

## Research Article

# A New Adaptive Hungarian Mating Scheme in Genetic Algorithms

Chanju Jung,<sup>1</sup> Yong-Hyuk Kim,<sup>2</sup> Yourim Yoon,<sup>3</sup> and Byung-Ro Moon<sup>1</sup>

<sup>1</sup>Department of Computer Science and Engineering, Seoul National University, 1 Gwanak-ro, Gwanak-gu, Seoul 08826, Republic of Korea

<sup>2</sup>Department of Computer Science and Engineering, Kwangwoon University, 20 Kwangwoon-ro, Nowon-gu, Seoul 01890, Republic of Korea

<sup>3</sup>Department of Computer Engineering, Gachon University, 1342 Sengnamdaero, Sujeong-gu, Seongnam-si, Gyeonggi-do 13120, Republic of Korea

Correspondence should be addressed to Byung-Ro Moon; moon@snu.ac.kr

Received 10 December 2015; Revised 1 March 2016; Accepted 17 March 2016

Academic Editor: Lu Zhen

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

In genetic algorithms, selection or mating scheme is one of the important operations. In this paper, we suggest an adaptive mating scheme using previously suggested Hungarian mating schemes. Hungarian mating schemes consist of maximizing the sum of mating distances, minimizing the sum, and random matching. We propose an algorithm to elect one of these Hungarian mating schemes. Every mated pair of solutions has to vote for the next generation mating scheme. The distance between parents and the distance between parent and offspring are considered when they vote. Well-known combinatorial optimization problems, the traveling salesperson problem, and the graph bisection problem are used for the test bed of our method. Our adaptive strategy showed better results than not only pure and previous hybrid schemes but also existing distance-based mating schemes.

## 1. Introduction

Mating scheme or selection is one of the important operations in genetic algorithm (GA). Most operations in GAs are closely related to the performance. These operations interact with each other. A small change of a key operation may cause a dramatic change in result. Ochoa et al. [1] presented that assortative mating is a good choice when the mutation rate is high, while disassortative mating is a good choice when the mutation rate is low.

In mating or selection stage, the methods of mating are classified into three groups. The first one gives preference to similar solutions [2, 3]. This method focuses on exploitation. It assumes that similar solutions have higher chance to make better solutions because they use discovered good schemata.

The second one is dissimilar mating. This method focuses on exploration, and it tries to evade a premature convergence

and a fast diversity consumption of similar mating. It is realized mostly by a restriction. Ramezani and Lotfi [4] restricted mating between family solutions such as parent and offspring. They solved function optimization problems and obtained good results. Fernandes et al. [5] reported that dissimilar mating outperforms a simple GA or similar mating in vector quantization problem.

The last group tries to find a better mating scheme by combining two or more mating schemes. Ishibuchi et al. [6, 7] considered the number of mating candidates. They used their parameters to control similarity of a mated solution. They proposed a method of changing its controlling parameter at a middle of running. They also presented that the changing parameter made better results (nearer to Pareto-optimal solutions) than fixed parameters.

Galán et al. [8] proposed a mating scheme where each individual has its mating preference value to balance exploitation and exploration. A low value of mating preference

makes a match between solutions close to each other, while a high mating preference makes a match between solutions far from each other. The preference is inherited or mutated like a normal gene. They tested their scheme in various environments of function optimization. They showed that their scheme outperformed random mating or a scheme with a fixed preference value.

In this paper, we propose a new adaptive Hungarian mating scheme. Our hybrid scheme adaptively selects a pure scheme for the next generation with voting. The traveling salesman problem (TSP) and the graph bisection problem are the test problems of our adaptive scheme. We summarize our contributions as follows. (i) We propose a new adaptive mating scheme, (ii) we provide a reasonable explanation for our scheme, (iii) we show that our adaptive hybrid scheme is more effective than any pure Hungarian scheme and a previous simple hybrid scheme for the two test problems, and (iv) finally we show that our scheme changes its action as the size of problem space changes.

In comparison with the preliminary version of this paper [9], we provide the following extended information: (i) empirical observations and theoretical proof of the recommended parameter settings, (ii) