methodologies in optimization field, that is, deterministic and stochastic optimization algorithms. Evolutionary computation can be partially categorized into the later one, that is, stochastic optimization algorithm. However, since chaotic phenomenon and mechanism were found [3, 4], chaos and chaotic system are as well a tool to describe and study the nature besides deterministic and stochastic systems. In optimization field, there must be a set of chaotic optimization algorithms inspired and normalized by chaotic philosophy and methodology. This is the fundamental motivation that prompts us to discover and study new optimization algorithms in the viewpoint of chaotic optimization.

The concepts of "evolution" and "chaos" have more the same characteristics in common [6]. First, both words refer to the phenomena that need to be explained and to the theories that are supported to do the explaining [10]. Second, there must be an iteration in both evolution and chaos processes, so the phenomena of them can appear. Third, the concepts and theories of evolution and chaos influence and enhance all research works by introducing and fusing the fundamentals of their concepts and approaches.

Inspired from chaotic motion of nonlinear system that has an ergodicity property, a chaotic ergodicity based EC algorithm, *chaotic evolution*, was proposed and studied recently [6]. It simulates the chaotic motion in a search space for implementing optimization. Because chaotic motion has an ergodicity property, the proposed algorithm can guarantee its global convergence partially. The scalability of the algorithm is better than other EC algorithms by introducing different chaotic systems.

Suppose that an array on the left side of Figure 1 presents individuals, contour lines at the right side are a fitness landscape, and circles on the landscape are the individuals. CE algorithm for one search generation is described below. It is repeated until a satisfied solution is found or the search reaches to the desired generations.

(1) Choose one individual as a target vector.

(2) Obtain a chaotic parameter from a chaotic system

$$\mathrm{CP}_i = \text{Chaotic System}\left(\mathrm{CP}_{i-1}\right). \quad (1)$$

(3) Make a mutant vector by

$$\text{Mutant Vec.}_i = \text{Target Vec.}_i * \left(1 + D_i * \mathrm{CP}_i\right). \quad (2)$$

(4) Generate a chaotic vector by crossing the target vector and the mutant vector

$$\text{Chaotic Vec.}_i = \mathrm{Cr}\left(\text{Mutant Vec.}_i, \text{Target Vec.}_i\right). \quad (3)$$

(5) Compare the target vector and the chaotic vector and choose whichever a better one as offspring in the next generation.

(6) Go to (1) and generate other offspring until all individuals are replaced with offspring in the next generation.

The terms of *vector* and *individual* mean the same search points. (1)–(4) are summarized as in (1), (2), and (3). This sketches the chaotic evolution algorithm which is easily implemented, where D_i is called a direction factor and CP_i is a chaotic parameter. There are several chaotic evolution variations in (1) for selecting different chaotic systems, crossover methods in (4), and others. After initial population needs fitness evaluation once, only the chaotic vector in (5) needs a fitness evaluation in every generation.

2.2. Interactive Chaotic Evolution Algorithm. Interactive evolutionary computation (IEC) is a niche research field in EC community [11]. The primary objective of IEC purposes solving the optimization problems that should embed subjective evaluation of real human into the optimized solution. Most of the evaluation spaces of IEC application are hard to be explicitly expressed or cannot be represented at all, so the real human's evaluation is necessary to achieve the final optimal solution. However, human fatigue is a serious issue for an IEC optimization application, because a real human has his or her evaluation limitation, and physiological or psychological tolerance capacity comparing with a tirelessness computer. So to relieve human's fatigue is a practical problem in IEC algorithm study and application.

From a framework viewpoint, there are three main parts in an IEC based optimization system. They are (1) a target system that is optimized, (2) an IEC algorithm (including an IEC interface) that conducts actual optimization function, and (3) a real human who provides his or her evaluation. The target system and the real human are fixed in an IEC optimization application, so there is a study subjective to relieve human's fatigue in the IEC algorithm part. One is to enhance optimization performance of IEC algorithm. The other is to improve IEC interface for a user friendly interaction.

If we implement a CE algorithm in an IEC application and replace the fitness function with a real human's

evaluation, the IEC framework is an implementation of interactive chaotic evolution (ICE). In the nature of ICE algorithm, there is a paired comparison-based mechanism when surviving an offspring between chaotic vector and target vector. In an IEC application, these two vectors present two IEC solutions for a real human to perform evaluations. Comparing with other IEC algorithms, such as interactive genetic algorithm (IGA), this paired comparison mechanism benefits real human's evaluation rather than evaluating all individuals together as in IGA. If the ICE algorithm can present significantly optimization performance compared to other paired comparison-based IEC algorithms, such as interactive differential evolution (IDE) [12] or its variations [13], ICE can be a practical algorithm implementation in IEC framework and enrich IEC algorithm family. We will investigate this subject in this paper.

2.3. Chaotic Systems. The search function of CE is implemented by generating a chaotic vector. It is decided by chaotic parameter from a chaotic system, which has the ergodicity property. By applying a variety of chaotic systems in the CE algorithm framework, we can implement its variations in different way. In this study, we introduce four chaotic systems, that is, logistic map, tent map, Gaussian map, and Hénon map to investigate the optimization performances of CE with these four chaotic maps. Here, we make a brief review on these chaotic maps.

2.3.1. Logistic Map. The logistic map is a polynomial mapping with two degrees that can arise from a chaotic phenomenon by a simple nonlinear system easily [14]. Equation (4) shows its mathematical expression, where μ is a parameter that decides the behaviour of system; μ is usually set at $(0, 4]$. For the values of $\mu > 4$, the map does not project interval $[0, 1]$ into itself. When $\mu = 4$, the system arises from chaotic phenomenon. The bifurcation diagram of logistic map is shown in Figure 2(a) that presents the whole system behaviour:

$$x_n = \mu * x_{n-1} * (1 - x_{n-1}). \tag{4}$$

2.3.2. Tent Map. The tent map is defined as in (5) whose function curve is like the tent-like shape. The parameter μ is within $(0, 2]$. Depending on the value of μ, the tent map demonstrates a range of dynamical behaviours from predictable to chaotic (see its bifurcation diagram in Figure 2(b)). When μ equals 2, the map becomes chaotic:

$$x_n = \begin{cases} \mu * x_{n-1} & \text{if } 0 \le x_n < 0.5, \\ \mu * (1 - x_{n-1}) & \text{if } 0.5 \le x_n \le 1. \end{cases} \tag{5}$$

2.3.3. Gaussian Map. The Gaussian map is a nonlinear map that is given by a Gaussian function (equation (6)), whose bifurcation diagram resembles a mouse (Figure 2(c)). The system can become chaotic, when the parameters α and β are set at 6.20 and −0.5, respectively:

$$x_n = \exp\left(-\alpha * x_{n-1}^2\right) + \beta. \tag{6}$$

Table 1: Test functions (Uni = unimodal, Multi = multimodal, Sh = shifted, Rt = rotated, GB = global on Bounds, HC = hybrid composition, and NM = number matrix).

Number	Type	Characteristic	Bounds	Optimum fitness
f_1	Uni	Sh sphere	$[-100, 100]$	−450
f_2		Sh Schwefel 1.2		−450
f_3		Sh Rt Elliptic		−450
f_4		f_2 with noise		−450
f_5		Schwefel 2.6 GB		−310
f_6	Multi	Sh Rosenbrock	$[-100, 100]$	390
f_7		Sh Rt Griewank	$[0, 600]$	−180
f_8		Sh Rt Ackley GB	$[-32, 32]$	−140
f_9		Sh Rastrigin	$[-5, 5]$	−330
f_{10}		Sh Rt Rastrigin	$[-5, 5]$	−330
f_{11}		Sh Rt Weierstrass	$[-0.5, 0.5]$	90
f_{12}		Schwefel 2.13	$[\pi, \pi]$	−460
f_{13}		Sh expanded F8F2	$[-3, 1]$	−130
f_{14}		Sh Rt Scaffer F6	$[-100, 100]$	−300
f_{15}	Hybrid	HC function	$[-5, 5]$	120
f_{16}		Rt HC function 1		120
f_{17}		f_{16} with noise		120
f_{18}		Rt HC function 2		10
f_{19}		f_{18} with basin		10
f_{20}		f_{18} with GB		10
f_{21}		Rt HC function 3		360
f_{22}		f_{21} with NM		360
f_{23}		NC Rt f_{21}		360
f_{24}		Rt HC function 4		260
f_{25}		f_{24} without bounds		260

2.3.4. Hénon Map. The Hénon map has two points (x_n, y_n) in the plane, and it maps them into a new point (equation (7)). The system behaviour depends on its parameters, that is, a and b. Different from the other chaotic maps, Hénon map is a two-dimensional nonlinear system. It is chaotic when its parameters have values of $a = 1.4$ and $b = 0.3$, which is a classic parameter setting. For other values of a and b, the map may be chaotic or intermittent or converge to a periodic orbit (Figure 2(d)). The Hénon map does not have a strange attractor for all values of the parameters a and b, but there are invariant points on the attractor; $x = (\sqrt{609} - 7)/28 = 0.631354477$ and $y = (3\sqrt{609} - 7)/280 = 0.189406343$ are examples of them:

$$\begin{aligned} x_n &= y_n + 1 - a * x_n^2, \\ y_n &= b * x_n. \end{aligned} \tag{7}$$

3. Chaotic Evolution Optimization Evaluation Analysis and Discussion

3.1. Experimental Settings and Results. We use 25 benchmark functions from [15] to evaluate our proposed CE algorithms with new introduced chaotic systems. Table 1 presents their type, characteristic, search bound, and optimum fitness value

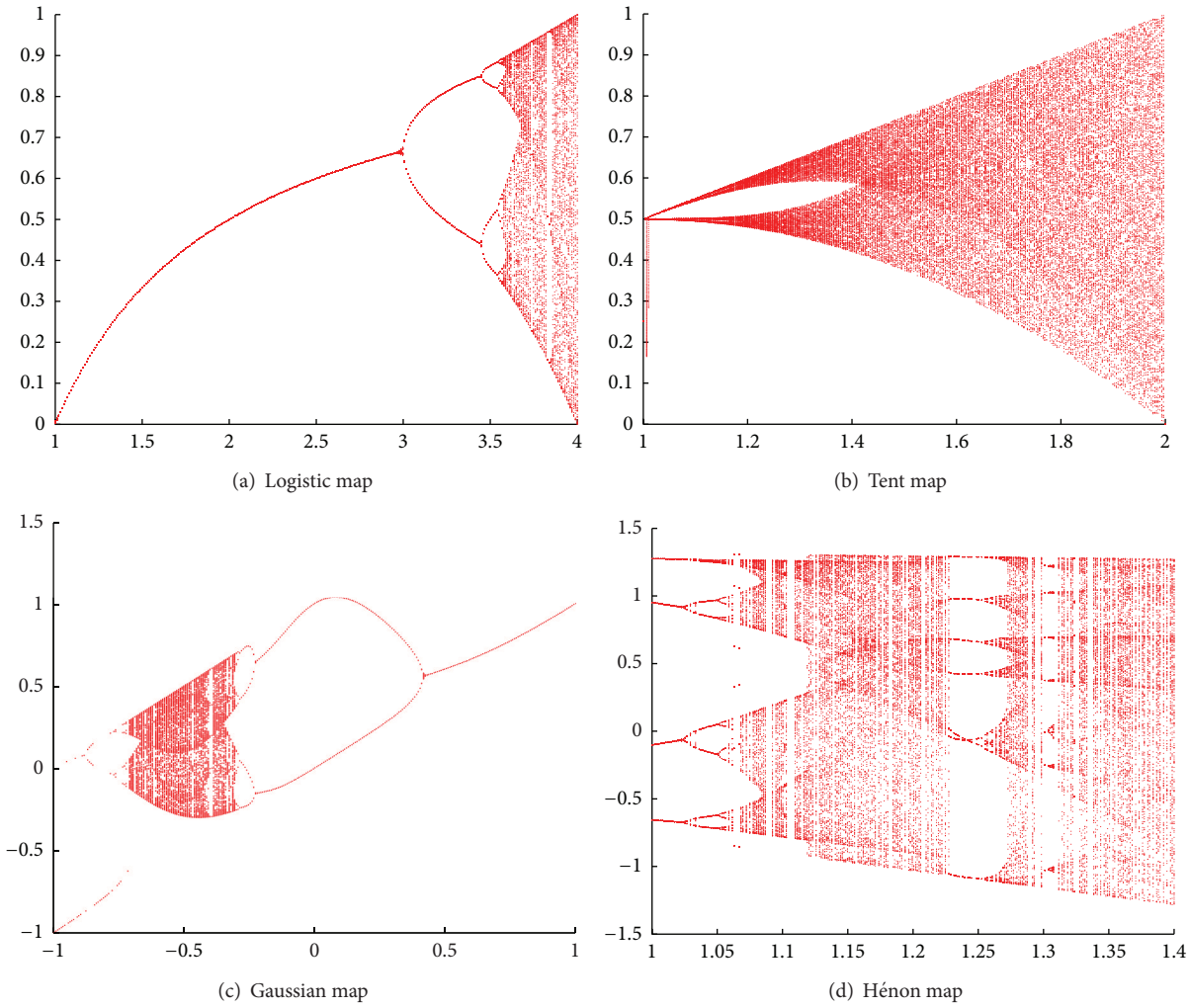


FIGURE 2: Bifurcation digrams of (a) logistic map, (b) tent map, (c) Gaussian map, and (d) Hénon map. Parameter μ in logistic map can be set at $(0, 4]$; chaos happens when $\mu = 4$ in (a). Parameter μ in tent map can be set at $[1, 2]$; chaos happens when $\mu = 2$ in (b). Parameter β in tent map can be set at $[-1, +1]$; chaos happens when $\alpha = 6.20$ and $\beta = -0.5$ in (c). Parameters $a = 1.4$ and $b = 0.3$ are a classic setting in Hénon map; one of its invariant points on the attractor, $x = (\sqrt{609} - 7)/28 = 0.631354477$ and $y = (3\sqrt{609} - 7)/280 = 0.189406343$, can be used as an initial value setting.

of these benchmark functions. For all benchmark functions, its dimensional settings are 10 and 30. For all algorithms, it runs up to 1000 generations with 30 trails running. The population size is $5 * \text{dimension}$; that is, 50 and 150 are for 10D and 30D benchmark functions, respectively.

There are five CE algorithms in our evaluation, that is, CE-logistic, CE-tent, CE-Gauss, CE-Hénon-rand, and CE-Hénon-attrac. Two DE algorithms (DE/best/1/bin and DE/rand/1/bin) are used to be compared with CE algorithms [16]. Abbreviations and their meaning are in Table 2. CE with Hénon map has two initialization methods. One uses random value within $(0, 1]$ (CE-Hénon-rand), and the other uses an attractor, $x = (\sqrt{609} - 7)/28 = 0.631354477$ and $y = (3\sqrt{609} - 7)/280 = 0.189406343$ (CE-Hénon-attrac.). Other chaotic systems of chaotic evolution algorithms use the uniform random number within $(0, 1]$ as their initial value. The evaluation hardware environment is PC with Windows 8.1 (x64) on Intel(R) Core(T) i7-4500 (CPU@1.80 GHz, 2.39 GHz, 4 GRAM); the algorithms are implemented in MATLAB (R2011b).

We apply Wilcoxon sign-ranked test and Friedman test on the fitness value at 1000th generation and make two groups with and without DE algorithms to rank the algorithms. We use Bonferroni-Dunn's test to check the significance of algorithm rank in both groups. Tables 3 and 4 are the fitness mean value of ALL algorithms at 1000th generation for 10D and 30D benchmark functions, respectively.

3.2. Chaotic Evolution Optimization Performance. One of the objectives in this paper is to investigate the optimization performance by using different chaotic systems in CE algorithm framework. Tables 3 and 4 show the mean of fitness value at the 1000th generation of each algorithm for 10D and 30D benchmark functions. The fitness values with bold font and italic font indicate the best and the worst fitness values among five proposed CE algorithms. From these two tables, we can

Table 2: Algorithm parameter setting and abbreviations of the algorithms used in evaluation.

Abbreviation	Meaning
CE-logistic	Chaotic evolution with logistic map (random initialization within $(0, 1]$, $\mu = 4$).
CE-tent	Chaotic evolution with tent map (random initialization within $(0, 1]$, $\mu = 2$).
CE-Gauss	Chaotic evolution with Gaussian map (random initialization within $(0, 1]$, $\alpha = 6.2$, $\beta = -0.5$).
CE-Hénon-rand	Chaotic evolution with Hénon map (random initialization within $(0, 1]$, $a = 1.4$, $b = 0.3$).
CE-Hénon-attrac.	Chaotic evolution with Hénon map (initialization with attractors explained in Figure 2, $a = 1.4$, $b = 0.3$).
CE crossover rate	1
DE/best/1/bin	DE with the best individual as base vector.
DE/rand/1/bin	DE with random individual as base vector.
DE scale factor F	1
DE crossover rate	1

conclude that CE with tent map obtains the worst fitness value for most of 10D and 30D benchmark functions. CE with Gaussian map and Hénon map using attractor initialization obtains the best results in 10D and 30D benchmark functions. It presents the fact that CE algorithm with tent map is less useful for most of benchmark tasks practically.

We apply Wilcoxon sign-ranked test ($P < 0.05$) on results with the best and the worst fitness values; most of the results indicate that there is a significant difference between all pairs. It addresses the fact that the optimization performance of CE is influenced significantly by the chaotic system.

Figure 2 shows the output ranges of four bifurcation diagrams of chaotic system used in our proposed CE algorithms. For further investigating the output distribution of each chaotic system, we calculate their output distribution in equal intervals and their percentages in Table 5.

There are some relationships between output distribution characteristic of chaotic system and optimization performance of CE. In logistic map, most of the system outputs cover the intervals of $(0, 0.1]$ and $(0.9, 1]$, whose percentages are up to 20.55% and 20.52%, respectively. When making a mutation vector from a target vector, the mutation vector will locate in the position near the target vector with adding the length of $(0, 0.1] * \text{target}$ and far from it with adding that of $(0.9, 1] * \text{target}$. It presents exploitation and exploration functions of CE algorithm with the logistic map. Most of the output values locate in the interval $(0, 0.1]$ in tent map, which indicates that the exploitation capability of CE with tent map is better than the others. However, its evolution speed is slow due to lack of exploration function. This is a primary reason why CE with the tent map has the worst optimization results in most of benchmark tasks. It is a disadvantage of CE-tent algorithm. The outputs of Gaussian map and Hénon map seem averagely to locate in the intervals among $(0, 0.5]$ and $(0, 1]$, respectively. These distributions ensure the exploitation and exploration functions of CE with Gaussian map and Hénon map. So they show the better optimization performance among the five CE algorithms averagely.

3.3. Comparison with Differential Evolution. We select two DE algorithms (DE/best/1/bin and DE/rand/1/bin) as two competitors to make a comparative evaluation with our proposed five chaotic evolution algorithms. The † and ‡ marks in Tables 3 and 4 show that our proposed CE algorithms have significantly better or equal performance to the evaluation performances of DE/best/1/bin and DE/rand/1/bin from the Wilcoxon sign-ranked tests ($P < 0.05$), respectively. By comparing the winner number of CE algorithms in Tables 3 and 4, CE algorithms, which are applied to higher dimensional benchmark functions (30D), can obtain better optimization performance than that applied to lower dimensional benchmark functions (10D). Particularly for the benchmark tasks, F11, F13, F14, F18, F19, F20, and F25, whose fitness landscapes are more complex, our proposed algorithms can obtain significantly better optimization performance.

3.4. Algorithm Ranking. We apply Friedman test on our proposed five CE algorithms and two DE algorithms to make an algorithm rank. Table 6 is the rank results of seven competitive algorithms. We can observe that DE/best/1/bin is the winner algorithm for both 10D and 30D benchmark functions averagely. However, for the F10–F14, F18–F20, its ranks become down because optimization results of our proposed algorithms are better than those in these benchmark tasks. Proposed CE algorithms have powerful optimization capability in the benchmark tasks with more complex fitness landscape from Table 6, especially for the higher dimensional tasks (30D). From evaluation metric of the average rank, the proposed CE algorithms have almost the same rank for some pairs, for example, CE-logistic and CE- Hénon-rand and CE-Gauss and CE-Hénon-attrac. in the 10D group. The same observations can be found in the 30D group as well. Table 7 is the algorithm rank only among five CE algorithms. CE-Gauss and CE-CE-Hénon-attrac. are the winner algorithm from the metric of average rank. The rank scores among five algorithms are not different so much. We suppose that the comparison within five CE algorithms and the comparison within CE and DE (seven algorithms) may not present significant difference when they are applied to a variety of benchmark tasks, that is, in a large sample condition.

We apply Bonferroni-Dunn's tests on the results of Friedman test to evaluate our hypothesis whether these algorithms have a significant difference. The tests are applied on a group with DE to compare CE and DE and on a group without DE to compare the CE algorithms with different chaotic systems. Both tests are for the 10D and 30D benchmark tasks. The evaluation metrics of critical difference (CD) are calculated

Table 3: Mean fitness values of F1–F25 with 10D. The fitness values in bold and italic font are the best and worst optimization results among the five chaotic evolution algorithms, respectively. The (†) and (‡) marks mean that they are significantly better than or as the same as DE/best/1/bin and DE/rand/1/bin, respectively, by Wilcoxon sign-ranked test ($P < 0.05$).

Function	DE-best	DE-rand	CE-logistic	CE-tent	CE-Gauss	CE-Hénon-rand	CE-Hénon-attrac.
F1	−449.998	−414.47	2998.806	*5171.277*	1925.366	1768.834	**1661.428**
F2	−440.516	−421.736	4219.061	*7585.492*	**2667.176**	3855.883	3568.253
F3	1583.076	215058.7	3624018	26036921	**2686083**	4846787	*3973677*
F4	−283.931	−402.19	6487.652	*8960.625*	**4226.79**	4723.953	5358.734
F5	−310	−309.972	3775.085	*9448.24*	3986.86	4148.611	**1892.268**
F6	446.6181	35496.74	287017050.30	628335744.87	272134693.96	*78198866*	**22561790**
F7	−177.747	−177.865	−97.4097	*−22.6933*	**−127.847**	−105.609	−121.71
F8	−119.992	−119.893	−119.635	*−119.38*	−119.62	−119.604	**−119.644**
F9	−308.575	−306.239	**−284.906**	*−269.758*	−282.345	−284.484	−282.21
F10	−291.876	−282.316	−254.408	*−232.193*	−261.758	−258.089	**−264.328**
F11	98.34748	94.5809	98.25204†	*99.94741*	**97.20956**†	99.0253†	97.52258
F12	321.4523	4968.164	20706.57	53091.15	33802.12	28609.12	**16818.88**
F13	−126.975	−126.213	−125.596	*−123.459*	**−126.954**†‡	−124.246	−126.658†‡
F14	−296.159	−296.567	−296.327†	*−295.874*	**−296.44**†‡	−296.269†	−296.384†‡
F15	476.4005	557.9363	573.2322‡	*746.0844*	612.5314‡	621.6522‡	**547.8405**‡
F16	302.9829	337.3688	378.9592	*452.6815*	377.3656	391.311	**356.5135**
F17	323.7027	326.4775	409.4423	*474.5775*	401.6362	426.4379	**381.8373**
F18	869.8405	578.9428	**910**	**910**	**910**	*910.1011*	**910**
F19	869.4931	591.2802	910	910	**905.6865**	*910.5839*	910
F20	866.7625	610.5591	910	910	**905.6865**	*910.5858*	910
F21	1079.758	929.7804	1623.005	*1691.75*	**1545.881**	1657.526	1574.536
F22	1222.383	1164.546	1320.278	*1406.045*	1307.327	1309.518	**1298.61**
F23	1066.17	970.7627	1647.835	*1701.935*	**1539.408**	1649.8	1563.626
F24	824.7887	494.8996	1450.171	*1498.591*	1376.52	1398.889	**1270.075**
F25	1989.595	2001.341	2010.443	**2009.772**	2018.292	*2021.227*	2014.102

Table 4: Mean fitness values of F1–F25 with 30D. The explanations of special marks are as the same as in Table 3.

Function	DE-best	DE-rand	CE-logistic	CE-tent	CE-Gauss	CE-Hénon-rand	CE-Hénon-attrac.
F1	97.24339	10626.17	33437.54	*44676.41*	**22489.18**	34997.74	27669.15
F2	1513.336	22808.96	28931.33	*38786.76*	**23073.34**‡	31553.89	26444.3
F3	3203063	91581615	155259859.97	*435879301.22*	**101212084.10**	212799570.04	139163745.08
F4	10475.36	31198.64	32279.57‡	*43413.27*	**28657.47**‡	37707.88	33388.45
F5	3753.477	12014.48	18617.58	*27065.35*	19258.04	21922.04	**18336.66**
F6	10041703	917911267.58	11016735211.66	*18010453533.81*	6390791853.98	11349392992.19	**6359560226.21**
F7	−165.651	98.12331	1042.799	*1718.531*	**803.677**	1145.698	996.1877
F8	−119.769	−119.03	−119.027‡	*−118.906*	−119.022‡	**−119.041**‡	−119.031‡
F9	−224.472	−80.3587	−4.98262	*17.25496*	**−26.9584**	−15.6137	−20.0503
F10	−157.64	−16.854	170.6209	*238.9339*	156.3653	133.2562	**132.4319**
F11	124.2699	129.963	125.7166†‡	*130.6959*‡	**123.5876**†‡	130.007‡	124.5107†‡
F12	39573.98	945070.8	722534‡	*1120996*	1097621	875280.5‡	**678438.7**‡
F13	−94.4806	−72.5022	−78.4622‡	−64.6016	**−102.475**†‡	*−58.4634*	−94.5819†
F14	−286.7	−286.454	−286.813†‡	*−286.369*‡	−286.822†‡	−286.57†‡	**−286.962**†‡
F15	712.6463	679.5186	907.8008	*1046.292*	**832.5582**	943.7358	899.282
F16	394.5135	445.3347	645.2657	*811.5967*	633.273	684.447	**610.1459**
F17	456.3692	489.36	698.1601	*882.8512*	725.3444	715.6857	**664.8518**
F18	924.3655	976.0293	**910**†‡	**910**†‡	**910**†‡	*1029.592*	**910**†‡
F19	927.3029	973.5324	**910**†‡	**910**†‡	**910**†‡	*1026.967*	**910**†‡
F20	927.4663	971.5176	**910**†‡	**910**†‡	**910**†‡	*1014.319*	**910**†‡
F21	1335.783	1319.538	1665.937	*1697.951*	**1638.145**	1657.629	1657.872
F22	1260.922	1312.237	1552.663	*1698.287*	**1545.124**	1569.841	1559.818
F23	1418.319	1374.563	1667.109	*1695.831*	**1633.517**	1659.605	1662.561
F24	1218.33	1210.63	1618.306	*1638.226*	1614.015	**1593.555**	1621.32
F25	1890.716	1944.179	1891.82†‡	**1889.155**†‡	1912.471‡	*1951.921*	1891.442†‡

Table 5: System output total number and its percentage in each interval of logistic map, tent map, Gaussian map, and Hénon map with attractor initiation after 10^5 iterations. The output of Hénon map with random initiation is out of the interval $(0, 1]$.

Interval	Logistic		Tent		Gaussian		Hénon-attrac.	
$(0, 0.1]$	20552	20.55%	99952	99.95%	7319	10.77%	2193	4.52%
$(0.1, 0.2]$	9051	9.05%	6	0.01%	13830	20.36%	3272	6.74%
$(0.2, 0.3]$	7396	7.40%	9	0.01%	10366	15.26%	3263	6.72%
$(0.3, 0.4]$	6674	6.67%	3	0.00%	13236	19.48%	7039	14.49%
$(0.4, 0.5]$	6353	6.35%	7	0.01%	23191	34.13%	5410	11.14%
$(0.5, 0.6]$	6473	6.47%	6	0.01%	0	0.00%	5505	11.34%
$(0.6, 0.7]$	6691	6.69%	1	0.00%	0	0.00%	7601	15.65%
$(0.7, 0.8]$	7296	7.30%	3	0.00%	0	0.00%	5606	11.54%
$(0.8, 0.9]$	8996	9.00%	6	0.01%	0	0.00%	3402	7.01%
$(0.9, 1]$	20518	20.52%	7	0.01%	0	0.00%	5272	10.86%

with (8), in which k is the number of algorithms and N is the number of benchmark functions:

$$\mathrm{CD} = q * \sqrt{\frac{k * (k+1)}{6 * N}}. \tag{8}$$

In our evaluation, $k = 7$ and $k = 5$ are for the groups with and without DE and $N = 25$ for both groups. Parameter q is critical value for nonparametric multiple-comparison testing with a control [17]; we use $q_{\alpha=0.01,k=7} = 3.144$ and $q_{\alpha=0.05,k=7} = 2.639$ and $q_{\alpha=0.01,k=5} = 3.024$ and $q_{\alpha=0.05,k=5} = 2.498$ in our calculations of CD. The results are $\mathrm{CD}_{q_{\alpha=0.01,k=7}} = 1.92$ and $\mathrm{CD}_{q_{\alpha=0.05,k=7}} = 1.61$ and $\mathrm{CD}_{q_{\alpha=0.01,k=5}} = 1.35$ and $\mathrm{CD}_{q_{\alpha=0.05,k=5}} = 1.11$. By comparing with the average values of mean ranking, there are not any CD values excluding average values in each group (taking the algorithm with the less rank value as a control algorithm). These results indicate that (1) CE algorithms have the same optimization capabilities comparing with the DE algorithms, and (2) there is not any significant difference among CE algorithms with different chaotic systems setting, when CE is applied to a variety of benchmark tasks (in the viewpoint of large sample condition).

4. Interactive Chaotic Evolution Optimization Evaluation Analysis and Discussion

4.1. Experimental Settings and Results. We propose using CE algorithm framework to implement a new IEC algorithm, interactive chaotic evolution (ICE). It is one of originalities and contributions in this paper. We use a Gaussian mixture model (equation (9)) as a pseudouser to evaluate ICE algorithm. Dimension setting of the Gaussian mixture model is 3, 5, 7, 10, 13, 15, 17, and 20. The population size of the algorithms is 20 and we evaluate them with 20 generations to simulate the characteristic of IEC algorithms, that is, with less population size and less generation. Other parameter settings are as well in Section 3; we use interactive version of DE, that is, IDE, to compare it with the new proposed ICE. Table 8 is the mean value of all the algorithms at 20th generation; we apply Wilcoxon sign-ranked test to evaluate the significance of results. Figure 3 shows convergent curves of 5D, 10D, 15D, and 20D Gaussian mixture models:

$$\mathrm{GMM}(\mathbf{x}) = \sum_{i=0}^{k} a_i \exp\left(-\sum_{j=0}^{n} \frac{\left(x_{ij} - \mu_{ij}\right)^2}{2\sigma_{ij}^2}\right), \tag{9}$$

where

$$\sigma = \begin{pmatrix} 1.5 & 1.5 & 1.5 & 1.5 & 1.5 & 1.5 & 1.5 & 1.5 & 1.5 & 1.5 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \end{pmatrix},$$

$$\mu = \begin{pmatrix} -1 & 1.5 & -2 & 2.5 & -1 & 1.5 & -2 & 2.5 & -1 & 1.5 \\ 0 & -2 & 3 & 1 & 0 & -2 & 3 & 1 & 0 & -2 \\ -2.5 & -2 & 1.5 & 3.5 & -2.5 & -2 & 1.5 & 3.5 & -2.5 & -2 \\ -2 & 1 & -1 & 3 & -2 & 1 & -1 & 3 & -2 & 1 \end{pmatrix},$$

$$a_i = \left(3.1,\ 3.4,\ 4.1,\ 3.0\right)^T. \tag{10}$$

4.2. Interactive Chaotic Evolution and Its Paired Comparison Mechanism. We analyse the optimization results of CE algorithms at the 1000th generation in Section 3. From the evaluation results and statistical tests, our proposed CE algorithms have better optimization performance in the initial 10 or 20 generations than that of DE algorithms in some benchmark tasks. This feature of CE algorithm can benefit the IEC [11], so ICE algorithm should be a powerful algorithm to the IEC application.

One of the characteristics in IEC is that it is required to run with less generation and less population size to solve the user fatigue problem. IEC user compares each individual with others kept in their memory and the mental load and fatigue increase. Reference [18] pointed out that human cannot process more than five to nine different items simultaneously. The population sizes of many IEC applications frequently exceed this memory limitation, and displaying more than five to nine sounds or movies, that is, time series object, to an IEC user, is not practical. Paired and multiple comparison-based IEC solves this problem by replacing the comparison of all individuals with paired or multiple comparisons [13]. It is therefore expected to reduce IEC user fatigue.

Table 6: Algorithms' rank with DE by Friedman test for F1–F25 of 10D and 30D, respectively. The abbreviation meanings are in Table 2.

Function	DE-best	DE-rand	Logistic	Tent	Gauss	Hénon-rand	Hénon-attrac.
				10-dimensional function			
F1	1.00	2.00	5.07	6.93	4.33	4.47	4.20
F2	1.03	1.97	5.17	6.87	3.73	4.73	4.50
F3	1.03	1.97	4.40	6.90	3.77	5.33	4.60
F4	1.70	1.30	5.30	6.60	4.03	4.50	4.57
F5	1.00	2.00	4.63	6.83	4.83	5.00	3.70
F6	1.00	2.00	5.20	6.77	4.50	5.03	3.50
F7	1.33	1.67	5.17	6.77	3.83	5.03	4.20
F8	1.07	2.23	4.30	6.70	4.43	4.93	4.33
F9	1.53	1.67	4.37	6.70	4.70	4.27	4.77
F10	1.33	2.17	5.07	6.83	4.13	4.50	3.97
F11	4.37	1.23	4.37	6.47	2.80	5.63	3.13
F12	1.13	2.00	3.87	6.83	5.63	5.10	3.43
F13	2.43	3.53	4.43	6.67	2.50	5.83	2.60
F14	5.17	2.80	3.50	6.70	2.47	4.30	3.07
F15	2.43	3.83	3.37	6.77	4.03	4.50	3.07
F16	1.43	2.60	4.53	6.90	4.13	5.10	3.30
F17	1.67	1.60	4.70	6.87	4.27	5.30	3.60
F18	4.90	1.27	3.83	5.38	3.83	4.95	3.83
F19	4.67	1.23	3.92	5.53	3.78	4.95	3.92
F20	4.70	1.50	3.87	5.40	3.77	4.90	3.87
F21	1.67	1.63	5.03	6.57	3.43	5.47	4.20
F22	2.17	1.10	4.97	6.97	4.37	4.37	4.07
F23	1.37	1.77	5.17	6.67	3.90	5.07	4.07
F24	2.07	1.20	5.43	6.17	4.60	4.73	3.80
F25	1.17	2.70	4.07	3.80	5.70	5.93	4.63
Ave.	**2.13**	**1.96**	**4.55**	**6.46**	**4.06**	**4.96**	**3.88**
				30-dimensional function			
F1	1.00	2.00	5.20	6.93	3.10	5.77	4.00
F2	1.00	2.87	4.77	6.97	2.70	5.77	3.93
F3	1.00	2.50	4.63	6.93	2.97	5.80	4.17
F4	1.03	3.33	3.97	6.87	2.57	5.80	4.43
F5	1.00	2.03	4.03	6.90	4.33	5.67	4.03
F6	1.00	2.00	5.47	6.90	3.50	5.43	3.70
F7	1.00	2.00	4.63	6.93	3.60	5.33	4.50
F8	1.00	3.93	4.37	6.77	4.27	3.60	4.07
F9	1.00	2.10	5.37	6.60	3.73	4.73	4.47
F10	1.03	1.97	5.23	6.87	4.90	3.90	4.10
F11	2.77	5.73	3.30	6.20	1.90	5.77	2.33
F12	1.00	4.80	2.80	6.33	6.20	4.37	2.50
F13	2.60	4.60	4.13	5.83	1.60	6.57	2.67
F14	3.93	5.67	2.90	6.20	2.67	4.80	1.83
F15	2.03	1.47	4.67	6.70	3.20	5.40	4.53
F16	1.53	1.70	4.37	6.80	4.40	5.33	3.87
F17	1.43	1.70	4.40	6.73	5.03	4.80	3.90
F18	5.00	6.30	2.02	3.95	2.02	6.70	2.02
F19	5.20	6.23	2.02	3.95	2.02	6.57	2.02
F20	5.17	6.33	2.02	3.95	2.02	6.50	2.02
F21	1.70	1.30	5.23	6.87	3.47	4.70	4.73
F22	1.07	1.93	4.30	6.97	4.27	4.87	4.60

Table 6: Continued.

Function	DE-best	DE-rand	Logistic	Tent	Gauss	Hénon-rand	Hénon-attrac.
F23	1.63	1.37	5.27	6.63	3.40	4.83	4.87
F24	1.37	1.63	5.10	6.30	4.67	3.77	5.17
F25	2.53	6.37	2.80	2.07	4.97	6.63	2.63
Ave.	**1.96**	**3.27**	**4.12**	**6.17**	**3.50**	**5.34**	**3.64**

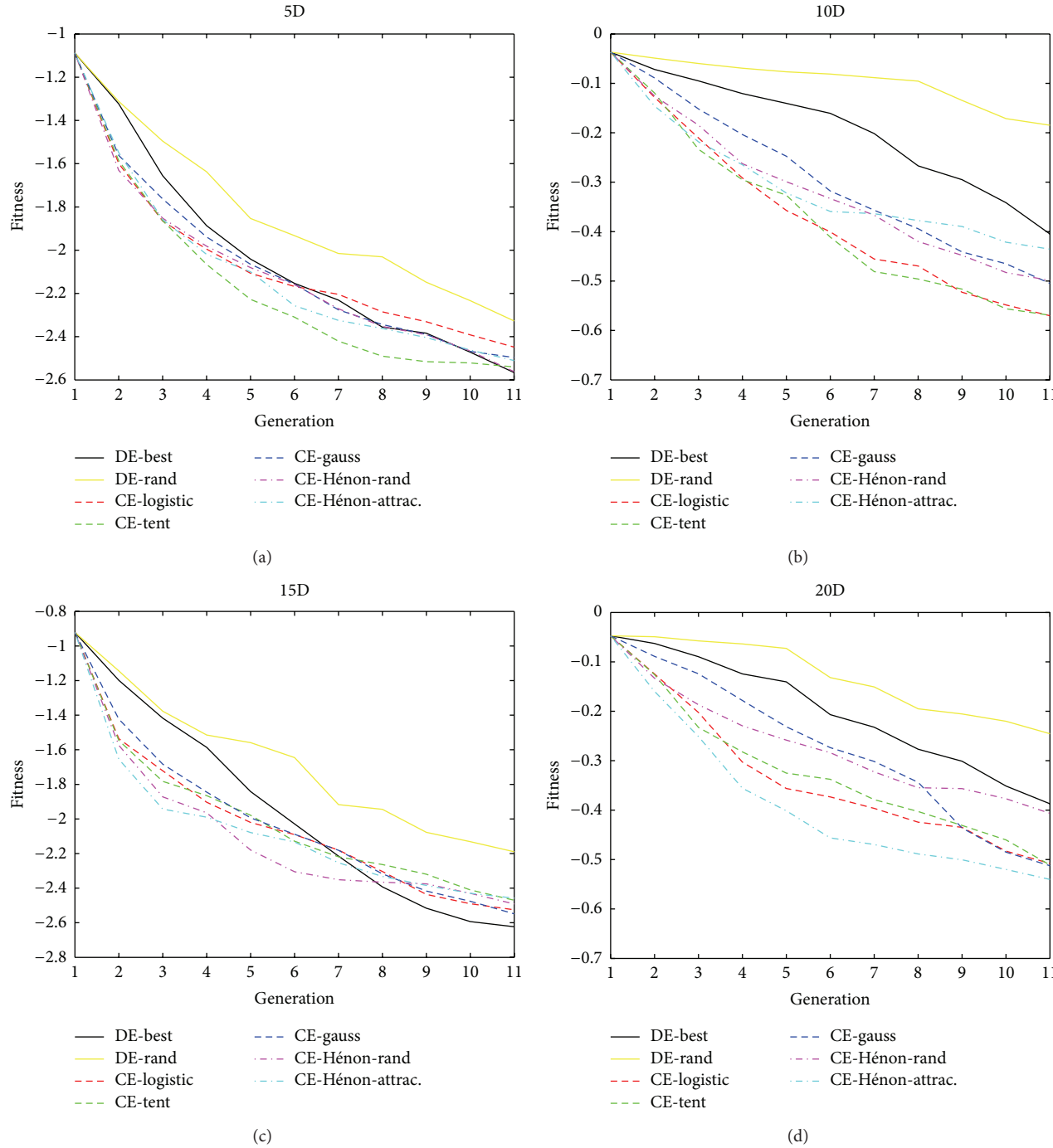


Figure 3: Interactive chaotic evolution convergent curves of 5D, 10D, 15D, and 20D Gaussian mixture model. All the ICE algorithms present fast convergent speed, especially in 10D and 20D tasks. IDE/best can obtain better final results than ICE, but it needs more fitness evaluation (i.e., more human subjective evaluation in IEC application) to find the best base vector. From the metric of fitness evaluation time, ICE is significantly better than IDE/best when these two algorithms call the same fitness function time.

Table 7: Algorithms' rank without DE by Friedman test for F1–F25 of 10D and 30D, respectively. The abbreviation meanings are in Table 2.

Function	Logistic	Tent	Gauss	Hénon-rand	Hénon-attrac.
			10-dimensional function		
F1	3.07	4.93	2.33	2.47	2.20
F2	3.17	4.87	1.73	2.73	2.50
F3	2.40	4.90	1.77	3.33	2.60
F4	3.30	4.60	2.03	2.50	2.57
F5	2.63	4.83	2.83	3.00	1.70
F6	3.20	4.77	2.50	3.03	1.50
F7	3.17	4.77	1.83	3.03	2.20
F8	2.33	4.73	2.57	2.97	2.40
F9	2.40	4.70	2.77	2.37	2.77
F10	3.13	4.83	2.20	2.57	2.27
F11	2.90	4.73	1.53	3.97	1.87
F12	1.93	4.83	3.63	3.10	1.50
F13	3.00	4.70	1.50	4.03	1.77
F14	2.67	5.00	1.73	3.27	2.33
F15	2.27	4.90	2.77	3.13	1.93
F16	2.67	4.90	2.47	3.23	1.73
F17	2.73	4.87	2.33	3.30	1.77
F18	2.47	4.02	2.47	3.58	2.47
F19	2.48	4.10	2.42	3.52	2.48
F20	2.50	4.03	2.43	3.53	2.50
F21	3.07	4.57	1.53	3.57	2.27
F22	3.00	4.97	2.43	2.43	2.17
F23	3.17	4.67	1.97	3.07	2.13
F24	3.43	4.17	2.70	2.77	1.93
F25	2.27	2.10	3.73	4.03	2.87
Ave.	**2.77**	**4.58**	**2.33**	**3.14**	**2.18**
			30-dimensional function		
F1	3.20	4.93	1.10	3.77	2.00
F2	2.87	4.97	1.20	3.83	2.13
F3	2.67	4.93	1.33	3.80	2.27
F4	2.37	4.87	1.17	3.87	2.73
F5	2.07	4.90	2.33	3.67	2.03
F6	3.47	4.90	1.50	3.43	1.70
F7	2.63	4.93	1.60	3.33	2.50
F8	2.73	4.80	2.70	2.17	2.60
F9	3.37	4.60	1.73	2.77	2.53
F10	3.23	4.87	2.90	1.90	2.10
F11	2.70	4.67	1.50	4.33	1.80
F12	1.73	4.47	4.33	3.03	1.43
F13	3.03	4.23	1.07	4.73	1.93
F14	2.50	4.77	2.23	3.87	1.63
F15	2.73	4.70	1.50	3.43	2.63
F16	2.43	4.80	2.43	3.33	2.00
F17	2.43	4.77	3.03	2.80	1.97
F18	2.02	3.95	2.02	5.00	2.02
F19	2.02	3.95	2.02	5.00	2.02
F20	2.02	3.95	2.02	5.00	2.02
F21	3.23	4.87	1.47	2.70	2.73

Table 7: Continued.

Function	Logistic	Tent	Gauss	Hénon-rand	Hénon-attrac.
F22	2.30	4.97	2.27	2.87	2.60
F23	3.27	4.63	1.40	2.83	2.87
F24	3.10	4.30	2.67	1.77	3.17
F25	2.27	1.60	4.00	5.00	2.13
Ave.	**2.66**	**4.53**	**2.06**	**3.53**	**2.22**

Table 8: Means of Gaussian mixture models with 3D, 5D, 7D, 10D, 13D, 15D, 17D, and 20 D. The abbreviation meanings are in Table 2. The (†) and (‡) marks mean that the proposed algorithms are significantly better than or as the same as IDE/best/1/bin and IDE/rand/1/bin, respectively, by Wilcoxon sign-ranked test ($P < 0.05$).

	DE-best	DE-rand	Logistic	Tent	Gauss	Hénon-rand	Hénon-attrac.
3D	−5.32	−5.05	−4.81	$-4.96^{\dagger\ddagger}$	$-5.36^{\dagger\ddagger}$	−4.86	$-5.08^{\ddagger}$
5D	−2.47	−2.23	$-2.39^{\dagger\ddagger}$	$-2.52^{\dagger\ddagger}$	$-2.47^{\dagger\ddagger}$	$-2.47^{\dagger\ddagger}$	$-2.46^{\dagger\ddagger}$
7D	−1.50	−1.00	$-1.17^{\ddagger}$	$-1.13^{\ddagger}$	$-1.36^{\dagger\ddagger}$	$-1.25^{\dagger\ddagger}$	$-1.31^{\dagger\ddagger}$
10D	−0.34	−0.17	$-0.55^{\ddagger}$	$-0.56^{\ddagger}$	$-0.46^{\ddagger}$	$-0.48^{\ddagger}$	$-0.42^{\ddagger}$
13D	−5.20	−5.17	−4.73	$-4.98^{\ddagger}$	$-5.32^{\dagger\ddagger}$	−4.81	$-5.09^{\ddagger}$
15D	−2.59	−2.13	$-2.49^{\dagger\ddagger}$	$-2.41^{\ddagger}$	$-2.48^{\dagger\ddagger}$	$-2.43^{\dagger\ddagger}$	$-2.43^{\dagger\ddagger}$
17D	−1.32	−0.87	$-1.23^{\dagger}$	$-1.24^{\dagger\ddagger}$	$-1.30^{\dagger\ddagger}$	$-1.28^{\dagger\ddagger}$	$-1.29^{\dagger\ddagger}$
20D	−0.35	−0.22	$-0.48^{\dagger\ddagger}$	$-0.46^{\dagger\ddagger}$	$-0.48^{\dagger\ddagger}$	$-0.38^{\dagger\ddagger}$	$-0.52^{\dagger\ddagger}$

In the running process of CE algorithm framework (Section 2.1), there are paired comparison mechanisms in its principle when surviving the target vector and the chaotic vector into the next generation. If we apply the ICE algorithm for a concrete IEC application, it is expected that the optimization performance of ICE may be better than that of IDE from the initial optimization results from Section 3. Table 8 sketches that ICE algorithms are significantly better than or are as the same as IDE/best/1/bin and IDE/rand/1/bin. In the viewpoint of fitness evaluation number, ICE algorithms are as the same as IDE/rand/1/bin but less than IDE/best/1/bin, because human user must compare all individuals together to select the best individual as the base vector. ICE presents better optimization performance than that of IDE/best/1/bin. That is, it can reduce human evaluation time and obtain optimization results better or at least as the same as IDE/best/1/bin algorithms, especially in initial generation shown in Figure 3.

4.3. Fusion of Chaotic Evolution and Differential Evolution. Evaluation results indicate that the convergent speeds of CE algorithms become down in some benchmark functions, especially in 15D tasks (Figure 3). However, convergence speeds of DE algorithms seem to be with the same tendency in Figure 3. If we fuse CE and DE together to make a hybrid algorithm optimization framework, the performance of hybrid algorithm should expectedly be better than any of the single algorithms. For example, when the convergence speeds of DE become slow, we can apply the CE algorithm with multiple chaotic systems to increase its population diversity and vice versa. Fusion algorithm can be implemented in dimension level, individual level, and generation level. This is a promising topic for the further investigation.

There are multiple implementations of the CE algorithm framework. We initially have investigated their optimization performance and output distribution of chaotic systems. Another opportunity for enhancing optimization performance of CE is to fuse multiple chaotic systems in one CE algorithm. It is as well a promising research topic for further study. There are three methods to implement this algorithm framework. First is to apply one chaotic system for a certain generation and change another for the following generations by observing the algorithm convergent speed or other evaluation metrics. Second is to fuse CE with multiple chaotic systems in individual level; that is, some of individuals are searching by the law of one chaotic system and some of the others by that of other chaotic systems. Third is to apply different chaotic systems on different dimensions of one individual. The landscape or search situation is different from all dimensions, and this must be matched by different dimensional searches with different chaotic systems. We can design a CE algorithm with multiple chaotic systems by obtaining the fitness landscape information in the whole or the particle dimensions. Some methods of analysing and approximating the fitness landscape information can be found in [19, 20]. This is a research subject in our future work.

4.4. Algorithmic Notation of Chaotic Evolution. We develop an algorithmic notation system for a better explanation and further development of CE algorithms. The actual search function of chaotic algorithms is implemented by a chaotic system, in which there are some parameters. This is one element in chaotic evolution. Another parameter of chaotic evolution is the crossover rate.

We abbreviate chaotic evolution as CE and use the notation format CE/x/y/z to make a note of a certain chaotic evolution algorithm, where x, y, and z present a chaotic system, the parameters of chaotic system, and a crossover rate, respectively. For example, the CE algorithm with Gaussian

map in this paper can be a notation as "CE/Gaussian/($\alpha = 6.2$, $\beta = -0.5$, random initialization)/1."

5. Conclusion and Future Works

In this paper, we develop a chaotic optimization algorithm that is theoretically supported by the fundamental of chaos theory. We introduce four chaotic systems in a well-designed CE algorithm framework to implement several CE algorithms, that is, CE-logistic, CE-tent, CE-Gauss, CE-Hénon-rand, and CE-Hénon-attrac. We analyse the optimization performances of these developed algorithms. A comparative evaluation is conducted by applying the Wilcoxon sign-ranked test and the Friedman test with two DE algorithms. We propose a new IEC algorithm, that is, ICE that has a paired comparison mechanism in its search scheme. A series of topics on optimization performances of chaotic evolution algorithms, comparison with DE algorithms, algorithms rank, interactive chaotic evolution, and fusion of these algorithms for enhancing performance are analysed and discussed.

In this paper, we do not pursue obtaining the results of the best winner algorithm but analyse and discuss the algorithm optimization mechanism of CE and the philosophy behind it. In the modern scientific world, there are two primary philosophies and methodologies to describe and study nature world from the determinism and probability viewpoint. In the optimization field, there are corresponding deterministic and stochastic optimization algorithms that are supported by these two methodologies. EC is one of stochastic optimization algorithms. However, after the discoveries of chaotic phenomena and systems, chaos becomes the third methodology to study the nature world. Chaotic optimization algorithm should therefore be researched and developed in optimization field. Chaotic evolution is one of the implementations, although it has many disadvantages in its search scheme, such as searching without considering fitness landscape, exploration, and exploitation depending on chaotic system rather than adaptive mechanism. In the theoretical analysis of CE, we will analyse concrete CE algorithm with Markov chain [6] and other mathematical models. They are interesting research topics on evaluating CE performance by a variety of benchmark functions and comparing it with other metaheuristics algorithms [7–9, 21]. We will further investigate these topics in our future work and hopefully discover new knowledge behind CE algorithm. It should benefit both chaos theory and evolutionary optimization hopefully.

Conflict of Interests

The author declares that there is no conflict of interests regarding the publication of this paper.

References

[1] P. Simon, M. de Laplace, F. W. Truscott, and F. L. Emory, *A Philosophical Essay on Probabilities*, vol. 166, Dover, New York, NY, USA, 1951.

[2] Y. Pei and H. Takagi, "A survey on accelerating evolutionary computation approaches," in *Proceedings of the International Conference of Soft Computing and Pattern Recognition (SoCPaR '11)*, pp. 201–206, October 2011.

[3] F. Diacu and P. Holmes, *Celestial Encounters: The Origins of Chaos and Stability*, Princeton University Press, Princeton, NJ, USA, 1996.

[4] H. Poincare, *Les méthodes nouvelles de la mécanique céleste: Méthodes de MM. Newcomb, Glydén, Lindstedt et Bohlin. Volume 2*, Gauthier-Villars et fils, 1893.

[5] L. D. S. Coelho, T. C. Bora, and L. Lebensztajn, "A chaotic approach of differential evolution optimization applied to loudspeaker design problem," *IEEE Transactions on Magnetics*, vol. 48, no. 2, pp. 751–754, 2012.

[6] Y. Pei, "Chaotic evolution: fusion of chaotic ergodicity and evolutionary iteration for optimization," *Natural Computing*, vol. 13, no. 1, pp. 79–96, 2014.

[7] G.-G. Wang, L. Guo, A. H. Gandomi, G.-S. Hao, and H. Wang, "Chaotic krill herd algorithm," *Information Sciences*, vol. 274, pp. 17–34, 2014.

[8] G.-G. Wang, A. H. Gandomi, and A. H. Alavi, "A chaotic particle-swarm krill herd algorithm for global numerical optimization," *Kybernetes*, vol. 42, no. 6, pp. 962–978, 2013.

[9] G. Wang, L. Guo, H. Wang, H. Duan, L. Liu, and J. Li, "Incorporating mutation scheme into krill herd algorithm for global numerical optimization," *Neural Computing and Applications*, vol. 24, no. 3-4, pp. 853–871, 2014.

[10] L. Smith, *Chaos: A Very Short Introduction*, Oxford University Press, Oxford, UK, 2007.

[11] H. Takagi, "Interactive evolutionary computation: fusion of the capabilities of EC optimization and human evaluation," *Proceedings of the IEEE*, vol. 89, no. 9, pp. 1275–1296, 2001.

[12] H. Takagi and D. Pallez, "Paired comparison-based interactive differential evolution," in *Proceedings of the World Congress on Nature & Biologically Inspired Computing (NaBIC '09)*, pp. 475–480, Coimbatore, India, December 2009.

[13] Y. Pei and H. Takagi, "Triple and quadruple comparison-based interactive differential evolution and differential evolution," in *Proceedings of the 12th Workshop on Foundations of Genetic Algorithms*, pp. 173–182, ACM, 2013.

[14] R. M. May, "Simple mathematical models with very complicated dynamics," *Nature*, vol. 261, no. 5560, pp. 459–467, 1976.

[15] P. N. Suganthan, N. Hansen, J. J. Liang et al., "Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization," KanGAL Report 2005005, 2005.

[16] R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[17] J. H. Zar, *Biostatistical Analysis*, Pearson Education, Delhi, India, 2007.

[18] G. A. Miller, "The magical number seven, plus or minus two: some limits on our capacity for processing information," *Psychological Review*, vol. 63, no. 2, pp. 81–97, 1956.

[19] Y. Pei and H. Takagi, "Fourier analysis of the fitness landscape for evolutionary search acceleration," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '12)*, pp. 1–7, June 2012.

[20] Y. Pei and H. Takagi, "Accelerating IEC and EC searches with elite obtained by dimensionality reduction in regression spaces," *Evolutionary Intelligence*, vol. 6, no. 1, pp. 27–40, 2013.

[21] G. Wang and L. Guo, "A novel hybrid bat algorithm with harmony search for global numerical optimization," *Journal of Applied Mathematics*, vol. 2013, Article ID 696491, 21 pages, 2013.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2015, Article ID 903624, 7 pages
http://dx.doi.org/10.1155/2015/903624

Research Article

Reverse Engineering of Free-Form Surface Based on the Closed-Loop Theory

Xue Ming He,[1,2] Jun Fei He,[2] Mei Ping Wu,[1,2] Rong Zhang,[3] and Xiao Gang Ji[2]

[1] *Department of Jiangsu Province Key Laboratory of Advanced Food Manufacturing Equipment and Technology, Jiangnan University, Wuxi 214122, China*
[2] *School of Mechanical Engineering, Jiangnan University, Wuxi 214122, China*
[3] *School of Science, Jiangnan University, Wuxi 214122, China*

Correspondence should be addressed to Xue Ming He; hxuem2003@163.com

Received 6 July 2014; Revised 19 November 2014; Accepted 27 November 2014

Academic Editor: S. N. Deepa

To seek better methods of measurement and more accurate model of reconstruction in the field of reverse engineering has been the focus of researchers. Based on this, a new method of adaptive measurement, real-time reconstruction, and online evaluation of free-form surface was presented in this paper. The coordinates and vectors of the prediction points are calculated according to a Bézier curve which is fitted by measured points. Final measured point cloud distribution is in agreement with the geometric characteristics of the free-form surfaces. Fitting the point cloud to a surface model by the nonuniform B-spline method, extracting some check points from the surface models based on grids and a feature on the surface, review the location of these check points on the surface with CMM and evaluate the model, and then update the surface model to meet the accuracy. Integrated measurement, reconstruction, and evaluation, with the closed-loop reverse process, established an accurate model. The results of example show that the measuring points are distributed over the surface according to curvature, and the reconstruction model can be completely expressed with micron level. Meanwhile, measurement, reconstruction and evaluation are integrated in forms of closed-loop reverse system.

1. Introduction

Free-form curves change freely in complex form, not like the law curve which can be described by the analytic functions. So, free-form surfaces also change in complex form; the reverse of objects with free-form surface is a research hot spot [1, 2]. The current reverse of products can be able to get the CAD model after data collection, data processing, and model reconstruction. These processes are sequentially performed from data acquisition to model reconstruction in the open-loop state, which makes data collection error be always present in the whole process of reverse engineering and be passed to the final CAD model [3, 4].

Huang and Qian [5, 6] proposed the reconstructed method of dynamic surface model, based on the point clouds of curve that have been achieved, using Kalman filtering method to guide B-spline surface model reconstruction dynamically. That is, the acquisition of point data is dynamically combined with B-spline surface reconstruction. This method can make the data point parameterization, dynamically determine next best measurement points and can effectively measure and reconstruct model with low differences. However, the whole process has a large amount of computation, data redundancy, and low efficiency and cannot evaluate the quality of the reconstructed model in time. Evaluating the quality of reconstruction and modification can only be carried on after the reconstruction model was manufactured, by comparing this production with primary product. If the reconstruction model can be directly evaluated, the accuracy can be ensured, the costs can be significantly reduced and the cycle of the reverse engineering is shortened as well.

Based on this, we propose a closed-loop reverse method which includes measurement, reconstruction, and evaluation, using OpenGL 3D graphics library to realize visualization in Visual C++ compiler environment. First, the contact CMM adaptive captures point cloud data based on curvature

characteristics of the tested surface. Then fit point cloud to obtain surface model and calculate the error of check points on the fitting model and the corresponding points on the physical. If the error exceeds the threshold (the threshold is set according to the required accuracy), the actual measured value of the check points will be added to old point cloud, fitting model and checking again and updating fitted model until it meets the accuracy requirements. This method can reduce the error of data collected from the source and the error of reconstruction model and reverse free-form surfaces with high-precision and high efficiency.

2. Free-Form Surfaces Reverse Based on the Closed-Loop Theory

2.1. The Process of Reverse Designing Based on the Closed-Loop Theory. Conventional reverse process is an open-loop process as shown in Figure 1. Data collection and model reconstruction are independent of each other and are done in sequence.

This paper will bring the closed-loop theory into reverse process in order to contact data acquisition with CAD model reconstruction by online evaluation, which can make measurement and reconstruction with closed-loop. Model reconstruction is based on measured point cloud, meanwhile the model guides data supplement.

The concrete practice of the reverse engineering based on the theory of closed-loop is as following. Reconstruct surface model by using the initial measured point cloud and then extract some check points from the reconstructed surface model and measure actual value of those check points on the physical surface. Judge the accuracy of model whether to meet request by the contrast analysis of the check point error between the theoretical value and the actual value. If the error is less than the threshold, reconstructed model can be considered reliable and its accuracy can meet the requirements. Otherwise, the actual value of the check points must be added to the existing point cloud and fitted again, increase the density of new check points. Calculate theoretical value of those new check points and then measure those new check points and calculate its error of the theoretical value and the actual value; repeat this process until the model accuracy can meet the requirements. This method can improve model accuracy and avoid feature omissions. The principle of this method is shown in Figure 2.

This method can do online evaluation in the situations that the coordinate system for the measurement does not change. The evaluation results can visually reflect whether reconstructed surface is faithless and whether some feature data points are lost or away from the group. It can also reflect the effect of noise and can help extract unreliable model timely. The reverse process can be able to ensure the quality and accuracy, at same time its cycle will be shortened.

Figure 3 shows a part with free-form surface characteristics and will be used in this method. It contains three steps: CMM adaptive measurement, surface model reconstruction, and online evaluation.

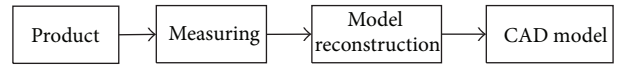


Figure 1: Traditional reverse process.

2.2. CMM Adaptive Measurement. The main types of data acquisition in the reverse process are contact and noncontact. Though noncontact measurement has high-efficiency, but it has lower accuracy. In this paper, touch-trigger CMM will be used to obtain point; the type of point data is scan lines style.

During the measurement, use the measured point fit a curve. Calculate the coordinates and vector direction of the predicted point according to the curvature characteristic of the curve. Then guide the CMM accurate measurement and obtain the actual value of the predicted point automatically.

Quintic Bézier curve has second derivative at any point of the curve and sequential, suitable for fitting scanning lines point cloud on most product surface. Figure 4 is the schematic of the Bézier curvature continuous adaptive measurement. Before adaptive measurement, some boundary points, the highest point, and the lowest point of the target surface should be measured by hand, the boundary points will construct four boundaries, and all adaptive measurement points must in the range of the boundaries. The coordinate of scan lines are calculated according to these boundaries. Assume initial measured points of every scan line are p_{i-5}, p_{i-4}, p_{i-3}, p_{i-2}, p_{i-1}, p_i, and these points are equally spaced, the distance of every two points and the vector direction of measurement of these six initial points are defined by operator, for example, Z direction. The six initial points p_{i-5}, p_{i-4}, p_{i-3}, p_{i-2}, p_{i-1}, p_i would be fitted to form a quintic Bézier curve [7], as follows, t is Parameter and $t \in [0, 1]$:

$$C(t) = \sum_{i=0}^{5} \frac{5!}{(5-i)!i!} (1-t)^{5-i} t^i p_i = \sum_{i=0}^{5} B_{5,i}(t)\, p_i. \quad (1)$$

This curve is assumed in plane YOZ, it can be decomposed in Y, Z directions as follows:

$$y = \alpha(t), \quad z = \beta(t), \quad t \in [0, 1]. \quad (2)$$

Curvature at any point on the curve:

$$k = \frac{\left|\alpha'(t)\beta''(t) - \alpha''(t)\beta'(t)\right|}{\left[\alpha'^2(t) + \beta'^2(t)\right]^{3/2}}. \quad (3)$$

The radius of curvature is as follows:

$$\rho = \frac{1}{k}, \quad t \in [0, 1]. \quad (4)$$

The p_{i+1} is predicted point, and it is calculated by the fitted curve. From the nature of Bézier curve we know that first control point is the curve starting point and end control point is the curve end point. When $t = 1$, the curvature radius of

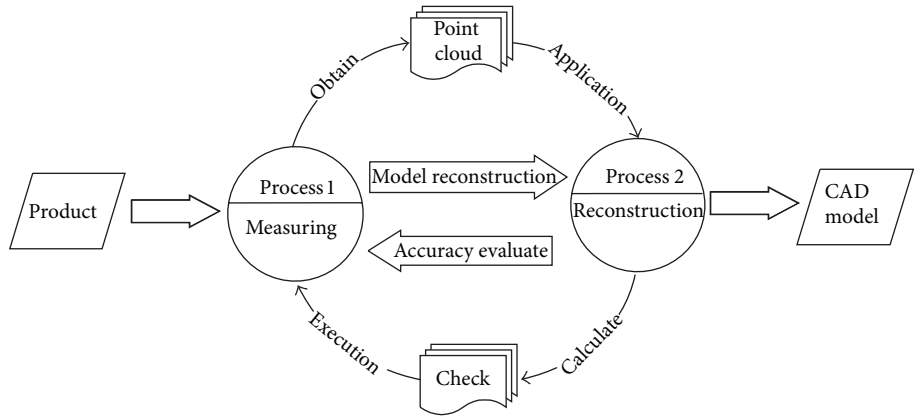


Figure 2: Reverse process based on the theory of closed-loop.

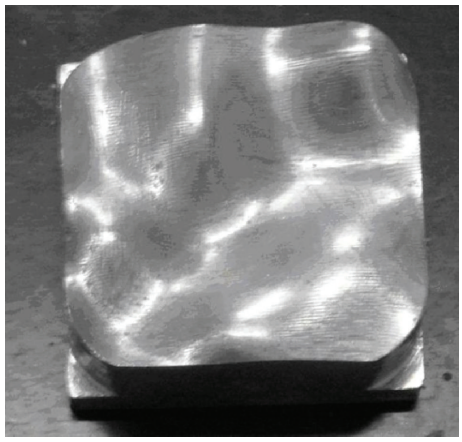


Figure 3: The product has free-form surface features.

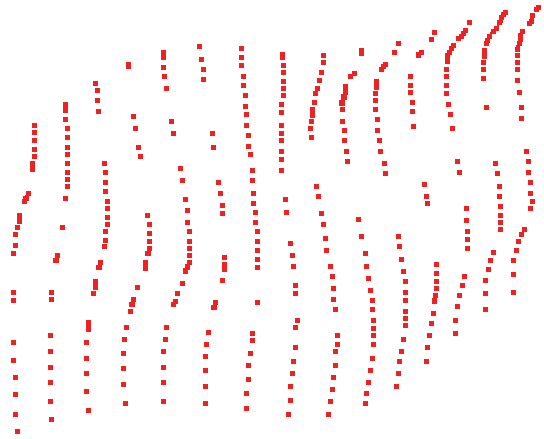


Figure 5: Point cloud by CMM adaptive measurement.

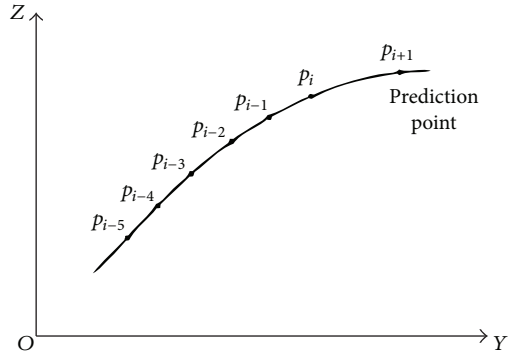


Figure 4: Adaptive measurement principle.

the curve can be obtained at the curve end point. The larger the radius of curvature, the smaller the sampling step. The smaller the radius of curvature is, the larger the sampling step is.

The Bézier curve is tangent to first and end of the edge of characteristic polygon. Assume the normal of point p_i is the vector direction of the predicted point p_{i+1}. After actual value of point is measured by CMM, then delete point p_{i-5}. The latest six points are p_{i-4}, p_{i-3}, p_{i-2}, p_{i-1}, p_i, p_{i+1}. Fit a new five Bézier curve and calculate point p_{i+2}. Repeat this process until this scan line is measured completely and then start to measure other scan line. The measuring point cloud shows in Figure 5.

Distribution of the measured point cloud is based on surface characteristics. Measuring points distribute densely in changeful areas and are small and sparsely in gentle areas. The number of initial measurement points is 419. Thus, the most complete information can be expressed with minimal points [8].

2.3. Real-Time Surface Model Reconstruction. Low-level B-spline surfaces are closer to data points relative to the high prices without shock and warp and better reflect the actual characteristics of the real thing. Therefore, this study uses three B-spline curves and surfaces to fit point clouds directly and obtain surface model [9]. In order to meet the conditions of fitting nonuniform B-spline surface, this paper interpolates each scan line point cloud with some nonuniform B-spline curves and resample in each interpolation curve to obtain data points with evenly distribution and quantity consistently.

If data points $(m+1)\times(n+1)$ were topology rectangular array on the scanning lines $n+1$, direction between the scanning lines and the scan lines of Figure 4 is assumed to be u direction and along the direction of the scan lines is v direction. The frequency of u and v direction is marked k and l and assumes that the number of k and l is three. It can define a 3×3 nonuniform B-spline surface as follows [10]:

$$p(u,v)=\sum_{i=0}^{m}\sum_{j=0}^{n}d_{i,j}N_{i,3}(u)\,N_{j,3}(v),\quad u,v\in[0,1]. \tag{5}$$

Use De Boer recursive formula [11] to calculate B-spline basis $N_{i,3}(u)$ $(i=0,1,\ldots,m)$ and $N_{j,3}(v)$ $(j=0,1,\ldots,m)$ according to knot vector u and v.

Construct nonuniform knot vector by using cumulative chord length method to guarantee the same type of distribution of knot vector and data points, that is, to ensure that the reconstruction of B-spline curves has higher quality. To ensure the characteristics of data points with interpolation, take clamped condition of quadruple knot endpoint. Mathematical expression of cumulative chord length parameterization is as follows:

$$\begin{gathered}u_0=u_1=u_2=u_3=0,\\ u_j=\frac{\sum_{i=1}^{j-3}L_i}{S},\quad j=4,5,\ldots,m+2,\\ u_{m+3}=u_{m+4}=u_{m+5}=u_{m+6}=1.\end{gathered} \tag{6}$$

u_j is a knot. L_i is distance between adjacent data points and S is the total length of the polyline posed by data points. Knot vector and topology rectangular array point cloud will decide the surface control point grid $d_{i,j}$ $(i=0,1,\ldots,m;\ j=0,1,\ldots,n)$ by (5) [12] and the reconstruction surface's control grid is as shown in Figure 6.

2.4. Online Evaluation. In this paper, an evenly distributed method is proposed to determine the check points, dividing the curved surface model into the grid type, extracting the center of each grid as check point. The number of check points H is determined by the amount of scanning lines L and check times r, as the specific relationship of (7). When the check times increased once, the number of check points has about a fourfold increase. The check precision increases and the time becomes longer with the addition to the quantity of checks [13, 14], and check times up to three times can reach micron-level precision in experimental process.

$$H=\left(L\times 2^{r-1}\right)\left(L\times 2^{r-1}\right). \tag{7}$$

Suppose that $\widehat{p}_i$ $(i=0,1,\ldots,H)$ represent check points extracted from reconstructed model, of which the coordinates and the normal vector information of curved surface where the points locate are used to instruct CMM to measure, getting the actual values of these points represented as p_i $(i=0,1,\ldots,H)$, Subsequently, the error of reconstruction model is as follows:

$$e=\frac{\sum_{i=0}^{n}\left|p_i-\widehat{p}_i\right|}{H+1}. \tag{8}$$

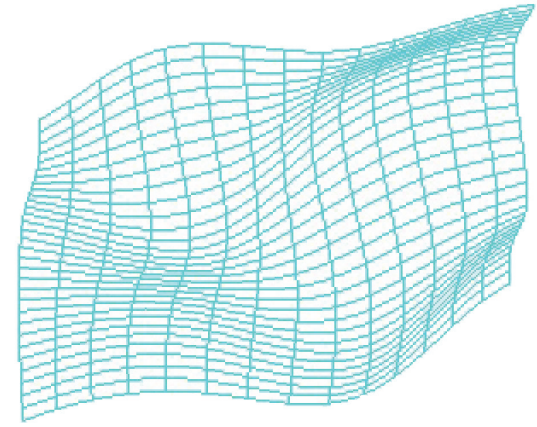


Figure 6: The surface's control grid.

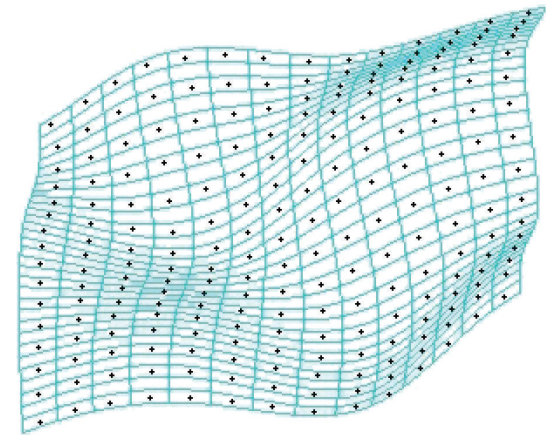


Figure 7: First check points.

Figure 7 shows the check points (black points) extracted from the curved surface model for the first time. In order to ensure the accuracy of inspection, check points cannot be same as the data points which are used for fitting.

Online accuracy evaluation of model is an important part of closed-loop reverse. If e is less than threshold (as in this example the threshold is 0.007 mm), it indicates that the reconstruction model is reliable and can be accepted. If e is greater than the threshold, the actual measured value of the check points in Figure 7 must be added to point cloud in Figure 5 to fit model again. Then, the curved surface grid is subdivided further according to (7) new check point for another inspection and evaluation is extracted.

This process is repeated until e becomes less than the threshold, and the closed-loop reverse process is completed. In this case, the initial evaluation result of mouse surface is that

e_1 is greater than the threshold, and the number of first check points is 225. The second evaluation result is $e_2 = 0.0068 < 0.007$, and the number of second check points is 900, and the overall operating time is 126 minutes, so this model is considered acceptable, as shown in Figure 8. The specific procedure of free-form curved surface measurement, reconstruction and evaluation of the closed-loop reverse system is shown in Figure 9.

3. Experimental Results

3.1. Free-Form Surface Reconstruction Algorithm Verification. Because the mathematical model of saddle surface is known, it is chosen as test object in order to inspect the reliability of the free-form surface reconstruction algorithm. The mathematical equations for the saddle surface are

$$z = \frac{(x-15)^2}{16} - \frac{(y-21.8301)^2}{25} + 5 \quad (x \in [0,30],\ y \in [0,43.6602]). \tag{9}$$

Figure 10(a) shows that a group of original points are calculated by mathematical saddle surface equation to reconstruct surface. First, the original points are fitted into a surface with B-spline curve and the surface reconstruction algorithm. Then, select a group of check points on this surface model. Finally, compare the check points with the corresponding points calculated by mathematical equation and get the error. The comparison results are shown in Figure 10(b). The maximum error is 0.00617 mm, which is micron-level. We can arrive at a conclusion from the results that this reconstruction algorithm is applicable to reverse modeling free-form surface products.

3.2. Examples of Applications. Figure 11 shows a precise arc surface cam. Its face area is G1 continuous, and side area is G2 continuous, while the intersection is G0 continuous. All of these form a complex composite surface. The working surface should be divided into some patches according to the principle that the curvature is continuous and then measured by the adaptive measurement method with CMM. The side can be measured adaptively due to curvature being continuous.

Figure 12 shows the point cloud data result from measuring the arc surface cam, and four patches are fitted with these point cloud data (for the convenience of reading, Figure 12 shows only about 1/3 of the point cloud and patches). The number of measurement points is 2655 and the operating time is 216 minutes. After fitting, the patches are evaluated for the first time, the number of first check points is 1532 and operating time is 125 minutes, the maximum e of working surface is 0.1070 mm, and the maximum e of side is 0.2944 mm. As a result, the first reconstruction patches are acceptable. Blend the six top patches to make a working surface, and the side patches to make a side surface and then extend and trim the working surface and side surface to obtain an entire model. Because the boundary of the cam cannot be measured by CMM, the quality and accuracy of blend faces cannot be controlled; just revise the blend faces according to the patches and ensure the entire surface's smoothness. The final CAD model is shown in Figure 13.

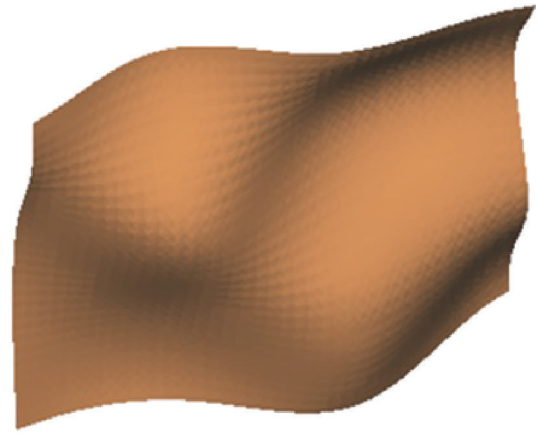


Figure 8: Final surface model.

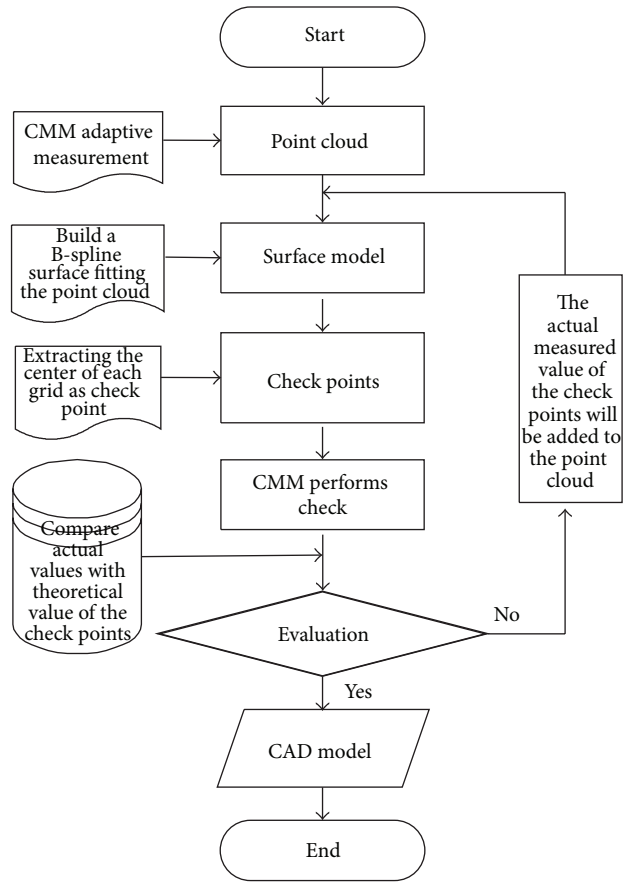
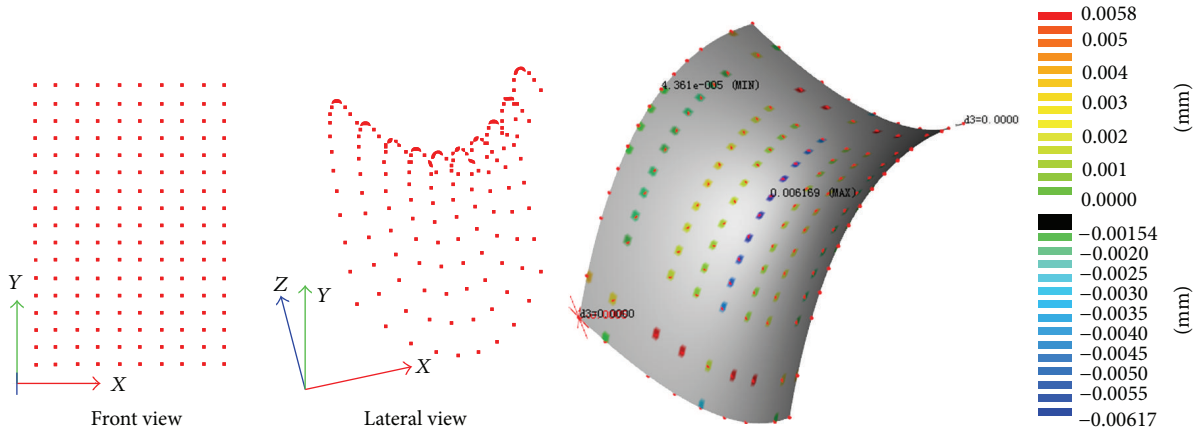


Figure 9: Flowchart of the closed-loop reverse engineering.



(a) Point cloud of the saddle surface (b) Reconstruction error results

Figure 10: Reconstruction algorithm verification.

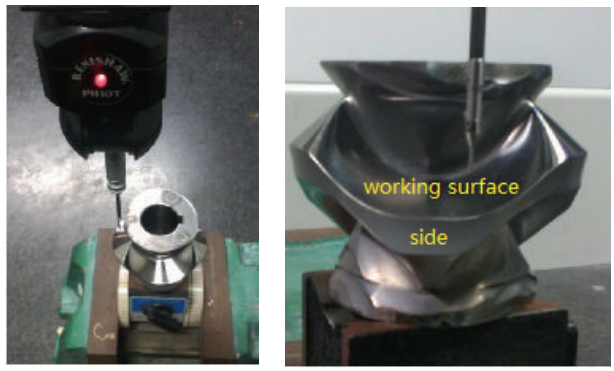
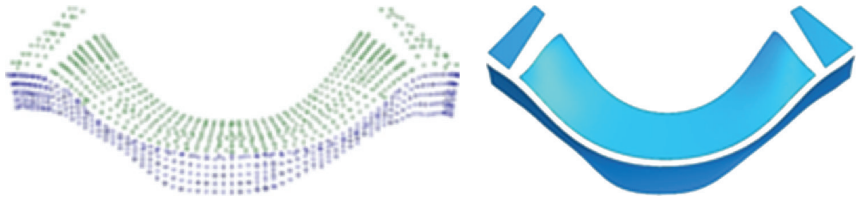


Figure 11: The arc surface cam.



(a) (b)

Figure 12: Point cloud and patches. (a) Measured points by CMM. (b) Patches by fitting the point cloud.

4. Conclusions

This paper has proposed a new reverse method based on the theory of closed-loop, achieving the information interaction between the measurement, reconstruction and evaluation. It avoids the separation between actual measurement, reconstruction and evaluation in traditional reverse process, and comparative evaluation can be carried out without reconstructed model manufactured. Aimed at the objects with free-form surface, that the geometric characteristics of real shape guide CMM to measure adaptively can come true, which results in the point cloud data with distribution reasonable and appropriate number. The surface model is obtained by using the method of non-uniform B-spline, on-line evaluated and updating model instead of evaluation after manufacturing. Finishing the process, precise measurement point cloud and accurate CAD model is obtained ultimately. The experimental results indicate that measurement accuracy and reconstruction accuracy of this method can reach micron-level. Additionally, the cost is evidently reduced, and the cycle is shortened obviously as well. Moreover, there is no need to make reconstruction model to compare and evaluate.

Figure 13: CAD model.

Conflict of Interests

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

Acknowledgments

This project is funded by Natural Science Foundation of China (51275210), (51105175), and Industry-University-Research Foundation of Jiangsu Province (BY2013015-30).

References

[1] P. Matej, I. Tomaz, B. Joze et al., "Reverse engineering of parts with optical scanning and additive manufacturing," *Procedia Engineering*, vol. 69, pp. 795–803, 2014.

[2] X. Zhao, C. Zhang, L. Xu, B. Yang, and Z. Feng, "IGA-based point cloud fitting using B-spline surfaces for reverse engineering," *Information Sciences*, vol. 245, pp. 276–289, 2013.

[3] A. Afeez and A. Kumar, "Application of CAD and reverse engineering methodology for development of complex assemblies," *Journal of Engineering, Design and Technology*, vol. 11, no. 3, pp. 375–390, 2013.

[4] T. L. Ruey and J. S. Fang, "Calculation of the unit normal vector using the cross-curve moving mask method for probe radius compensation of a freeform surface measurement," *Measurement*, vol. 43, no. 4, pp. 469–478, 2010.

[5] Y. Huang and X. Qian, "Dynamic B-spline surface reconstruction: closing the sensing-and-modeling loop in 3D digitization," *CAD Computer Aided Design*, vol. 39, no. 11, pp. 987–1002, 2007.

[6] Y. Huang and X. Qian, "A dynamic sensing-and-modeling approach to three-dimensional point- and area-sensor integration," *Journal of Manufacturing Science and Engineering*, vol. 129, no. 3, pp. 623–635, 2007.

[7] Z. M. Huo, H. Y. Quan, and S. K. Wang, "Five rational Bézier arc curve represents," *University Mathematics*, vol. 24, no. 1, pp. 39–46, 2008.

[8] X. He, C. Li, Y. Hu, P. Qu, and W. Li, "Continuous curvature adaptive planning of the measuring path for CMM," *Journal of Tsinghua University*, vol. 47, no. 2, pp. 1835–1839, 2007.

[9] M. Zhou, "A new approach of composite surface reconstruction based on reverse engineering," *Procedia Engineering*, vol. 23, pp. 594–599, 2011.

[10] L. Chen and D. T. Qing, "Algorithm and implementation of complex surface reconstruction of reverse engineering," *China Mechanical Engineering*, vol. 13, no. 6, pp. 505–508, 2002.

[11] X. Wu M and J. Wen, "Surface reconstruction based on CMM point cloud," *Journal of Engineering Graphics*, vol. 32, no. 2, pp. 68–72, 2011.

[12] J. Qu, J. Li, and Y. Ke, "Measured data supplement and surface reconstruction in reverse engineering of sculpture surface," *Chinese Journal of Mechanical Engineering*, vol. 38, no. 9, pp. 110–117, 2002.

[13] P. Pedone and D. Romano, "Designing small samples for form error estimation with coordinate measuring machines," *Precision Engineering*, vol. 35, no. 2, pp. 262–270, 2011.

[14] R. Ascione, G. Moroni, S. Petrò, and D. Romano, "Adaptive inspection in coordinate metrology based on kriging models," *Precision Engineering*, vol. 37, no. 1, pp. 44–60, 2013.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2015, Article ID 148010, 11 pages
http://dx.doi.org/10.1155/2015/148010

Review Article

State of the Art of Fuzzy Methods for Gene Regulatory Networks Inference

Tuqyah Abdullah Al Qazlan,[1] Aboubekeur Hamdi-Cherif,[2] and Chafia Kara-Mohamed[1]

[1]*Information Technology Department, Computer College, Qassim University, Buraydah 51452, Saudi Arabia*
[2]*Computer Science Department, Computer College, Qassim University, Buraydah 51452, Saudi Arabia*

Correspondence should be addressed to Aboubekeur Hamdi-Cherif; elhamdi62@gmail.com

Received 27 July 2014; Accepted 3 December 2014

Academic Editor: Albert Victoire

To address one of the most challenging issues at the cellular level, this paper surveys the fuzzy methods used in gene regulatory networks (GRNs) inference. GRNs represent causal relationships between genes that have a direct influence, trough protein production, on the life and the development of living organisms and provide a useful contribution to the understanding of the cellular functions as well as the mechanisms of diseases. Fuzzy systems are based on handling imprecise knowledge, such as biological information. They provide viable computational tools for inferring GRNs from gene expression data, thus contributing to the discovery of gene interactions responsible for specific diseases and/or *ad hoc* correcting therapies. Increasing computational power and high throughput technologies have provided powerful means to manage these challenging digital ecosystems at different levels from cell to society globally. The main aim of this paper is to report, present, and discuss the main contributions of this multidisciplinary field in a coherent and structured framework.

1. Introduction

Life in all living organisms is perfectly controlled by rigorous processes, thus sustaining stability in highly complex ecosystems from cellular level to society. The unprecedented amount of data generated by the Human Genome Project and the powerful sequencing and microarray technologies has to be carefully managed. *Ecosystem management combines the structuring and understanding of ecological information to facilitate the decision making in order to meet the society goals* [1]. Among the key elements in the cellular ecosystem we address, we emphasize gene regulatory networks (GRNs) as suitable means for representing causal relationships between genes. It is now widely accepted that GRNs have a direct influence on the survival and development of living organisms. Furthermore, GRNs offer a useful support for the understanding of the cellular functions and the mechanism of the diseases because they are able to reflect all interactions between genes with their products that determine the spatial and temporal expression patterns of a set of genes [2]. Gene interactions research studies have provided several useful applications such as new drugs discovery that act on regulatory pathways and the development of tracking methods in the dynamics of disease evolution within cells. GRNs inference is based on determining the genes that affect the expression of other genes and on adequately describing these effects.

Several methods have been developed for inferring GRNs and producing hypotheses about the presence or absence of interactions among genes, hypotheses that can later be tested by laboratory experiments [3]. In this context, various models have been proposed, such as Boolean networks and Bayesian networks for discrete models and ordinary differential equations (ODEs) and weight matrices for continuous models [4]. The ODEs representation was the most common model used while the piecewise linear differential equations (PLDEs) have been proposed for some approximations and simplifying nonlinear models. At a higher level, we attempt to show how machine learning can help in developing better bioinformatics methods and tools in a coherent manner by focusing on soft computing methods [5], with fuzzy methods being one of these.

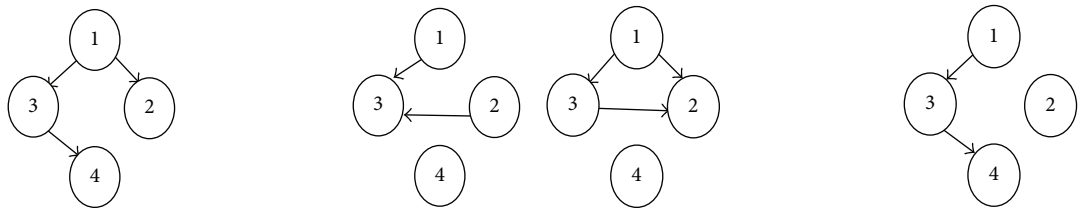


Figure 1: (a) Two GRN structures before inference. (b) GRN structures after evolvement during the inference process, whereby arcs are swapped between the nodes 3-4 and 2-3.

Although the gene regulation has an inherent fuzzy nature, most of GRN studies were based on crisp algorithms [6]. The main motivation for using fuzzy methods lies in the fact that uncertainty and imprecision are inherent in biological data. This is due to the nature of biological experiments themselves. Therefore, the fuzzy approach, as a qualitative computational approach, can be considered as a suitable formalism to deal with the imprecision intrinsic to many biological problems, knowing that GRN inference is one of the most challenging ones [7].

Fuzzy logic, as a fundamental component of the fuzzy approach, is a combination of various mathematical principles for representing the knowledge depending on a gradual degree of membership instead of crisp membership available in Boolean logic. Fuzzy logic consists of several sets that can be used to map a certain input to an output, a process referred to as fuzzy inference. The two most well-known inference techniques are those due to Mamdani and Tagaki-Sugeno. These models depend on using simple language of IF-THEN rules in the description of the required system response as a function of several linguistic variables. Furthermore, fuzzy models are extremely robust and the input data can work with small or no tuning at all. Indeed, in the fuzzy models, there is no need for accurate inputs since it is naturally imprecise. Conversely, when the number of inputs and outputs increases, its complexity unfortunately increases too. The main benefits of fuzzy methods are based on the generality of function estimators, clarity, modularity, ability to be explained, easy handling of uncertainty, and parallel processing of rules. On the other hand, the main limitations that restrict the use of these systems are the high computational costs, comprehensibility, and optimization. These limitations are inherent in bioinformatics settings [8].

Despite the fact that fuzzy systems present powerful framework for GRN inference, the corresponding literature is diversified and scattered in numerous journals, conference proceeding, and Web sites. It is urgent that these contributions be compiled in a coherent and structured body of knowledge; hence this paper attempts to bridge this gap.

The paper is organized as follows. Section 2 describes the GRN inference problem and the fuzzy approach to solve it. Section 3 discusses mainstream fuzzy methods used in the GRN inference and Section 4 is devoted to hybrid methods. Section 5 summarizes the different methods. Section 6 discusses an example of GRN inference for *Saccharomyces cerevisiae*. The paper ends with a conclusion highlighting the main contribution and pointing toward eventual further improvements.

2. GRN and Fuzzy Approach

In this section, the basic GRN inference process is outlined. To do so, the emphasis is made on the utilization of the fuzzy inference approach.

2.1. Main GRN Inference Issue. GRNs inference is based on determining the genes that affect the expression of other genes and on describing these effects. At the abstract level, genes are usually represented by nodes and eventually interacting with each other through arcs, representing casual interactions between genes. The arrow (i, j) indicates that gene i regulates gene j. The process of inference consists in finding suitable GRNs that reproduce available data and extends this knowledge by predicting relations not explicitly given by actual data. For reasons of simplicity and without loss of generality, most studies concentrate on directed acyclic graphs (DAGs).

Figure 1(a) shows 2 hypothetical GRNs with 4 genes each. Figure 1(b) shows the same 2 GRNs after evolvement during the process of inference. An increase (resp., decrease) of a cellular component, such as RNA (ribonucleic acid) or protein, in response to an external variable is called upregulation (resp., downregulation). No arc means that the genes are not directly correlated. We use −1 to indicate downregulation and 1 to show upregulation between genes in the inferred GRN, as shown in Figure 3.

2.2. Example of Inference. Figure 1 shows a 4-gene GRN to be improved through the inference process.

The two GRNs in Figure 1(a) represent two initial solutions whose links are generated by some process (e.g., randomly, by *a priori* knowledge, among others).

In Figure 1(b), the resulting GRNs are obtained after the inference process is achieved. A performance measure, such as a fitness function in evolutionary algorithms settings, is used to estimate the quality of the solution.

Additionally, Figure 3 shows another example of GRN, resulting from an inference process with cases of up- and downregulation for the yeast cell cycle.

```
// Basic fuzzy approach for GRN Inference //
Fuzzification stage
    Define fuzzy membership values for each group
    of fuzzy sets for gene expression values from
    the microarray data.
Inference stage
    Estimate fuzzy rules related to gene
    expression with a specific optimal gene
    expression level;
    Choose rules from a fuzzy rule set to infer
    the predicted fuzzy expression profile.
Defuzzification stage
    Defuzzify inferred profile into a group of
    numbers that have the same scale of the main
    microarray data.
```

Algorithm 1: Fuzzy approach in GRN inference.

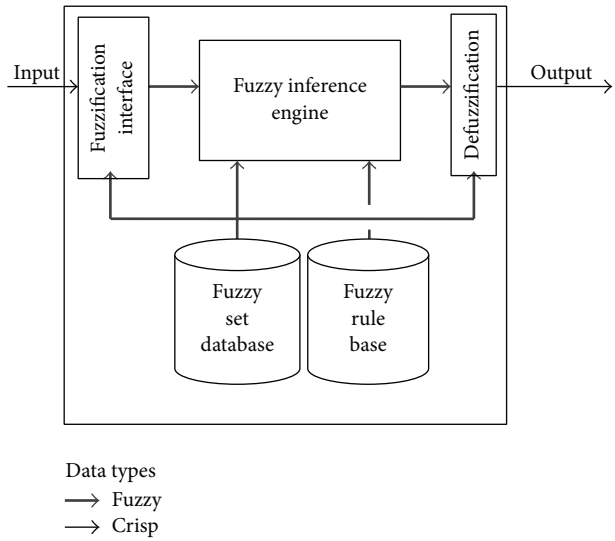


Figure 2: The basic fuzzy approach.

2.3. Fuzzy Approach

2.3.1. Steps of Fuzzy Approach. The basic fuzzy approach is depicted in Figure 2. Crisp values are given as inputs and crisp values are also obtained in the output after the fuzzy inference is accomplished. Fuzzification transforms crisp inputs and makes them amenable to inference by using fuzzy logic operators such as AND, OR, and NOT. Rules are evaluated and aggregated and then defuzzified to give a crisp result. As a summary, any fuzzy logic algorithm consists of three main traditional stages, fuzzification, inference, and defuzzification.

2.3.2. Fuzzy Approach in GRN Inference. The general steps of fuzzy logic algorithm for inferring GRNs from microarray gene expression data are slowly emerging [4]. The main steps are described in Algorithm 1.

The fuzzy approach has an essential role in combining biology-based models with logical techniques for reconstructing the underlying GRN. Biological relations in the best-fitting fuzzy GRN models can productively recover both the direct and indirect interactions from preceding knowledge to offer best understanding of biological background about the transcriptional and regulatory mechanisms [9]. That is why one of the main methods that have emerged to infer GRNs is the fuzzy approach model, which can provide a basic rule structure due to the categorization of observations. These models are flexible and can be adjusted for several regulatory models and inferential rule groups. Furthermore, the nonlinear impacts between regulatory genes can be added, based on additional knowledge elicited from human specialists or automatically derived from specialized knowledge bases and converting data about gene relations into suitable fuzzy logic model.

Because the use of fuzzy logic allows the fusion of high-level, human-like reasoning in constructing and structuring the rule-based construction of the GRN, it is therefore advantageous for the biologist or subject matter expert who have extensive knowledge about existing regulatory mechanisms to express this information in an intuitive fashion using the semantics of fuzzy logic [4].

2.4. Complexity of Fuzzy Approach in GRN Inference. One of the main drawbacks of the fuzzy approach is its computational complexity. The Mamdani fuzzy inference involves a large computational burden. On the other hand, the computational efficiency of fuzzy inference can be improved using the Tagaki-Sugeno method, which has great attractiveness in control, especially for dynamic nonlinear systems, and works well with adaptive and optimization techniques [8]. One of the most challenging issues remaining is the intervention *via* an external control action on the GRN structure in order to enforce corrective therapeutic actions [10]. Unfortunately, with the number n of genes used for the model, the complexity of the inferred GRN increases in $O(n^3)$, if we use a triplet of genes. More complex models such as

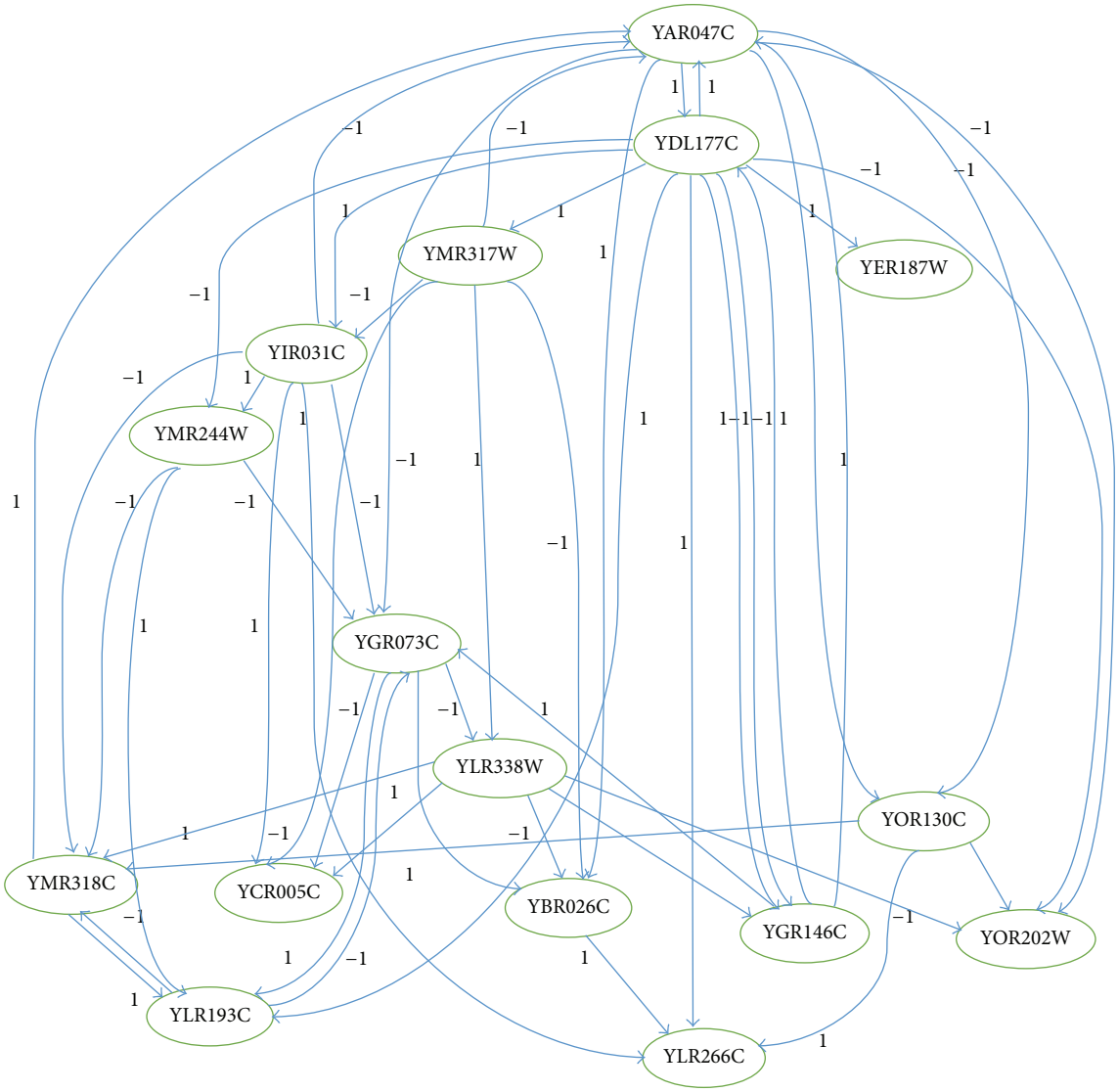


Figure 3: Inferred GRN of yeast cell cycle genes dataset with 16 genes and 50 direct links showing upregulation and downregulation. The notation (−1) is used to indicate downregulation and (1) to show upregulation between genes.

a model with coactivators and corepressors would have an $O(n^5)$ complexity. Accordingly, the algorithm can only focus on simple regulation patterns but cannot scale well to more complex models whose implementation time would be on the scale of years instead of hours.

Therefore, the computation time has to be improved. One way of doing so is by introducing a temporal gene expression clustering as a data processing step into the algorithm. In terms of computation time of experimental results for several real gene expression datasets, it is shown that the improved fuzzy approach is more efficient for reconstructing GRNs while the Mamdani model, although slow because of suffering from time complexity, has given the best performance in terms of the resilience to noise [7].

3. Discussion of Mainstream Fuzzy Methods in GRN Inference

In this section, we describe the mainstream methods in GRN inference based on the fuzzy approach, highlighting their merits and limits.

3.1. Detecting GRNs from Microarray Data. The appearance of microarray technology motivated computer scientists to improve algorithms for inferring GRNs and provided the possibility of simultaneously monitoring the expression levels for hundreds of thousands of genes. In addition, the DNA (deoxyribonucleic acid) microarray technology allowed the measurement of the RNA amount that is associated with a

number of genes in parallel and decided which of them were expressed in a specific type of cells. This technology might face different types of difficulties in understanding the gene interaction. The basic reasons of these difficulties are the thousands of genes contained in microarray databases with few time steps in addition to the expression values that are subjected to noise [11].

Further, discovering GRNs using microarray data has some additional inherent difficulties. Various stages are used to regulate gene expression, including DNA transcription, RNA processing and transport, RNA translation, and post-translational adjustment of proteins. The biological activity in the first of these stages is captured by DNA microarrays while the changes in mRNA transcript abundance do not necessarily reflect the regulation in the later stages. More experimental tests are therefore required for determining the validity of genetic interactions predicted by network models [4].

The use of fuzzy approach gave the ability to search for the potential regulatory interactions and comprehend reverse engineering of regulatory pathways related to a subset of interesting genes. The objectives of the proposed fuzzy approaches range from examining the entire genome of microarray data for small regulatory units to an exhaustive reconstruction of gene interactions in a certain pathway. These methods were effectively applied to both the simulated/experimental data related to the RAF (rapidly accelerated fibrosarcoma) signaling pathway and the real microarray data related to the yeast cell cycle, and the results were compared with other well-known algorithms. When making simulations, missing values were assigned using the k-nearest-neighbor (kNN) algorithm. Two ways were used to analyze the datasets.

First, eliminate all genes with expression levels below the noise threshold and that did not vary significantly during the cell cycle. Particularly, remove all genes with a maximum expression level below the first quartile in any of the four datasets or those that exhibited less than a threefold difference between maximum and minimum expression values in all four datasets. Using the fuzzy model, span the resulting dataset of 1737 genes and assess the gene triplets for enrichment of transcription factors and cell cycle regulated genes.

In the second analysis, select 12 genes from the yeast cell cycle pathway taken from KEGG (Kyoto Encyclopedia of Genes and Genomes http://www.genome.jp/kegg/) and run the fuzzy logic genetic search algorithm model on the gene expression values corresponding to these 12 genes. Check for consistency of the estimated GRN with the pathway depicted in KEGG. While the method is primarily used as a probing tool, the increase in computational overhead must be evaluated against the added flexibility of the model [4].

3.2. Predicting Changes in Expression Level. Fuzzy rules are used for finding gene regulation patterns from gene expression data using an activator-repressor GRN model. First, fuzzify gene expression profiles and then convert them into qualitative descriptors such as low, medium, and high. Then, a set of fuzzy rules is used to test all the probable combinations of triplets, target, repressors, and activators. The output obtained from the proposed system is then compared with the optimal gene expression level to discover whether they effectively fit the proposed fuzzy model. The difference between the achieved expression level and the target one is then used to rank the regulation combinations that have a small error and fit the majority of fuzzy rules. These combinations are inferred to exhibit the relations of the target, repressor, and activator combinations. An example of fuzzy rules is given in Algorithm 2.

```
// Examples of rules for inferring GRNs from
            expression data //
Rule 1
If activator_expression is high
and repressor_expression is low
then target_gene_expression is high
Rule 2
if the repressor_concentration is high
and activator_concentration is low
then target_gene_expression is low
```

Algorithm 2: Predicting changes in gene expression level.

Since the genes have three different states, the proposed method supports the concept of generalization in GRN inference by allowing a broader search space for inferring regulatory relationships and predicting changes in expression level of the target gene through interval time points. As a result, the expected regulation patterns of this model compose the GRN. The model is tested using yeast expression data that are obtained from *Saccharomyces cerevisiae* (yeast) cell cycle expression dataset [12].

3.2.1. Fuzzy Clustering Method. Fuzzy logic can also be efficiently used for modeling the interaction and regulation of genes to precisely reflect the fundamental biology and to provide the complete analysis of GRN from clustering to evaluating the credibility of the network. A multiscale fuzzy clustering method using k-means algorithm allows the genes interacting among regulatory pathways across several conditions at various levels of detail. The causal relationships among groups of coregulated genes can be detected using fuzzy cluster centers. In this context, fuzzy method can measure the weight of expert knowledge and assist in quantifying uncertainty about the genes functions with the use of Gene Ontology (GO) (http://www.geneontology.org/) database in order to emphasize certain interactions.

The experiment on carbohydrate metabolism in the model plant (*Arabidopsis thaliana*) is used for gene expression to illustrate the proposed model. The information from the Gene Ontology (GO) database is used to evaluate the main gene regulatory relationships. A novel regulatory relationship about carbohydrate metabolism *trehalose* regulation was detected in the resulting network. The GRN inference algorithm has been efficiently evaluated to provide the information for time delay using the cluster centers. The feedback in the networks can also be allowed by using the algorithm

that makes the results more reliable. The suggested work can be enhanced by incorporating the GRN model with accessible metabolic networks in the simulation of certain cellular processes [13].

3.2.2. Fuzzy Approach in Modeling Gene Expression and Analyzing Protein Network. Because some genes interact between many regulatory pathways, soft clustering algorithm can be used for the description of these interactions. From all the reviewed applications in the study, it was found that the fuzzy approach-based models and fuzzy clustering techniques are powerful and efficient in many aspects and they are expected to play an essential role in integrating, analyzing, and modeling large amount of microarray datasets and complex biological systems.

Its prohibitive computational time remains one of the challenges because all the probable combinations of triplets must be examined. For example, the analysis of the relationships between 1,898 genes required 200 hours of computation on an 8-processor SGI Origin 2000 system. The results of experiments on many real gene expression datasets have illustrated the superior effectiveness for the improved fuzzy logic model in reconstructing GRNs in terms of computation time and the superior performance for Mamdani's model in terms of the ability for adapting to noise [7].

3.3. Inferring GRNs from Time Series Expression Data. Many developed analysis methods on reverse engineering of GRNs were proposed as a result of the increase in the number of time series gene expression experiments. The published methods evaluated all combinations of gene interactions in a gene expression time series database for those genes that directly fit a gene regulation. In order to reduce the computational cost, the search space of the least probable candidates is pruned in order to fit the required regulatory model. The method is used in evaluating numerous gene interactions for validating specific regulatory events based on filtering out the potential regulatory event candidates using fuzzy logic.

We can describe the changes that occurred in various expression levels and represent them as directional slope data while discarding combinations whose slope reaction is outside the proposed regulatory relation model scope. Several cheap hit vector calculations can be carried out to evaluate the ability of each gene relation triplet to fit in a more practical regulatory method.

The pruning process is performed effectively with the smooth response surface (SRS) algorithm on three *Saccharomyces cerevisiae* datasets, without significantly affecting the sensitivity of the SRS algorithm. The results gave 70% reduction in the computational time that was required for the processing of three datasets for *Saccharomyces cerevisiae* microarray time series [14]. However, it is not clear whether the proposed method can successfully be applied to more complex GRNs.

3.3.1. Fuzzy Relational System for GRN Inference. Fuzzy techniques can be used to discover the GRNs from time series gene expression data. Fuzzy rules are designed depending on fuzzy set theory and expressing levels of gene. The fuzzy dependency relationships in noisy and high-dimensional data are also discovered. An artificial bee colony-based (ABC) search algorithm is presented to discover the potential constructions for a GRN which correctly fit the time series data and to discover the probable gene relations. Thus, the technique defines the fuzzy dependency measure between genes based on converting the values of the quantitative expression into linguistic terms, and the gene interaction is explained using a group of fuzzy relational matrices. The proposed technique is based on the fact that the measured time points are restricted. It uses an *Escherichia coli* SOS (when DNA is damaged, the SOS response is activated by stopping the cell cycle and inducing DNA repair and mutagenesis) DNA that includes around 30 genes regulated at the level of transcription. Results demonstrate that we are able to detect the genes that have an impact on a target gene, recognizing if the target gene is activated or inhibited, based on the detected dependency relations. Moreover, the simulation results for both real and artificial data show that this technique can effectively capture the GRNs' nonlinear dynamics and reveal the possible gene interaction relationship [15].

3.3.2. Fuzzy Data Mining Technique for Inferring GRNs from Time Series Expression Data. Fuzzy data mining techniques are capable of extracting the high-dimensional and noisy expression data in order to construct relevant fuzzy sequential relations among genes. In addition, they are also capable of discovering the dependency of genes with each other and the possibility of a gene to be activated or inhibited. As an example, one technique can also predict the influence of genes on a target gene in an unseen sample. The proposed method is evaluated with the use of real expression data where the results demonstrate that the use of a fuzzy model for analyzing gene expression data is very efficient [3].

3.3.3. Inferring GRNs from Expression Data by Discovering Fuzzy Dependency Relationships. The detected fuzzy dependency relations of genes can uncover the biologically significant gene regulatory relations, which can then be utilized in inferring the fundamental structures of these networks [16]. The proposed method can be enhanced by relying on the naturally parallel nature of this method in detecting the dependency relation. This improvement will in its turn make this method able to tackle highly complex gene datasets.

4. Discussion of Hybrid Methods for GRN Inference

The benefits of combining fuzzy logic with other methods (e.g., soft computing methods) are well recognized. Ultimately, fuzzy models can be optimized using hybrid techniques based on artificial neural networks and/or genetic algorithms. This improvement can be obtained at different levels, not only for forming membership functions but also for choosing rules to be used in the fuzzy logic system. We briefly describe examples of hybridized fuzzy methods.

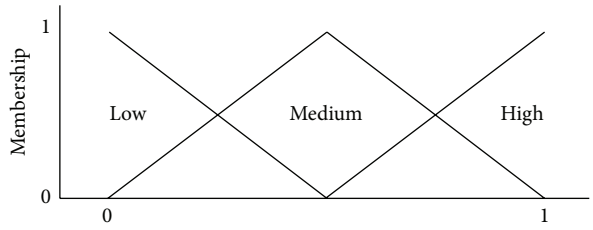


Figure 4: Example of fuzzy expression levels.

4.1. Fuzzy Inference Systems (FISs) and ODEs

4.1.1. ANFIS and ODEs. A model for representing a small GRN can be combined with the features of both fuzzy inference systems (FISs) and ordinary differential equations (ODEs) models. A neural network is used to train the FIS as a part of an adaptive network-based fuzzy inference system (ANFIS). The neural network is also used for allowing the output and membership functions from FIS to be adapted, based on the training data which in turn reduce the previous knowledge needed for modeling the phenomenon. The proposed model is used for describing the *lac operon* (*lactose* operon) in *Escherichia coli* and then compared with other similar models. (The operation of *lac operon* can be summarized as follows. When the glucose which is the main carbon source in the *Escherichia coli* is not present in the environment cell, it is then formed by the bacterium using the lactose. This process indicates that there is a regulatory operation that allowed synthesizing the required enzymes for obtaining glucose.) The fuzzy logic allowed the expression of the rules by linguistic labels for integrating the expert knowledge in an easy way. Thus, the fuzzy rules were used for describing the behavior of the *allolactose* based on two inputs: internal lactose and *β-galactosidase*. Therefore, three fuzzy sets were identified as low, medium, and high. Eight possible rules related to the two inputs are produced and implemented [17]. Figure 4 shows a typical profile of fuzzy expression levels used in fuzzy rules.

4.1.2. Advantages of Integrating ANFIS and ODEs. The behavior of nonlinear system is similar to the one obtained with ODE and enhanced by the flexibility of fuzzy logic and the ability of training provided by ANFIS. The ANFIS network allowed the obtainment of all data needed for describing the transient state of implemented experiments and gave a large quantity of ODEs stable points. This indicates the flexibility of FIS and the significant training capability of the network. However, the training process of the ANFIS network caused errors. These latter ones were then compared with the outcome errors of the models based on ODEs. Further, the proposed PLDE-based model can be improved using optimization approaches and heuristic methods.

This work showed that the fuzzy approach proved to be an important tool due to its ability to characterize nonlinear systems, its human-like language to define knowledge, and its facility to include and edit fuzzy rules. To obtain a more powerful method, fuzzy approach can be supplemented by ANFIS for obtaining knowledge from experimental data in addition to ODEs.

Table 1: Comparison of pure FIS versus ANFIS and ODEs.

Characteristics	Methods	
	FIS	ANFIS/ODEs
Characterization of nonlinear systems	Yes	Yes
Automatic training/learning	No	Yes
Reduction of knowledge needed for modeling biological phenomenon	Needs human expert. It is tedious and scarce	Automatically done directly from datasets
Adaptation of output and membership functions from FIS	No adaptation	Automatic adaptation

4.1.3. Disadvantages of Using ANFIS and ODEs. There seems to be an issue in making a tradeoff between acceptable representations of biological phenomena without compromising the viability of its computational implementation. In addition to that, there is an urgent need in maintaining a simple and comprehensible language that allows systems analysis from a qualitative standpoint.

4.1.4. Comparison of Pure FIS versus ANFIS/ODEs. Table 1 summarizes the characteristics that are supported by FIS, on the one hand, and/or by ANFIS and ODEs, on the other hand.

4.2. Modeling GRNs Using Fuzzy Petri Network (FPN)

4.2.1. Basic Fuzzy Petri Network (FPN). Since GRN is a model of gene interactions at the expression level, the microarray technology has an essential role in the reconstruction of GRN. The GRN inference has challenged computer scientists to develop more enhanced algorithms for modeling the underlying regulatory relations among genes. One of these methods is the fuzzy Petri net (FPN) which consists of a fuzzy reasoning process that is able to search the microarray databases for activator-repressor regulatory relations between genes.

(1) FPN Method. The aim of the fuzzy Petri net (FPN) method is to construct a rule-based reasoning system that takes into account the gene triplets from the microarray data and searches for activator-repressor regulatory relationships among those genes. The proposed method uses the ways of predicting changes in expression level in the target gene based on input expression level. This method removed the probable false predictions from the traditional fuzzy model and offered an open search space to infer the regulatory relations. An activator-repressor relationship was shown by the genes that most likely fit the model. The main properties of the proposed method were its ability to explain the classical fuzzy reasoning procedure and the visualization of the structure of a rule-based fuzzy system. Furthermore, the method provides the reasoning stages required to predict changes in the gene

Table 2: Comparison of pure FIS versus FPN and FRBPN.

Characteristics	Methods		
	FIS	FPN	FRBPN
Predicting changes in expression level of the target gene	No	Yes	Yes
Need determination of truth degree of a proposition in advance	Yes (need of human expert)	Yes (need of human expert)	No (no need of human expert)
Need determination of the confidence degree of a rule in advance	Yes (need of human expert)	Yes (need of human expert)	No (no need of human expert)

expression levels for validation. The results show that the proposed method is suitable and practicable for predicting changes in expression level of the target gene. The proposed work can be enhanced by integrating neural network for modeling GRN [18].

(2) Limits of FPN Method. The main limitation of the FPN method is that the truth degree of a proposition and the confidence degree of a rule should both be determined in advance. The characterization of these two degrees usually relies on experts' experience. This latter might be unavailable or difficult to obtain. Even if the experience is available, the modeling will induce some uncertainty in the reasoning.

4.2.2. Enhancing FPN through FRBPN. An extension of the FPN for modeling and analysis of a GRN uses a fuzzy reasoning Boolean Petri network (FRBPN) method and the dynamical behavior of gene. The proposed model was tested on a GRN responsible for the carbon starvation nutritional stress response in the bacterium *Escherichia coli* cells with the use of a comprehensive database. The results showed that there are six genes specified, with their interactions that play an important role in this process. Six modules are identified that correspond to the truth tables describing the Boolean behavior of each one of the regulatory entities in the nutritional stress response network of carbon starvation [19].

4.2.3. Comparison of FIS versus FPN and FRBPN. See Table 2.

4.3. Other Hybrid Methods

4.3.1. AFEGRN. The adaptive fuzzy evolutionary GRN (AFEGRN) method is destined to infer GRNs using fuzzy approach. Specifically, it uses data distribution for automatically determining model parameters, for example, number of clusters for fuzzy c-means and estimation of Gaussian distribution algorithm. The method is tested for both normal and cancerous breast GRNs. The results are consistent with the biological knowledge and demonstrate that the differentially expressed (DE) genes were the cause of most of the cancers related to GRN changes. This result might corroborate the use of the AFEGRN framework for modeling any GRN [6].

4.3.2. Coalesce GRN (CGRN) Reconstruction Framework. The need for cross platform GRN fusion is highly motivated by the noisy nature of microarray data and platform bias. Since the produced gene expressions by various platforms are not directly comparable, the cross platform GRN fusion becomes a difficult task. The coalesce GRN (CGRN) reconstruction framework integrates a cross platform GRN in order to eliminate the experimental and platform bias with the use of Dempster-Shafer theory of evidence. The common cancer related regulatory links in 10 databases are found using the CGRN framework. These databases resulted from several microarray platforms, like Affymetrix (http://www.affymetrix.com/estore/) and cDNA arrays. The results illustrate that the proposed CGRN framework can efficiently be applied for cross platform GRNs fusion to find out tumor specific links and unknown pathways [21].

4.3.3. Combining Fuzzy Clustering and Bayesian Networks for Modeling GRNs. The Bayesian network (BN) approach is mainly used in the expectation of the GRNs from the expression data. Two main problems deeply undermine the efficiency of BN methods. One is the excessive computational time. The other major issue is the search space of possible BNs, which is very large because of the very large number of genes. Due to the fact that both the hierarchy and modularity are main biological networks features, global networks can be gained by collecting local components which consist of genes that have the same function and may have the same or a similar expression pattern. Combining fuzzy clustering and Bayesian networks for modeling is used in order to reduce the search space. In the learning process of local networks, the existence of gaps among the number of genes and samples is minimized and the search space is reduced. Simulation results confirm that the proposed approach can effectively predict GRNs and obtain noticeable improvement in the accuracy and the reduction of computational time in comparison with other available BN approaches [11].

4.3.4. Inferring Fuzzy Cognitive Maps for GRNs. Fuzzy cognitive maps (FCMs) are used for representing GRNs. The method is based on an ant colony optimization (ACO) and relies on a decomposed method for reducing the dimension of the problem. Thus, the FCM learning algorithm becomes more scalable. Gene expression data are represented by fuzzy variables, and the relationships between genes are modeled through the fuzzy relations in FCMs.

Since the previously proposed FCM learning algorithms can only learn less than 40 nodes, the extended algorithm is able to learn FCMs with more than a hundred nodes. It can

Table 3: Comparison of FIS versus other hybrid methods.

Characteristics	Methods				
	FIS	AFEGRN	CGRN	Fuzzy clustering and BNs	FCMs
Automatic model parameters estimation, for example, number of clusters for fuzzy *c*-means	No	Yes	NT*	NT*	NT*
Cross platform GRNs fusion	No	NT*	Yes	NT*	NT*
Eliminate the experimental and platform bias	No	NT*	Yes	NT*	NT*
Reduction of search space/complexity	No	No	NT*	Yes (better than traditional BNs)	Yes (better than BNs and ODEs)

*NT means "not tested" (information not available): the corresponding method has not been tested against the specific characteristic mentioned in Table 3.

Table 4: Main fuzzy methods in GRN inference.

Dataset	Modeling method	Tool/technique software/database	References
Saccharomyces cerevisiae (yeast)	Fuzzy logic + clustering	Preprocessing algorithm coded in MATLAB	Ram et al. [9]
Saccharomyces cerevisiae (yeast) cell cycle	Fuzzy rules	GeneChip + SAGE	Woolf and Wang [12]
Saccharomyces cerevisiae	Fuzzy logic + clustering	DSOM approach + ART	Ressom et al. [20]
Breast cancer (cancerous cells in human)	AFEGRN	AFEGRN framework	Sehgal et al. [6]
10 different cancer datasets (cancerous cells in human)	Coalesce GRN (CGRN)	Cross platform GRN fusion	Sehgal et al. [21, 22]
Bacterium *Escherichia coli*	FRBPN	FRBPN technique	Hamed [19]
Saccharomyces cerevisiae	Combined fuzzy clustering and Bayesian networks (FCBN)	Software platform MATLAB and BNT package	Wang et al. [11]
Lac operon in *Escherichia coli*	Combined ODEs models and FIS	Simulink, fuzzy logic Toolbox and Optimization Toolbox in MATLAB	Muñoz et al. [17]
Arabidopsis thaliana plants, Affymetrix *Arabidopsis* ATH1 genome with 22K genes	Fuzzy cognitive map (FCM) model + clustering	FCModeler tool	Du et al. [13]
Escherichia coli bacteria SOS DNA consisting of 30 genes	Fuzzy logic + artificial bee colony (ABC)	ABC- and DE-based simulations, on Pentium	Das et al. [15]
Saccharomyces cerevisiae	Fuzzy Petri net (FPN)	FPN graphical tool	Hamed et al. [18]
Saccharomyces cerevisiae	Fuzzy logic based modeling	Fuzzy data mining technique	Ma and Chan [3]
Escherichia coli and *Saccharomyces cerevisiae* (210 time points in 100-gene networks), 10 DREAM-4 datasets	Fuzzy cognitive maps (FCMs) model and ant colony optimization (ACO)	Simulation using stochastic differential equations, DREAM project	Chen et al. [2]
Saccharomyces cerevisiae microarray time series (three datasets)	Activator-repressor regulatory model, SRS biological model	SRS program, regulatory-fit scoring method	Volkert and Malhis [14]
Saccharomyces cerevisiae genes time series gene expression data (about 800 cell cycle regulated genes)	Fuzzy data mining model	C4.5, SVM, and FID, 10-fold cross validation strategy	Ma and Chan [16]
Yeast cell cycle (cell-synchronized datasets with 14 time points)	Fuzzy logic genetic search algorithm model	KEGG database, *k*-nearest-neighbor (*k*NN) algorithm	Brock et al. [4]

also avoid the discretizing of expression data, allowing the representation of genes dynamics more accurately. The basic problem of optimizing the whole weight matrix is reduced to smaller optimizing problems of one column of the weight matrix. Evaluation is done using 10 DREAM-4 datasets, from the DREAM Project (http://www.the-dream-project.org/). The results show that the FCM-based method outperforms other methods such as ODEs and Bayesian networks in part of the 10-gene network problems and in all of the 100-gene network problems. The suggested work can be further developed for application to larger networks [2].

4.3.5. Comparison of FIS and Other Hybrid Methods. See Table 3.

5. Summary of Fuzzy Contributions for GRNs Inference

The main contributions of GRN inference based on the fuzzy approach and its extensions are summarized in Table 4.

6. Example of GRN Inference

The biograph in Figure 3 shows genes interactions in the dataset of *Saccharomyces cerevisiae* (yeast) showing 16 genes represented as nodes and gene-gene interactions drawn as 50 links (directed edges) between genes. For example, gene YAR047C is

(i) downregulated by genes YIR0317C and YMR317W (−1 on the arrow);

(ii) upregulated by genes YDL177C and YMR318C (1 on the arrow).

7. Conclusion

Fuzzy approach provides useful means for GRNs inference as reported in the diversified methods described in this paper. This is due to the fact that the underlying biological processes are best described using approximate features and not crisp values. The main contribution of the paper is that it gives an up-to-date hierarchical structuring of the various applications of fuzzy systems to GRNs; contributions now spread in diversified and scattered literature. We focus not only on the general concepts of fuzzy systems but also on their effective applications in GRNs inference, stressing both merits and limitations of each method. However, we do not enclose any biological aspects of the methods described or case studies on particular datasets as these are detailed in each method referenced. Despite the computational power offered by fuzzy systems in dealing with GRNs inference, better results are obtained using fuzzy method combined with other soft computing methods such as neural networks, genetic search, and Petri nets, among others. An interesting avenue for future research is to rank all the methods described above and others in terms of suitability for GRN inference.

Conflict of Interests

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

Acknowledgment

This work is supported by the Deanship of Scientific Research at Qassim University, Saudi Arabia, as part of the research project with ID no. 1223-2012.

References

[1] V. Adriaenssens, B. de Baets, P. L. M. Goethals, and N. De Pauw, "Fuzzy rule-based models for decision support in ecosystem management," *Science of the Total Environment*, vol. 319, no. 1–3, pp. 1–12, 2004.

[2] Y. Chen, L. J. Mazlack, and L. J. Lu, "Inferring fuzzy cognitive map models for gene regulatory networks from gene expression data," in *Proceedings of the IEEE International Conference on Bioinformatics and Biomedicine (BIBM '12)*, pp. 598–601, Philadelphia, Pa, USA, October 2012.

[3] P. C. H. Ma and K. C. C. Chan, "A fuzzy data mining technique for the reconstruction of gene regulatory networks from time series expression data," in *Proceedings of the 3rd Computational Intelligence in Bioinformatics and Computational Biology Symposium (CIBCB '06)*, pp. 1–8, September 2006.

[4] G. N. Brock, V. Pihur, and L. Kubatko, "Detecting gene regulatory networks from microarray data using fuzzy logic," in *Fuzzy Systems in Bioinformatics and Computational Biology*, Y. Jin and L. Wang, Eds., vol. 242 of *Studies in Fuzziness and Soft Computing*, pp. 141–163, Springer, Berlin, Germany, 2009.

[5] A. Hamdi-Cherif, "Integrating machine learning in intelligent bioinformatics," *WSEAS Transactions on Computers*, vol. 9, no. 4, pp. 406–417, 2010.

[6] M. S. B. Sehgal, I. Gondal, L. Dooley, and R. Coppel, "AFE-GRN: adaptive fuzzy evolutionary gene regulatory network reconstruction framework," in *Proceedings of the IEEE International Conference on Fuzzy Systems*, pp. 1737–1741, Vancouver, Canada, July 2006.

[7] S. Zhang, R. S. Wang, S. X. Zhang, and L. Chen, "Fuzzy system methods in modeling gene expression and analyzing protein networks," in *Fuzzy Systems in Bioinformatics and Computational Biology*, vol. 242 of *Studies in Fuzziness and Soft Computing*, pp. 165–189, Springer, Berlin, Germany, 2009.

[8] A. Hamdi-Cherif, "Type-2 fuzzy control for bioinformatics—a systems approach," *International Journal of Computer Science and Network Security*, vol. 10, no. 7, pp. 69–75, 2010.

[9] E. Ram, M. Chetty, and T. I. Dix, "Fuzzy model for gene regulatory network," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '06)*, pp. 1450–1455, July 2006.

[10] A. Hamdi-Cherif, "Intelligent control and biological regulation for bioinformatics," *International Journal of Mathematical Models and Methods in Applied Sciences*, vol. 4, no. 2, pp. 93–104, 2010.

[11] F. Wang, D. Pan, and J. Ding, "A new approach combined fuzzy clustering and Bayesian networks for modeling gene regulatory networks," in *Proceedings of the International Conference on BioMedical Engineering and Informatics (BMEI '08)*, May 2008.

[12] P. J. Woolf and Y. Wang, "A fuzzy logic approach to analyzing gene expression data," *Physiological Genomics*, vol. 3, no. 1, pp. 9–15, 2000.

[13] P. Du, J. Gong, E. S. Wurtele, and J. A. Dickerson, "Modeling gene expression networks using fuzzy logic," *IEEE Transactions on Systems, Man, and Cybernetics B: Cybernetics*, vol. 35, no. 6, pp. 1351–1359, 2005.

[14] L. G. Volkert and N. Malhis, "An efficient method for fuzzy identification of regulatory events in gene expression time series data," in *Proceedings of the IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB '04)*, pp. 17–24, October 2004.

[15] P. Das, P. Rakshit, A. Konar, M. Nasipuri, and A. K. Nagar, "Fuzzy relational system for identification of gene regulatory network," in *Proceedings of the World Comp*, 2009.

[16] P. C. H. Ma and K. C. C. Chan, "Inferring gene regulatory networks from expression data by discovering fuzzy dependency relationships," *IEEE Transactions on Fuzzy Systems*, vol. 16, no. 2, pp. 455–465, 2008.

[17] C. Muñoz, F. Vargas, J. Bustos, M. Curilem, S. Salvo, and H. Miranda, "Fuzzy logic in genetic regulatory network models," *International Journal of Computers, Communications and Control*, vol. 4, no. 4, pp. 363–373, 2009.

[18] R. I. Hamed, S. I. Ahson, and R. Parveen, "A new approach for modelling gene regulatory networks using fuzzy petri nets," *Journal of Integrative Bioinformatics*, vol. 7, no. 1, article 113, 2010.

[19] R. I. Hamed, "Computational modeling and dynamical analysis of genetic networks with FRBPN algorithm," in *Advances in Computing, Communication and Control*, vol. 125 of *Communications in Computer and Information Science*, pp. 49–55, Springer, Berlin, Germany, 2011.

[20] H. Ressom, D. Wang, R. S. Varghese, and R. Reynolds, "Fuzzy logic-based gene regulatory network," in *Proceedings of the 12th International Conference on Fuzzy Systems (FUZZ '03)*, vol. 2, pp. 1210–1215, May 2003.

[21] M. S. B. Sehgal, I. Gondal, L. Dooley, and R. Coppel, "Coalesce gene regulatory network reconstruction: a cross-platform transcriptional gene network fusion framework," in *Proceedings of the IEEE Region 10th Conference (TENCON '06)*, pp. 1–4, IEEE, Hong Kong, November 2006.

[22] M. S. B. Sehgal, I. Gondal, and L. S. Dooley, "CF-GeNe: fuzzy framework for robust gene regulatory network inference," *Journal of Computers*, vol. 1, no. 7, pp. 1–8, 2006.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2015, Article ID 680681, 13 pages
http://dx.doi.org/10.1155/2015/680681

Research Article

Enhancing the Selection of Backoff Interval Using Fuzzy Logic over Wireless Ad Hoc Networks

Radha Ranganathan[1] and Kathiravan Kannan[2]

[1]*Department of Information Technology, Easwari Engineering College, Chennai 600089, India*
[2]*Department of Computer Science and Engineering, Easwari Engineering College, Chennai 600089, India*

Correspondence should be addressed to Radha Ranganathan; radhupriya@yahoo.co.in

Received 2 August 2014; Revised 21 November 2014; Accepted 6 December 2014

Academic Editor: Albert Victoire

IEEE 802.11 is the de facto standard for medium access over wireless ad hoc network. The collision avoidance mechanism (i.e., random binary exponential backoff—BEB) of IEEE 802.11 DCF (distributed coordination function) is inefficient and unfair especially under heavy load. In the literature, many algorithms have been proposed to tune the contention window (CW) size. However, these algorithms make every node select its backoff interval between [0, CW] in a random and uniform manner. This randomness is incorporated to avoid collisions among the nodes. But this random backoff interval can change the optimal order and frequency of channel access among competing nodes which results in unfairness and increased delay. In this paper, we propose an algorithm that schedules the medium access in a fair and effective manner. This algorithm enhances IEEE 802.11 DCF with additional level of contention resolution that prioritizes the contending nodes according to its queue length and waiting time. Each node computes its unique backoff interval using fuzzy logic based on the input parameters collected from contending nodes through overhearing. We evaluate our algorithm against IEEE 802.11, GDCF (gentle distributed coordination function) protocols using ns-2.35 simulator and show that our algorithm achieves good performance.

1. Introduction

Ad hoc network is a collection of dynamic, self-configured, and radio equipped nodes without any infrastructure. Ad hoc networks require every intermediate node to act as routers, receiving and forwarding data to every other node. This type of network is prevalently deployed in various scenarios wherein instantaneous connectivity becomes the need of the hour, either in emergency situations like a disastrous evacuation situation or in a casual get-together for presentations.

IEEE 802.11 MAC is the predominant protocol used over ad hoc networks for medium access. Binary exponential backoff algorithm (BEB) has been used by IEEE 802.11 DCF for collision avoidance. Whenever a node wants to transmit a packet, it starts sensing the medium. If the medium is idle for distributed interframe space (DIFS) period, then the node generates a backoff counter which is set for a random value between $[0, \mathrm{CW}]$. After that, the backoff counter is decremented by one for every idle slot. If the channel is busy, the backoff counter is paused until the next DIFS free period. When the backoff counter reaches zero, the node starts transmission. Here the minimum and maximum values of CW are called $\mathrm{CW}_{\min}$ and $\mathrm{CW}_{\max}$ with the default values of 31 and 1023, respectively. CW is initially set to $\mathrm{CW}_{\min}$ and after every unsuccessful transmission CW is doubled with the maximum limit of $\mathrm{CW}_{\max}$. Upon a successful transmission, CW is reset to $\mathrm{CW}_{\min}$.

Bianchi [1] analyzed the saturated throughput using Markov chain model and showed that the throughput increases with the smaller number of active nodes and small CW. When the number of active nodes increases smaller CW can lead to high collision. Larger CW improves the fairness among flows but reduces the overall throughput. Random BEB algorithm of IEEE 802.11 failed to improve the fairness and throughput over heavily congested ad hoc networks. Many algorithms have been developed to tune the contention window (CW) according to the congestion status.

These proposals targeted for improving either throughput or fairness or both.

The algorithms proposed in the literature can be put under two categories. They are overhearing-based and non-overhearing-based solutions. In overhearing-based solutions, each node collects information such as rate, channel utilization, and buffer status from neighbors and adapts to the new contention window (CW) according to some policy. But it is well known that contention loss occurs mainly due to hidden terminals, whereas overhearing is limited to neighbors. These methods fail to consider the status of hidden terminals. Nonoverhearing solutions enable the nodes to utilize their local information like number of idle slots, busy slots, Tx failures, and so forth, to tune its contention window (CW). However, nodes within the same transmission range assess the channel in the same way. It does not help the nodes to have differentiation and get fair channel access.

The proposed solutions realize the channel congestion status either through overhearing or local information and tunes contention window size according to some policy. After tuning, they tend to select the backoff interval randomly between $[0, \mathrm{CW}]$. In one way, this random selection helps to avoid collision between the nodes that are using the same CW. But, in another way, this randomness badly changes the order and frequency of medium access among nodes due to the zero lower bound. Random selection even leads to collisions under heavy load. This scenario affects the throughput and fairness by increasing the delay and collisions. Instead of random selection, node differentiation based on their individual parameters can optimally order the medium access among competing nodes. Optimal scheduling of the medium access is a challenging task over ad hoc networks due to its distributive and dynamic nature. Every node needs to collect information about the status of contending nodes to schedule itself to access the medium. The collected information is dynamic and vague due to the ever-changing topology of the network.

In this paper IEEE 802.11 binary exponential backoff algorithm is used for selecting contention window size (CW). To the best of our knowledge, this is the first time we introduce an algorithm that computes the backoff interval within the CW limit such that it schedules the medium in a fair and efficient way without involving much overhead. It also controls the contention by ordering the nodes according to its waiting time. We make use of fuzzy logic to compute the unique backoff interval between $[0, \mathrm{CW}]$. Fuzzy logic is a simple problem solving methodology that accepts vague or ambiguous input values and makes us arrive at a definite and crisp output using simple if. . .then. . .rules. These rules should reflect the exact behavior of the system.

Each node collects the input parameters queue length and waiting time from contending nodes through overhearing and stores them in a neighbor table. Every node is also responsible for advertising its input parameters (queue length—myqlen and waiting time—mywt) through request to send (RTS) message. When a node overhears this RTS message, it gains knowledge about its neighbors. This learning starts from training phase and continues. During transmission each node has to choose a count between $[0, \mathrm{CW}]$ and set it for the backoff counter by comparing its own input parameters with the collected information. This dynamic and vague information is applied to the membership functions to derive fuzzy variables. These fuzzy variables are fed to the fuzzy inference engine to get fuzzy output. Finally, defuzzification helps to derive at the crisp and unique backoff counter value between $[0, \mathrm{CW}]$.

The rest of the paper is organized as follows. In Section 2 we describe the related work and in Section 3, we brief about fuzzy logic and the steps involved and the key elements of our design in detail. We evaluate our performance against IEEE 802.11 and GDCF in Section 4 and finally we conclude our paper in Section 5.

2. Related Work

BEB algorithm of IEEE 802.11 [2] suffers from severe performance degradation under heavy traffic over wireless ad hoc network. It is well accepted that contention window plays vital role in improving the aggregate throughput and fairness. In this section, we review the proposals that tune the backoff interval with the goal of achieving good throughput or fairness or both. In [3], authors derived an analytical model to find optimal P value that reaches the theoretical throughput limit for P-persistent IEEE 802.11 protocol. To perform this, each node must have known the exact number of stations in the network and it depends on feedback information. To overcome this drawback, asymptotically optimal backoff (AOB) has been proposed by authors [4] to dynamically tune CW size according to the channel contention level. They probabilistically postpone transmission based on slot utilization factor. They show that their algorithm achieves theoretical capacity. In [5], authors tune the contention window based on the bit error rate of the medium. Both of the above methods need to estimate the number of active stations.

The authors of [6] use linear programming algorithm to optimize the minimum contention window size based on the channel condition (signal to noise ratio) and number of competing stations. Authors choose the access mode and $\mathrm{CW}_{\mathrm{min}}$ with analytical approach to optimize the throughput. They depend on network feedback to collect the channel condition status. Virtual backoff algorithm (VBA) [7] was developed using sequencing technique to reduce the number of collisions, thereby improving the throughput. However, VBA works well only in steady state where the number of nodes is fixed. VBA suffers from collisions in a dynamic scenario. In [8], authors analytically derive contention window size based on slot utilization and optimize the throughput in both saturated and nonsaturated conditions. It utilizes only local information like busy slots and free slots and does not require estimating the number of active stations. The authors of [9] propose an algorithm GDCF wherein they perform gentle decrease of contention window to reduce collision probability. They do not reset the contention window size after every successful transmission. Instead, they find optimal counter c and the contention window is halved after c consecutive successful transmission. This method reduces

the collision when the number of nodes is large. Nodes need to know the number of nodes in the network to find optimal value of c. Authors improve both fairness and throughput using this algorithm.

In [10], authors propose a control theoretic approach to tune contention window based on the locally available information. By comparing the average number of consecutive idle slots between two transmissions against the optimal set point, this method tunes CW and achieves optimal throughput and fairness. This method uses local information, but it also depends on the number of active stations. In [11] authors achieve fairness and weighted fairness among nodes using proposed increase with synchronized multiplicative decrease that supports background transmission. In [12], channel capacity is distributed among contending nodes through overhearing. This helps in improving the fairness among nodes. MadMAC protocol in [13] achieves both fairness and throughput using limited local information like number of experienced collisions and carrier sensing information. In [14], authors use fuzzy logic to tune the contention window based on the fuzzy parameters such as busy degree of the medium and number of neighbor nodes. This approach reduces collision probability and improves throughput and also fairness. Simplified backoff algorithm (SBA) [15] uses only local information like success, collision probability to tune CW. There are only two possible sizes for CW called CW_{min} (31) and CW_{max} (1023). CW is assigned to CW_{min} and CW_{max} during light load and heavy load, respectively. Authors claim to improve fairness and throughput. But this algorithm increases the delay due to the large CW. All of the above algorithms concentrate on tuning contention window according to the congestion level of the medium and they finally select the backoff interval randomly between $[0, CW]$.

In [16], authors change the lower bound and upper bound of the backoff interval based on the number of one-hop neighbors and number of transmission attempts. They prove that their algorithm reduces the number of collisions. Authors of [17] enable the nodes to change the upper and lower bounds based on the current network load and past history. In [18], authors introduce different subranges for backoff interval with respect to different network contention levels. Although these methods change their lower and upper bounds, final selection is done randomly within the new bound.

In this paper, we use BEB to tune the CW according to the current contention level. After tuning the CW, we introduce a new method of assigning backoff interval between $[0, CW]$. The individual parameters of each node like waiting time and queue length are taken into account to compute the backoff interval. These parameters help us to allocate a fair and effective medium access among the nodes. We ensure that unique backoff value is assigned to each node so as to avoid collision. Fuzzy logic is a simple and promising approach that extracts crisp and definite output from vague and ambiguous input parameters. Fuzzy logic has been widely used in wireless communication across various layers for computing, control, and decision making [19]. In [20] authors use fuzzy logic to calculate backoff interval to reduce contention over vehicular ad hoc networks. They control the current backoff interval using the past interval and success ratio of the node. Authors of [21] used fuzzy logic controller for early detection and prevention of congestion at the router buffer. They used delay rate and average queue length as input parameter and produced packet dropping probability as the crisp output. Authors of [22] have active router queue management based on conditions derived from Lyapunov stability theory. They used fuzzy congestion controller for the same.

In our method, each node collects input parameters from contending neighbors. The collected information is processed along with node's own attributes and applied to membership functions to get fuzzy input parameters. By applying these fuzzy variables to the rules base, we can derive the backoff interval as crisp output.

3. System Architecture

Problem with IEEE 802.11. In IEEE 802.11, the following steps are executed whenever a node wants to transmit a packet.

(i) Node senses the medium.

(ii) If the medium is idle for distributed interframe space (DIFS) period, then

  (a) the node generates a backoff counter randomly between $[0, CW]$;

  (b) the backoff counter is decremented by one for every idle slot;

  (c) if the channel is busy, the backoff counter is paused until the next DIFS free period;

  (d) when the backoff counter reaches zero, the node starts transmission.

Each node uses BEB algorithm to find out the current contention window size (CW). The value of CW reflects the contention status of the channel. The minimum and maximum values of CW are called CW_{min} and CW_{max} with the default values of 31 and 1023, respectively. IEEE 802.11 updates CW as follows.

(1) CW is initially set to CW_{min}.

(2) After every unsuccessful transmission CW is doubled with the maximum limit of CW_{max}.

(3) Upon a successful transmission, CW is reset to CW_{min}.

We note that the backoff value is randomly chosen between $[0, CW]$ irrespective of the value of CW. Lower bound 0 changes the optimal order and frequency of channel access among nodes [16–18]. Previous studies have revealed that it greatly affects the average delay and throughput of the individual nodes. For larger number of nodes with heavy traffic, the number of collisions is more which leads to larger value of CW resulting in unfairness [23]. BEB never considers the traffic status (like waiting time or queue length) of the contending nodes for allocating the medium. The number of collisions can be reduced when the contending nodes are

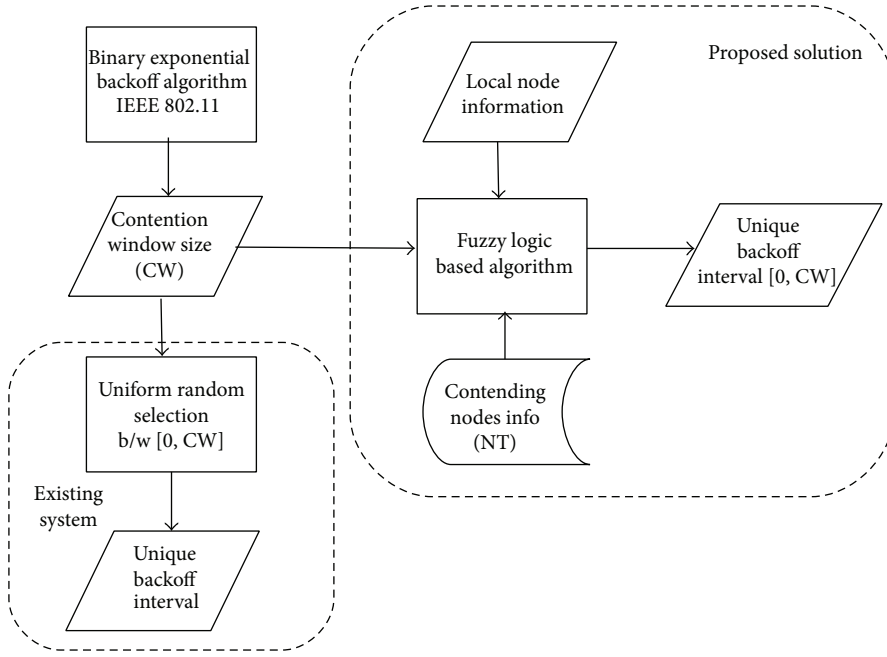


Figure 1: System architecture.

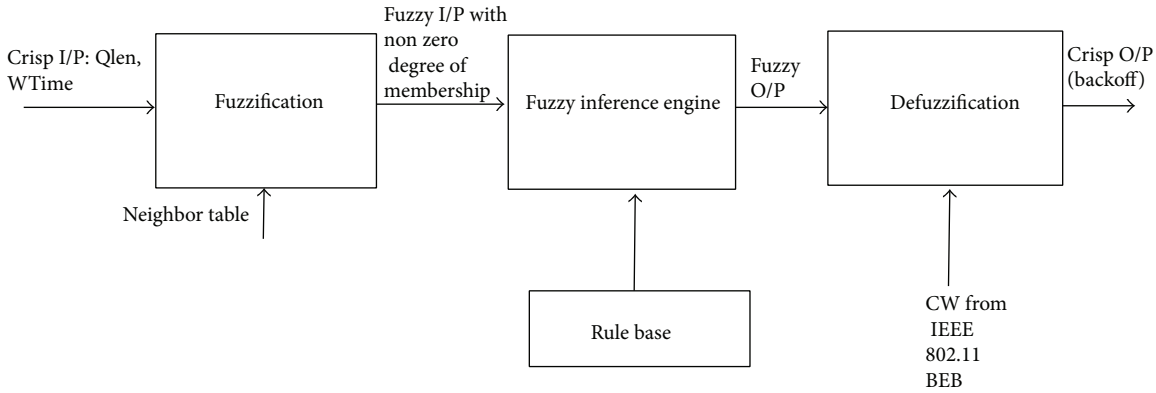


Figure 2: Steps in fuzzy logic.

assigned with unique backoff value according to its traffic status.

Proposed Design. Our proposed algorithm enhances IEEE 802.11 DCF with additional level of contention resolution that prioritizes the contending nodes according to its queue length and waiting time. Each node learns about the contending nodes and computes a unique backoff interval between [0, CW] for itself. Contention window size (CW) is updated using IEEE 802.11 BEB. Each node needs to compute unique backoff interval by comparing its own data with the input parameters collected from contending nodes. In a dense network with larger nodes and heavy traffic, the collected input can be huge and vague. We make use of fuzzy logic at each node to find out its order for accessing medium.

Our system architecture is shown in Figure 1. It is clear that our fuzzy logic algorithm replaces random backoff selection and enhances IEEE 802.11. We specify that our algorithm can be incorporated along with the existing backoff algorithms [4, 6, 7, 9] too. In the next section, we explain our fuzzy logic algorithm in detail.

3.1. Fuzzy Logic. Fuzzy logic is a promising approach that produces a definite output from vague parameters. The following steps are involved in our algorithm and are shown diagrammatically in Figure 2.

(1) Collect the input parameters.

(2) Apply the input parameters to the triangular membership functions and retrieve the degree of membership on each fuzzy input variable—fuzzification.

(3) Apply the fuzzy values to each rule in the rule base and retrieve the fuzzy output from fuzzy inference engine using Mamdani's method.

TransmitRTS:
(i) If the medium is Free for DIFS period then
    (a) If the backoff counter is in freeze state then do continue with old backoff counter
    (b) else if node is in training Phase then
        (i) Generate Backoff counter randomly between (0, CW)
    (c) else if node is not in training phase then
        (i) Apply local data, input from NT to Fuzzy logic algorithm and find out backoff between (0, CW).
(ii) Decrement backoff counter by one for every idle slot.
(iii) If the back off counter reaches zero,
    (a) Store current queue length and waiting time of this node
    (b) Start transmission.
(iv) else if the channel is busy, the backoff counter is paused until the next DIFS free period.
ReceiveRTS:
(i) Receive RTS and increment RTS count by one.
(ii) If RTS count is greater than 15 then training phase is set to false
(iii) If the packet is destined to current node
    (a) update neighbor table and process the packet.
(iv) If the packet is not destined to current node.
    (a) update the neighbor table and NAV (Network Allocation Vector) timer.
    (b) drop the packet.

Algorithm 1: Steps to be followed while transmitting and receiving RTS.

(4) Defuzzify the fuzzy output to retrieve the crisp output using center of gravity method.

Each step is discussed in detail in the following section.

3.1.1. Collecting Input Parameters. Every node is in the training phase for specific time period after joining the network. When a node wants to transmit a packet during training phase, it executes random BEB for choosing backoff interval between [0, CW]. Each node is responsible for advertising its local data (queue length—myqlen and waiting time—mywt) along with request to send (RTS) message. Nodes that overhear or receive RTS will store the queue length and waiting time of this neighbor in its neighbor table (NT). This NT is dynamically updated every time while receiving or overhearing RTS to reflect the current status of the contending nodes. After the training phase, nodes start using our fuzzy logic algorithm to compute unique backoff interval that is explained in the next section. Updating of NT continues even after training phase. We should ensure that the duration of the training phase is sufficient for each node to overhear information from its neighbors. We have set the training phase based on the number of RTS messages overheard or received. It is set to 15 in our simulation. The steps executed by a node during transmission or reception of RTS packet are shown in Algorithm 1.

In our work, queue length and waiting time have been identified as the input parameters for medium scheduling. Input parameters play an important role in improving the performance of the algorithm. The objective of our algorithm is to have fair and quick ordering among the nodes to access the medium. Competing nodes that are possessing the same contention window size must be given the medium access based on their waiting time. It can definitely improve the fairness and delay. If the node with larger queue length suffers from getting to the media access, it can cause unexpected long delay for the packets waiting in the queue which leads to unnecessary timeout at the respective TCP sources. So, we have taken waiting time and queue length as two input parameters to our fuzzy logic based algorithm. Every node advertises the following information through its RTS message.

Waiting Time. Waiting time is defined as the interval between the time at which the current packet entered the queue and time at which it leaves front of the queue.

Queue Length. Queue length is defined as the number of packets waiting in the interface queue for transmission. It is measured when the current packet leaves the node and it reflects the total number of packets waiting for medium access.

Whenever a node overhears or receives RTS message from its one-hop neighbors, it updates the collected information in the neighbor table (NT) as shown in Table 1.

3.2. Fuzzification. Fuzzification is the process of converting the crisp or scalar input into fuzzy input parameters. Waiting time (mywt) and queue length (myqlen) of the current node are applied as the crisp input to the fuzzification phase. Fuzzy set of our crisp input waiting time takes three values {less, average, more} and fuzzy set of queue length takes the fuzzy values {short, moderate, long}. Fuzzification process applies every crisp input parameter into the membership function of each value of the fuzzy set and finds out degree of membership. It yields us the percentage of membership of the crisp input in each fuzzy value of corresponding fuzzy set.

Table 1: Neighbor table.

Neighbor ID	Queue length	Waiting time
1	QL_1	WT_1
2	QL_2	WT_2
3	QL_3	WT_3
⋮	⋮	⋮
n	QL_n	WT_n

Triangular membership function has been defined for each fuzzy value {short, moderate, long} of queue length in (1). It is used to measure the degree of membership of the given queue length for each fuzzy value. The triangular membership diagram for queue length is shown in Figure 3. In the following equation, myqlen represents the queue length of the local node when the current packet is in front of the queue:

$$\mu_s(\text{myqlen}) = \begin{cases} 1 & \text{if myqlen} \le \min Q_l \\ \dfrac{((\min Q_l + \max Q_l)/2) - \text{myqlen}}{((\min Q_l + \max Q_l)/2) - \min Q_l} & \\ \qquad \text{if } \min Q_l > \text{myqlen} < \dfrac{(\min Q_l + \max Q_l)}{2} & \\ 0 & \text{if myqlen} \ge \dfrac{(\min Q_l + \max Q_l)}{2} \end{cases}$$

$$\mu_m(\text{myqlen}) = \begin{cases} \dfrac{\text{myqlen} - \min Q_l}{((\min Q_l + \max Q_l)/2) - \min Q_l} & \\ \qquad \text{if } \min Q_l \ge \text{myqlen} < \dfrac{(\min Q_l + \max Q_l)}{2} & \\ 1 \quad \text{if myqlen} = \dfrac{(\min Q_l + \max Q_l)}{2} & \\ \dfrac{\max Q_l - \text{myqlen}}{\max Q_l - ((\min Q_l + \max Q_l)/2)} & \\ \qquad \text{if } \dfrac{(\min Q_l + \max Q_l)}{2} > \text{myqlen} \le (\max Q_l) \end{cases}$$

$$\mu_l(\text{myqlen}) = \begin{cases} 0 \quad \text{if myqlen} < \dfrac{(\min Q_l + \max Q_l)}{2} & \\ \dfrac{\text{myqlen} - ((\min Q_l + \max Q_l)/2)}{\max Q_l - ((\min Q_l + \max Q_l)/2)} & \\ \qquad \text{if } \left(\dfrac{(\min Q_l + \max Q_l)}{2} \right) \le \text{myqlen} < \max Q_l & \\ 1 \quad \text{if myqlen} \ge \max Q_l. \end{cases} \tag{1}$$

In (1), $\min Q_l$ and $\max Q_l$ represent the minimum queue length and maximum queue length among contending nodes and they act as the boundary value between fuzzy values. The scalar value of myqlen is compared with $\min Q_l$ and $\max Q_l$ and a degree of membership between 0 and 1 under each fuzzy value is obtained. $\min Q_l$ and $\max Q_l$ are calculated from the neighbor table as shown in the following equation and this calculation is done when a node needs backoff counter value just before transmission:

$$\begin{aligned} (\min Q_l &= \text{minimum}(QL_1, QL_2, QL_3, \ldots, QL_n)), \\ (\max Q_l &= \text{maximum}(QL_1, QL_2, QL_3, \ldots, QL_n)). \end{aligned} \tag{2}$$

In (2), n represents the number of neighbors of current node. QL_i represents the queue length in neighbor i. An example for finding out the degree of membership is shown in Figures 3(a) and 3(b). The membership function accepts the queue length of the current node (myqlen) and finds out its degree of membership for each fuzzy value {short, moderate, long} with respect to its contending nodes ($\min Q_l$ and $\max Q_l$). In this example, myqlen = 14; $\min Q_l$ and $\max Q_l$ are assumed as 10 and 32, respectively. These values are applied to (1) and the degree of membership for {short, moderate, long} is found to be $\{0.64, 0.36, 0\}$. The membership functions for waiting time can also be derived in the same manner.

3.3. Fuzzy Inference Engine. Fuzzy inference engine maps the fuzzy input parameters into fuzzy output parameters with the help of rule base. Here, fuzzy input parameter means the fuzzy values that have nonzero degree of membership (e.g., short, moderate from the previous example of Figures 3(a) and 3(b)) from each input parameter. The fuzzy set of our output (medium access) takes four linguistic values {immediate, fast, moderate, slow}. The triangular membership diagram for the fuzzy output is shown in Figure 4. x-axis represents the crisp value (backoff) of the fuzzy output. The input parameters to the fuzzy inference engine are shown with an example in Figure 5.

The rule base is the set of rules that reflects the exact behavior of the system. In the rule base, fuzzy value of queue length and waiting time are related to fuzzy value of our output medium access. The rule base is constructed and shown in Table 2. The row represents the possible fuzzy values of waiting time and column represents the fuzzy value of queue length. Each cell represents the fuzzy value of the output parameter medium access which will be mapped to crisp output called backoff interval. The rule is of the form

if (queue length is long and waiting time is more), then medium access is immediate.

Since the rule involves two input parameters, we need to evaluate the antecedent of each rule and obtain the membership value for the same. We use conjunction to connect the two parts of the rule. So, we can derive the consequent of each rule by applying the "and" operator to the antecedent.

We recorded the value of input parameters at a specific node during the experiment and we have shown that example

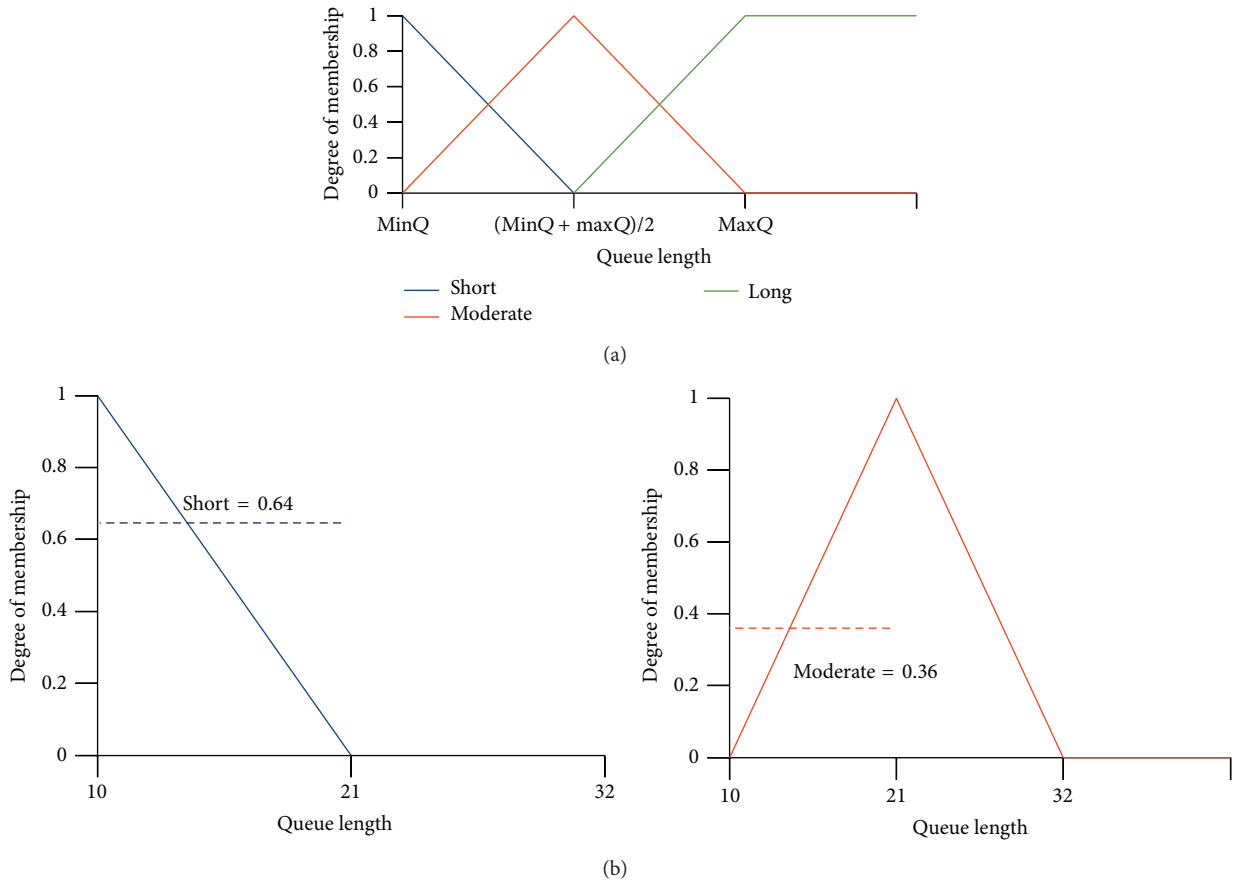


Figure 3: Membership diagram for queue length. (a) and (b) Degree of membership for short and moderate with input myqlen = 14, min $Q = 10$, and max $Q = 32$.

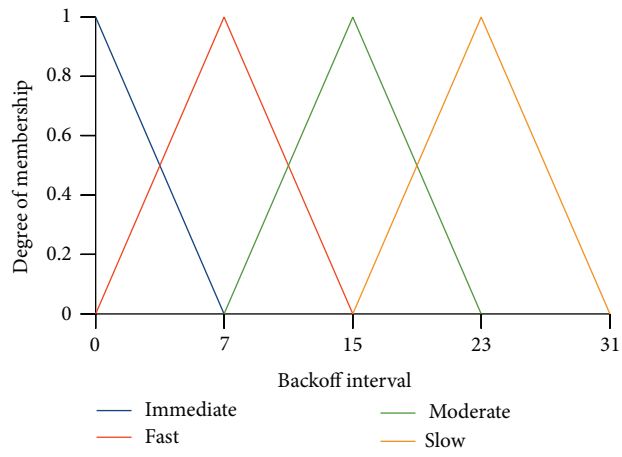


Figure 4: Membership diagram of output parameter backoff interval assuming that CW = 31.

diagrammatically for better understanding. It is drawn in Figure 6. We have derived all possible rules of the rule base for the given example. The degree of membership for each fuzzy value of the inputs is applied in the consequent and antecedent is found by applying min operator to the two parts of the consequent.

Table 2: Rule base for medium access (backoff interval).

		Myqlen		
		Short	Moderate	Long
Mywt	Less	*Slow*	*Moderate*	*Fast*
	Average	*Slow*	*Moderate*	*Immediate*
	More	*Moderate*	*Fast*	*Immediate*

3.4. Defuzzification. It is the process of mapping the fuzzy output parameters into the crisp output. Our fuzzy output parameter "medium access" takes four linguistic values {immediate, fast, moderate, slow}. The crisp values for these fuzzy set are derived from the CW. Maximum crisp value associated with each fuzzy value of medium access are $\{0, (CW/4), (CW/2) - 1, 3/4 * CW\}$. The degree of membership for each value has been obtained as the result of evaluating the rules. These membership values are applied to the centroid method of defuzzification to retrieve the crisp output. The formula for the centroid method is shown in (3). Here x represents the maximum crisp output value at which the degree of membership is 1 for this fuzzy output and $\mu(x)$

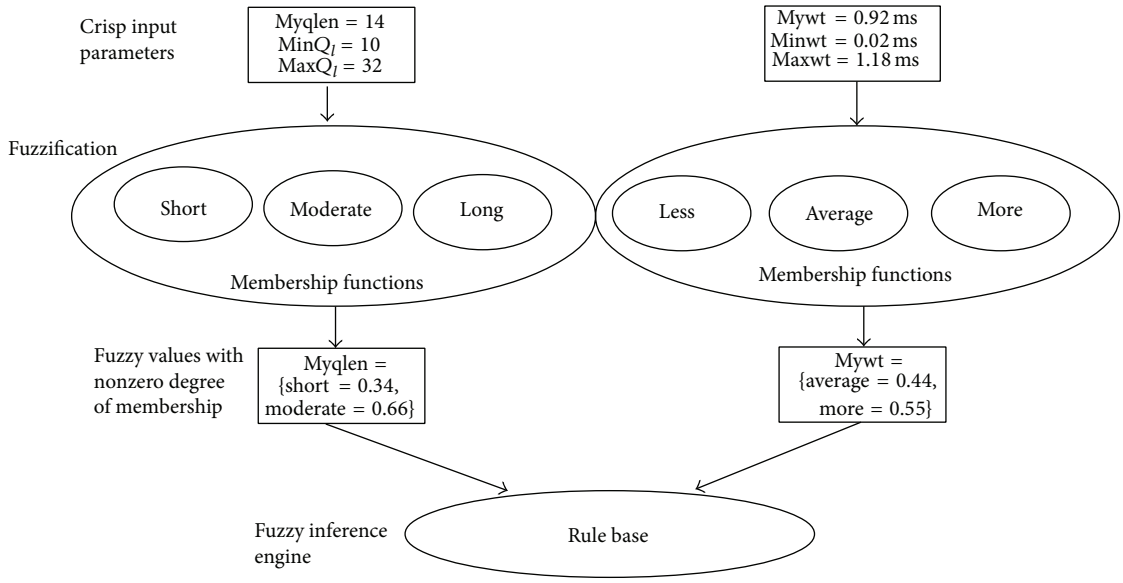


Figure 5: Input to fuzzy inference engine.

is the membership value of the fuzzy output evaluated from the rule. It is diagrammatically shown in Figure 7:

$$\text{Backoff} = \frac{\int x \cdot \mu_{(r)}(x)\, dx}{\int \mu_{(r)}(x)}. \quad (3)$$

4. Performance Evaluation

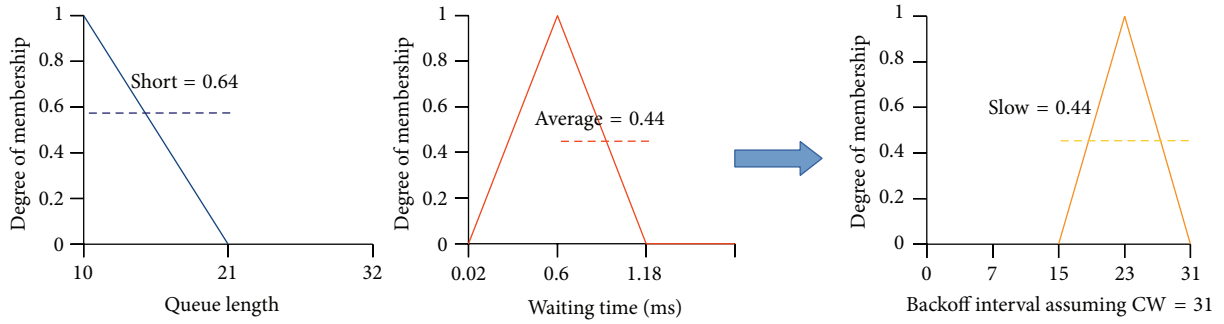
In this section, we present the simulation results of our proposed system. Our fuzzy approach can be used along with any of existing backoff algorithm to generate a unique backoff interval within CW limit where CW is dynamically updated by backoff algorithm based on channel status. We have incorporated our fuzzy logic algorithm with (i) IEEE 802.11 BEB and named it IEEE802.11 + fuzzy and (ii) GDCF [4] and named it GDCF + fuzzy. Fuzzy approach is implemented as part of MAC 802.11 layer on each node. We have shown the comparison among IEEE802.11, IEEE802.11 + fuzzy, GDCF, and GDCF + fuzzy for various parameters. The simulation is done using ns-2.35 simulator. We have considered two scenarios. In the first scenario 50 nodes are randomly deployed within the area of 500 ∗ 500. The second scenario is deployed over 1000 ∗ 1000 with 50 nodes. In the third scenario we have tested our protocol by varying both the number of nodes and number of flows over 2000 m ∗ 2000 m area. We have tested our performance for varying loads by changing the number of TCP (transmission control protocol) connections which uses FTP application. The simulation scenario is described in Table 3. We have considered the following parameters for performance evaluation.

4.1. Average Throughput. Throughput is defined as the number of bits received per second by the destination. Average throughput gives us the mean value of throughput for the destination nodes scattered in the network. We measure throughput in terms of kilobits per second. Average throughput of the fuzzy algorithm is better than IEEE 802.11 and GCF. We have plotted the average throughput of scenario 1 and scenario 2 in Figures 8 and 9, respectively. In scenario 1 the nodes are closely placed which results in severe contention and limited spatial reuse. We also observe that throughput does not follow a uniform increase or decrease. In scenario 3, overall throughput is high due to spatial reuse in 2000 ∗ 2000. It is shown in Figure 10. Our method performs better than IEEE 802.11 and also GDCF.

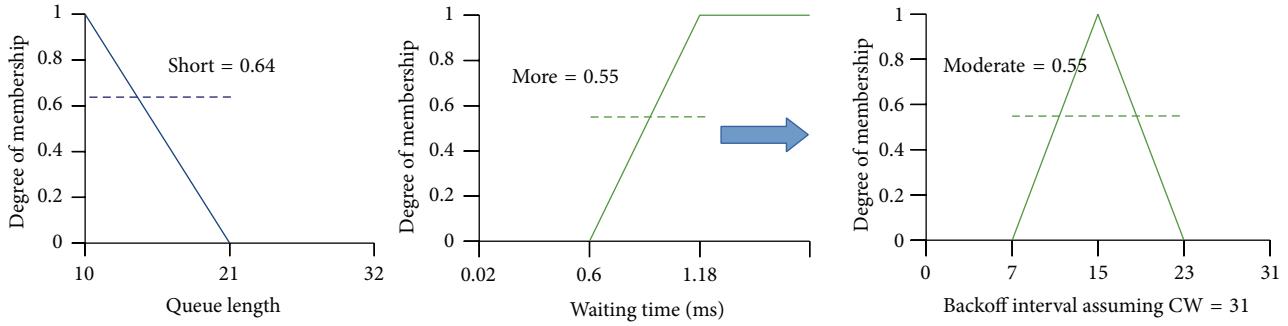
Table 3: Simulation setup.

Channel bit Rate	1 Mbps
PLCP data rate	1 Mbps
Backoff slot time	20 μs
CW_{min}	31
CW_{max}	1023
SIFS	10 μs
DIFS	50 μs
Data packet size	8000 bits
RTS packet size	160 bits + 20 bits additional overhead
Number of nodes	10 to 100

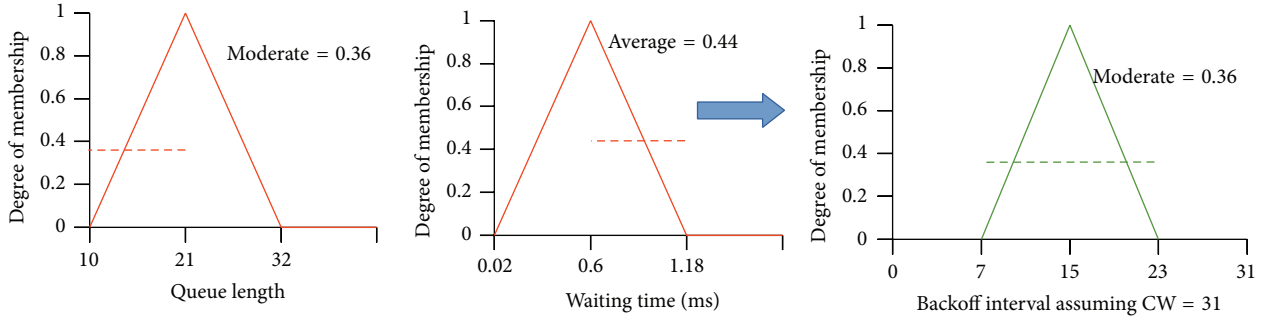
4.2. Average End to End Delay. End to end delay is defined as the time period taken for the packet to reach the destination from the source. Average delay gives us the mean delay of the packets transmitted in the end to end path. It is measured in terms of miliseconds. We observe that fuzzy based approaches outperform the others. Contention window size reflects the contention status. CW size will be more when the contention is heavy and less during low traffic.



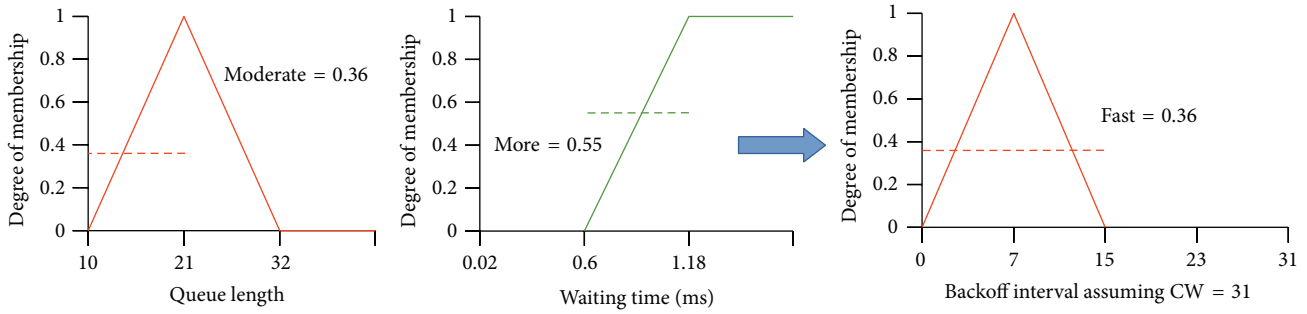
(a) If queue length is short and waiting time is average, then medium access (backoff interval) is slow



(b) If queue length is short and waiting time is more, then medium access (backoff interval) is moderate



(c) If queue length is moderate and waiting time is average, then medium access (backoff interval) is moderate



(d) If queue length is moderate and waiting time is more, then medium access (backoff interval) is immediate

FIGURE 6: Evaluating the rule base with input parameters myqlen = 14, mywt = 0.92 msec, CW = 31, $\min Q_l = 10$, $\max Q_l = 32$, min WT = 0.02 msec, and max WT = 1.18 msec.

After assigning the CW, nodes are ordered based on their waiting time and queue status. Due to the proper ordering in accessing the medium, every packet is transmitted in a quick manner without much waiting that leads to reduced end to end delay. Approximately our fuzzy methods show 50% reduction in delay in both scenarios as shown in Figures 11 and 12.

4.3. Packet Delivery Ratio. Packet delivery ratio (PDR) represents the ratio of the packets received to the packets

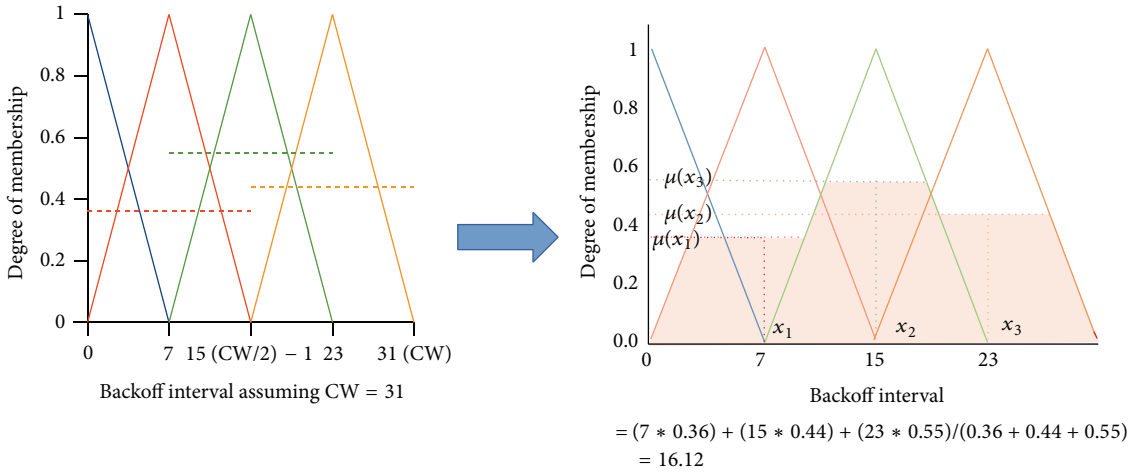


Figure 7: Aggregating the result of Rules with centroid method.

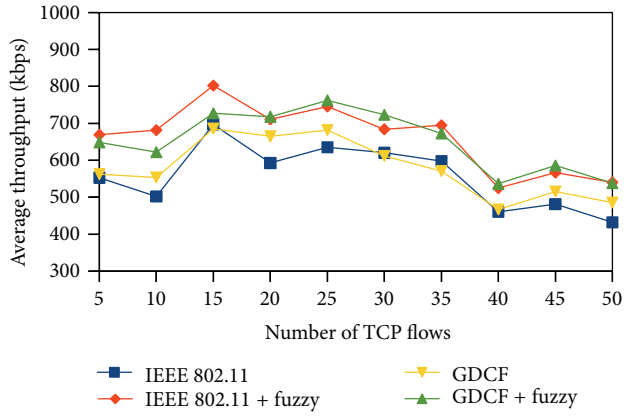


Figure 8: Average throughput for 500 ∗ 500 scenario.

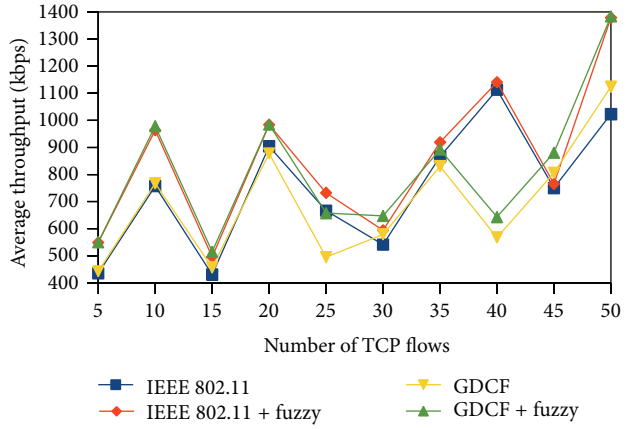


Figure 9: Average throughput for 1000 ∗ 1000 scenario.

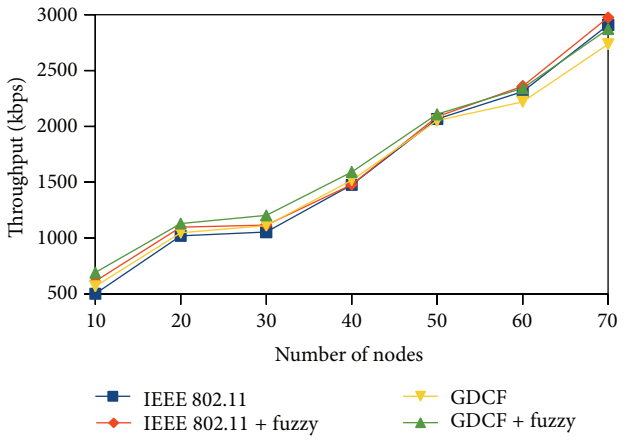


Figure 10: Average throughput for varying nodes.

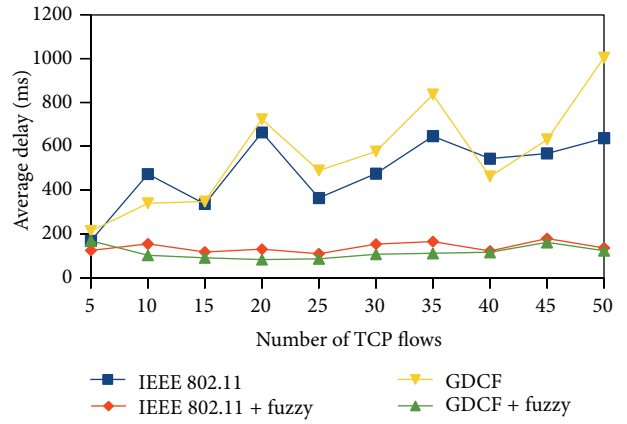


Figure 11: Average end to end delay for 500 ∗ 500 scenario.

transmitted. It is plotted in y-axis. Packet delivery ratio reflects the percentage of successful transmission. PDR of fuzzy based approaches is high than IEEE 802.11 and GDCF. It is diagrammatically shown in Figures 13, 14, and 15.

4.4. Number of Collisions. Number of collisions is a factor to measure contention in the network. Collisions are more when there is a heavy contention among the nodes to access the medium. It gets increased under increasing loads. These

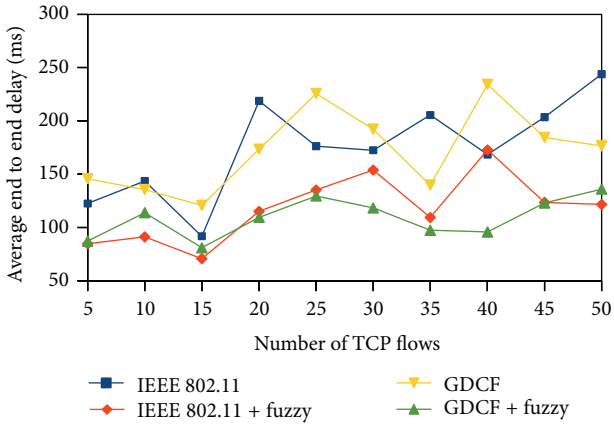


Figure 12: Average end to end delay for 1000 ∗ 1000 scenario.

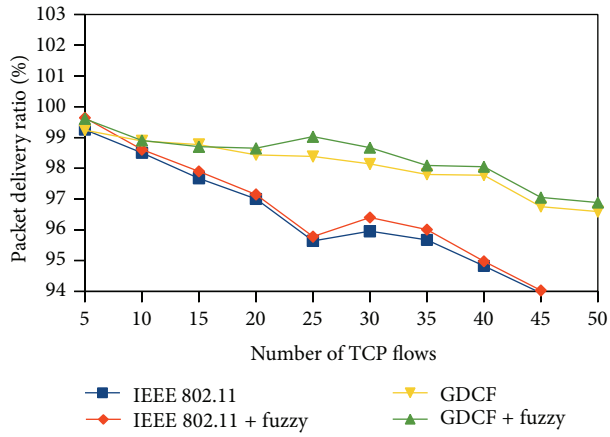


Figure 13: Packet delivery ratio for 500 ∗ 500 scenario.

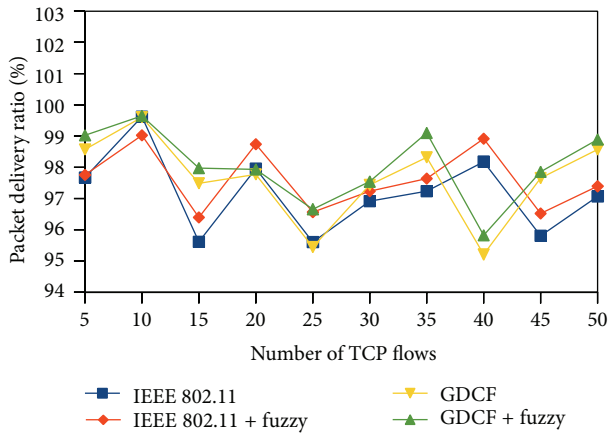


Figure 14: Packet delivery ratio for 1000 ∗ 1000 scenario.

collisions result in unsuccessful transmission that makes the node double the contention window size. Larger contention window size reduces the throughput and increases the delay. The number of collisions in our algorithm is reduced by allocating backoff interval based on nodes buffer status.

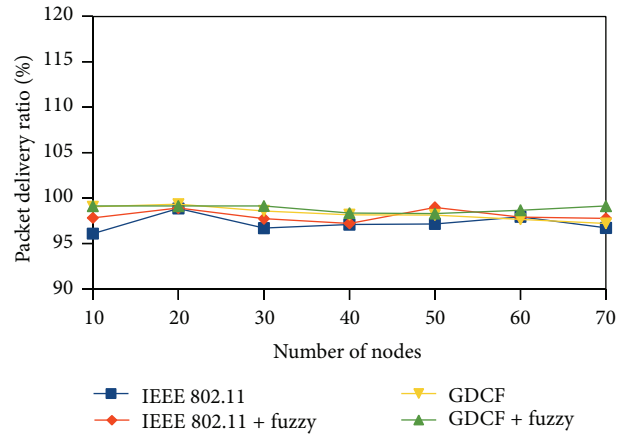


Figure 15: Packet delivery ratio for variable nodes and traffic.

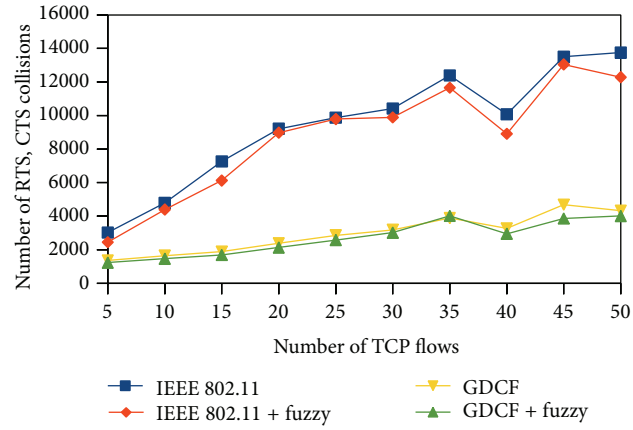


Figure 16: Number of RTS, CTS collisions for 500 ∗ 500 scenario.

Nodes with smaller buffer and smaller waiting time make themselves wait, thereby reducing the contention.

The number of collisions also denotes the interferences happened while transmitting two packets at the same time. Collisions also reflect the uniqueness of our fuzzy approach. It is an important factor that determines the uniqueness of backoff interval value generated from our algorithm. It is calculated using the number of collision drops while transmitting request to send (RTS) or clear to send (CTS).

We represent the number of collisions on y-axis. Here, we observe that the number of collisions happening in the network is less than random backoff generation algorithms of IEEE 802.11 and GDCF. It is diagrammatically shown in Figures 16, 17, and 18.

5. Conclusion

The collision avoidance mechanism of IEEE 802.11 DCF makes it inefficient and unfair especially under heavy load. IEEE 802.11 BEB algorithm makes every node select its backoff interval between [0, CW] in a random and uniform manner. But this random backoff interval can change the

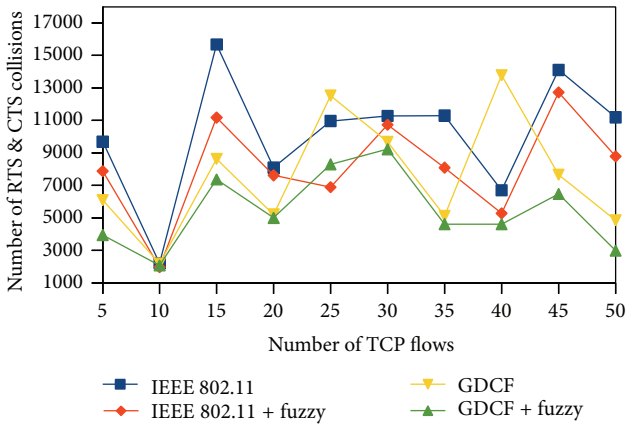


Figure 17: Number of RTS, CTS collisions for 1000 ∗ 1000 scenario.

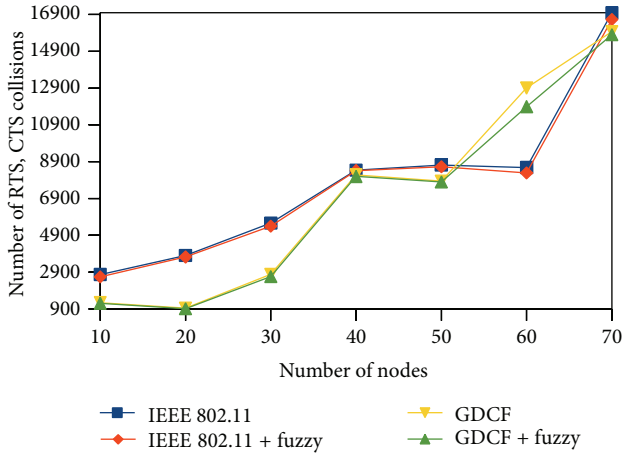


Figure 18: Number of collisions for varying traffic and nodes.

optimal order and frequency of channel access among competing nodes which result in unfairness and increased delay. We proposed an algorithm that enables each node to compute its unique backoff interval using fuzzy logic based on the input parameters collected from contending nodes through overhearing. Every node in the network finds its order to access the medium. Our algorithm makes sure that nodes waiting for a long time with more packets get to the medium quickly and nodes with small number of packets and less waiting time get to the medium later. We control the channel contention by ordering the nodes according to their waiting time. Our future work would be to test the performance for dynamic and mobile scenarios and to work on the issues related.

Conflict of Interests

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

References

[1] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 3, pp. 535–547, 2000.

[2] IEEE Standard for Information Technology, "Telecommunications and information exchange between systems- local and metropolitan area networks—specific requirements—part 11: wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications," Tech. Rep. ANSI/IEEE Std 802.11:1999, 2003.

[3] F. Calì, M. Conti, and E. Gregori, "Dynamic tuning of the IEEE 802.11 protocol to achieve a theoretical throughput limit," *IEEE/ACM Transactions on Networking*, vol. 8, no. 6, pp. 785–799, 2000.

[4] L. Bononi, M. Conti, and E. Gregori, "Runtime optimization of IEEE 802.11 wireless LANs performance," *IEEE Transactions on Parallel and Distributed Systems*, vol. 15, no. 1, pp. 66–80, 2004.

[5] D.-J. Deng, C.-H. Ke, H.-H. Chen, and Y.-M. Huang, "Contention window optimization for ieee 802.11 DCF access control," *IEEE Transactions on Wireless Communications*, vol. 7, no. 12, pp. 5129–5135, 2008.

[6] S. Y. Chang and H.-C. Wu, "Novel Adaptive dcf protocol using the computationally-efficient optimization with the feedback network information for wireless local-area networks," *IEEE Transactions on Wireless Communications*, vol. 8, no. 6, pp. 2827–2830, 2009.

[7] P. V. Krishna, S. Misra, M. S. Obaidat, and V. Saritha, "Virtual backoff algorithm: an enhancement to 802.11 medium-access control to improve the performance of wireless networks," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 3, pp. 1068–1075, 2010.

[8] K. Hong, S. K. Lee, K. Kim, and Y. H. Kim, "Channel condition based contention window adaptation in IEEE 802.11 WLANs," *IEEE Transactions on Communications*, vol. 60, no. 2, pp. 469–478, 2012.

[9] C. Wang, B. Li, and L. Li, "A new collision resolution mechanism to enhance the performance of IEEE 802.11 DCF," *IEEE Transactions on Vehicular Technology*, vol. 53, no. 4, pp. 1235–1246, 2004.

[10] Q. Xia and M. Hamdi, "Contention window adjustment for IEEE 802.11 WLANs: a control-theoretic approach," in *Proceedings of the IEEE International Conference on Communications (ICC '06)*, pp. 3923–3928, July 2006.

[11] Y. Jian and S. Chen, "Can CSMA/CA networks be made fair?" in *Proceedings of the 14th Annual International Conference on Mobile Computing and Networking (MobiCom '08)*, pp. 235–246, ACM, San Francisco, Calif, USA, September 2008.

[12] S. Chen and Z. Zhang, "Localized algorithm for aggregate fairness in wireless sensor networks," in *Proceedings of the 12th Annual International Conference on Mobile Computing and Networking (MOBICOM '06)*, pp. 274–285, ACM Press, New York, NY, USA, September 2006.

[13] T. Razafindralambo and I. Guérin-Lassous, "Increasing fairness and efficiency using the MadMac protocol in ad hoc networks," *Ad Hoc Networks*, vol. 6, no. 3, pp. 408–423, 2008.

[14] J. Chen and W. Wu, "Dynamic contention window selection scheme to achieve a theoretical throughput limit in wireless networks: a fuzzy reasoning approach," in *Proceedings of the 60th IEEE Vehicular Technology Conference (VTC '04)*, vol. 5, pp. 3196–3200, September 2004.

[15] T. Razafindralambo and I. Guérin Lassous, "SBA: a simple backoff algorithm for wireless Ad Hoc networks," in *NETWORKING 2009: Proceedings of the 8th International IFIP-TC 6 Networking Conference, Aachen, Germany, May 11–15, 2009*, L. Fratta, H. Schulzrinne, Y. Takahashi, and O. Spaniol, Eds., vol. 5550 of *Lecture Notes in Computer Science*, pp. 416–428, Springer, Berlin, Germany, 2009.

[16] L. Gannoune and S. Robert, "Dynamic tuning of the contention window minimum (CWMIN) for enhanced service differentiation in IEEE 802.11 wireless ad-hoc networks," in *Proceedings of the IEEE 15th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC '04)*, pp. 311–317, IEEE, September 2004.

[17] A. Balador, A. Movaghar, S. Jabbehdari, and D. Kanellopoulos, "A novel contention window control scheme for IEEE 802.11 WLANs," *IETE Technical Review*, vol. 29, no. 3, pp. 202–212, 2012.

[18] A. Ksentini, A. Nafaa, A. Gueroui, and M. Naimi, "Determinist contention window algorithm for IEEE 802.11," in *Proceedings of the IEEE 16th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC '05)*, vol. 4, pp. 2712–2716, September 2005.

[19] S. N. Sivanandam, S. Sumathi, and S. N. Deepa, *Introduction to Fuzzy Logic Using MATLAB*, Springer, Secaucus, NJ, USA, 2006.

[20] T. Abdelkader and K. Naik, "A localized adaptive strategy to calculate the backoff interval in contention-based vehicular networks," *IEEE Access*, vol. 2, pp. 215–226, 2014.

[21] M. Baklizi, H. Abdel-Jaber, A. A. Abu-Shareha, M. M. Abualhaj, and S. Ramadass, "Fuzzy logic controller of gentle random early detection based on average queue length and delay rate," *International Journal of Fuzzy Systems*, vol. 16, no. 1, pp. 9–19, 2014.

[22] W.-J. Chang, P.-H. Chen, and C.-T. Yang, "Robust fuzzy congestion control of TCP/AQM router via perturbed Takagi-Sugeno fuzzy models," *International Journal of Fuzzy Systems*, vol. 15, no. 2, pp. 203–213, 2013.

[23] Z. Li, S. Nandi, and A. K. Gupta, "Achieving MAC fairness in wireless ad-hoc networks using adaptive transmission control," in *Proceedings of the 9th International Symposium on Computers and Communications (ISCC '04)*, pp. 176–181, July 2004.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2015, Article ID 475806, 9 pages
http://dx.doi.org/10.1155/2015/475806

Research Article

A Double Herd Krill Based Algorithm for Location Area Optimization in Mobile Wireless Cellular Network

F. Vincylloyd[1] and B. Anand[2]

[1]*RVS Technical Campus, Coimbatore 641 402, India*
[2]*Hindusthan College of Engineering and Technology, Coimbatore 641 032, India*

Correspondence should be addressed to F. Vincylloyd; vincylloyd@gmail.com

Received 20 August 2014; Revised 27 October 2014; Accepted 28 October 2014

Academic Editor: S. N. Deepa

In wireless communication systems, mobility tracking deals with determining a mobile subscriber (MS) covering the area serviced by the wireless network. Tracking a mobile subscriber is governed by the two fundamental components called location updating (LU) and paging. This paper presents a novel hybrid method using a krill herd algorithm designed to optimize the location area (LA) within available spectrum such that total network cost, comprising location update (LU) cost and cost for paging, is minimized without compromise. Based on various mobility patterns of users and network architecture, the design of the LR area is formulated as a combinatorial optimization problem. Numerical results indicate that the proposed model provides a more accurate update boundary in real environment than that derived from a hexagonal cell configuration with a random walk movement pattern. The proposed model allows the network to maintain a better balance between the processing incurred due to location update and the radio bandwidth utilized for paging between call arrivals.

1. Introduction

Location areas represent a significant strategy of location management, used to reduce signalling traffic imposed by location updating and paging messages in mobile cellular networks. Due to the increasing dimension spaces to be searched, the location of optimal LAs represents a NP-hard optimization problem. In contrast to a landline telephonic network, mobile wireless cellular network (MWCN) accommodates dynamically relocatable service users with whom location uncertainty is always associated. To reduce this location uncertainty, each mobile terminal has to report its location information in regular interval, which is called an LU procedure. In dynamic LU scheme, the frequency of LU performed by a mobile terminal (MT) depends upon a stochastic phenomenon, which is user's movement behavior [1–6].

Upon the arrival of a mobile-terminated call, it is the responsibility of the network to search for the terminal for delivering the call successfully. This search is an iterative process, which continues until the terminal is successfully located. The frequency of paging to be performed by the network, per user, depends upon another stochastic phenomenon, which is incoming call arrival process for each user [7].

Since LU and paging process both consume sufficient amount of radio resource, cost is incurred for performing an LU as well as for paging. Both of these processes are coupled in a sense that there is an inherent trade-off between these two cost components, and these two together determine the total network cost. The size of the LA, in particular, affects the signalling load generated due to paging and LU. From a designer's point of view, it is required to find out an optimum size of LA such that the desired cost effectiveness can be achieved.

More precisely, the location area (LA) planning represents a vital role in cellular networks because of the trade-off created by paging and registration signaling. The upper bound on the size of an LA is the service area of a mobile switching center (MSC). In that extreme case, the cost of paging is at its highest, but no registration is needed. On

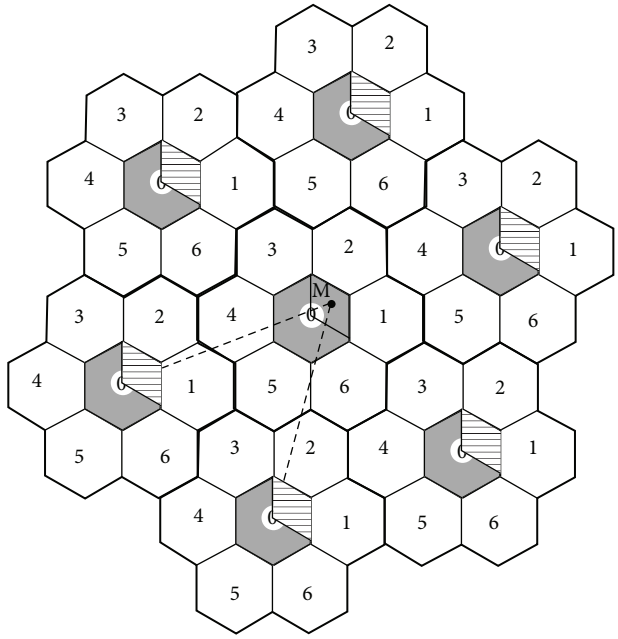


Figure 1: Cellular network architecture for the movement of MTs.

the other hand, if each cell is an LA, the paging cost is minimal, but the registration cost is the most. In general, the most important component of these costs is the load on the signaling resources. Between the extremes lie one or more partitions of the MSC service area that minimize the total cost of paging and registration. The present work falls into the class of location area planning (LAP) problem [8, 9].

For hexagonal cell configuration, [10] had tried to find out optimum movement threshold value, which would, in effect, determine the number of cells, which collectively could be considered as a dynamic optimal LA (as it depends upon two stochastic phenomena, namely, call arrival pattern and microscopic behavioural pattern of terminal mobility). Due to the uniqueness of design considerations and problem formulation in [10], they used GA and SA method to solve the problem and have compared the quality of solutions found by each for different inputs.

In fact, there is a growing body of literature in the application of emerging heuristics to solve the optimizing problems in various fields of science and engineering, but there is a huge vacuum in the application of heuristics for the location area (LA) problem. This paper proposes a Krill Herd based algorithm for minimizing total network cost, comprising location update (LU) cost and cost for paging. Krill herd algorithm (KHA) is a recently developed powerful evolutionary algorithm proposed by Gandomi and Alavi [11]. The KHA is based on the herding behaviour of krill individuals. Each krill individual modifies its position using three processes, namely, (1) movement induced by other individuals, (2) foraging motion, and (3) random physical diffusion.

Although some may disagree that a suitable algorithm design would assure a high probability of finding solution, population size does indirectly contribute to the effectiveness and efficiency of the performance of an algorithm [12]. The prime deciding factor of population size on any population-based heuristic algorithms is the execution cost. If an algorithm involves large population size, it will search thoroughly and increase the chance of exploring the entire search space and locating possible good solutions but unavoidably bear an unwanted and high computational cost. The other version is if an algorithm with small population size may suffer from premature convergence or may search partially the search space. Perhaps suggesting heuristically a suitable population size may be adequate because one need not know the exact fitness landscape to solve a complex optimization problem. Hence, a compromised, yet effective, solution would be dynamically adjusting the population size to explore the search space in balance between computational cost and the attained performance [13].

In this paper, the basic KHA is enhanced by incorporating a dual population criterion to find an optimal solution of the above problem. There are few literatures that tackle the issue of population size with various heuristics [14–16]. The rest of paper is organized as follows. In Section 2, the system as well as the proposed model in [10] is revisited. In Section 2.3, the constrained cost optimization problem for LA planning is mathematically formulated. In Section 3.1, the KHA technique is overviewed in general and then the KHA based algorithm is proposed for solving the problem of interest. In Section 3.1.1, the dual herd KHA based algorithm is discussed. In Section 4.2, some representative results are presented. Section 5 concludes the present work.

2. System and Model Description

2.1. System Description. Figure 1 shows that the cellular network coverage area is comprised of hexagonal shaped cells.

The entire coverage area is partitioned into rings of cells. The center cell is defined to be a cell where an MT has performed the last LU. An MT resides in each cell it enters, for a generally distributed time interval and then it can move to any of the neighbouring cells. The movement of an MT is assumed to be a simple random walk [17]. The next LU is performed by the MT, when the number of cell boundary crossings, since the last LU, equals a threshold value d. It is also assumed that MTs move in a radial direction as shown in Figure 1.

If an MT makes, say, d movements in one particular radial direction, as shown in Figure 1, then the LA will be defined as the area within $(d-1)$ rings from the center cell. If d assumes an optimum value, then the corresponding LA will be an optimum LA [18].

If the LA consists of D rings, then

$$D = (d-1). \tag{1}$$

The number of cells within the LA, L_D will be

$$N(D) = 3 * (D+1) * D + 1. \tag{2}$$

The perimeter $L(D)$ of the LA L_D, can be calculated as

$$L(D) = (12D+6) * R, \tag{3}$$

where R denotes the radius of the circle inscribing a hexagonal cell and the area of each cell is $(3\sqrt{3}/2)R^2$ and

$$d = 2\sqrt{3}R. \tag{4}$$

The area $S(D)$ of the LA L_D is

$$S(D) = [3D * (D+1) + 1] * (2.6) * R. \tag{5}$$

Next, it is assumed that the incoming call arrivals to each MT follow a Poisson process. The paging area is the entire LA that is the area within $(d-1)$ rings from the center cell.

2.2. Analytical Model. Let $\alpha(k)$ be the probability that there are K boundary crossings performed by an MT between two successive call arrivals. If the probability density function of cell residence time t_m has the Laplace-Stieltjes transform $F_m^*(s)$ MT and mean $1/\lambda_m$, the call arrival to each terminal follows Poisson process with rate λ_c. Based on these assumptions, the expression of $\alpha(k)$ can be derived as follows [6]. Here, θ is the call-to-mobility ratio (CMR), where CMR is defined as λ_c/λ_m [6]:

$$\alpha(k) = \begin{cases} 1 - \dfrac{1}{\theta}\left[1 - F_m^*(\lambda_c)\right], & K = 0, \\ \dfrac{1}{\theta}\left[1 - F_m^*(\lambda_c)\right]^2 \left[F_m^*(\lambda_c)\right]^{K-1}, & K > 0. \end{cases} \tag{6}$$

Considering that cell residence time t_m follows a Gamma distribution, we get [6]

$$F_m^*(s) = \left[\frac{(\lambda_m \gamma)}{(s+\lambda_m \gamma)}\right]^{\gamma}, \tag{7}$$

where

$$\gamma = \frac{1}{(\nu_{\mathrm{ar}} \lambda_m^2)}. \tag{8}$$

Let $\beta(k, K)$ denote the probability that the MT is k rings away from the center cell, given that the mobile user has already performed K number of cell boundary crossings. To depict the mobility pattern of an MT, a 2D random walk model is considered. Let us assume that P_K denotes the $K \times K$ state transition matrix, where an element $p_{i,j,k}$ in P_K gives us the probability that a mobile terminal moves from one ith ring cell to one jth in single step. Then, the probability $\beta(k, K)$ comes out to be [6]

$$\beta(k, K) = \begin{cases} P_K; & \text{Single step state transition matrix,} \\ P_K^* P_K^{(n-1)}; & n \text{ step transition matrix.} \end{cases} \tag{9}$$

Since we want to find out the optimum value of d, we express LU cost C_u as a function of d and the expression is as follows (detailed derivation is given in [19]):

$$C_u(d) = \frac{U^* \left(Y\left[1 - F_m^*(\lambda_C)\right]/\theta\right) * \left(\left[F_m^*(\lambda_c)\right]^{d-1}\right)}{\left(1 - \left[F_m^*(\lambda_C)\right]^d\right)^2}. \tag{10}$$

Similarly, we derive paging cost C_v as a function of d and it is as follows (detailed derivation is in [19]):

$$C_v(d) = V * \sum_{k=0}^{d-1} \rho_{-k} N(k) \pi \phi_k, \tag{11}$$

where Y is the number of MT attached to the network within the LA and U and V are respective cost coefficients for performing an LU and paging and ϕ_k is the probability of finding the MT within the LA, L_k; the density of the MT in that area is denoted by ρ_k. So the total cost is

$$C_T(d) = C_u(d) + C_v(d) \tag{12}$$

or

$$C_T(d) = \frac{U * \left(Y\left[1 - F_m^*(\lambda_C)\right]/\theta\right) * \left(\left[F_m^*(\lambda_C)\right]^{d-1}\right)}{\left(1 - \left[F_m^*(\lambda_C)\right]^d\right)^2} + V \sum_{k=0}^{d-1} \rho_{-k} N(k) \phi_k. \tag{13}$$

2.3. Problem Formulation. The constrained optimization problem can be stated mathematically as

$$\text{Minimize} \quad (P) : C_T(d) = C_u(d) + C_v(d) \tag{14}$$

$$\text{Subject to:} \quad 0 < p < 1 \tag{15}$$

$$q_k S(k) \rho_k = N_k p \tag{16}$$

$$R < R_{\max}. \tag{17}$$

In constraint (15), p denotes the penetration factor. Constraint (16) gives the total number of MTs in a service area where ρ_k is the density of the MTs in an LA L_k, and there are q_k switches. This number must be greater than or equal to the total number of attached mobile users. In this constraint N_k denotes population size within the LA. The maximum radius is to be considered so that the constraint on system power budget is not violated with respect to (17).

3. Solution Methodology

The high complexity associated with the optimization problem (P) necessitates computationally efficient and robust tools. Traditional techniques like gradient search, linear programming, quadratic programming, and so forth were not fruitful due to the complexity involved in the problem. This encouraged us to propose two heuristics which are based on krill herd algorithm and are presented in the next section. However, the quality of solution (compared to classical optimization techniques) is often traded off against computation time, and one can always reach a near optimal solution within bounded computation time.

3.1. Krill Herd Algorithm: An Overview. Krill herd algorithm (KHA) is a recently developed heuristic algorithm based on the herding behavior of krill individuals. It has been first proposed by Gandomi and Alavi [11]. It is a population-based method consisting of a large number of krill in which each krill moves through a multidimensional search space to look for food. In this optimization algorithm, the positions of krill individuals are considered as different design variables and the distance of the food from the krill individual is analogous to the fitness value of the objective function. In KHA, the individual krill alters its position and moves to the better positions. The movement of each individual is influenced by the three processes, namely, (i) induction process, (ii) foraging activity, and (iii) random diffusion. These operators are briefly explained and mathematically expressed as follows.

(i) Induction. In this process, the velocity of each krill is influenced by the movement of other krill individuals of the multidimensional search space and its velocity is dynamically adjusted by the local, target, and repulsive vector. The velocity of the ith krill at the mth movements may be formulated as follows [19]:

$$v_i^m = \alpha_i v_i^{\max} + \omega_n v_i^{m-1},$$
$$\alpha_i = \sum_{j=1}^{N_S} \left[\frac{f_i - f_j}{f_w - f_b} \times \frac{Z_i - Z_j}{\left|Z_i - Z_j\right| + \text{rand}\,(0,1)} \right] + 2\left[\text{rand}\,(0,1) + \frac{i}{i_{\max}}\right] f_i^{\text{best}} Z_i^{\text{best}}, \tag{18}$$

where $V_i^{\max}$ is the maximum induced motion; V_i^m and V_i^{m-1} are the induced motion of the ith krill at the mth and $(m-1)$th movement; ω_n is the inertia weight of the motion induced; f_w and f_b are the worst and the best positions, respectively, among all krill individuals of the population; f_i and f_j are the fitness value of the ith and jth individuals, respectively; N_S is the number of krill individuals surrounding the particular krill; i and $i_{\max}$ are the current iteration and the maximum iteration number.

A sensing distance (SD_i) parameter is used to identify the neighboring members of each krill individual. If the distance between the two individual krill is less than the sensing distance, that particular krill is considered as neighbor of the other krill. The sensing distance may be represented by [19]

$$\text{SD}_i = \frac{1}{5n_p} \sum_{k=1}^{n_p} \left|Z_i - Z_k\right|, \tag{19}$$

where n_p is the population size; Z_i and Z_k are the position of the ith and kth krill, respectively.

(ii) Foraging Action. Each individual of krill updates its foraging velocity according to its own current and previous food location. The foraging velocity of the ith krill at the mth movement may be expressed by [19]

$$V_{f_i}^m = 0.02\left[2\left(1 - \frac{i}{i_{\max}}\right) f_i \frac{\sum_{k=1}^{N_S} (Z_k/f_k)}{\sum_{k=1}^{N_S} (1/f_k)} + f_i^{\text{best}} X_i^{\text{best}}\right] + \omega_x V_{f_i}^{m-1}, \tag{20}$$

where ω_X is the inertia weight of the foraging motion; $V_{f_i}^{m-1}$ and $V_{f_i}^m$ are the foraging motion of the ith krill at the $(m-1)$th and m movement.

(iii) Random Diffusion. In KHA algorithm, in order to enhance the population diversity, random diffusion process is incorporated in krill individuals. This process maintains or increases the diversity of the individuals during the whole optimization process. The diffusion speed of krill individuals may be expressed as follows [19]:

$$v_{D_i}^m = \mu V_D^{\max}, \tag{21}$$

where $V_D^{\max}$ is the maximum diffusion speed; μ is a directional vector uniformly distributed between $(-1, 1)$.

(iv) Position Update. In KHA, the krill individuals fly around in the multidimensional space and each of krill adjusts its position based on induction motion, foraging motion, and diffusion motion. In this way, KHA combines local search with global search for balancing the exploration and exploitation. The updated position of the ith krill may be expressed as [19]

$$Z_i^{m+1} = Z_i^m + \left(V_i^m + V_{f_i}^m + V_{D_i}^m\right) P_t \sum_{j=1}^{N_d} \left(u_j - l_j\right), \tag{22}$$

where N_d is the number of control variables; u_j and l_j are the maximum and minimum limits of the jth control variable; P_t is the position constant factor. The above procedure will be used to optimize (14) for location area (LA).

In order to speed up the convergence property and to find better results, the crossover and mutation operations of DE are combined with the proposed algorithm to utilize the exploration ability of DE. These two operators are briefly described below.

3.1.1. Mechanism of the Dual Herd Algorithm. This mechanism is adopted from [15], which uses this for the differential evolution (DE) algorithm. Specifically, this search mechanism is adopted to maintain the population diversity of the krill herd. Also, the original krill herd algorithm is a good local search optimizer [11].

Step 1 (initialization). This step initializes the population for exploration of the search space or else an equal of the herd truncated from the previous initialization will be taken.

Step 2. Evaluating the new positions using

$$Z_i^{m+1} = Z_{r1}^m + F * \left(Z_{r2}^m - Z_{r3}^m\right), \tag{23}$$

where $F = \text{rand}(0.01, 1)$ and $r1$, $r2$, and $r3$ are random integers generated from truncated herd.

Step 3. If $\text{rand}(0, 1) \geq \text{CR}$ and $j \neq k$, then set $Z_i^{m+1} = Z_i^m$, where $\text{CR} = \text{rand}(0, 1)$.

Step 4. Or if $\text{rand}(0, 1) < \text{PM}$, then $Z_i^{m+1} = \text{rand}(\text{LB}_j, \text{UB}_j)$, where $\text{PM} = 0.5$.

Step 5. Repeat from Step 2, until all the individuals in the truncated herd are updated.

The above procedure will ensure the algorithms thorough search of the solution space as it confirms a random choice of candidates [16] compared to a directional policy.

3.1.2. A Double Herd Krill (DHK) Algorithm Developed for Constrained Optimization Problem (P). The proposed double herd KH (DHKH) algorithm problem of interest in this paper is described as follows.

Pseudocode for the DHK Algorithm

Step 1 (data structures). Defining the simple bounds, determining algorithm parameter(s), and so forth.

Step 2 (initialization). Randomly creating the initial population in the search space.

Step 3 (fitness evaluation). Evaluation of each krill individual according to its position.

Step 4. Motion calculation.

Step 5. Motion induced by the presence of other individuals:

foraging motion,

physical diffusion.

Implement the updating steps of the dual herd algorithm (Section 3.1.1).

Step 6 (updating). Updating the krill individual position in the search space.

Step 7 (repeating). Go to Step 3 until the stop criteria are reached.

Step 8. End

Krill herd algorithm is applied twice in order to maintain the diversity in the search process. In a single herd algorithm, the search process towards the location area attempts to get hold with that of the local optimal before reaching the convergent solution. As a result, krill herd algorithm is carried out twice which involves diversity in search process and increases the exploration capability during search mechanism resulting in better convergence on the solution in comparison with that of the single krill herd algorithm considered for finding the optimal solution.

4. Performance Analysis

In this section, first we have varied some system parameters to observe their reflections on the model behavior, without considering any constraints. Next, we have run our optimization algorithms and have presented some representative results. Lastly, we have compared the performance of the heuristics proposed.

4.1. Model Variation. To validate the analytical model, we have varied the value of movement threshold d and obtained nature of variation of LU cost, paging cost, and total cost (Figures 2–4), without considering any constraint. The objective of this endeavour is to validate our model as well as our proposed heuristics with respect to that cited in [6]. Through this effort, we find out the region within the entire search space where optimal or near optimal solution could be found.

We take three different values of CMR to demonstrate the effect of changing mobility and call arrival patterns. For a very small value of d, the LU cost is very high as lesser number of d implies that the frequency of LU is higher and vice versa. This fact can be corroborated from Figure 2. From Figure 2, it has also been seen that, for different CMR values, the LU cost is different. Low CMR means the probability of boundary crossing is high by an MT between two successive calls, which results in higher location update cost. For higher CMR values, the situation is just the reverse. Paging cost also varies as d changes. If the value of d is small, the size of the LA will be small and the number of cells within the LA will be less. It has been observed in Figure 3 that, for small values of d, paging cost is minimal.

As we expected, the paging cost also varies with CMR. If the CMR is high, call arrival rate is high and for each call meant for a mobile terminal, the network will have to go through the call delivery process, which will raise the paging cost significantly.

The total cost C_T varies as d changes, which is plotted in Figure 4. For smaller value of d, the total cost is high. As we have already explained, even though the paging cost is less, the LU cost is very high. For certain value of d, the total cost

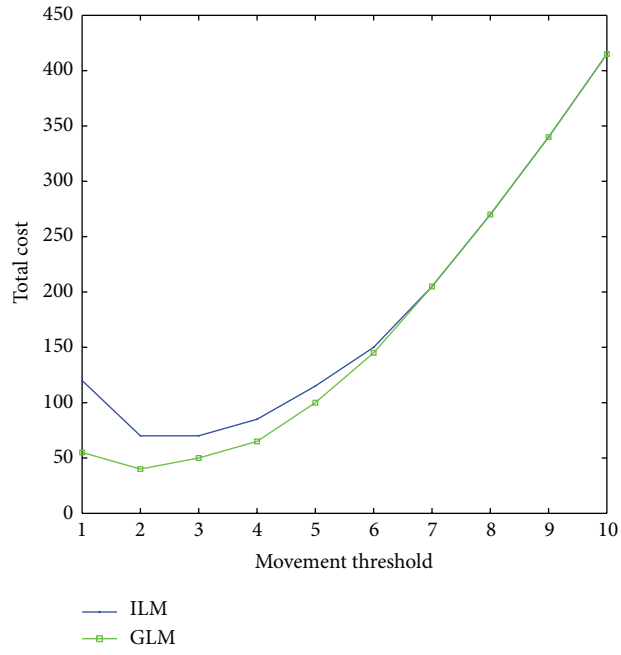


Figure 2: LU cost for various CMR values.

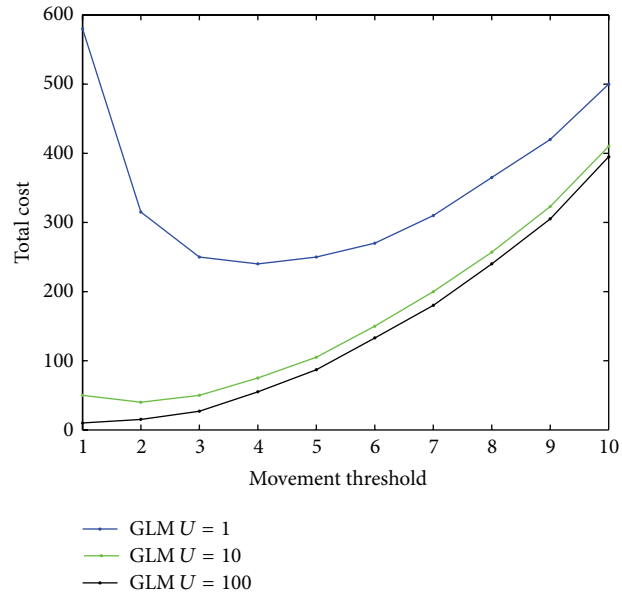


Figure 3: Paging cost for various CMR values.

attains a minimum value, which is the optimum value of d. From Figure 4, we find that, for the unconstrained problem for three different values of CMR, the optimum value of d lies in the interval [2, 4] when we used our heuristics. The nature of the cost variation with d as well as with CMR is the same as presented in [6] with delay equal to one.

4.2. Results on Optimization of (P). In this section, we present the results obtained after solving the constrained optimization problem (P) using KHA and DHKA. To evaluate the

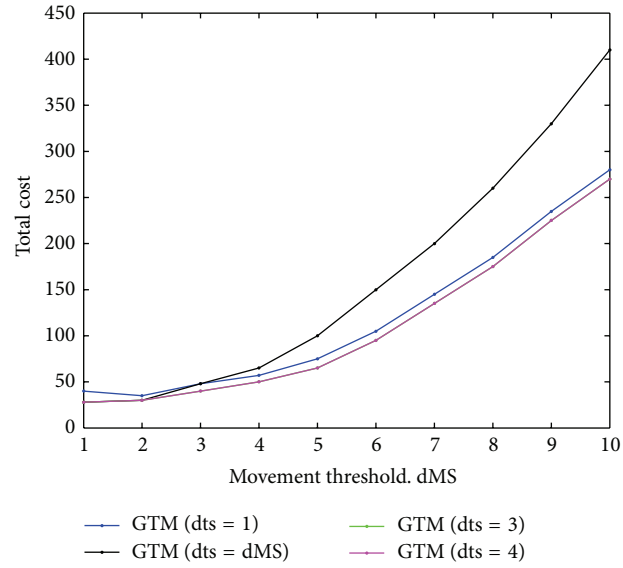


Figure 4: Total cost for various CMR values.

Table 1: Parameter settings for the DHK algorithm.

Parameter	Value
Number of krill in the population (KN)	40
Number of iterations (IN)	6000
Foraging speed V_f	0.02
Diffusion speed $D_{\max}$	0.006
Initial violation tolerance (ε)	1.0
Decrement (dec)	1.002
$\Omega_{\min}$	0
$\omega_{\max}$	1

performance of the LAP problem, we have varied the population and network size as well as CMR and have attempted to find out optimum cost for these varying input parameters. We have assumed that in our microcellular configuration each hexagonal cell has a radius of 1 km. Moderate user density (50 users/km^2) is considered. To calculate the cost coefficient for LU (U) is 1 and that for paging (V) is 1. The parameter settings for the proposed krill herd algorithm are delineated in Table 1. These values are commonly used for both krill herd algorithms used in this research.

Figure 5 shows the variation of perimeter of optimum LA with varying population size. If the population size is increased, optimum value of d increases and hence the size of the LA is increased. It is obvious that if larger number of MTs, serviced by a single MSC, are to be accommodated in a single LA, the number of MTs residing closer to the boundary of an LA will be high. Due to this, the possibility of number of boundary crossings by mobile terminals also increases. To minimize the total cost, the movement threshold value d would also increase so that after each boundary crossing one LU does not take place. However, Figure 5 shows that DHKA results in slightly smaller LA. For a population size of 800, DHKA finds an optimum d value, which is 5.26% less

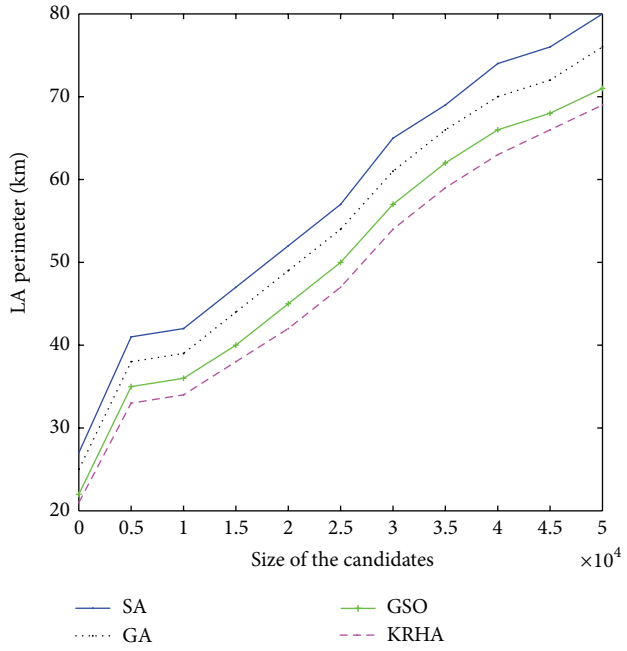


Figure 5: Variation of perimeter of location area for change in candidate size.

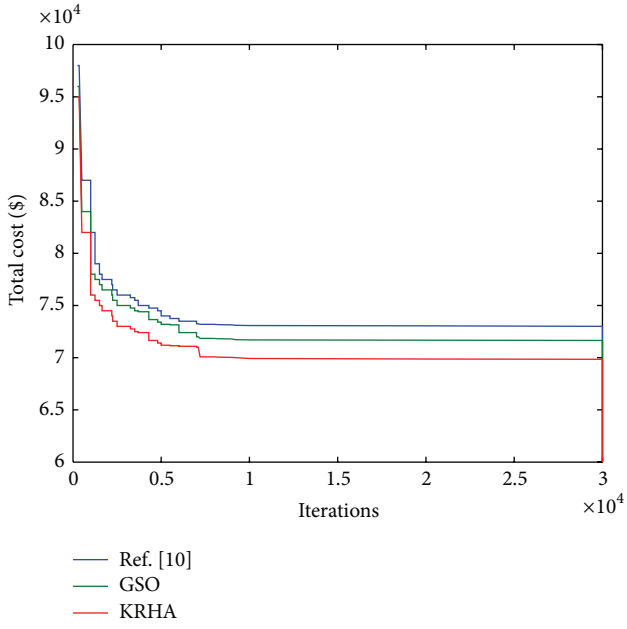


Figure 6: Variation of the optimum total cost for various CMR values.

than that generated by SA. However, for a population size of 20,000, using GA, the optimum value that we obtain is 7.9% less than that obtained by running SA.

Figure 6 shows the nature of variation of the optimum LU cost, paging cost, and the total cost with number of cells per LA with $\lambda_C = 0.1$ and CMR = 0.1. In this case, as done in [10], the simulation set-up varied the network size consisting of 7 cells to 469 cells corresponding to movement threshold

Table 2: Results obtained for LA minimization problem using various methods.

No. of cells	Method	LU cost	Paging cost	Total cost
13	[10]	189521	18547	208068
	KHA	188654	18332	206986
	DHKA	188551	18254	206805
19	[10]	87661	19365	107026
	KHA	87421	19122	106543
	DHKA	87225	19085	106310
25	[10]	46442	22541	68983
	KHA	46102	22269	68371
	DHKA	45894	21687	67581
31	[10]	25647	44221	69868
	KHA	25362	44095	69457
	DHKA	25112	43885	68997
37	[10]	17889	53252	71141
	KHA	17567	53054	70621
	DHKA	17339	52985	70324

values in the interval [1, 17]. However, while presenting the results, we have rounded up the values of the number of cells per LA to avoid any confusion. For cell size 37 onwards, the LU cost is almost insignificant. However, the paging cost increases as the size of an LA increases.

From Table 2, it is apparent that an LA consisting of cells in the interval [19, 20] results in minimum total cost. Performancewise, both algorithms generate results which are comparable; average paging cost with DHKA came out to be 0.13% less than that obtained by KHA; average total cost only varies by 0.12%. Variation in LU cost is almost negligible. Here, all the LU cost, paging cost, and subsequently the total cost are comparatively less when the cell number is 25. Unlike the results in [10], this research has identified a new set of results, confirming that the location area problem minimization is much effectively solved using the DHKA algorithm compared to the GA and SA methods in [10].

Next, we have varied the value of CMR to investigate its impact on optimum total cost per call arrival. The nature of the variation is shown in Figure 7, where the optimum cost per call arrival decreases with increase in CMR. It is apparent that, for low CMR, call arrival rate is low and the mobility rate of the MT is high. In this situation, the possibility of boundary crossings by an MT increases. This would increase the LU cost significantly, whereas paging cost will be minimal. As CMR increases, the LU cost drastically decreases and so does the total cost. Optimum movement threshold value d and hence the perimeter of an LA obtained DHKA come out to be less than those SSA calculates. This pattern becomes apparent as CMR value increases from 0.025 onwards.

4.3. Comparative Study of the Performance of the Proposed Solution Methodology. To assess the performance of the two heuristics we have used in this research, we have compared

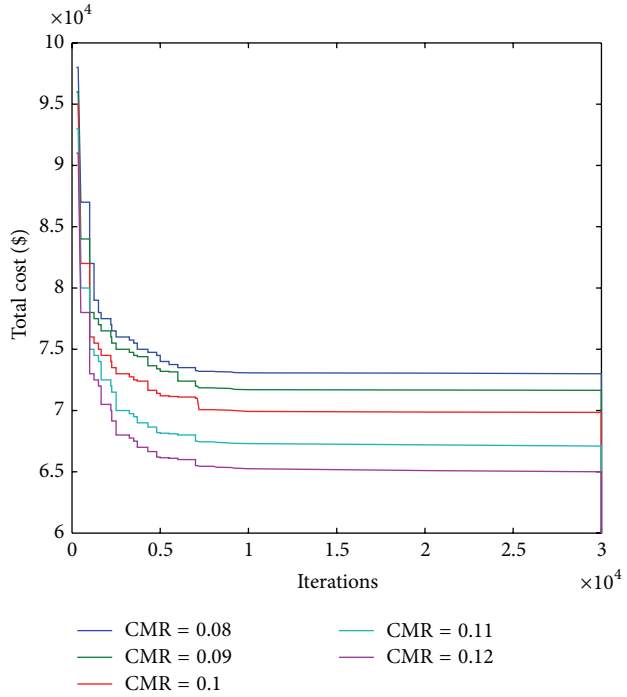


Figure 7: Variation of the optimum total cost using various techniques.

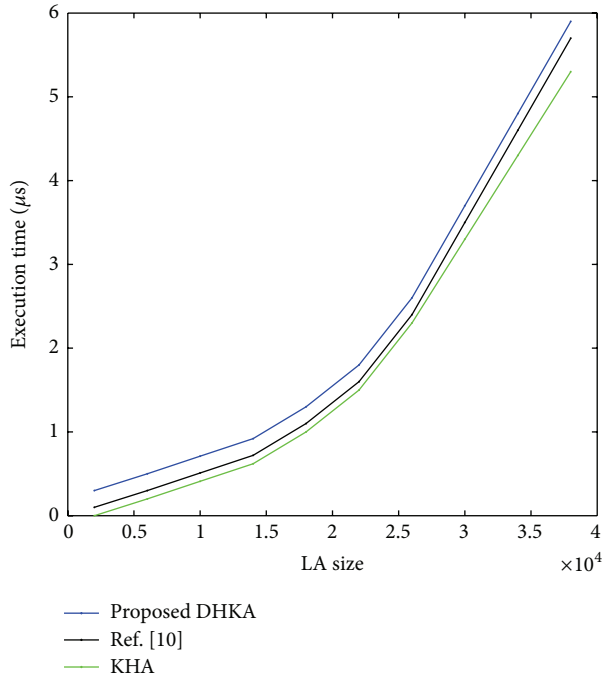


Figure 8: Simulation time of the proposed techniques with respect to LA size (Dummy).

the computation time taken by each algorithm to converge by varying population size and network as well.

Figures 8 and 9 show that, even as the network size grows, the proposed KHA based algorithm tends to converge within tolerable run time, whereas DHKA takes much more time to converge. DHKA basically has twice the size of the herd

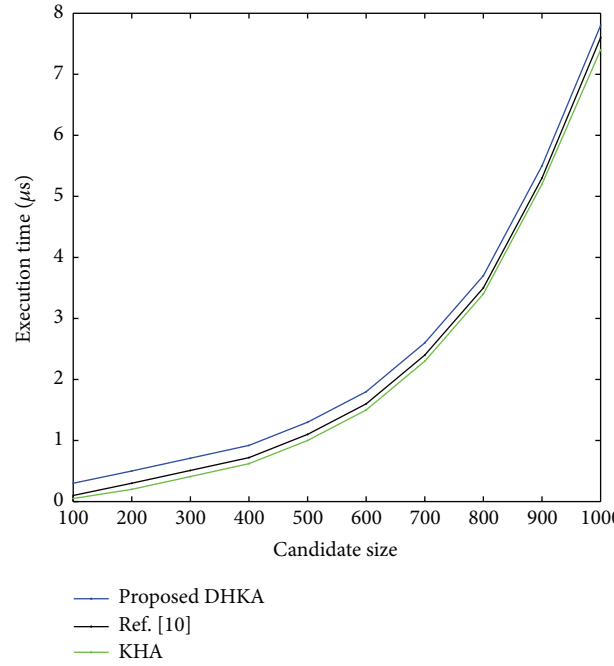


Figure 9: Simulation time of the proposed techniques with respect to candidate size (Dummy).

compared to KHA and it continues to search for the best member scanning through generations. The population size and the maximum number of generations conventionally assume large values for DHKA to work properly. On average, KHA based algorithm takes about 0.05 μsec whereas DHKA takes around 4.9 μsec. Despite this fact, DHKA is a strong contender as a combinatorial optimization tool because of its robustness.

5. Conclusion

This paper discusses the problem of ever increasing concern for reduction of signalling loads generated in future mobile wireless cellular network (MWCN) due to paging and LU. In this regard, the implication of proper planning for location area (LA) towards optimality is demonstrated. The optimal LA problem was overviewed formally and mathematically formulated considering various practical constraints. Based on various mobility patterns of users and network architecture, the design of the LR area is formulated as a combinatorial optimization problem. The nondifferentiable nature of the problem of interest in this paper has created room for some efficient, robust, and nontraditional search algorithms. This research has proposed a double krill herd algorithm wherein the krill herd is divided based on their global search and local search mechanisms derived from DE. Numerical simulation witnesses promising results, with the condition of widely varying input parameters, some of which are documented. Thus, the proposed DHKA appeared to be a strong alternative for a complex, hard NP-complete optimization problem, such as LAP.

Conflict of Interests

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

References

[1] K.-Y. Chung, J. Yoo, and K. J. Kim, "Recent trends on mobile computing and future networks," *Personal and Ubiquitous Computing*, vol. 18, no. 3, pp. 489–491, 2014.

[2] I. Demirkol, C. Ersoy, M. U. Caglayan, and H. Delic, "Location area planning in cellular networks using simulated annealing," in *Proceedings of the 20th Annual Joint Conference on the IEEE Computer and Communications Societies (INFOCOM '01)*, vol. 1, pp. 13–20, April 2001.

[3] A. Bar-Noy, P. Cheilaris, Y. Feng, and M. J. Golin, "Paging mobile users in cellular networks: optimality versus complexity and simplicity," *Theoretical Computer Science*, vol. 470, pp. 23–35, 2013.

[4] S. K. Sen, A. Bhattacharya, and S. K. Das, "A selective location update strategy for PCS users," *Wireless Networks*, vol. 5, no. 5, pp. 313–326, 1999.

[5] D. Feng, C. Jiang, G. Lim, L. J. Cimini Jr., G. Feng, and G. Y. Li, "A survey of energy-efficient wireless communications," *IEEE Communications Surveys and Tutorials*, vol. 15, no. 1, pp. 167–178, 2013.

[6] G. P. Pollini and I. Chih-Lin, "A profile-based location strategy and its performance," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 8, pp. 1415–1423, 1997.

[7] J. S. Ho and I. F. Akyildiz, "Mobile user location update and paging under delay constraints," *Wireless Networks*, vol. 1, no. 4, pp. 413–425, 1995.

[8] A. Bhattacharya and S. K. Das, "LeZi-update: an information-theoretic approach to track mobile users in PCS networks," in *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '99)*, pp. 1–12, Seattle, DC, USA, August 1999.

[9] I. F. Akyildiz, J. S. M. Ho, and Y.-B. Lin, "Movement-based location update and selective paging for PCS networks," *IEEE/ACM Transactions on Networking*, vol. 4, no. 4, pp. 629–638, 1996.

[10] M. Maitra, R. K. Pradhan, D. Saha, and A. Mukherjee, "Optimal location area planning for mobile cellular network using evolutionary computing methods," *IETE Journal of Research*, vol. 51, no. 3, pp. 235–244, 2005.

[11] A. H. Gandomi and A. H. Alavi, "Krill herd: a new bio-inspired optimization algorithm," *Communications in Nonlinear Science and Numerical Simulation*, vol. 17, no. 12, pp. 4831–4845, 2012.

[12] K. C. Tan, T. H. Lee, and E. F. Khor, "Evolutionary algorithms with dynamic population size and local exploration for multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 5, no. 6, pp. 565–588, 2001.

[13] W.-F. Leong and G. G. Yen, "PSO-based multiobjective optimization with dynamic population size and adaptive local archives," *IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics*, vol. 38, no. 5, pp. 1270–1293, 2008.

[14] J. Arabas, Z. Michalewicz, and J. Mulawka, "GAVaPS-a genetic algorithm with varying population size," in *Proceedings of the IEEE World Congress on Computational Intelligence: Evolutionary Computation*, pp. 73–78, 1994.

[15] J.-H. Zhong, M. Shen, J. Zhang, H. S.-H. Chung, Y.-H. Shi, and Y. Li, "A differential evolution algorithm with dual populations for solving periodic railway timetable scheduling problem," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 4, pp. 512–527, 2013.

[16] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 398–417, 2009.

[17] C. Rose and R. Yates, "Minimizing the average cost of paging under delay constraints," *Wireless Networks*, vol. 1, no. 2, pp. 211–219, 1995.

[18] W. Wang, I. F. Akyildiz, G. L. Stüber, and B.-Y. Chung, "Effective paging schemes with delay bounds as QoS constraints in wireless systems," *Wireless Networks*, vol. 7, no. 5, pp. 455–466, 2001.

[19] S.-S. Kim, J.-H. Byeon, J. Taheri, and H. Liu, "Swarm intelligent approaches for location area planning," *Journal of Multiple-Valued Logic and Soft Computing*, vol. 22, no. 3, pp. 287–306, 2014.

[20] B. Krishnamachari, R.-H. Gau, S. B. Wicker, and Z. J. Haas, "Optimal sequential paging in cellular wireless networks," *Wireless Networks*, vol. 10, no. 2, pp. 121–131, 2004.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 840305, 19 pages
http://dx.doi.org/10.1155/2014/840305

Research Article

A Partition-Based Active Contour Model Incorporating Local Information for Image Segmentation

Jiao Shi,[1] Jiaji Wu,[1] Anand Paul,[2] Licheng Jiao,[1] and Maoguo Gong[1]

[1] *Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, Institute of Intelligent Information Processing, Xidian University, Xi'an, Shaanxi 710071, China*
[2] *School of Computer Science Engineering, Kyungpook National University, Daegu 702-701, Republic of Korea*

Correspondence should be addressed to Anand Paul; paul.editor@gmail.com

Received 3 March 2014; Revised 2 June 2014; Accepted 5 June 2014; Published 24 July 2014

Academic Editor: S.N. Deepa

Active contour models are always designed on the assumption that images are approximated by regions with piecewise-constant intensities. This assumption, however, cannot be satisfied when describing intensity inhomogeneous images which frequently occur in real world images and induced considerable difficulties in image segmentation. A milder assumption that the image is statistically homogeneous within different local regions may better suit real world images. By taking local image information into consideration, an enhanced active contour model is proposed to overcome difficulties caused by intensity inhomogeneity. In addition, according to curve evolution theory, only the region near contour boundaries is supposed to be evolved in each iteration. We try to detect the regions near contour boundaries adaptively for satisfying the requirement of curve evolution theory. In the proposed method, pixels within a selected region near contour boundaries have the opportunity to be updated in each iteration, which enables the contour to be evolved gradually. Experimental results on synthetic and real world images demonstrate the advantages of the proposed model when dealing with intensity inhomogeneity images.

1. Introduction

Image segmentation is one of the fundamental tasks in the fields of computer vision and image processing. It has been successfully applied to a variety of realistic problems, such as medical imaging, object recognition, and synthetic aperture radar (SRA) image understanding [1–4]. Currently, various image segmentation techniques have been proposed [5–8], such as histogram thresholding, clustering [7], active contour models [8], and graph-cut methods [5]. However, due to the complexity of several images, designing a robust and efficient segmentation method is still a common goal and challenge for researchers.

Since the introduction of snakes, active contour models (ACMs) [8] have been applied to many fields, in which image segmentation is one of the most important applications [9–14]. The core part of ACMs for image segmentation is that a curve which evolves subject to image characteristics is employed, and then the desired object can be extracted by optimizing an energy function. Nevertheless, the performance of ACMs which use energy functions on the basis of edge information is inadequate, as only objects with edges defined by gradient can be detected [15–18].

Enhanced techniques have been proposed to overcome the limitations of traditional ACMs, especially on designing complex region-based energy functions. Region-based models [19–25] utilize not only the image information near the evolving contour, but also the image statistics information deriving from both sides of the contour. They are less sensitive to noise and more likely to detect weak boundaries when compared to edge-based models. In the Mumford-Shah model [20], an image is decomposed into some regions, and then each region is approximated by a smooth function. However, it is difficult to minimize the function which is

not convex in generality. The Chan-Vese (CV) [9] model was proposed based on a region-based energy function inspired by a simplified Mumford-Shah function. The CV model can detect the object with boundaries not necessarily defined by gradient. However, the computational cost of the CV model is rather expensive due to the complicated procedures involved, which limits its application. After the CV model, a broad range of variations have been proposed for reducing computational cost [21].

Traditional ACMs are called piecewise-constant models, since they are designed on the assumption that image intensities are statistically homogeneous of each region over the whole image. However, the assumption is so strict for real world images which are not always statistically homogeneous. In fact, intensity inhomogeneity is frequently observed in real world images and challenging in image segmentation. Due to technical limitations or artifacts introduced by the object being imaged, intensity inhomogeneities often occur in real world images from different modalities and may cause considerable difficulties in image segmentation. For example, intensity inhomogeneity in magnetic resonance (MR) images arises from the nonuniform magnetic fields produced by radio-frequency coils as well as from variations in object susceptibility. It often appears as the variation of intensities from the same tissue type over the locations in an image. In addition, since aurorae are often imaged at night, intensity inhomogeneity in aurora images is usually due to technical limitations of sensor. Segmenting such MR and UVI images has been a challenge. Without an effective preprocessing step such as intensity inhomogeneity correction or histogram equalization, segmentation is difficult to implement.

To deal with intensity inhomogeneity, an effective way is taking local information of the segmented image into account. Because of using local region information, the distance regularized level set evolution (DRLSE) model can cope with intensity inhomogeneity [14]. In addition, some related methods were proposed in [25–27] which have similar capability of handling intensity inhomogeneity as the DRLSE model. However, these local information based methods are sensitive to initial condition to some extent, which holds back their practical applications.

Fuzzy logic which has the ability to flexible process information is widely studied and successfully applied in many real applications [28–30]. The fuzzy logic provides a balanced technique with a strong ability to reject "weak" local minima. Fuzzy energy-based active contour model (FAC) was proposed by Krinidis and Chatzis [31], which first combines fuzzy logic with the active contour methodology. Fuzzy active contour models have a strong ability to reject local minima and can detect objects with smooth or discontinuous boundaries. Besides, to reduce the computational cost, the associated Euler-Lagrange equations were replaced by pseudo-level set functions. Thus, fuzzy active contour models have the properties of less sensitivity to initial conditions and high computation speed. Some related methods which have similar capability of rejecting local minima as the FAC model were proposed in [26, 28, 31].

Although fuzzy active contour models have made great progress, they still have some disadvantages. (1) Traditional fuzzy active contour models fail to process images with intensity inhomogeneity by using distinct means of pixel intensities, one representing the objects region and the other representing the background. (2) The real intention of ACMs is to evolve the region near contour boundaries, which cannot be achieved by the traditional fuzzy active contour models. In fuzzy active contour models, the membership degrees of all the pixels are updated at each iteration, which may cause incorrect results, especially when the contour is not continuous enough [28]. To deal with the difficulties caused by intensity inhomogeneity, a partition-based fuzzy active contour model is proposed for image segmentation. In particular, the following techniques are introduced. (1) By introducing image local characteristics, distinct means of pixel intensities are replaced by spatially varying ones which is better suited for images with intensity inhomogeneity. (2) Regions near contour boundaries are detected with the help of shadowed sets. Only the pixel within the selected region has the opportunity to be updated, which enables the contour to be evolved gradually. The main advantages of the proposed model can be concluded as (1) computational simplicity (the calculation of each step only includes the computation of pixels within the region near the contour boundary); (2) flexibility (it has less sensitivity to initial conditions with the help of fuzzy technique); (3) being suitable for images with intensity inhomogeneity by considering image local characteristics.

The rest of this paper is organized as follows. In Section 2, the main ideas of the proposed model and our motivation will be introduced. Section 3 will describe the proposed model in detail. In Section 4, experimental results on synthetic images, medical images, and natural images will be described. Conclusions will be drawn in Section 5.

2. Motivation

The basic idea of traditional ACMs is to look for the partition of a given image I into two regions which have distinct means of pixel intensities, one representing the objects region and the other representing the background. First, a random initial partition of the image is provided. This initial partition defines a curve C that will be iteratively evolved according to the image characteristics in the image domain Ω by a minimization process. The evolving curve C defines a boundary of the segmentation region. The object to be detected is represented by the region inside C, while the background is represented by the region outside C. In ACMs, the segmentation process is defined as the minimization of the distances between pixels and cluster prototypes, taking into account the length term as a regularization term.

The segmentation result of traditional ACMs is highly dependent on initial conditions. Soft computing technique which has the ability to flexibly process information has been successfully applied in many real applications [28–30], but not in active contour methods. Fuzzy energy-based active contour model (FAC) was proposed by Krinidis and Chatzis [31], which first combines fuzzy logic with the active contour methodology. In the FAC model, the segmentation process

is defined as the minimization of a fuzzy energy function [26, 28, 31]:

$$\begin{aligned} F(C, c_1, c_2, u) &= \mu \cdot \text{Length}(C) + \lambda_1 \int_\Omega [u(x,y)]^m \\ &\quad \times |I(x,y) - c_1|^2 dx\, dy + \lambda_2 \int_\Omega [1 - u(x,y)]^m \\ &\quad \times |I(x,y) - c_2|^2 dx\, dy, \end{aligned} \tag{1}$$

where c_1 and c_2 are the average intensities of regions inside and outside of the evolving curve C, respectively, (x, y) is the spatial coordinate of a pixel, $u(x, y)$ is the membership degree of the pixel (x, y) belonging to the inside of C, $m \geq 1$ is the fuzzy coefficient, $I(x, y)$ is the gray value of the pixel (x, y), and λ_1, $\lambda_2 > 0$ and $\mu \geq 0$ are fixed parameters. The first term in (1) is the length term, which accounts for smoothing the curve. μ is used to control the effect of the length term. According to [26, 28, 31], the length term is not important for a clean image. For simplicity, without losing the generality, the length term has not been considered during minimizing the fuzzy energy function. λ_1 and λ_2 are used to control the weights of the distances between pixels and average prototypes of the image regions inside and outside C, respectively. λ_1 and λ_2 are generally set to 1 for balancing the weights of two regions in (1). c_1 and c_2 are defined as follows:

$$\begin{aligned} c_1 &= \frac{\int_\Omega [u(x,y)]^m I(x,y)\, dx\, dy}{\int_\Omega [u(x,y)]^m dx\, dy}, \\ c_2 &= \frac{\int_\Omega [1 - u(x,y)]^m I(x,y)\, dx\, dy}{\int_\Omega [1 - u(x,y)]^m dx\, dy}. \end{aligned} \tag{2}$$

Keeping c_1 and c_2 fixed and minimizing (1), the membership degree of each pixel is computed as follows:

$$u(x,y) = \frac{1}{1 + \left(\lambda_1 (I(x,y) - c_1)^2 / \lambda_2 (I(x,y) - c_2)^2\right)^{1/(m-1)}}. \tag{3}$$

Fuzzy active contour models have a strong ability to reject local minima and can detect objects with smooth or discontinuous boundaries. Besides, to reduce the computational cost, the associated Euler-Lagrange equations were replaced by pseudo-level set functions. Although fuzzy active contour models have made great progress, they still have some difficulties in processing images with intensity inhomogeneous, which will be analyzed as follows.

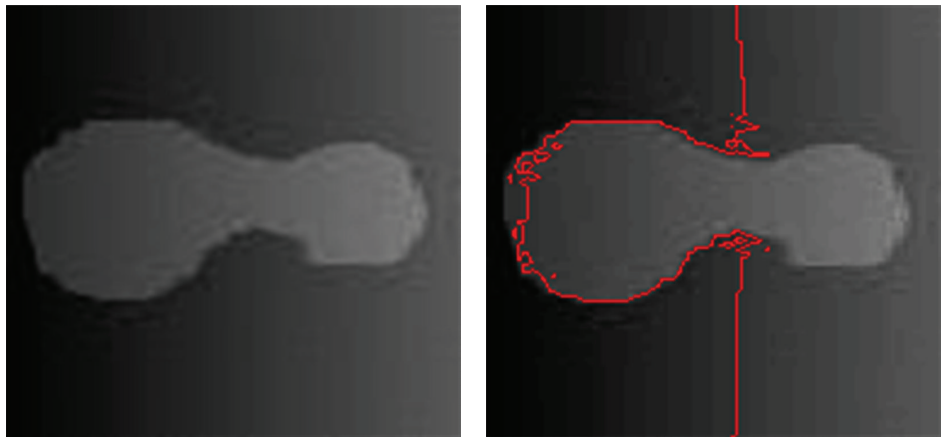
2.1. Motivation of Introducing Local Region Information. ACMs are called piecewise-constant models, since they are designed on the condition that images are approximated by regions with piecewise-constant intensities. It can be seen from (2) that the prototypes are constant, representing the average intensities of pixels within regions inside and outside of the curve, respectively. However, regions are not always statistically homogeneous in real world images. It is inappropriate to employ constant prototypes for describing the image where the intensity distributions overlap between the object region and the background.

Figure 1 shows the segmentation result of the FAC model on a synthetic image with intensity variation. The segmentation result shows that certain parts of the object with weak intensities are submerged in the background. It is impossible for the FAC model to obtain a satisfactorily result on the synthetic image with intensity variation. Figure 2(b) depicts a 10 × 10 window shown in a red square extracted from Figure 2(a). The intensity of pixels within this window ranges from 60 to 112. The average intensities of regions inside and outside of the curve obtained by the FAC model are 74.96 and 24.89, respectively. Since the intensities of pixels within this window are more similar to those in the region inside the curve, all the pixels within this window are partitioned into the object region, which will incorrectly partition some pixels within the background into the object region.

Naturally, each pixel has connections with its neighbors to some extent and is reasonable to be approximately described by taking its local immediate neighborhood into consideration. Generally, the local information is an important feature to describe the relationship of pixels within a local region. For a point x, its intensity can be approximated by a weighted average of the pixel intensity $I(y)$ where y is the neighborhood of x. By introducing image local characteristics, a mild assumption that the image is approximated by regions with piecewise-constant intensities within a local region is more suitable for real world images. In this study, we try to incorporate local region information into the original piecewise-constant models for processing the image with intensity inhomogeneity. The proposed technique is described in Section 3.1.

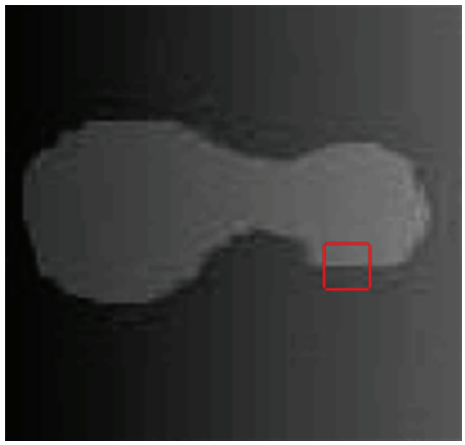
2.2. Motivation of Detecting the Contour Boundary. In fuzzy active contour models, the membership degree measures the degree of a pixel belonging to the object region. The range of the membership degree is [0, 1]. For a binary classification problem, pixels with membership degrees equal to 0.5 form the contour boundary. If the membership degree of a pixel is larger than 0.5, the pixel belongs to the object region. If the membership degree of a pixel is smaller than 0.5, the pixel belongs to the background region. A large uncertainty may exist when assigning pixels with membership degrees near 0.5, because they have nearly the same degrees of belonging to the object region and the background. Thus, it is natural to make great effort to process pixels with large uncertainties within certain selected region of interest.

In addition, ACMs use a high order function to define the contour boundary and originally intend to evolve the region near the contour boundary. However, in traditional fuzzy active contour models, membership degrees of pixels are updated over the whole image domain in each iteration. These approaches fail to only evolve the regions near contour boundaries. Therefore, it is necessary to design a narrow band strategy for making the contour evolve gradually.



(a) The synthetic image (b) The FAC model result

Figure 1: The segmentation result of the FAC model on a synthetic image.



102	103	104	105	107	108	109	110	109	109
101	102	103	104	106	108	109	109	110	109
100	100	102	103	105	107	108	109	110	109
99	99	101	102	104	106	107	108	110	110
96	97	99	102	104	107	109	110	112	112
111	111	110	110	110	109	109	109	103	104
62	62	63	63	64	64	64	65	72	74
64	65	66	67	69	71	72	72	67	67
64	65	65	65	65	65	65	65	73	72
60	60	62	63	65	67	68	69	65	67

(a) The window obtained from the image (b) The intensity value of pixels within the window

Figure 2: A 10 × 10 window obtained from a synthetic image marked with a red rectangle.

To confine the update area close to contour boundaries, it is natural to update the pixel with the membership degree of 0.5. However, it is not easy to accurately define contour boundaries by detecting the pixel with the membership degree of 0.5, because fuzzy energy will create a region with low pixel-density close to the membership degree of 0.5 [28]. An improved model that uses mathematical morphology to detect regions near contour boundaries is stated as follows:

$$u'(x,y) = \frac{\beta\left(H\left(u(x,y) - 0.5\right)\right) \oplus R}{1 + \left(\lambda_1 \left(I(x,y) - c_1\right)^2 / \lambda_2 \left(I(x,y) - c_2\right)^2\right)^{1/(m-1)}}, \tag{4}$$

where $H(\cdot)$ is the Heaviside function, $\beta(A) = A - (A \ominus B)$ is a morphological boundary extraction operation that uses a structural element B to cause erosion from the image A, and R is another structuring element for expanding the boundary to its neighborhood. The structuring elements B and R are a 3×3 square and a circle with radius 5, respectively [28]. The selection of structuring elements has a great impact on final segmentation results and requires a careful consideration in practical applications. It is difficult to select a suitable structuring element; different images usually require different structuring elements. According to what has been mentioned above, it is necessary to design a partition-based strategy for making the contour evolve gradually. The designed technique which adaptively detects the region near

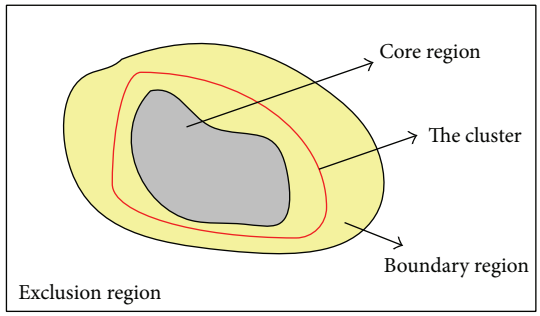


Figure 3: Three levels of belongingness with respect to a cluster.

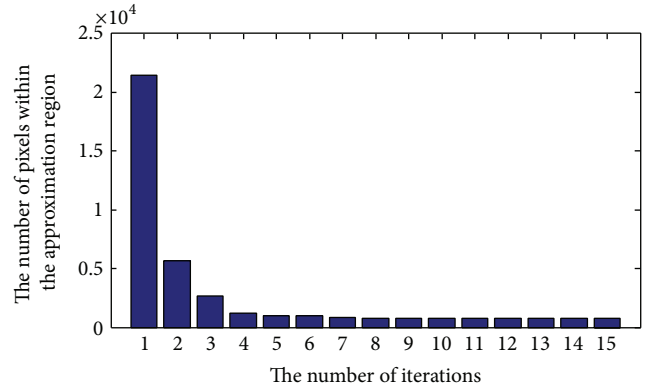


Figure 4: The number of pixels within approximation regions near the contour boundaries changes over time.

the contour boundary based on the intrinsic structure of the image is described in Section 3.2.

3. Methodology

In this section, an enhanced fuzzy active contour model that draws upon intensity information in local regions is proposed. In particular, the following techniques are designed. (1) The constant prototypes are replaced by the spatially varying ones for overcoming the difficulties caused by intensity inhomogeneities. (2) The region near contour boundaries is automatically detected and only the pixels within the selected region are updated, which enables the contour to evolve gradually. The details are described as follows.

3.1. Spatial Varying Prototypes. Traditional ACMs are designed on the assumption that the image is approximated by regions with piecewise-constant intensities over the whole image. However, the assumption is so strict for real world images which are not always statistically homogeneous. A milder assumption that the image is statistically homogeneous within a small local region may better suit real world images. Thus, it is more reasonable to utilize local region information for approximately describing the intensity of a pixel by the weighted average of its neighbors. With the incorporation of local region information, we replace the constants c_1 and c_2 by the spatial varying ones which are computed as follows:

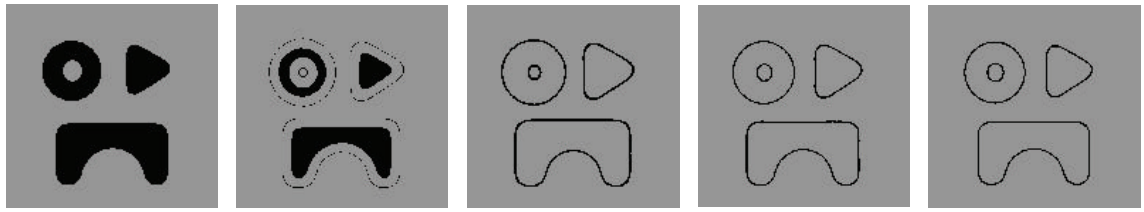
$$c_1(x,y) = \frac{\sum_\Omega [u(i,j)]^m g_k(i,j) I(i,j)}{\sum_\Omega [u(i,j)]^m g_k(i,j)},$$
$$c_2(x,y) = \frac{\sum_\Omega [1-u(i,j)]^m g_k(i,j) I(i,j)}{\sum_\Omega [1-u(i,j)]^m g_k(i,j)}, \quad (5)$$

where (x, y) is the spatial coordinate of the current pixel, (i, j) is the pixel falling into the local region around the current pixel, g_k is a kernel function which is taken as the weight coefficient assigned to a pixel within the local region around the current pixel ($g_k(i,j) = (1/\sqrt{2\pi}\sigma)e^{-d^2/2\sigma^2}$ with a standard deviation $\sigma > 0$, and $d = ((x-i)^2 + (y-j)^2)^{1/2}$ is the spatial distance between the pixel (i, j) and the current pixel (x, y)). We assume that one pixel is one unit length.

The kernel function is highly useful in clustering analysis and provides a robust property based on influence function analysis [32–35]. It generates larger weight coefficients to pixels closer to the current pixel and smaller weight coefficients to pixels far away from the current pixel. If the spatial distance between pixels (i, j) and (x, y) is larger than a level (i.e., (i, j) keeps away from (x, y)) [35], the weight coefficient will approach its minimum value. That means the contribution of $I(i, j)$ to $c_1(x, y)$ and $c_2(x, y)$ approaches zero as the pixel (i, j) lies far away from the current pixel. Due to the localization property of the kernel function, the intensity averages of local regions inside and outside of the curve are dominated by the intensities of pixels within the neighborhood of the current pixel [18]. The kernel function makes the influence of pixels within the local region on the current pixel change flexibly according to their spatial distances from the current pixel, allowing more image local characteristics to be obtained. With the incorporation of local region information, the assumption that the image is statistically homogeneous within different small local regions is more appropriate for the characteristics of images with intensity inhomogeneity.

3.2. Detecting the Approximation Region Near the Contour Boundary. With the introduction of spatial varying prototypes, the computational cost increases due to the computation of $c_1(x, y)$ and $c_2(x, y)$ for each pixel. It is crucial to make great effort to process pixels with a large uncertainty, thus restricting the computational cost to certain selected region of interest. In addition, since pixels with membership degrees near 0.5 form the region near contour boundaries, making great effort to such selected region of interest is benefit to evolve the contour boundary gradually. Thus, we try to confine the update area near the current contour boundary at each iteration, which enables the curve to evolve gradually and saves the computational resource as well.

The concept of shadowed sets was developed to improve the observation and the interpretability of fuzzy sets [36–39]. Shadowed sets help to restrict uncertainty of membership degrees within the whole fuzzy set to certain selected regions



(a) The variation of pixels within the approximation regions at the first five iterations

(b) The evolution of the contour boundary at the first five iterations

Figure 5: The evolution of the approximation region and the associated contour boundary with iteration.

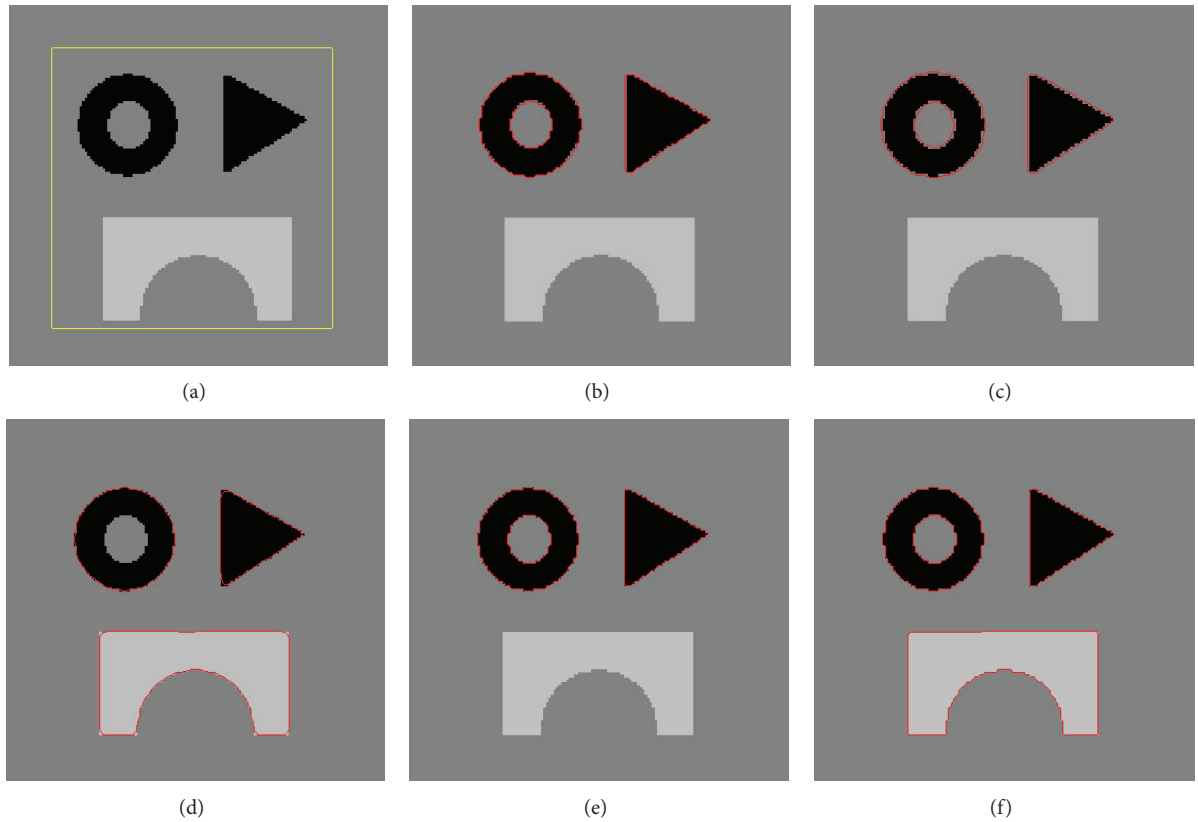
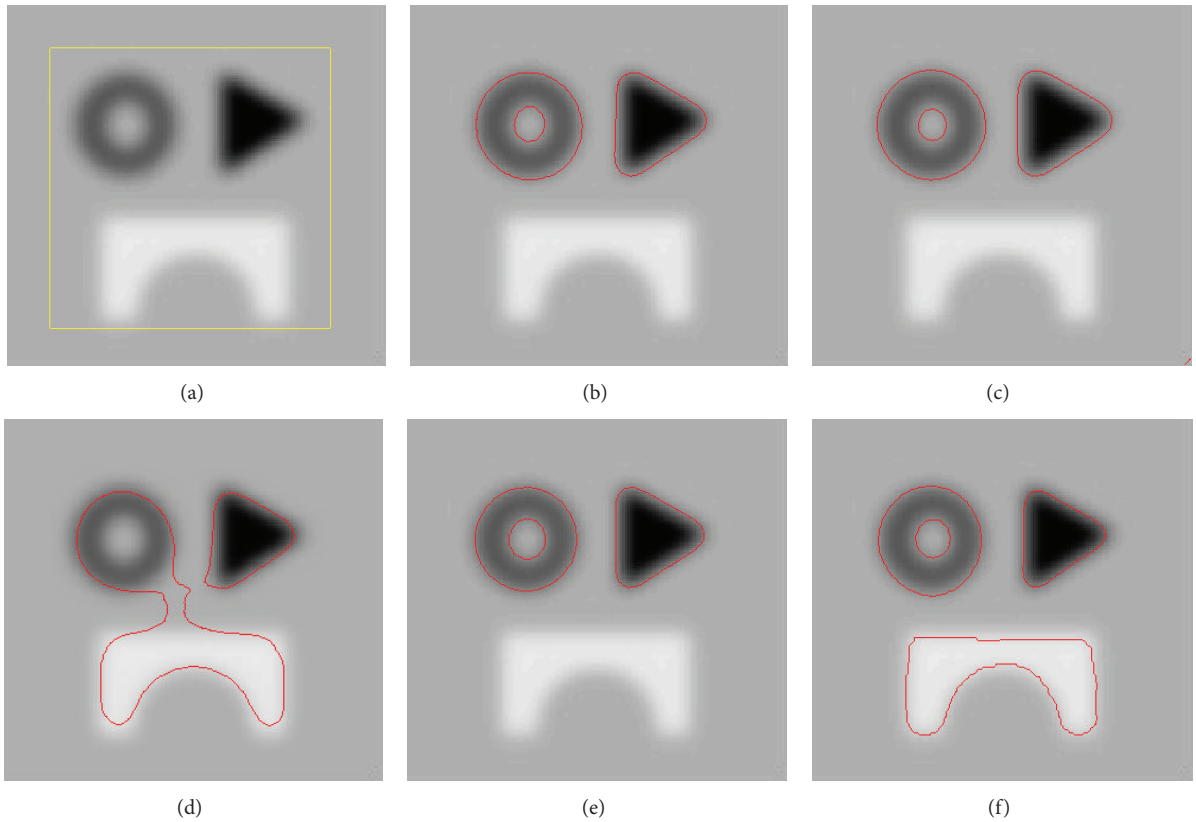


Figure 6: Segmentation results on a synthetic image: (a) the initial contour, (b) the CV model result, (c) the GACV model result, (d) the DRLSE model result, (e) the FAC model result, and (f) the LFAC model result.

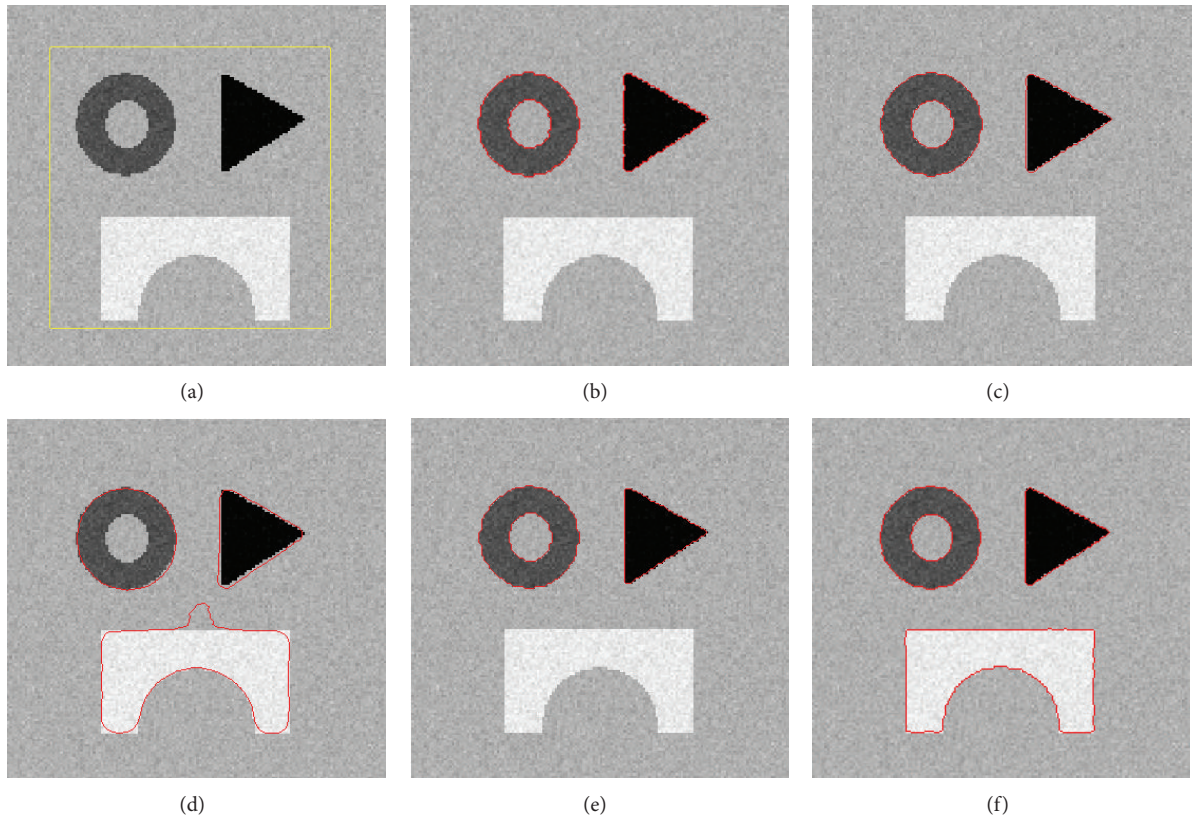
to denote patterns with large extent of vagueness and to elevate or reduce membership degrees in other regions. Shadowed sets partition the distribution of a target set into three regions: the core, boundary, and exclusion regions, as shown in Figure 3. The red curve describes the real contour of the cluster. Patterns within the core region belong to the cluster, those within the exclusion region do not belong to the cluster, and those within the boundary region possibly belong to the cluster and come with a certain component of uncertainty. According to shadowed sets theory, great effort will be made to process patterns with large extent of vagueness within the boundary region.



(a) (b) (c)

(d) (e) (f)

Figure 7: Segmentation results on a synthetic image: (a) the initial contour, (b) the CV model result, (c) the GACV model result, (d) the DRLSE model result, (e) the FAC model result, and (f) the LFAC model result.



(a) (b) (c)

(d) (e) (f)

Figure 8: Segmentation results on the synthetic image corrupted by Gaussian noise: (a) the initial contour, (b) the CV model result, (c) the GACV model result, (d) the DRLSE model result, (e) the FAC model result, and (f) the LFAC model result.

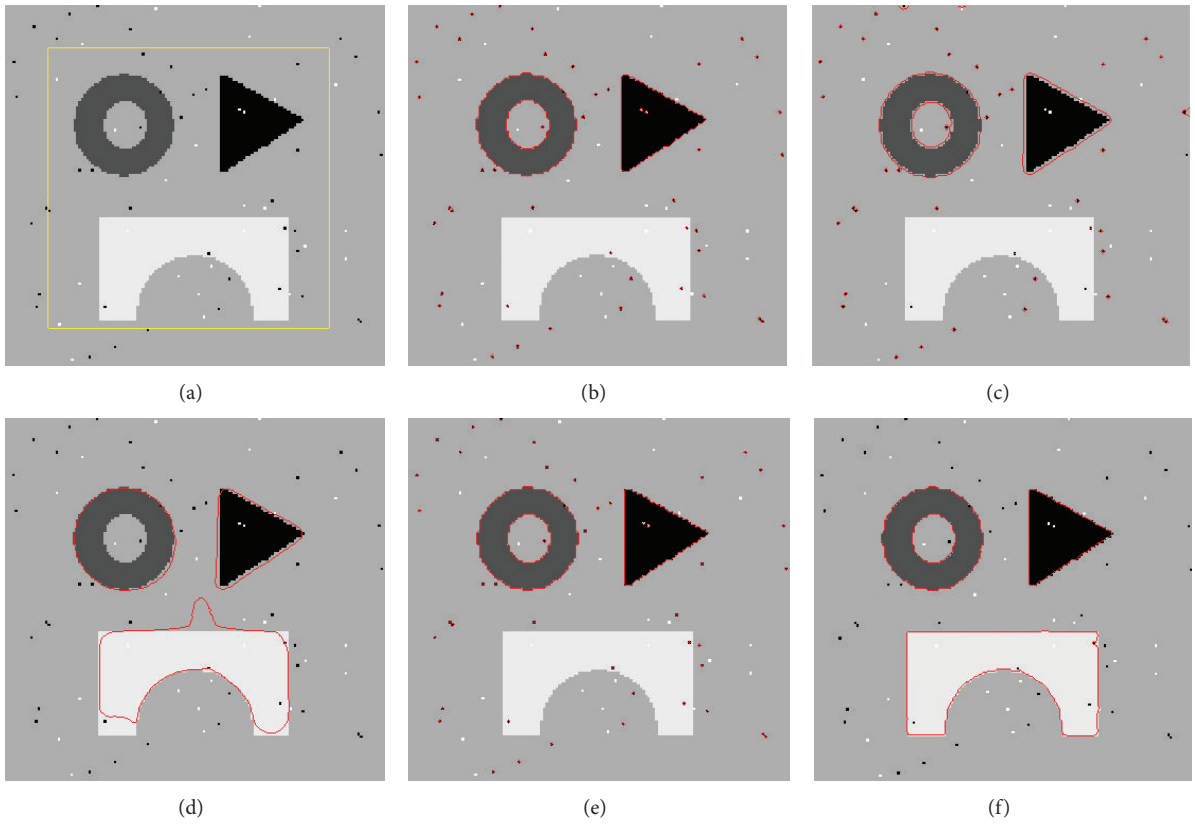


Figure 9: Segmentation results on the synthetic image corrupted by salt and pepper noise: (a) the initial contour, (b) the CV model result, (c) the GACV model result, (d) the DRLSE model result, (e) the FAC model result, and (f) the LFAC model result.

A specific threshold is needed for partitioning the distribution of fuzzy set into three regions. The threshold of object region is computed by minimizing the following function [39]:

$$V(\alpha) = \left|\psi_1 + \psi_2 + \psi_3\right|,$$

$$\psi_1 = \sum_{u(x,y)\le\alpha} u(x,y),$$

$$\psi_2 = \sum_{u(x,y)\ge(U_{\max}-\alpha)} \left(U_{\max} - u(x,y)\right),$$

$$\psi_3 = \operatorname{card}(\Delta), \tag{6}$$

where $\Delta = \{(x,y) \mid \alpha < u(x,y) < (U_{\max} - \alpha)\}$, $\operatorname{card}(\Delta)$ is the number of pixels in Δ, $u(x,y)$ denotes the membership degree of pixel (x,y) belonging to the object region, $U_{\max}$ and $U_{\min}$ are the largest and the smallest membership degrees of all the pixels belonging to the object region, respectively, ψ_1 denotes the reduction of membership degrees, ψ_2 means the elevation of membership degrees, and ψ_3 represents the region with the greatest uncertainty. The three terms on the right side of (6) correspond to the three regions shown in Figure 3 [39]. The optimal threshold can be obtained by satisfying the requirement $\alpha_{\text{opt}} = \arg\min_{\alpha} V(\alpha)$. The values of α are suggested in the range of $[U_{\min}, (U_{\max} + U_{\min})/2]$ with an interval of 0.001 [39].

The membership degrees of pixels belonging to a certain region can be considered as a fuzzy set. After obtaining the thresholds of the object region and the background, the approximate region near the contour boundary can be detected according to the following functions:

$$R_a = S_1 \cup S_2,$$

$$S_1 = \left\{(x,y) \mid \alpha_{1\text{opt}} < u(x,y) < \left(\max\left(u(x,y)\right) - \alpha_{1\text{opt}}\right)\right\},$$

$$S_2 = \left\{(x,y) \mid \alpha_{2\text{opt}} < (1 - u(x,y)) < \left(\max\left(1 - u(x,y)\right) - \alpha_{2\text{opt}}\right)\right\}, \tag{7}$$

where $\alpha_{1\text{opt}}$ and $\alpha_{2\text{opt}}$ are the adaptive thresholds of the object region and background and are updated in each iteration and S_1 and S_2 are boundary regions inside and outside of the evolving curve, respectively. R_a is the approximation region near the contour boundary, which is adaptively detected with the help of shadowed sets theory.

To demonstrate the effectiveness of detecting the approximation region near the contour boundary, a synthetic image with size 143×150 is employed. The variation in the number of pixels within the approximation region is depicted in Figure 4. It can be seen that the number of pixels within the

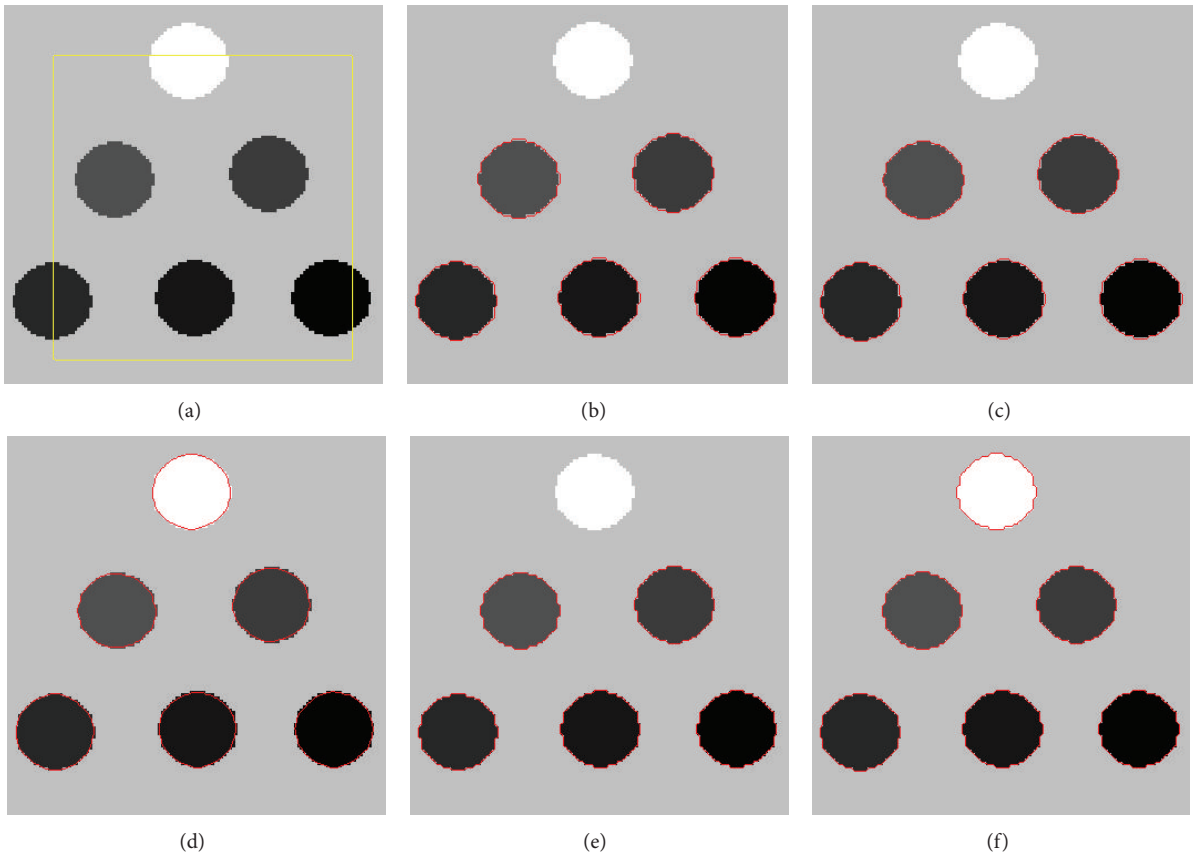


Figure 10: Segmentation results on the synthetic image: (a) the initial contour, (b) CV result, (c) GACV result, (d) DRLSE result, (e) FAC result, and (f) LFAC result.

approximate region becomes smaller with time increasing. Figure 5(a) shows the variation of the approximate region, in which pixels within the approximation region are shown in black and other pixels are shown in gray. The evolution of the corresponding contour boundary (red curve) is depicted in Figure 5(b). It can be found that pixels within the region near the contour boundary are effectively identified, and the approximate region can more accurately approach to the real contour of objects with time increasing.

3.3. General Framework of the Proposed Model. We present a partition-based fuzzy active contour model with incorporating local information, termed as LFAC for short. The proposed model designs an enhanced fuzzy energy function which is written in the summation sign form for indicating the discrete nature of the image data. The fuzzy energy function is introduced as follows:

$$F = \sum_{\Omega} \left(u\left(x,y\right)\right)^{m} \left(I\left(x,y\right) - c_1\left(x,y\right)\right)^2 + \sum_{\Omega} \left(1 - u\left(x,y\right)\right)^{m} \left(I\left(x,y\right) - c_2\left(x,y\right)\right)^2, \tag{8}$$

where $c_1(x, y)$ and $c_2(x, y)$ are the spatial varying prototypes of the pixel (x, y), Ω denotes the image domain, $u(x, y)$ is the membership degree of the pixel (x, y) belonging to the object region, $I(x, y)$ is the gray value of the pixel (x, y), and $m \geq 1$ is the fuzzy coefficient. The main steps of the proposed model are presented as follows.

Step 1. Set fuzzy coefficient m and the stopping condition $\varepsilon = 10^{-2}$.

Step 2. Provide an initial partition of the image; set $u > 0.5$ for the object region and $u < 0.5$ for the background.

Step 3. Detect the approximation region close to contour boundaries according to (7).

Step 4. Compute the spatial varying prototypes for each pixel within the approximation region according to (5).

Step 5. Assume that the membership degree of the current pixel is u_o. Compute the new membership degree u_n for the pixel according to (3). Then, compute the difference between the old and the new energy ΔF which is derived in the Appendix:

$$\Delta F = \left(u_n^m \left(\frac{s_1}{s_1 + u_n^m - u_o^m} \right)^2 - u_o^m \right) \left(I_o - c_1\left(x,y\right)\right)^2$$

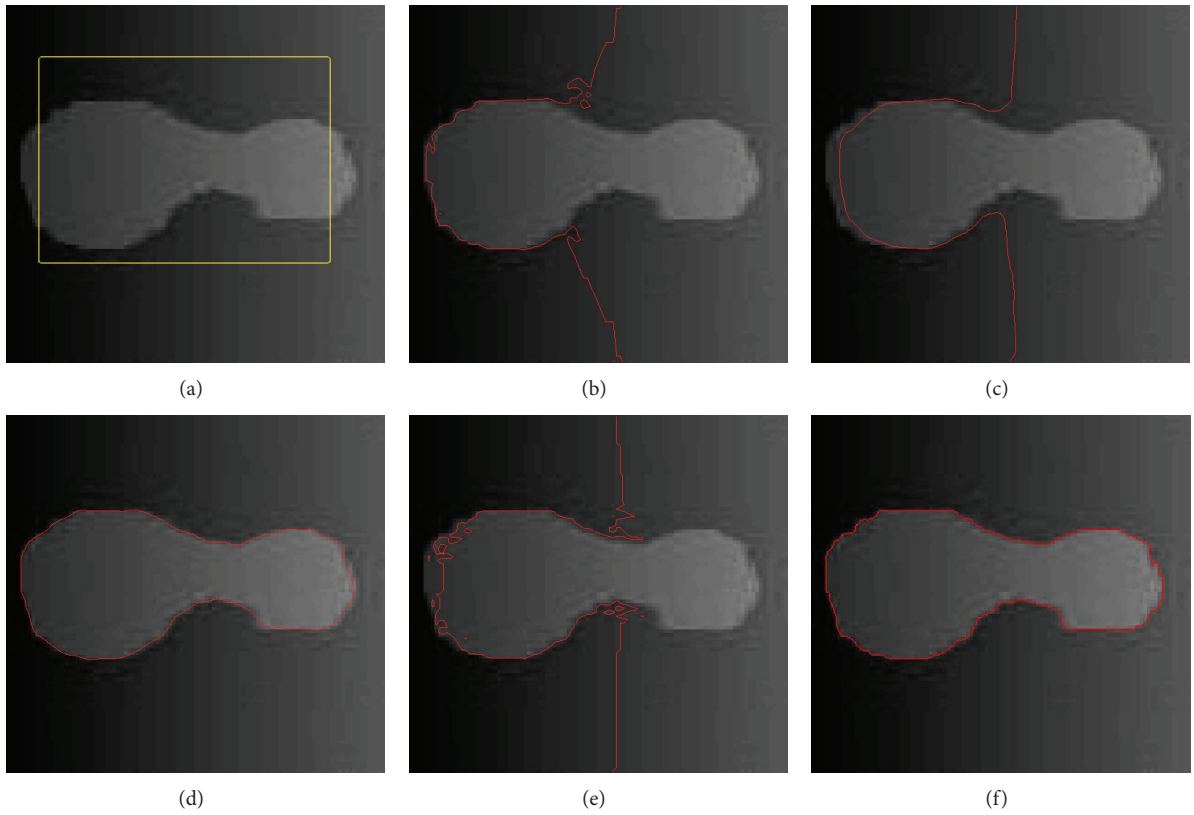


Figure 11: Segmentation results on the synthetic image: (a) the initial contour, (b) CV result, (c) GACV result, (d) DRLSE result, (e) FAC result, and (f) LFAC result.

$$+\left((1-u_n)^m\left(\frac{s_2}{s_2+(1-u_n)^m-(1-u_o)^m}\right)^2 -(1-u_o)^m\right)(I_o-c_2(x,y))^2, \tag{9}$$

where $s_1 = \sum_\Omega [u(i,j)]^m g_k$ and $s_2 = \sum_\Omega [1-u(i,j)]^m g_k$. If $\Delta F < 0$, u_o is replaced by u_n and vice versa.

Step 6. Compute the total energy F according to (8). If the change of F is smaller than ε, stop. Otherwise, return to Step 3.

4. Experimental Results

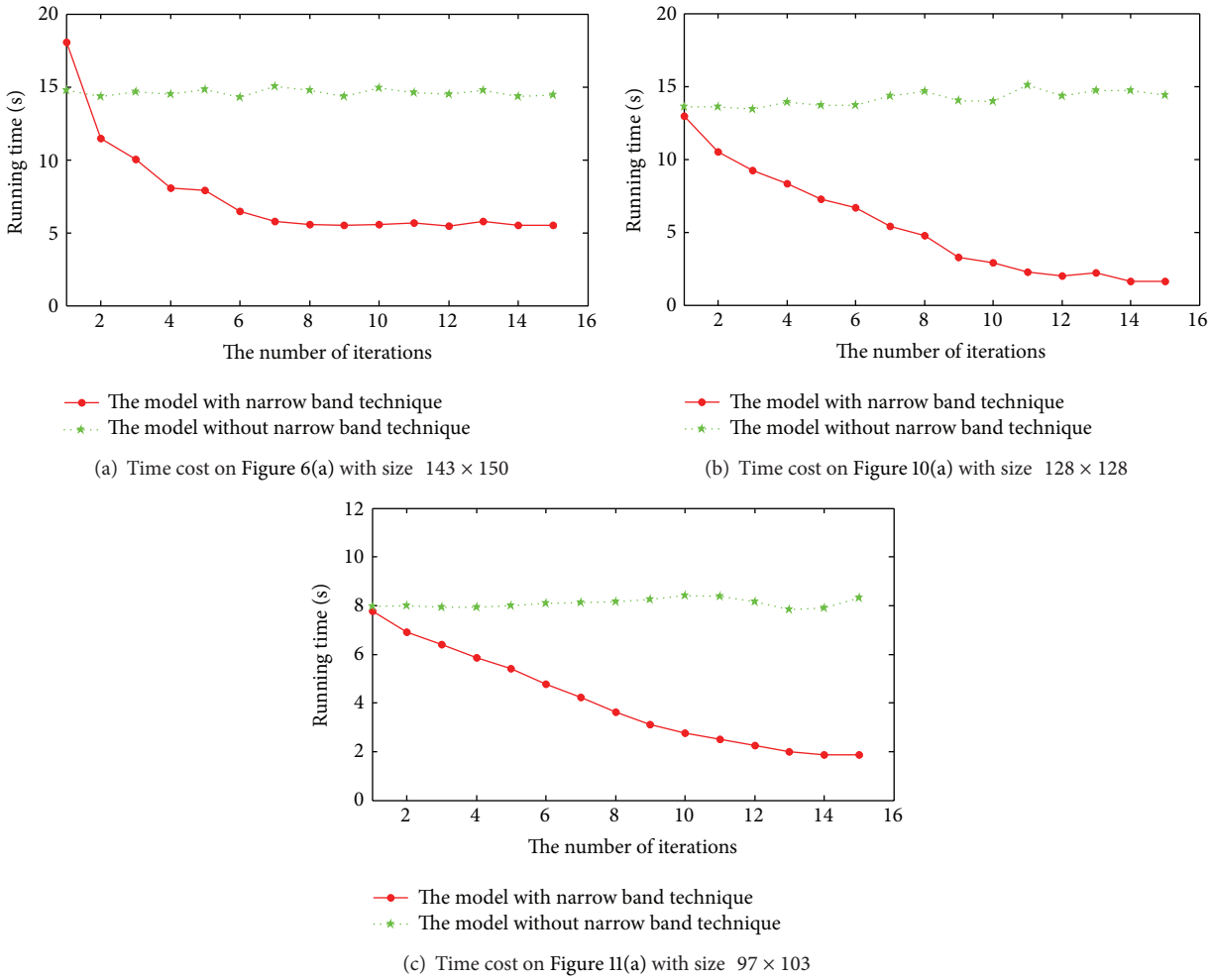
In order to assess the effectiveness of the proposed method, both synthetic and real world images which are widely used in the field of image segmentation algorithms are used in the experiments [14, 18, 25, 26, 31]. We compare the proposed model with state-of-art models including the CV model [9], the Geodesic-Aided C-V method (GACV) [6], the distance regularized level set evolution model (DRLSE) [14], and the FAC model [31]. The initial contours are set to be the same for all the compared models in Sections 4.2 and 4.3.

4.1. Measuring Segmentation Accuracy. In our experimental study, the results are exhibited in two ways: the final segmentation results in figure form and the criteria in tabular form. Since the ground truth of the synthetic images can be obtained, the performances of the compared models were compared with respect to segmentation accuracy. To evaluate segmentation quality, a region-based segmentation accuracy measurement P_{mp} is employed [40]. P_{mp} which measures the variation of the extracted region from the desired region is the percentage of mislabelled pixels and is defined as

$$P_{\text{mp}} = \frac{|R_{\text{mb}}| + |R_{\text{ms}}|}{|R_b| + |R_s|} \times 100\%, \tag{10}$$

where R_b and R_s are the sets of pixels within the desired and extracted regions, respectively, x is the pixel in the image, $R_{\text{mb}} = \{x : x \in R_b \wedge x \notin R_s\}$ is the set of pixels in R_b, but not in R_s, and $R_{\text{ms}} = \{x : x \notin R_b \wedge x \in R_s\}$ is the set of pixels in R_s but not in R_b. The smaller P_{mp} is the better the result is assessed.

4.2. Results on Synthetic Images. The first experiment applies the compared models on synthetic images with different characteristics, in which the intensities of the object regions to be detected are varying. Figure 6 illustrates that the CV model, the GACV model, and the FAC model fail to detect the object at the bottom, while the proposed model and



(a) Time cost on Figure 6(a) with size 143 × 150

(b) Time cost on Figure 10(a) with size 128 × 128

(c) Time cost on Figure 11(a) with size 97 × 103

FIGURE 12: Time cost of the model with and without partition-based technique on three images.

the DRLSE model succeed. The curve obtained by the CV model or the FAC model better depicts the real contour than that of the GACV model. The DRLSE model can detect the object at the bottom, but the region inside the ring cannot be segmented. As the fact that the intensity of the arch is more similar to that of background, the arch is easily submerged into background when adopting constant prototypes as in the CV model, the GACV model, and the FAC model. The comparison between Figures 6(f) and 6(e) demonstrates that the proposed model detects all the objects by introducing the spatial varying prototypes.

We next consider a synthetic image with blurred boundaries as shown in Figure 7(a). Figures 7(b), 7(c), and 7(e) show that similar segmentation results are obtained by the CV model, the GACV model, and the FAC model. The object at the bottom cannot be easily detected by the three models. The curve obtained by the FAC model can better match the real contour inside the ring compared with the CV model and the GACV model. Although the DRLSE model detects a portion of the object at the bottom as shown in Figure 7(d), its segmentation result is not satisfied. Figure 7(f) shows that the contour boundary of the proposed model can be better tailored to the real contour of three objects.

Furthermore, the performance of the compared models under noise conditions is investigated. A synthetic image is corrupted by Gaussian noise and salt and pepper noise. Segmentation results are shown in Figures 8 and 9, respectively. Figures 8(b), 8(c), and 8(e) show that the CV model, the GACV model, and the FAC model are not affected by Gaussian noise, but the object at the bottom cannot be detected by the three models. Figure 8(d) shows that the DRLSE model detects a large proportion of the object at the bottom. But its result is not satisfactory enough, as the contour inside the ring is unable to be detected. Figure 8(f) shows that the proposed model removes the added noise and well detects the object contour.

Figures 9(b), 9(c), and 9(e) show that the results obtained by the CV model, the GACV model, and the FAC model are affected by salt and pepper noise to some extent. The result obtained by the CV model is the most seriously affected by noise. The comparison between Figures 9(e) and 9(f) shows that the proposed model performs better than the

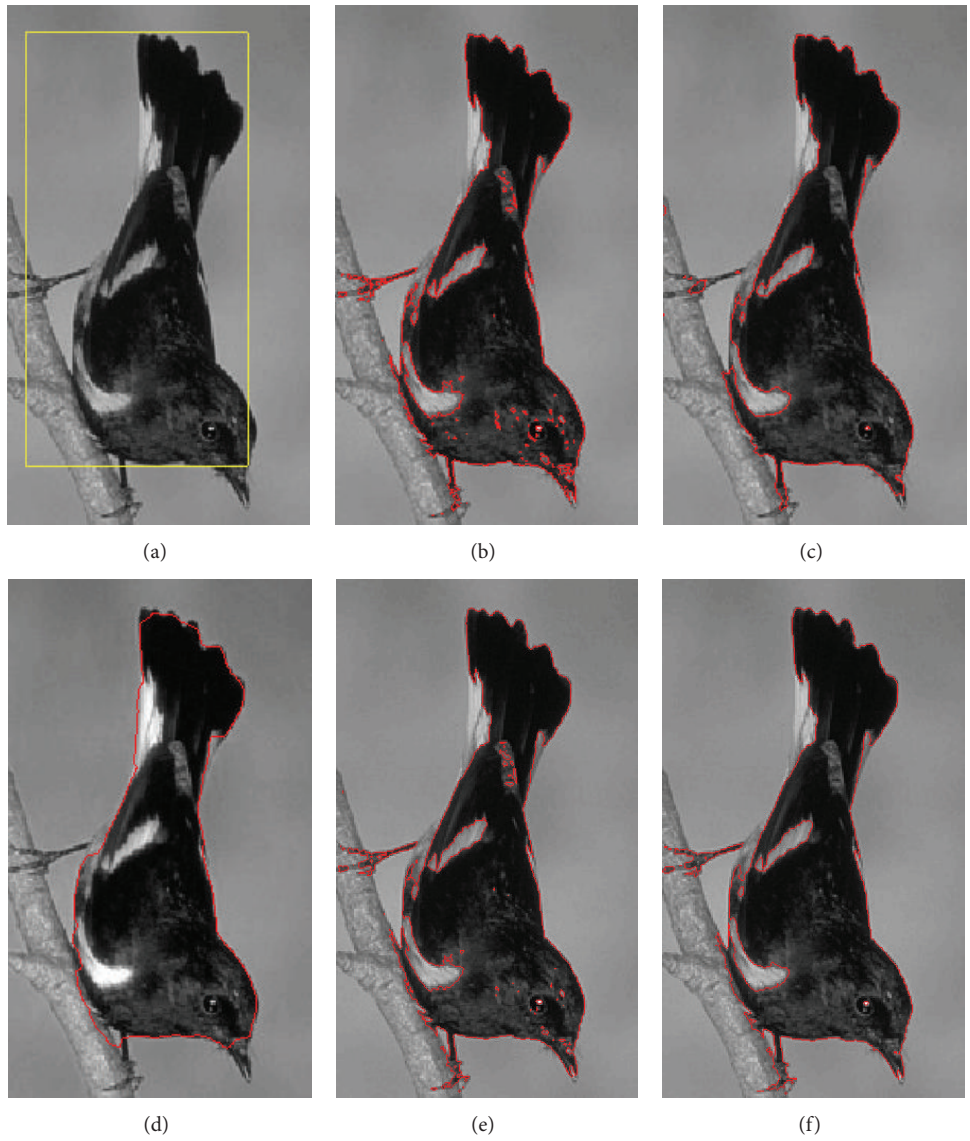


FIGURE 13: Segmentation results on the bird image: (a) the initial contour, (b) the CV model result, (c) the GACV model result, (d) the DRLSE model result, (e) the FAC model result, and (f) the LFAC model result.

FAC model. With the accurate estimation of the relationship among neighbors by employing the local region information, the proposed model obtains better performance than the original FAC model. Figure 9(d) shows that the DRLSE model removes a large proportion of the noise and keeps clear contour boundaries. However, it fails to detect the region inside the ring. Figure 9(f) shows that the proposed model not only obtains a desire partition but also becomes insensitive to noise.

Table 1 shows the segmentation accuracy in terms of P_{mp} obtained by each model. Apart from the proposed model, the DRLSE model obtains the lowest P_{mp} on four images among the remaining models. The FAC model performs slightly better than the CV model and the GACV model in terms of P_{mp}. The value of P_{mp} obtained by the proposed model is lower than that obtained by the FAC model. The comparison between the FAC model and the proposed model demonstrates that object regions which are not distinct from the background can be detected to a larger extent by the proposed model. The proposed model obtains the lowest P_{mp} on four images, which indicates that the object region detected by the proposed model is more similar to the actual object region compared with those of other four models.

Moreover, we apply the compared models on other two synthetic images. Since the ground truth of the two synthetic images is not given, only the segmentation result is shown. Figures 10(b), 10(c), and 10(e) show that the CV model, the GACV model, and the FAC model fail to obtain the right segmentation results, as the top dot cannot be detected. Visually, the DRLSE model and the proposed model are able to segment the dot in white as shown in Figures 10(f) and 10(d). The contour boundary of the proposed model can be better

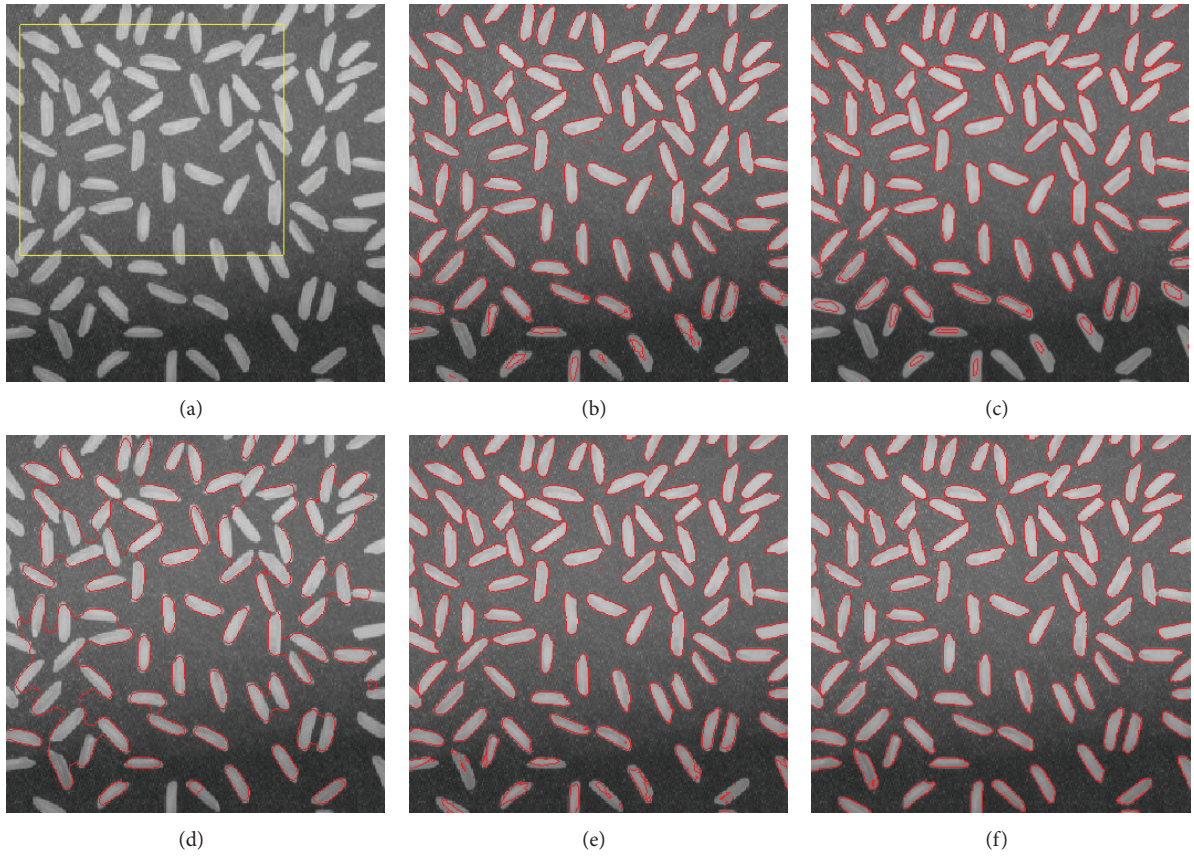


Figure 14: Segmentation results on the rice image: (a) the initial contour, (b) the CV model result, (c) the GACV model result, (d) the DRLSE model result, (e) the FAC model result, and (f) the LFAC model result.

Table 1: Quantitative result given by different models.

	CV	GACV	DRLSE	FAC	The proposed model
P_{mp} on Figure 6(a)	38.77%	41.91%	3.39%	38.77%	**0**
P_{mp} on Figure 7(a)	44.21%	48.85%	16.96%	42.66%	**9.67%**
P_{mp} on Figure 8(a)	38.77%	39.07%	4.57%	38.77%	**0.24%**
P_{mp} on Figure 9(a)	39.15%	41.54%	5.25%	39.15%	**0.32%**

tailored to the real contour of objects than that of the DRLSE model. Figure 11 shows the segment results on a synthetic image where the pixel intensities of background and the object to be detected present high variation. Since, the right part of the background has higher intensities than the left part of object, the FAC model, the CV model, and the GACV model cannot detect the contour of the object at the right side as shown in Figures 11(b), 11(c), and 11(e). Figures 11(d) and 11(f) show that DRLSE and LFAC can better detect the object than other three models.

Finally, Figure 12 illustrates the computational cost on three synthetic images used in this subsection with different sizes for the proposed model with and without partition-based technique. All experiments performed on a Pentium IV (3 GHz) workstation under Windows XP Professional using MATLAB. The model without using partition-based technique is computational consuming, since all the pixels are updated in each iteration. With the help of shadowed sets, only the pixels within the boundary region are updated in each iteration. Since the number of pixels within the region close to the contour boundary becomes smaller with time increasing, the time cost of the proposed model employing partition-based technique is less than the model without using partition-based technique as the result shown in Figure 12.

4.3. Results on Real World Images. In this part, two medical images and two natural images which are widely used in the field of image segmentation algorithms are used in the experiments [14, 18, 25, 26, 31]. Figure 13 shows that the five models achieve the similar results in general, except in the area near the tree branch and the head of the bird. Figure 13(b) shows that the CV model fails to detect

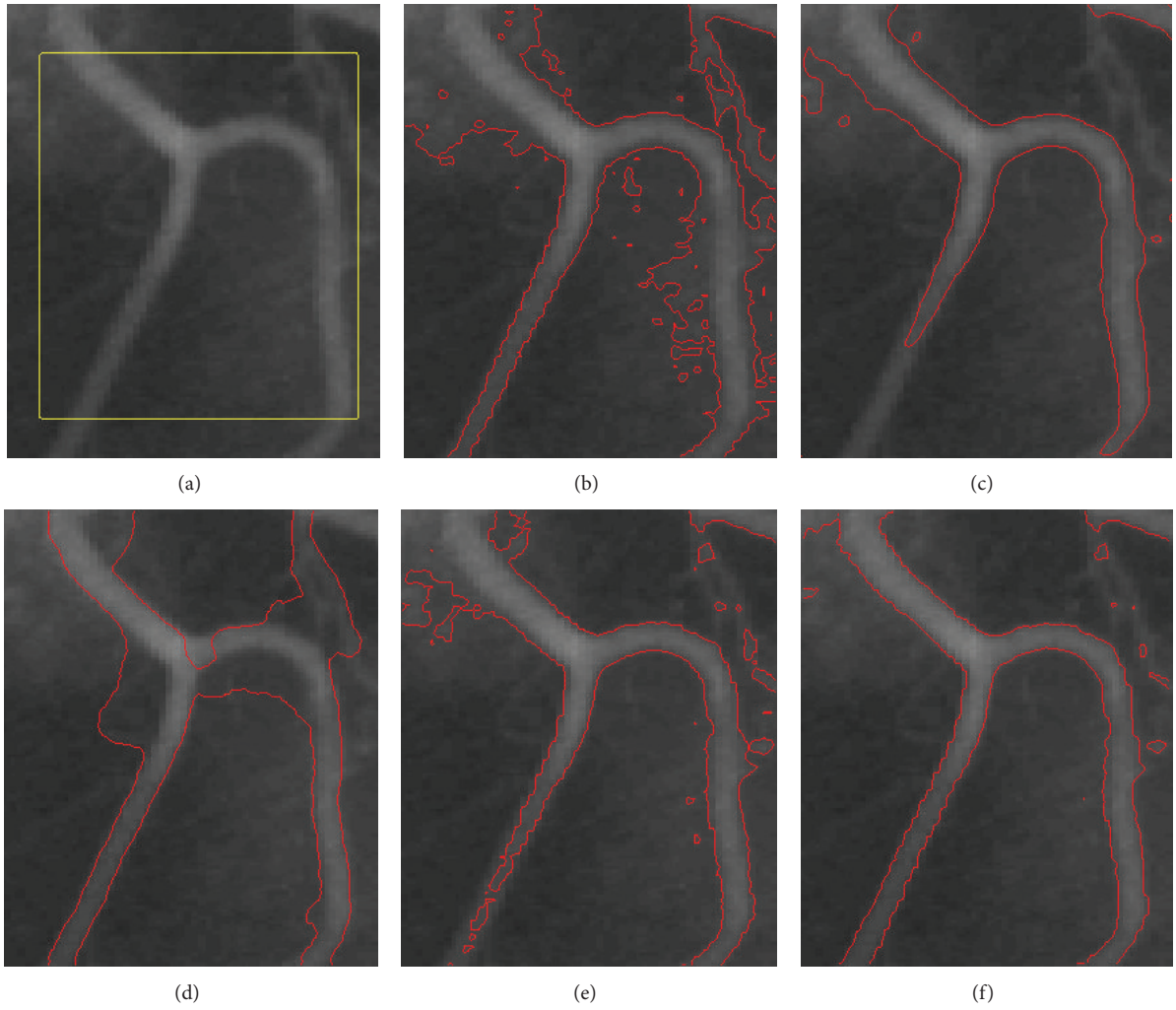


Figure 15: Segmentation results on the first vessel image: (a) the initial contour, (b) the CV model result, (c) the GACV model result, (d) the DRLSE model result, (e) the FAC model result, and (f) the LFAC model result.

the head and the claw. Two paws cannot be detected by the GACV model as shown in Figure 13(c). Figure 13(e) illustrates that the FAC model better detects the paws than the CV model and the GACV model. As shown in Figure 13(d), the DRLSE model achieves a clear contour, while the contour cannot depict real boundaries of the paws and the beak. The proposed model correctly detects the border between the bird and the tree branch. The final curves obtained by LFAC well approach the real contour of the bird, which can be seen from the tail, the paws, and the beak.

Figure 14(d) shows that the DRLSE model fails to segment the rice image. Figures 14(b), 14(c), and 14(e) depict that the CV model, the GACV model, and the FAC model are unable to correctly detect the real contour of all the rice grains, especially the rice grains at the bottom of the image. Besides, the results obtained by the CV model and the FAC model have some noises on the top of the image. The proposed model extracts the desired contour of all the rice grains as shown in Figure 14(f). With the help of spatial varying prototypes, even the rice grains at the bottom with little difference in terms of the intensity from the background can be well detected.

Furthermore, the performance of the compared models on two vessel images is evaluated. Figure 15 shows that the results obtained by the GACV model and the FAC model cannot detect the whole vessel, as the vessel at the bottom left of the image is submerged into the background. Although the results obtained by the CV model and the DRLSE model detect the vessel at the bottom, they fail to approach the real contour of the vessel as shown in Figures 15(f) and 15(e). Figure 15(f) indicates that that the final contour obtained by the proposed model well depicts the real border of the vessel. Figures 16(d) and 16(e) demonstrate that the results obtained by DRLSE and FAC cannot detect the vessel at the bottom left of the image. Although the segmentation results obtained by CV and GACV detect the vessel at the bottom of the image, they still fail to depict the real contour of the vessel on the right side. The vessel is well detected by the proposed model as shown in Figure 16(f). Moreover, the curve obtained by the proposed model well approach the real border of the vessel.

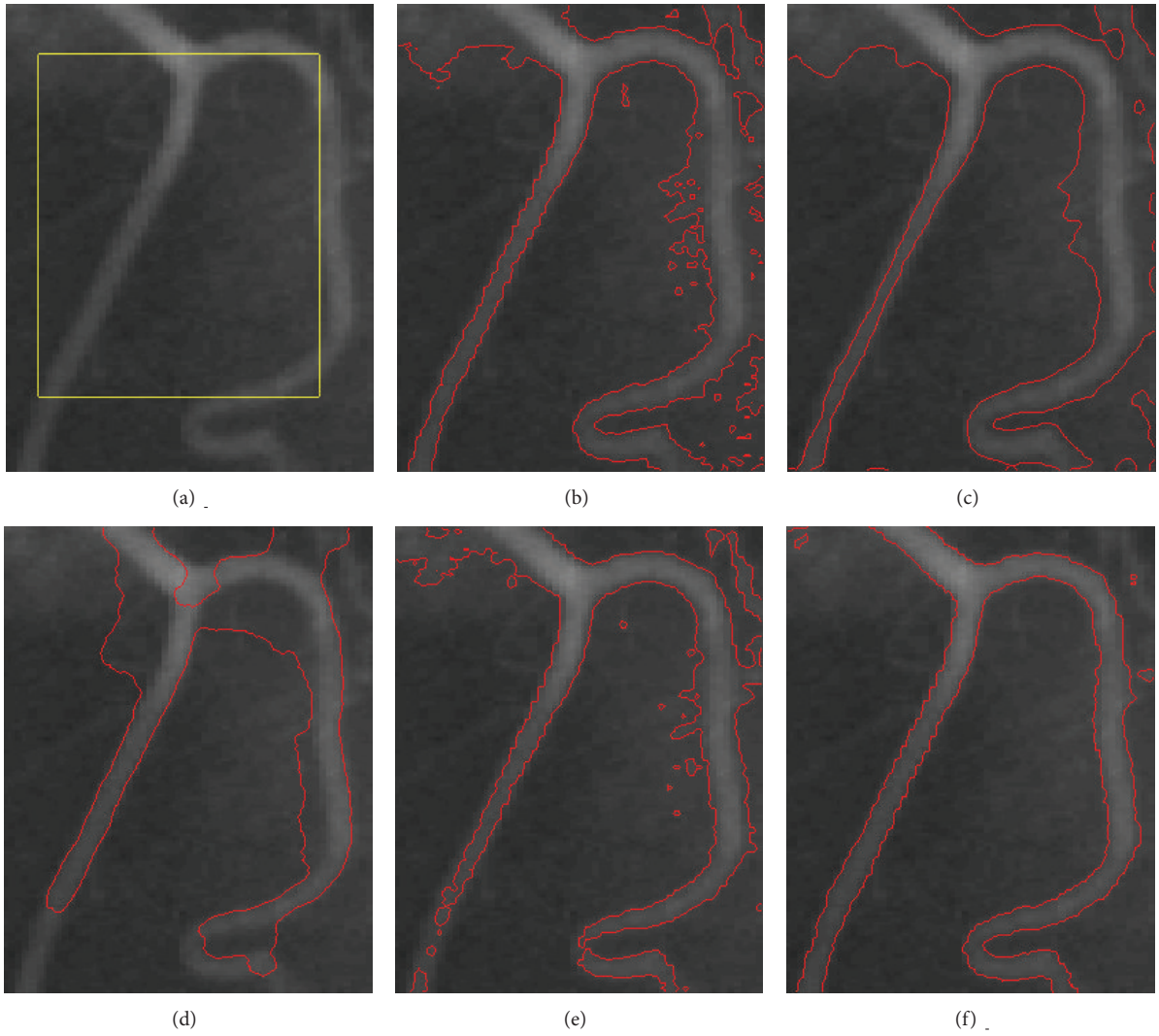


FIGURE 16: Segmentation results on the second vessel image: (a) the initial contour, (b) the CV model result, (c) the GACV model result, (d) the DRLSE model result, (e) the FAC model result, and (f) the LFAC model result.

4.4. The Effect of Different Initial Conditions. The effect of different initial conditions on the performance of the five models is investigated in this part. Different initial conditions are depicted in Figure 17(a). Segmentation results obtained by the five models are shown in Figures 17(b)–17(f), respectively. Segmentation results obtained by the CV model, the GACV model, and the DRLSE model are sensitive to different initial conditions. All the results obtained by the FAC model and the proposed model are the same as shown in Figures 17(e) and 17(f), which demonstrates the robustness of the two fuzzy active contour models. Moreover, compared with other four models, the proposed model can obtain the correct segmentation results under different initial conditions.

5. Concluding Remarks

In this paper, an enhanced partition-based fuzzy active contour model with incorporating local information for image segmentation is proposed. In the proposed model, the prototypes are not the average intensities of pixels inside and outside of the curve. Instead, they are spatially varying and are related to each pixel. By considering local image characteristics, the proposed model can efficiently segment images with intensity inhomogeneity. Moreover, to confine the update area at each iteration, shadowed sets theory is employed to adaptively detect the regions near contour boundaries, which enables the curve to evolve gradually and saves the computational resource.

The proposed approach is basically built on fuzzy active contour model with incorporating local information. The main advantages of the proposed model can be concluded as (1) computational simplicity (the calculation of each step only includes the computation of pixels within the interesting region); (2) flexibility (it has less sensitivity to initial conditions by incorporating fuzzy technique); (3) it can efficiently segment images with intensity inhomogeneity by considering image local characteristics. However, classical ACMs do not consider any spatial information in image context, which

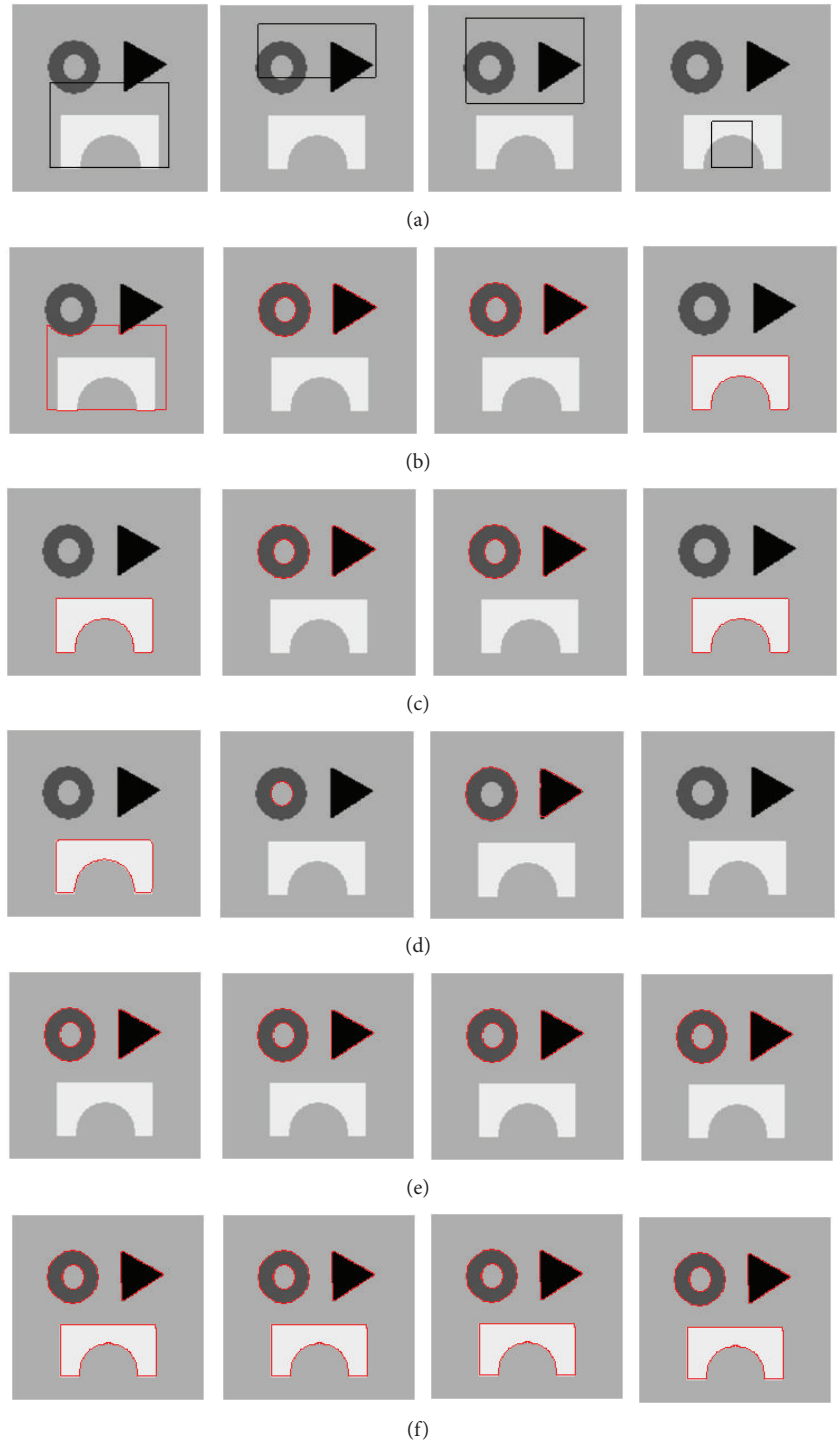


Figure 17: Detection of different objects from a synthetic image with various initial conditions: (a) the four different initial contours, (b) the CV model results of the corresponding initial conditions, (c) the GACV model results of the corresponding initial conditions, (d) the DRLSE model results of the corresponding initial conditions, (e) the FAC model results of the corresponding initial conditions, and (f) the LFAC model results of the corresponding initial conditions.

makes them sensitive to noise and other imaging artifacts to some extent. Recently, many researchers have incorporated local spatial information into segmentation algorithms to improve the performance of image segmentation. In the further work, we want to design a fuzzy active contour model which retains more image details and preserves robustness to noise by taking both spatial and gray constraints into consideration.

Appendix

The proofs of the center and energy differences are presented in this part. The spatial varying centers of each pixel are written in the following forms:

$$c_1(x,y) = \frac{\sum_\Omega [u(i,j)]^m g_k I(i,j)}{\sum_\Omega [u(i,j)]^m g_k},$$
$$c_2(x,y) = \frac{\sum_\Omega [1-u(i,j)]^m g_k I(i,j)}{\sum_\Omega [1-u(i,j)]^m g_k}, \tag{A.1}$$

where (x, y) is the spatial coordinate of the current pixel. (i, j) is the pixel in the set of neighbors falling into the local region around the current pixel. g_k is the kernel function, which can be taken as the weight assignment of the pixel within the neighbor of the current point. $u(i, j)$ is the degree of membership of pixel (i, j) belonging to the inside of the curve C. m is the fuzzy coefficient.

Assume that the gray value of the pixel is I_o. u_o is the degree of membership of the pixel. The new degree of membership of the pixel is u_n. The spatial varying centers of the pixel $\tilde{c}_1(x, y)$ and $\tilde{c}_2(x, y)$ will have a corresponding change. Considering that the membership value of only one pixel is changed, then $\tilde{c}_1(x, y)$ is given as follows:

$$\begin{aligned}
&\tilde{c}_1(x,y)\\
&= \frac{\sum_\Omega [\tilde{u}(i,j)]^m g_k I(i,j)}{\sum_\Omega [\tilde{u}(i,j)]^m g_k}\\
&= \frac{\sum_\Omega [u(i,j)]^m g_k I(i,j) + u_n^m I_o - u_o^m I_o}{\sum_\Omega [u(i,j)]^m g_k + u_n^m - u_o^m}\\
&= \frac{s_1 c_1(x,y) + I_o(u_n^m - u_o^m)}{s_1 + u_n^m - u_o^m}\\
&= \left(c_1(x,y)(s_1 + u_n^m - u_o^m) - c_1(x,y)(u_n^m - u_o^m)\right.\\
&\qquad \left. + I_o(u_n^m - u_o^m)\right)(s_1 + u_n^m - u_o^m)^{-1}\\
&= c_1(x,y) + \frac{u_n^m - u_o^m}{s_1 + u_n^m - u_o^m}(I_o - c_1(x,y)),
\end{aligned} \tag{A.2}$$

where $s_1 = \sum_\Omega [u(i, j)]^m g_k$.

The $\tilde{c}_2(x, y)$ will be given as the similar way:

$$\begin{aligned}
&\tilde{c}_2(x,y)\\
&= c_2(x,y) + \frac{(1-u_n)^m - (1-u_o)^m}{s_2 + (1-u_n)^m - (1-u_o)^m}(I_o - c_2(x,y)),
\end{aligned} \tag{A.3}$$

where $s_2 = \sum_\Omega [1 - u(i, j)]^m g_k$. Thus, the calculation of the new spatial varying centers can be easily achieved according to (A.2) and (A.3), respectively.

Besides, the change of the pixel in terms of the membership value and the spatial varying centers will cause a change in fuzzy energy. Considering that the membership value and the spatial varying centers of only one pixel are changed, the new fuzzy energy is given as follows:

$$\begin{aligned}
\tilde{F} = &\underbrace{\sum_\Omega [\tilde{u}(x,y)]^m (I(x,y) - \tilde{c}_1(x,y))^2}_{\tilde{A}}\\
&+ \underbrace{\sum_\Omega [1-\tilde{u}(x,y)]^m (I(x,y) - \tilde{c}_2(x,y))^2}_{\tilde{B}}.
\end{aligned} \tag{A.4}$$

$\tilde{A}$ and $\tilde{B}$ can be formulated separately as follows:

$$\begin{aligned}
&\tilde{A}\\
&= \sum_\Omega [\tilde{u}(x,y)]^m (I(x,y) - \tilde{c}_1(x,y))^2\\
&= \sum_\Omega [u(x,y)]^m (I(x,y) - c_1(x,y))^2\\
&\quad + u_n^m (I_o - \tilde{c}_1(x,y))^2 - u_o^m (I_o - c_1(x,y))^2\\
&= \mathrm{A} + u_n^m\\
&\quad \times \left(I_o - c_1(x,y) - \frac{u_n^m - u_o^m}{s_1 + u_n^m - u_o^m}(I_o - c_1(x,y))\right)^2\\
&\quad - u_o^m (I_o - c_1(x,y))^2\\
&= A + u_n^m \left(\frac{s_1}{s_1 + u_n^m - u_o^m}(I_o - c_1(x,y))\right)^2\\
&\quad - u_o^m (I_o - c_1(x,y))^2\\
&= A + \left(u_n^m \left(\frac{s_1}{s_1 + u_n^m - u_o^m}\right)^2 - u_o^m\right)\\
&\quad \times (I_o - c_1(x,y))^2,
\end{aligned} \tag{A.5}$$

where $A = \sum_\Omega [u(x, y)]^m (I_o - c_1(x, y))^2$.

In a similar way, it can be proven that

$$\widetilde{B} = B + \left((1-u_n)^m \left(\frac{s_2}{s_2 + (1-u_n)^m - (1-u_o)^m} \right)^2 - (1-u_o)^m \right) (I_o - c_2(x,y))^2, \tag{A.6}$$

where $B = \sum_\Omega [1-u(x,y)]^m (I_o - c_2(x,y))^2$.

As $\widetilde{F} = \underbrace{\sum_\Omega [\widetilde{u}(x,y)]^m (I(x,y) - \widetilde{c}_1(x,y))^2}_{\widetilde{A}} + \underbrace{\sum_\Omega [1-\widetilde{u}(x,y)]^m (I(x,y) - \widetilde{c}_2(x,y))^2}_{\widetilde{B}}$, the new energy is obtained by combining (A.5) and (A.6), which will be given as

$$\begin{aligned}
\widetilde{F} &= \sum_\Omega [u(x,y)]^m (I_o - c_1(x,y))^2 + \sum_\Omega [1-u(x,y)]^m \\
&\quad \times (I_o - c_2(x,y))^2 + \left(u_n^m \left(\frac{s_1}{s_1 + u_n^m - u_o^m} \right)^2 - u_o^m \right) \\
&\quad \times (I_o - c_1(x,y))^2 \\
&\quad + \left((1-u_n)^m \right. \\
&\qquad \left. \times \left(\frac{s_2}{s_2 + (1-u_n)^m - (1-u_o)^m} \right)^2 - (1-u_o)^m \right) \\
&\quad \times (I_o - c_2(x,y))^2 \\
&= F + \left(u_n^m \left(\frac{s_1}{s_1 + u_n^m - u_o^m} \right)^2 - u_o^m \right) (I_o - c_1(x,y))^2 \\
&\quad + \left((1-u_n)^m \left(\frac{s_2}{s_2 + (1-u_n)^m - (1-u_o)^m} \right)^2 \right. \\
&\qquad \left. - (1-u_o)^m \right) (I_o - c_2(x,y))^2.
\end{aligned} \tag{A.7}$$

Conflict of Interests

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

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant nos. 61377011, 61302063, 61273317, 61202176, and 61203303) and an EU FP7 IRSES Grant (no. 247619) on "Nature Inspired Computation and its Applications (NICaiA)."

References

[1] L. Ding, A. Yilmaz, and R. Yan, "Interactive image segmentation using Dirichlet process multiple-view learning," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 2119–2129, 2012.

[2] S. Chen and D. Zhang, "Robust image segmentation using FCM with spatial constraints based on new kernel-induced distance measure," *IEEE Transactions on Systems, Man, and Cybernetics B: Cybernetics*, vol. 34, no. 4, pp. 1907–1916, 2004.

[3] W. Cai, S. Chen, and D. Zhang, "Fast and robust fuzzy c-means clustering algorithms incorporating local information for image segmentation," *Pattern Recognition*, vol. 40, no. 3, pp. 825–838, 2007.

[4] L. Yi, G. Zhang, and Z. Wu, "A scale-synthesis method for high spatial resolution remote sensing image segmentation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, no. 10, pp. 4062–4070, 2012.

[5] C. Fowlkes, S. Belongie, F. Chung, and J. Malik, "Spectral Grouping Using the Nyström Method," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 2, pp. 214–225, 2004.

[6] L. Chen, Y. Zhou, Y. Wang, and J. Yang, "GACV: geodesic-aided C-V method," *Pattern Recognition*, vol. 39, no. 7, pp. 1391–1395, 2006.

[7] M. Gong, Y. Liang, J. Shi, and W. P. Ma, "Fuzzy C-means clustering with local information and kernel metric for image segmentation," *IEEE Transactions on Image Processing*, vol. 22, no. 2, pp. 573–584, 2013.

[8] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1988.

[9] T. F. Chan and L. A. Vese, "Active contours without edges," *IEEE Transactions on Image Processing*, vol. 10, no. 2, pp. 266–277, 2001.

[10] C. Li, R. Huang, Z. Ding, J. C. Gatenby, D. N. Metaxas, and J. C. Gore, "A level set method for image segmentation in the presence of intensity inhomogeneities with application to MRI," *IEEE Transactions on Image Processing*, vol. 20, no. 7, pp. 2007–2016, 2011.

[11] T. N. A. Nguyen, J. Cai, and J. Zhang, "Robust interactive image segmentation using convex active contours," *IEEE Transactions on Image Processing*, vol. 21, no. 8, pp. 3734–3743, 2012.

[12] M. B. Salah, A. Mitiche, and I. B. Ayed, "Effective level set image segmentation with a kernel induced data term," *IEEE Transactions on Image Processing*, vol. 19, no. 1, pp. 220–232, 2010.

[13] Y. Liu and Y. Z. Yu, "Interactive image segmentation based on level sets of probabilities," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 2, pp. 202–213, 2012.

[14] C. Li, C. Xu, C. Gui, and M. D. Fox, "Distance regularized level set evolution and its application to image segmentation," *IEEE Transactions on Image Processing*, vol. 19, no. 12, pp. 3243–3254, 2010.

[15] L. D. Cohen, "Multiple contour finding and perceptual grouping using minimal paths," *Journal of Mathematical Imaging and Vision*, vol. 14, no. 3, pp. 225–236, 2001.

[16] Ç. E. Erdem, A. M. Tekalp, and B. Sankur, "Video object tracking with feedback of performance measures," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 4, pp. 310–324, 2003.

[17] N. Xu, R. Bansal, and N. Ahuja, "Object segmentation using graph cuts based active contours," in *Proceedings of the IEEE*

Computer Society Conference on Computer Vision and Pattern Recognition, pp. 46–53, Madison, Wis, USA, June 2003.

[18] C. Li, C. Xu, C. Gui, and M. Fox, "Level set evolution without re-initialization: a new variational formulation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, vol. 1, pp. 430–436, June 2005.

[19] J. H. Zhao, B. Y. Chen, M. G. Sun, W. Y. Jia, and Z. Y. Yuan, "Improved algorithm for gradient vector flow based active contour model using global and local information," *The Scientific World Journal*, vol. 2013, Article ID 479675, 8 pages, 2013.

[20] D. Mumford and J. Shah, "Optimal approximations by piecewise smooth functions and associated variational problems," *Communications on Pure and Applied Mathematics*, vol. 42, no. 5, pp. 577–685, 1989.

[21] F. Gibou and R. Fedkiw, "A fast hybrid k-means level set algorithm for segmentation," in *Proceedings of the Annual Hawaii International Conference on Statistics and Mathematics*, pp. 281–291, 2005.

[22] L. Bertelli, B. Sumengen, B. S. Manjunath, and F. Gibou, "A variational framework for multiregion pairwise-similarity-based image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 8, pp. 1400–1414, 2008.

[23] K. Zhang, L. Zhang, H. Song, and W. Zhou, "Active contours with selective local or global segmentation: a new formulation and level set method," *Image and Vision Computing*, vol. 28, no. 4, pp. 668–676, 2010.

[24] J. Ning, L. Zhang, D. Zhang, and C. Wu, "Interactive image segmentation by maximal similarity based region merging," *Pattern Recognition*, vol. 43, no. 2, pp. 445–456, 2010.

[25] K. Zhang, H. Song, and L. Zhang, "Active contours driven by local image fitting energy," *Pattern Recognition*, vol. 43, no. 4, pp. 1199–1206, 2010.

[26] K. Shyu, V. Pham, T. Tran, and P. Lee, "Global and local fuzzy energy-based active contours for image segmentation," *Nonlinear Dynamics*, vol. 67, no. 2, pp. 1559–1578, 2012.

[27] X. Wang, D. Huang, and H. Xu, "An efficient local Chan-Vese model for image segmentation," *Pattern Recognition*, vol. 43, no. 3, pp. 603–618, 2010.

[28] C. L. Pereira, C. A. C. M. Bastos, T. I. Ren, and G. D. C. Cavalcanti, "Fuzzy active contour models," in *Proceeding of the IEEE International Conference on Fuzzy Systems (FUZZ '11)*, pp. 1621–1627, Taipei, Taiwan, June 2011.

[29] M. Gong, Z. Zhou, and J. Ma, "Change detection in synthetic aperture radar images based on image fusion and fuzzy clustering," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 2141–2151, 2012.

[30] X. Yang, G. Zhang, J. Lu, and J. Ma, "A kernel fuzzy c-Means clustering-based fuzzy support vector machine algorithm for classification problems with outliers or noises," *IEEE Transactions on Fuzzy Systems*, vol. 19, no. 1, pp. 105–115, 2011.

[31] S. Krinidis and V. Chatzis, "Fuzzy energy-based active contours," *IEEE Transactions on Image Processing*, vol. 18, no. 12, pp. 2747–2755, 2009.

[32] Y. Zhang, S. Wang, G. Ji, and Z. Dong, "An MR brain images classifier system via particle swarm optimization and kernel support vector machine," *The Scientific World Journal*, vol. 2013, Article ID 130134, 9 pages, 2013.

[33] R. Patra and S. K. Saha, "A kernel-based approach for biomedical named entity recognition," *The Scientific World Journal*, vol. 2013, Article ID 950796, 7 pages, 2013.

[34] X. Yin, S. Chen, E. Hu, and D. Zhang, "Semi-supervised clustering with metric learning: an adaptive kernel method," *Pattern Recognition*, vol. 43, no. 4, pp. 1320–1333, 2010.

[35] K. Wu and M. Yang, "Alternative c-means clustering algorithms," *Pattern Recognition*, vol. 35, no. 10, pp. 2267–2278, 2002.

[36] W. Pedrycz, "From fuzzy sets to shadowed sets: interpretation and computing," *International Journal of Intelligent Systems*, vol. 24, no. 1, pp. 48–61, 2009.

[37] G. Cattaneo and D. Ciucci, "An algebraic approach to shadowed sets," *Electronic Notes in Theoretical Computer Science*, vol. 82, no. 4, pp. 67–78, 2003.

[38] W. Pedrycz, "Shadowed sets: representing and processing fuzzy sets," *IEEE Transactions on Systems, Man, and Cybernetics B: Cybernetics*, vol. 28, no. 1, pp. 103–109, 1998.

[39] J. Zhou, W. Pedrycz, and D. Miao, "Shadowed sets in the characterization of rough-fuzzy clustering," *Pattern Recognition*, vol. 44, no. 8, pp. 1738–1749, 2011.

[40] C. Cao, T. S. Newman, and G. A. Germany, "New shape-based auroral oval segmentation driven by LLS-RHT," *Pattern Recognition*, vol. 42, no. 5, pp. 607–618, 2009.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 468405, 11 pages
http://dx.doi.org/10.1155/2014/468405

Research Article

A Procedure for Extending Input Selection Algorithms to Low Quality Data in Modelling Problems with Application to the Automatic Grading of Uploaded Assignments

José Otero,[1] Ana Palacios,[2] Rosario Suárez,[1] Luis Junco,[1] Inés Couso,[3] and Luciano Sánchez[1]

[1] *Computer Science Department, Universidad de Oviedo, Sedes Departamentales, Edificio 1, Campus de Viesques, 33203 Gijón, Spain*
[2] *Computer Science Department, Universidad de Granada, C/Periodista Daniel Saucedo Arana s/n, 18071 Granada, Spain*
[3] *Statistics Department, E. U. I. T. Industrial, Universidad de Oviedo, Módulo 1, Planta 4, Campus de Viesques, 33203 Gijón, Spain*

Correspondence should be addressed to Luciano Sánchez; luciano@uniovi.es

Received 11 March 2014; Revised 26 May 2014; Accepted 9 June 2014; Published 7 July 2014

Academic Editor: Anand Paul

When selecting relevant inputs in modeling problems with low quality data, the ranking of the most informative inputs is also uncertain. In this paper, this issue is addressed through a new procedure that allows the extending of different crisp feature selection algorithms to vague data. The partial knowledge about the ordinal of each feature is modelled by means of a possibility distribution, and a ranking is hereby applied to sort these distributions. It will be shown that this technique makes the most use of the available information in some vague datasets. The approach is demonstrated in a real-world application. In the context of massive online computer science courses, methods are sought for automatically providing the student with a qualification through code metrics. Feature selection methods are used to find the metrics involved in the most meaningful predictions. In this study, 800 source code files, collected and revised by the authors in classroom Computer Science lectures taught between 2013 and 2014, are analyzed with the proposed technique, and the most relevant metrics for the automatic grading task are discussed.

1. Introduction

Online courses are ubiquitous nowadays. Almost every institution, university, college, or high-school offers freely accessible online courses. Massive Open Online Courses (MOOCs) and Distance Learning have a large impact in developing countries, helping to improve education in poor regions.

Learning Management Systems or Content Management Systems are used to provide the students with different kinds of material and also allow students and teachers to interact via lectures, assignments, exams, or gradings. However, the resources needed for tracking students and taking examination are time consuming for the organizing institutions; thus, there is demand for intelligent techniques that help the instructor to manage large groups of students. In particular, procedures that partially or completely automate the grading process are sought, understood as taking standardized measurements of varying levels of achievement in a course [1].

There are topics, however, whose qualification is troublesome. Think for instance of computer programming, where the usual examination procedure consists in challenging the students with a set of problems to be solved. In online courses, the student's solutions, comprising one or more source code files, are uploaded to the platform, where the person at charge scores the task. This needs a long time and it is also difficult for the teacher to be objective and unbiased. If the grading depends not only on the program output correctness (using a set of sample data inputs) but also on the structure of the solution (data types, control flow, ans efficiency) or the documentation quality, the situation is even worse. In addition to this, the students should follow the usual software developing process and thus the solution of each assignment

should pass through several stages until it reaches a maturity level such that it can be submitted as a completed task. Those intermediate stages could provide a valuable feedback to the teacher, regarding individual students' needs and also teacher's lectures, materials or strategies quality.

1.1. Automatic Grading in Online Courses. The automatic grading is a problem that has been addressed by many researchers. To name some, in [2] a semiautomated system for task submission and grading is proposed, but the grading itself must be done manually by the teacher. The *WebToTeach* system [1], on the contrary, is able to check submitted source code automatically. Similar to this and focused on programming, the methods in [3] or [4] achieve an automatic grading by comparing the output of each student program with the output of a correct program. There is no measurement of the internals of the source code, which is labelled as correct if the output is correct, regardless of the solution strategy.

The *AutoLEP* system [5] is more recent. One of the salient points of this last work is a procedure to compare any implementation of an algorithm against a single model. Furthermore, in [6] a methodology is presented that accomplishes automatic grading by testing the program results against a predefined set of inputs and also by formally verifying the source code or by measuring the similarities between the control flow graph and the teacher's solution. The parameters of a linear model are found that averages the influence of the three techniques in order to match teacher's grade and automatic grading in a corpus of manually graded exercises. Finally, in [7], software metrics are used to measure the properties of the students' programs and a fuzzy rule-based system is used to determine how close the programs submitted by students and the solutions provided by the teacher are, partially achieving an automatic grading.

1.1.1. Automatic Grading and Continuous Assessment. The former approaches pay particular attention to exam grading. In the problem at hand, this consists in comparing the outputs of student programs to those of a correct program, but there are secondary aspects about the internals of the source code (i.e., code style, documentation, etc.) that must be assessed too. In some cases, software metrics provide an additional insight [7].

However, the purpose of following an online course is arguably not to obtain a certificate but to acquire knowledge. From the instructor's side, it is important that an early corrective action is taken if learning difficulties are detected and therefore a continuous assessment of the student must be carried out. In this case, the picture is completely different. It is hard to combine MOOCs and continuous assessment; however, it is clear that the evolution of each student could not be tracked down to a single exam. Incremental measurements of the levels of achievement of each programming concept should be taken, with the help of the many different assignments that the students upload to the server hosting the course. The number and size of these assignments depend on the student, and some of its elements might be missing; not all the online students finish their tasks, and the amount of work carried out by the students is largely different.

Therefore, different sets of assignments must be combined and jointly considered by the grading system. The combination procedure must be resilient to missing data and incomplete assignments, as students will be graded on the basis of sets of data of different sizes. Finally, if software metrics are used to assess the quality of the assignments, not all of them are equally informative for each programming concept. Because of the mentioned reasons, in this paper,

(i) a method is proposed for building a fuzzy compound value that summarizes the values of the software metrics of different source files that are related to the same programming concept. This compound value takes into account both the average value and the dispersion of the different metrics;

(ii) the relevance of the different metrics is assessed with an extension of a crisp feature selection algorithm to fuzzy data. It will be shown that the extension described in this paper exploits the available data in a real-world problem better than the alternatives;

(iii) a learning fuzzy system that can extract if-then rules from interval and fuzzy data is used to build the rule based system that performs the grading on the basis of the metrics that are selected in the preceding step.

This paper is organized as follows: in Section 2, the method for combining the values of a metric over a set of different source files and a method for ranking the importance of the fuzzy aggregated values are described. In Section 3, the rule learning algorithm is described. In Section 4, numerical results are provided that validate the claims of this paper with actual data collected in classroom lectures in 2013 and 2014. Section 5 concludes the paper and highlights future research lines.

2. Feature Selection for Regression with Vague Data

As mentioned, the grading process is intended to determine the level of achievement of each programming concept, which in turn is assessed by means of a set of source code files written by the students. The metrics of all files in these sets are jointly considered. Given that these sets are of different sizes for different students and some of its elements may be missing, a robust combination method is needed.

The proposed combination is based on the assumption that the application of a software metric to a given source code can be assimilated to the process of measuring the value of an observable variable or *item* that provides partial information to describe an unobservable or *latent* variable. In this case, the latent variable is the degree of assessment of a given programming concept. It is remarked that the information provided by different items may be in conflict.

The conversion of a set of items into a compound value that can be fed into a model has been solved in different ways in other contexts. For instance, in marketing problems, certain models have been designed where sets of items are

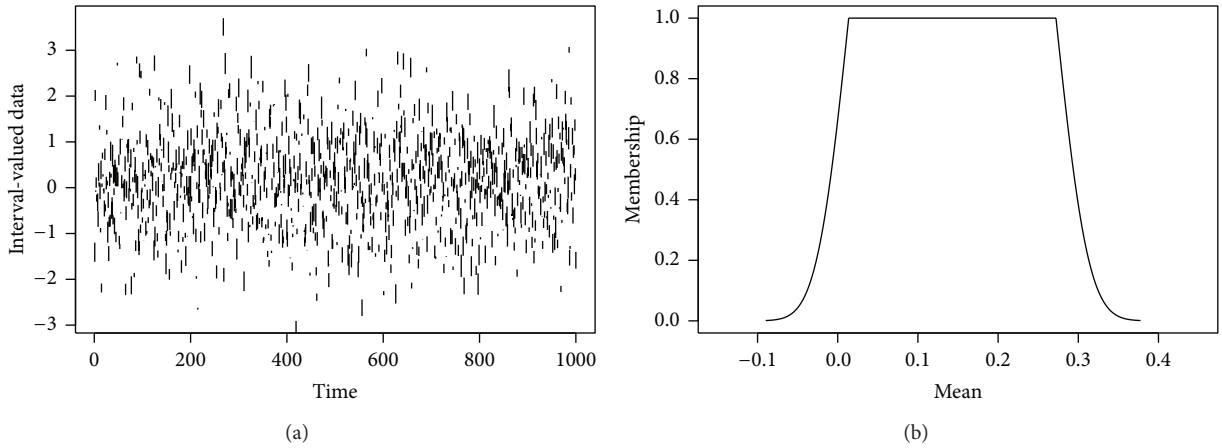


Figure 1: Fuzzy representation of LQD. (a) 1000 instances of an interval-valued database. (b) Fuzzy (possibilistic) representation of the mean value of the data in the left part. α-cuts of this fuzzy set are bootstrap-based confidence intervals of the interval valued data; that is, they are the smallest intervals containing at least a fraction $1 - \alpha$ of the data.

preprocessed and aggregated into a characteristic value [8]. The most commonly used aggregation operator is the mean, although many different functions may be used instead [9].

In [10], however, a different approach was used: it was assumed that there exists a true value for the latent variable, but also that this value cannot be precised further than a set that contains it. In this respect, it is widely known that uncertainty in databases encompasses probabilistic and incomplete data, the former being a refinement of the latter, but some categories of imprecise data are hardly addressed in this framework, such as censored or interval-valued data. Imprecise probabilities-based representations are better suited for these problems, because the available knowledge about the possible values of the data by means of families of probability distributions.

Possibilistic representations are a particular case of these, general enough for modeling a wide range of practical problems: incomplete databases can be represented by means of the vacuous belief function (the set of all probability distributions) that models full ignorance, and other types of uncertain data, such as the aforementioned interval-valued or censored data can also be easily represented. Moreover, a possibilistic view of uncertainty is compatible with the use of fuzzy sets for describing partial knowledge about the data, because the contour function of a possibility distribution is a fuzzy set [11]. In this context, α-cuts of fuzzy sets may be linked to confidence intervals about the unknown value of the feature with significance levels $1 - \alpha$ (see Figure 1 and reference [12]). This last property supports the use of intervals or fuzzy data for modelling the following types of uncertain or *low quality* data:

(i) unqualified sets of possible values, such as enumerations, $\{x_1, x_2, \ldots, x_N\}$, or interval-valued data, $[x_1, x_2]$. Note that missing data is represented by intervals spanning the whole domain of the variable. Censored data is represented in a similar way;

(ii) qualified sets of possible values, where each item is associated with a probability value, an upper probability, or an interval of probabilities. These constructs often arise from the fuzzification interfaces of fuzzy rule-based systems. For example, given two membership functions related to the linguistic concepts "Fast" and "Slow," a crisp value "5" might be mapped to a set {0.2/Fast + 0.8/Slow} and an interval "[4, 5]" might be mapped to the interval-valued fuzzy set {[0.15, 0.25]/Fast + [0.75, 0.85]/Slow} or to the fuzzy set {0.25/Fast + 0.85/Slow}, depending on the chosen method.

Following this section, a method for ranking the importance of the fuzzy aggregated metrics in relation to the grading problem is presented that allows applying an arbitrary deterministic or random feature selection algorithm to this problem. In short, each imprecise value in the training database will be regarded as a set of possible values. A standard feature selection algorithm will be repeatedly launched over different selections of the sample that comprise possible instantiations of the data. Each selection gives rise to a different ranking, and all of these will be aggregated into a fuzzy membership function, to which a possibilistic meaning is assigned, as described before. Finally, a ranking between these fuzzy memberships will be defined and used to select a relevant subset of features.

2.1. Random Feature Selection Extended to Vague Data. In the following, the grades and fuzzy aggregated metrics will be regarded as random and fuzzy random variables, respectively. A fuzzy random variable will be regarded as a nested family of random sets:

$$\left(\Lambda_\alpha\right)_{\alpha\in(0,1)}, \tag{1}$$

each one associated to a confidence level $1-\alpha$ [13]. A random set is a mapping where the images of the outcomes of the random experiment are crisp sets. A random variable X is a selection of a random set Γ when the image of any outcome by X is contained in the image of the same outcome by Γ.

For a random variable $X : \Omega \rightarrow \mathbf{R}$ and a random set $\Gamma : \Omega \rightarrow \mathscr{P}(\mathbf{R})$, X is a selection of Γ (written $X \in S(\Gamma)$) when

$$X(\omega) \in \Gamma(\omega) \quad \forall \omega \in \Omega. \tag{2}$$

In turn, a random set can be viewed as a family of random variables (its selections.)

Let be $M + 1$ paired samples $(X_1^k, X_2^k, \ldots, X_N^k)$ and $(Y_1, Y_2, \ldots, Y_N)$, with $k = 1, \ldots, M$, from $M + 1$ standard random variables $X^1, X^2, \ldots, X^M$ and Y. In this particular case, M is the number of metrics and N is the number of students. It will be assumed that all universes of discourse are finite. Let be assumed that a feature selection algorithm is a random mapping between the $M + 1$ paired samples and a permutation σ of $\{1, \ldots, M\}$ that sorts the metrics according to their relevance:

$$\sigma\left(X_1^1, X_2^1, \ldots, X_N^M, Y_1, Y_2, \ldots, Y_N, \omega\right) = (\sigma_1, \ldots, \sigma_M)(\omega), \tag{3}$$

where $p_{ik} = P(\sigma_i = k) = P(\omega \mid \sigma_i(\omega) = k)$ with $i, k = 1 \ldots, M$, is the probability that the kth random variable X^k is ranked as the ith most relevant feature. If the feature selection criterion is deterministic (e.g., a correlation or mutual information-based criterion [14]) then $p_{ik} \in \{0, 1\}$. In other cases, successive launches of the feature selection algorithm over the same sample will produce different permutations (think for instance of random forest-based feature importance measures [15]).

Now let $M + 1$ be fuzzy paired samples $(\widetilde{X}_1^k, \widetilde{X}_2^k, \ldots, \widetilde{X}_N^k)$ and an also paired crisp sample $(Y_1, Y_2, \ldots, Y_N)$ from $M + 1$ fuzzy random variables $\widetilde{X}^1, \widetilde{X}^2, \ldots, \widetilde{X}^M$ and the random variable Y. Let the list of fuzzy numbers $\widetilde{\sigma} = (\widetilde{\sigma}_1, \ldots, \widetilde{\sigma}_M)$ be defined as

$$\mu_{\widetilde{\sigma}_i}(k) = \sup\left\{\alpha \mid P\left(\sigma_i\left(X_1^1, X_2^1, \ldots, X_N^M, Y_1, Y_2, \ldots, Y_N\right) = k\right) \geq \epsilon \; X_i^k \in S\left(\left[\widetilde{X}_i^k\right]_\alpha\right),\ i, k = 1, \ldots, M\right\}, \tag{4}$$

for a given small value ϵ. It will be shown later in this paper that each fuzzy number $\widetilde{\sigma}$ models our incomplete knowledge about the possible ranks of each fuzzy aggregated metric $\widetilde{X}^k$; these metrics will be ordered according to a ranking between fuzzy numbers. In Section 5 a detailed practical case is worked.

3. Genetic Learning of Fuzzy Rules from Imprecise Data in Modeling Problems

It was mentioned that an importance-based ranking of features in fuzzy datasets is also uncertain. In the preceding section, a method for dealing with this imprecision was proposed. A secondary problem arises when the automatic grading system has to be learnt from this vague data; however, there exist machine learning algorithms that can be used with this purpose. In this paper, the fuzzy rule learning algorithm NMIC, introduced in [16], will be used. As this is not a widely known method, it is recalled and briefly described in this section for the convenience of the reader.

Fuzzy models, comprising R rules with the form

$$\texttt{If } x \texttt{ is } A_r \texttt{ then } y \texttt{ is } B_r \texttt{ with weight } w_r, \tag{5}$$

are used, where x and y are the feature and the output vectors, respectively, and A_r are conjunctions of linguistic labels, which in turn are associated with fuzzy sets. B_r is a singleton. The output $\widetilde{Y}$ of the fuzzy model for a fuzzy input $\widetilde{X}$ is defined through the membership function

$$\widetilde{Y}(y) = \sup\left\{\widetilde{X}(x) \mid Y = \frac{\sum_{r=1}^{R} f_r(x)}{\sum_{r=1}^{R} A_r(x)}\right\}, \tag{6}$$

where each function $f_r(x)$ is a product $\beta_r A_r(x)$. $A_r(x)$ is the membership of x to a linguistic expression whose terms are labels of the linguistic variables defined over the input variables, connected by the operators "AND" and "OR." β_r is the product of the centroid of B_r and the weight w_r assigned to the rule.

3.1. A Michigan-Style Genetic Fuzzy Model for Imprecise Data. The NMIC learning method is based on the hypothesis that the best model will comprise rules that are in nondominated sets under confidence and support measures [17]. An extension of the NSGA-II algorithm to fuzzy data is repeatedly launched to obtain Pareto fronts containing nondominated rules in terms of confidence and support. Every front is regarded as a population of a Michigan-type algorithm, where each individual is an antecedent of a fuzzy rule and the whole population is a fuzzy model. Individuals (rules) are weighted and selected by means of a procedure called *SVD select* (explained later in this section). The pseudocode of the NMIC algorithm is shown in Pseudocode 1. The codification and genetic operators are described in [16].

The fitness of an individual has three components: the support of the antecedent of the rule, its precision (dual concept of the confidence in classification), and the weight of the rule. The support is the fuzzy arithmetic-based sum of the memberships of the antecedent for all the points in the sample. The precision is understood as the inverse of the (fuzzy) variance of the examples covered by the rule, that is, all examples in the dataset, weighed by their memberships to the antecedent of the rule.

The following expression is used to compute the precision p of a rule (the lower the better):

$$\bigoplus\left[\widetilde{Y}_n \ominus pA_m\left(\widetilde{X}_n\right)\right]^2 \bigotimes A_m\left(\widetilde{X}_n\right) \quad \exists n \mid A_m\left(\widetilde{X}_n\right) > 0 \qquad \bigoplus\left[\widetilde{Y}_n\right]^2 \quad \text{otherwise}, \tag{7}$$

```
Initialize P
Evaluate confidence and support in P
SVD select P
Create Intermediate Population Q
while iter ≤ maxiter
    Evaluate confidence and support in P + Q
    SVD select P + Q
    non dominated sort P + Q
    compute crowding distance P + Q
    P ← selection of P + Q
    SVD select P
    non dominated sort P
    Create Intermediate Population Q
end while
Output the nondominated elements of P
```

Pseudocode 1: Pseudocode of the NMIC algorithm.

where p is the solution to the weighted least squares problem defined by the centers $\overline{X}_n$ of the data:

$$p = \frac{\sum_{n=1}^{N} Y_n A_m\left(\overline{X}_n\right)^2}{\sum_{n=1}^{N} A_m\left(\overline{X}_n\right)^3}. \tag{8}$$

Furthermore, observe that the cooperation of the rules only arises if the sum of the fitness values of the individuals is comonotonic with the fitness of the rule base they form. Otherwise, the genetic evolution would not improve the error or the model. The proposed precision and support measures are not enough to achieve cooperation by themselves, as each rule in the population models the part of the space covered by its antecedent, but nothing prevents that more than one rule covers the same area while other areas are uncovered. Therefore, a *SVD select* procedure is introduced that consists in assigning each rule in the Pareto front a weight, with the purpose of obtaining a compact rulebase, while at the same time these weights achieve the best matching between the data and the model.

These weights will be obtained by least squares. Let A be the matrix of the memberships of the antecedents of all rules in the Pareto front, at the centerpoint of the inputs. Let Y be a column vector with the centers of the desired outputs of the model, and let W be another column vector formed by the weights of these rules, those that we want to obtain. The assignment of weights that minimizes the error (and therefore solves the cooperation problem) is

$$K = \left(A^t A\right)^{-1} A^t Y \tag{9}$$

provided that the rank r_A of A coincides with its number of columns, the number of individuals in the Pareto front In most cases, r_A is lower than this; therefore, $C = A^t A$ does not have inverse. The common solution to this problem is to apply a singular value decomposition $C = UDV^t$, then cancel the eigenvalues of D lower than 10^{-6} times the highest, and take the inverse of the remaining ones, and by last define

$$K = \left[V \cdot \left(\frac{1}{D}\right) \cdot U^t\right] A^t Y. \tag{10}$$

While this assignment solves the cooperation problem, it does not solve the competition problem, because the redundant rules are not discarded. The value of K in the preceding equation is that of minimum norm, but what we really need is the definition of the matrix $(1/D)$ that produces *the most sparse* definition of K, not that with the lowest weights. For example, observe that the definition in (10) will assign the same weight to identical rules, but we want one of them to take all the credit. It is easy to purge the duplicated rules, but it is difficult to remove rules that are (almost) linear combination of others in the Pareto front.

As a matter of fact, the number of individuals we want to assign weights different than zero is the same as the number of not cancelled eigenvalues in D. Observe that the columns of the matrix U associated with null eigenvalues form a basis of the nullspace of A and that means that each individual (each column of A), if expressed in the base formed by the columns of U, will have at most r_A nonnull coefficients; that is, we will not find more than r_A independent elements in the Pareto front. Therefore, we know that we can set to zero the weights of all the rules but r_A. The problem here is how can we determine which columns of A will be set to nonzero weight.

The solution is trivial, but computationally intensive. Since the minimum norm is not searched but we are interested in an sparse matrix K (thus the number of fuzzy rules is kept as small as possible), in [16] it was proposed not to add a regularization term, that is, not to take the inverse of $A^t A + \lambda I$, but that the eigenvalues of all the submatrices A' of A formed by removing only one of its columns are computed. When a submatrix A' whose non null eigenvalues are the same as those of A is found, the column is removed and the process restarted from A'. At the end of the process, a matrix with r_A columns and full rank is obtained and (9) can be applied.

4. Numerical Results

In this section a real-world problem is worked; thus, the benefits of the proposed approach can be compared to the results of alternative approaches. A brief state of the art in software metrics is included first; thus, the experimental setup can be better understood. The experimental design and a description of the field work follow, and the section ends with some compared numerical results.

4.1. Software Metrics Used in This Experimentation. As mentioned, software metrics will be used to measure the quality of the students' submitted source code. It will be assumed that better coding leads to higher scoring, but static analysis is also used to obtain additional insights into the student's programs. There are many comprehensive surveys describing the best known software metrics; see for instance [18] or [19].

According to [20], the most relevant software metrics in this context are as follows.

(i) Number of lines of code: a naive measurement of the code size.

(ii) Ratio between lines of comments and lines of code: a measurement of code documentation.

(iii) Halstead metrics [21]: these metrics have been reported to be useful to evaluate students programs [22].

(iv) Cyclomatic number [23]: often related to *complexity* measures and thus usually referred to as cyclomatic complexity number, but there is not a full agreement about the subject [22]. The cyclomatic number is a graph theory concept that has been translated to software because a program can be modeled as a strongly connected graph [24].

Most of the software analysis tools (http://www.webappsec.org/) provide these features and many other indexes. As the number of software metrics increases, the odds that some of them are closely related or overlap in the property being measured increases too. For instance, it is hard to conceive an increase of the cyclomatic number without a simultaneous increase of the lines of code. Moreover, an increase in the value of a metric is not always consistent with an improvement in the code quality. For instance, it may happen that the complexity of a given problem solution is too low, because the student is unwinding a loop, or the complexity may be greater than expected because the student is using a quadratic algorithm for a problem with linear solution. In either case, the dependence between the metric value and the desired solution is highly problem dependent. Because of this, some researchers propose to use feature selection techniques for finding the best software metrics for the problem at hand. In [25], a stochastic procedure is employed to select the subset of quantitative measures that bring out the best software quality prediction. Another example is [26], where eighteen filter-based feature selection procedures are tested against sixteen software datasets, in this case searching for fault prone modules.

In this paper, a set of software metrics that are commonly used to measure student's source code properties [7] with some additions from [27] for extracting information related with style and structure will be used, and feature selection techniques will be developed that help to select the best set of metrics for each programming concept.

4.2. Experimental Design and Field Work. Forty-six volunteering students from the first course of an Engineering Degree in Computer Science at Oviedo University, Spain, participated in this study. The Python programming language was used. Students were allowed to upload as many source code files as they wished, ranging from none to more than a solution for each problem. 800 files were uploaded. Seven programming concepts were studied: Standard I/O, Conditionals, While loop, For loop, Functions, File I/O, and Lists. The evaluation of the students comprised both theoretical and practice skills, with two exams each, at the midterm and at the end of the term. The uploaded exercises were not part of the exams and had no impact on the final grading. 23 software metrics and properties were measured for each source file; thus, the feature selection stage has to choose between 161 different combinations of programming concept and software metric.

Table 1: Most relevant pairs of software metric/programming concept for the field study mentioned in the text. Observe that the programming concepts in this subset of metrics are "Conditional" and "File I/O," meaning that the correlation between the scores and "While loop," "Functions," "Lists," or "Standard I/O" is weaker.

Programming concept	Description of the metric	Rank 99%	Rank 80%
Conditional	COCOMO SLOC [29]	1 ± 0	11 ± 10
Conditional	Number of tokens	2 ± 0	8.5 ± 7.5
Conditional	Code ratio	4 ± 1	25 ± 24
File I/O	Number of characters	4 ± 1	11 ± 8
Conditional	Number of lines	7 ± 1	21 ± 19
Functions	Number of characters	4 ± 1	43 ± 37
Conditional	Number of keywords	7.5 ± 1.5	36 ± 33
Conditional	Number of comments	17 ± 6	48 ± 44
File I/O	Ratio of comments	17 ± 6	48 ± 44
File I/O	McCabe Complexity [23]	17 ± 9	30 ± 24
File I/O	Number of blocks	17 ± 9	31 ± 25

The feature selection algorithm to be extended is based on the random forest feature importance measures [15]. A fuzzy rank (see [28]) was used to sort the fuzzy rankings.

The most relevant pairs of found software metric-programming concept are displayed in Table 1. This is the subset for which the best model attained a minimum error; details are given later in this section. Two α-cuts of their fuzzy ranks are also given. In Figure 4(b), the whole membership function of the fuzzy rank is plotted for two particular metrics: "COCOMO SLOC" [29] and "McCabe Complexity" [23]. It is remarked that a high degree of overlapping between the memberships of the fuzzy ranks was found in this study.

Observe that the only programming concepts in this subset of metrics are "Conditional" and "File I/O," meaning that the correlation between the scores and "While loop," "Functions," "Lists," or "Standard I/O" is weaker. This is an unexpected result, since the latter five programming concepts intuitively convey more information about the grading than the two former ones, but this fact can be explained nonetheless if the particular circumstances of this experimentation are taken into account. Students uploaded more exercises at the beginning of the course than at the end; thus, the quality of the information about their capabilities is better for the initial problems. The ranking algorithm has therefore discovered that apparently relevant information may be discarded without lowering the prediction capabilities of the model and suggested to evaluate the performance of a student with the eleven metrics shown in Table 1.

In Figure 2, the rank of the most relevant metrics, according to the proposed algorithm, is graphically displayed.

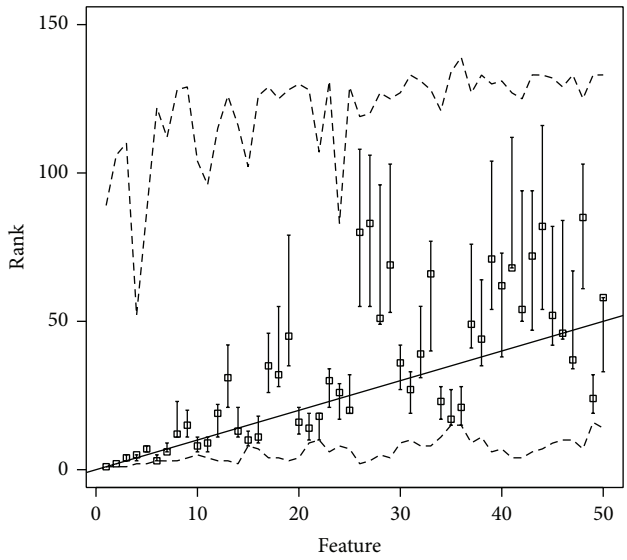


Figure 2: Supports (dashed lines), modal points (bars), fuzzy ranks (abscissa), and crisp ranks (ordinate of the squares) of the 50 most relevant metrics. Those metrics whose square is plotted below the diagonal line occupy a more relevant position under the fuzzy rank.

Supports (dashed lines), modal points (bars), fuzzy ranks (abscissa), and crisp ranks (ordinate of the squares) of the 50 most relevant metrics are displayed. Observe that those metrics whose square is plotted below the diagonal line occupy a more relevant position under the fuzzy rank than they were assigned by the crisp feature selection algorithm. Squares over the diagonal line, on the contrary, are assigned more weight by the crisp algorithm than they are with the fuzzy extension.

From a methodological point of view, the proposed technique is robust and the available information is better exploited with the combination of the fuzzy feature selection and NMIC than it is with standard feature selection and model learning algorithms. To prove this fact, regression trees [30], neural networks [31], support vector machines [32], random forests [33], and the NMIC algorithm were launched over subsets sweeping the range between 10 and 20 metrics, found by both the fuzzy extended feature selection algorithm and the original crisp version operating on the centerpoints of the aggregated data. In Table 2 these results are jointly displayed. In Figure 3, test errors corresponding to the selection of the most relevant variables with random forest feature importance measures (applied to the centerpoints of the fuzzy data) are drawn with dashed lines. The proposed extension of the same feature selection method to fuzzy data, followed by a learning with the same centerpoints for Regression Trees, Neural Networks, Support Vector Machines, and Random Forest, but the whole fuzzy data for NMIC, are drawn with solid lines. Observe that if the number of features associated with the lowest test error is chosen as a quality index, the proposed extension improved the accuracy of the grading system in 4 of 5 cases (all but the Regression Tree, with incidentally attained the worst results).

Table 2: Test error or the different regression methods for feature sets ranging from 10 to 20 variables.

Features	Multilayer Perceptron	SVM	Regression Tree	Random Forest	NMIC
10	7.703	7.185	8.574	6.671	7.011
11	7.729	7.093	8.574	7.285	**6.321**
12	7.823	7.128	8.185	6.802	6.629
13	**7.451**	6.911	8.185	6.742	6.678
14	7.641	6.914	8.185	6.854	7.073
15	7.472	6.670	**8.084**	7.617	7.056
16	7.661	6.652	8.185	**6.576**	7.236
17	7.785	6.791	8.185	7.716	6.764
18	7.838	6.728	8.703	7.285	8.399
19	8.195	**6.445**	8.703	7.256	7.256
20	8.672	6.648	8.909	8.357	7.451

The combination of the NMIC algorithm with fuzzy data was consistently better in all cases (statistically relevant results, according to Friedman/Wilcoxon tests, P value better than 0.05). In Figure 4(a) a set of boxplots is drawn, showing the statistical differences between the test error of the combination of Neural Networks (NN), Support Vector Machines (SVM), Regression Trees (RT), Random Forests (RF), and NMIC with the feature set computed as described in this paper. This last graphic is intended to show that NMIC exploits the imprecision in the information better than the alternatives, demonstrating that the fuzzy aggregation loses less information than the alternatives and also that the proposed method is able to exploit this extra information.

4.3. Comparison to Other Automatic Grading Systems. The closest to this proposal automatic grading algorithm found in the literature is [7]. In this reference, an online judge/tutor is presented. The purpose of that system is to provide the students of an introductory Computer Programming subject with a quality measure of the submitted code and also with some recommendations to improve it. The students interact with the system via an Eclipse plugin, comparing their solutions against the teacher's ones. The comparison has two parts: first a structural comparison is done; then an evaluation of the correctness using a set of test cases is performed.

This paper is related to the first task, where a fuzzy representation of the algorithm's structure is used. For each exercise the teacher writes his/her best solution and then computes several software metrics (McCabe cyclomatic complexity, COCOMO SLOC, etc.). Three fuzzy sets (Low, Normal, and High) are defined in order to allow certain degree of deviation with increasing penalization as the student's values depart from the teacher's ones. For each software metric and problem, the teacher manually defines each set. Later, the same software metrics are computed for the student's submitted source code. The aggregated membership to teacher's fuzzy sets values is taken as a student's solution quality measurement. The aggregation function has several manually selected weights to adjust the relative importance of the different software metrics for each assignment.

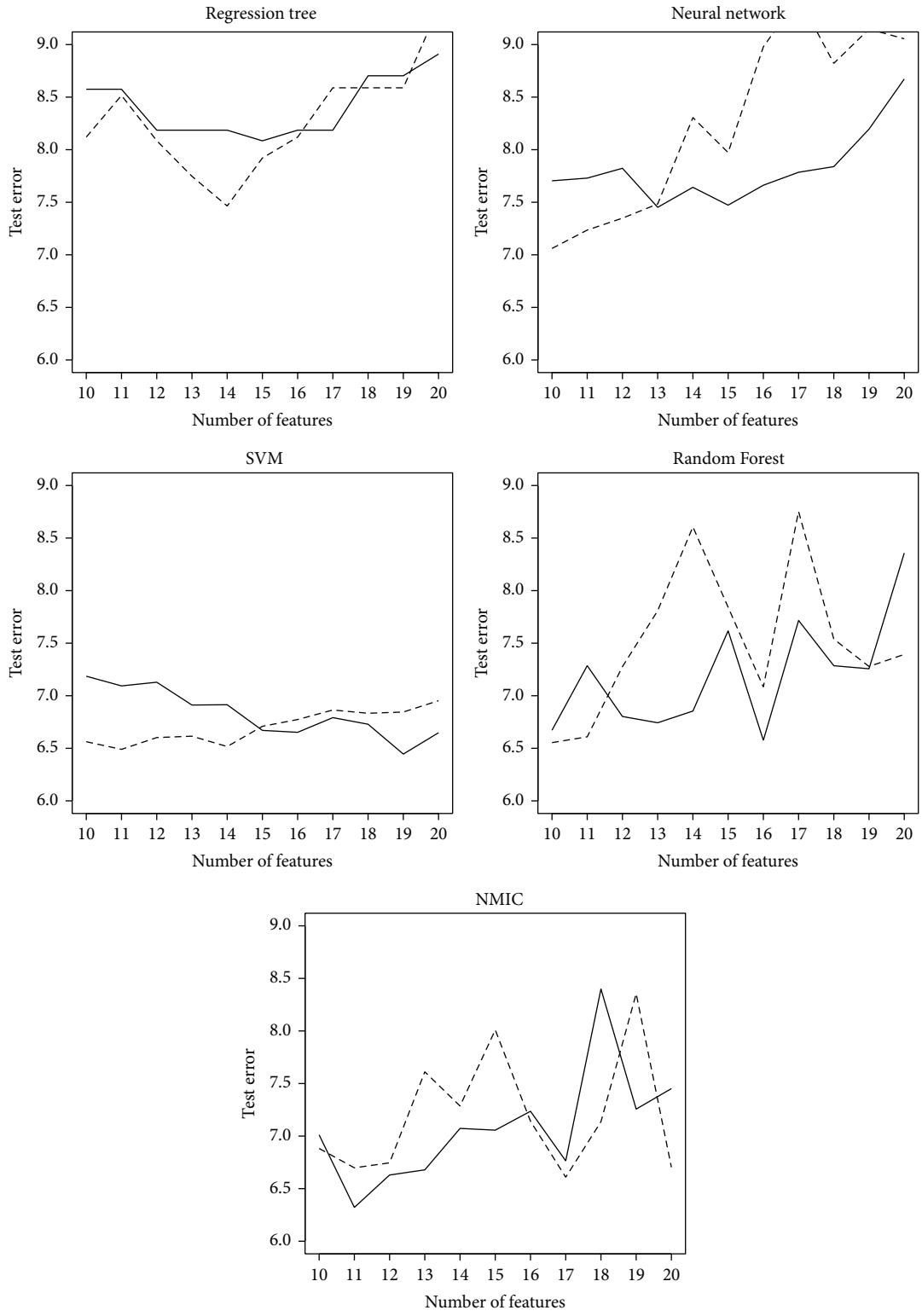


Figure 3: Test errors for feature subsets of sizes 10 to 20. The results associated to a random forest feature importance-based selection, applied to the centerpoints of the fuzzy data, are drawn with dashed lines. The extension of this method to fuzzy data, followed by a learning with the same centerpoints for Regression Trees, Neural Networks, Support Vector Machines, and Random Forest, but the whole fuzzy data for NMIC, are drawn with solid lines.

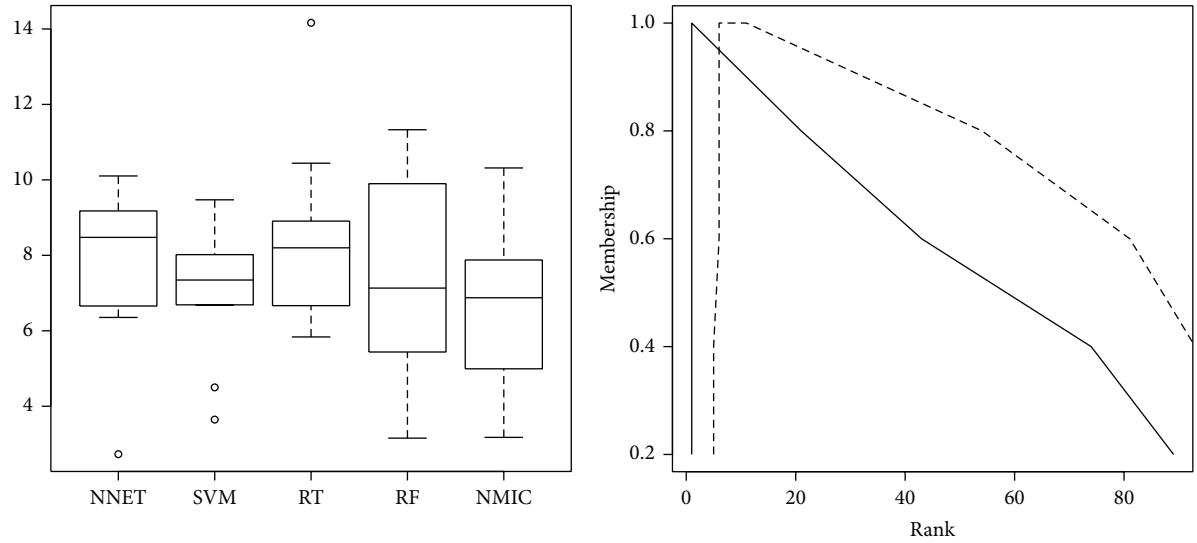


Figure 4: (a) Boxplot showing the statistical differences between the test error of the combination of Neural Networks (NN), Support Vector Machines (SVM), Regression Trees (RT), Random Forests (RF), and NMIC with the feature set computed as described in this paper. NMIC exploits the imprecision in the information better than the alternatives. (b) Membership functions of the ranks of the first (solid) and 10th (dashed) features, that is, COCOMO SLOC and McCabe complexity.

The main two differences between [7] and the method described in this paper are as follows.

(1) Only one submission is allowed in [7] for each assignment, and the solution to the problem must be provided by the instructor. In this paper, the metrics of a set of papers can be combined into a fuzzy value, allowing for multiple submissions, missing data, and partially incomplete assignments.

(2) The set of metrics is chosen beforehand in [7] but chosen from a pool of metrics in this proposal.

In order to compare both approaches, each of the submitted source codes was measured in five of the seven software metrics proposed in [7]; two of them were not applicable to Python and had to be removed. Some additional changes were effected: since the method in this paper supports using different source files vinculated to the same task, each source file had to be matched with a solution to the problem chosen by hand and also newly coded by the instructor.

A 10-cv experimental design was used, whose results are graphically shown in Figure 5. It can be stated that the proposed system is more accurate than [7]. This assert is supported by a statistical test where the hypothesis "both methods produce the same results" was rejected at the 95% confidence level (Wilcoxon test, paired data, P value = 0.036, alternative hypothesis: true location shift is greater than 0).

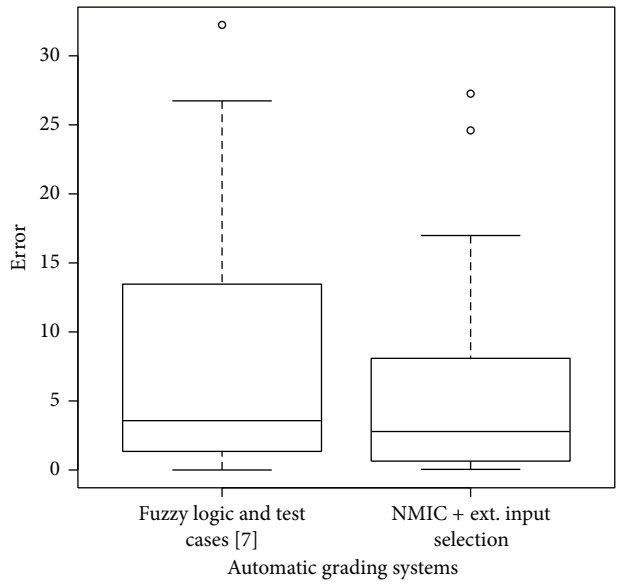


Figure 5: Boxplot comparing the test results (10-cv) of two different automatic grading systems: Fuzzy Logic and Test Cases [7], and NMIC + Extended Input Selection (this paper), showing a statistically relevant difference between both groups.

5. Concluding Remarks and Future Work

A method for ranking software metrics according to their relevance in an automatic grading system has been proposed. The main innovation of the new method lies in the development of a set of techniques that can make use of a fuzzy aggregation of the information contained in a variable number of exercises about the same learning subject.

From a methodological point of view, the new algorithm is a solid alternative. The combination of a learning algorithm for vague data and the extended feature selection proposed in this paper was shown to make a better use of the imprecision in the information than any of the alternatives, demonstrating

that the fuzzy aggregation keeps valuable information and also that the proposed method is able to exploit this. On the other hand, from the point of view of the automated grading techniques, it has been found that the most informative metrics are some measures of the cost and complexity of the code, followed by indicators related to the code size and quality of the documentation. However, there is still a margin for improving this knowledge, as the number of students participating in the study was small and further work is needed to build a larger corpus of hand-graded assignments.

Conflict of Interests

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

Acknowledgments

This work was supported by the Spanish Ministerio de Economía y Competitividad under Project TIN2011-24302, including funding from the European Regional Development Fund.

References

[1] D. Arnow and O. Barshay, "On-line programming examinations using Web to teach," in *Proceedings of the 4th Annual SIGCSE/SIGCUE ITiCSE Conference on Innovation and Technology in Computer Science Education (ITiCSE '99)*, pp. 21–24, 1999.

[2] K. A. Reek, "A software infrastructure to suppor t introductory computer science courses," in *Proceedings of the 27th SIGCSE Technical Symposium on Computer Science Education (SIGCSE '96)*, K. J. Klee, Ed., pp. 125–129, ACM, New York, NY, USA, 1996.

[3] A. Kurnia, A. Lim, and B. Cheang, "Online judge," *Computers & Education*, vol. 36, no. 4, pp. 299–315, 2001.

[4] B. Cheang, A. Kurnia, A. Lim, and W. Oon, "On automated grading of programming assignments in an academic institution," *Computers and Education*, vol. 41, no. 2, pp. 121–131, 2003.

[5] T. Wang, X. Su, P. Ma, Y. Wang, and K. Wang, "Ability-training-oriented automated assessment in introductory programming course," *Computers and Education*, vol. 56, no. 1, pp. 220–226, 2011.

[6] M. Vujosevic-Janicic, M. Nikolić, D. Tošić, and V. Kuncak, "Software verification and graph similarity for automated evaluation of students assignments," *Information and Software Technology*, vol. 55, no. 6, pp. 1004–1016, 2013.

[7] F. Jurado, M. Redondo, and M. Ortega, "Using fuzzy logic applied to software metrics and test cases to assess programming assignments and give advice," *Journal of Network and Computer Applications*, vol. 35, no. 2, pp. 695–712, 2012.

[8] J. Casillas, F. Martínez-López, and F. Martínez, "Fuzzy association rules for estimating consumer behaviour models and their application to explaining trust in internet shopping," *Fuzzy Economic Review*, vol. 9, no. 2, pp. 3–26, 2004.

[9] J. Casillas and F. J. Martínez-López, "Mining uncertain data with multiobjective genetic fuzzy systems to be applied in consumer behaviour modelling," *Expert Systems with Applications*, vol. 36, no. 2, pp. 1645–1659, 2009.

[10] L. Sanchez, I. Couso, and J. Casillas, "Genetic learning of fuzzy rules based on low quality data," *Fuzzy Sets and Systems*, vol. 160, no. 17, pp. 2524–2552, 2009.

[11] H. Prade and D. Dubois, "Fuzzy sets—a convenient fiction for modeling vagueness and possibility," *IEEE Transactions on Fuzzy Systems*, vol. 2, no. 1, pp. 16–21, 1994.

[12] I. Couso and L. Sanchez, "Higher order models for fuzzy random variables," *Fuzzy Sets and Systems*, vol. 159, no. 3, pp. 237–258, 2008.

[13] I. Couso, S. Montes, and P. Gil, "The necessity of the strong α-cuts of a fuzzy set," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 9, no. 2, pp. 249–262, 2001.

[14] R. Battiti, "Using mutual information for selecting features in supervised neural net learning," *IEEE Transactions on Neural Networks*, vol. 5, no. 4, pp. 537–550, 1994.

[15] Y. Saeys, T. Abeel, and Y. Peer, "Robust feature selection using ensemble feature selection techniques," in *Machine Learning and Knowledge Discovery in Databases*, W. Daelemans, B. Goethals, and K. Morik, Eds., vol. 5212 of *Lecture Notes in Computer Science*, pp. 313–325, Springer, Berlin, Germany, 2008.

[16] L. Sánchez and J. Otero, "Learning fuzzy linguistic models from low quality data by genetic algorithms," in *Proceedings of the IEEE International Conference on Fuzzy Systems*, pp. 1–6, July 2007.

[17] H. Ishibuchi, O. Kuwajima, and Y. Nojima, "Relation between pareto-optimal fuzzy rules and pareto-optimal fuzzy rule sets," in *Proceedings of the 1st IEEE Symposium of Computational Intelligence in Multicriteria Decision Making (MCDM '07)*, pp. 42–49, IEEE, Honolulu, Hawaii, April 2007.

[18] M. Abdellatief, A. B. M. Sultan, A. A. A. Ghani, and M. A. Jabar, "A mapping study to investigate component-based software system metrics," *Journal of Systems and Software*, vol. 86, no. 3, pp. 587–603, 2013.

[19] B. Kitchenham, "What's up with software metrics?—a preliminary mapping study," *Journal of Systems and Software*, vol. 83, no. 1, pp. 37–51, 2010.

[20] I. Samoladas, I. Stamelos, L. Angelis, and A. Oikonomou, "Open source software development should strive for even greater code maintainability," *Communications of the ACM*, vol. 47, no. 10, pp. 83–87, 2004.

[21] M. Halstead, *Elements of Software Science*, Elsevier, North-Holland, 1975.

[22] A. Abran, *Software Metrics and Software Metrology*, Wiley-IEEE Computer Society Press, 2010.

[23] T. J. McCabe, "A complexity measure," *IEEE Transactions on Software Engineering*, vol. SE-2, no. 4, pp. 308–320, 1976.

[24] A. H. Watson, *McCabe Complexity, Software Development Systems Management Development*, Auerbach, 1995.

[25] N. Pizzi, A. Demko, and W. Pedrycz, "The analysis of software complexity using stochastic metric selection," *Journal of Pattern Recognition Research*, vol. 6, no. 1, pp. 19–31, 2011.

[26] T. M. Khoshgoftaar, K. Gao, and A. Napolitano, "A comparative study of different strategies for predicting software quality," in *Proceedings of the 23rd International Conference on Software Engineering and Knowledge Engineering (SEKE '11)*, pp. 65–70, July 2011.

[27] P. Sallis, A. Aakjaer, and S. MacDonell, "Software forensics: old methods for a new science," in *Proceedings of the International Conference on Software Engineering: Education and Practice (SEEP '96)*, pp. 481–484, IEEE Computer Society, Washington, DC, USA, 1996.

[28] G. Bortolan and R. Degani, "A review of some methods for ranking fuzzy subsets," *Fuzzy Sets and Systems*, vol. 15, no. 1, pp. 1–19, 1985.

[29] B. Boehm, B. Clark, E. Horowitz et al., "Cost models for future softwa re life cycle processes: COCOMO 2.0.," *Annals of Software Engineering*, vol. 1, no. 1, pp. 57–94, 1995.

[30] L. Breiman, L. J. Friedman, A. Olshen, and C. Stone, *Classification and Regression Trees*, Wadsworth, 1984.

[31] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Prentice Hall, New York, NY, USA, 2nd edition, 1998.

[32] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and Computing*, vol. 14, no. 3, pp. 199–222, 2004.

[33] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.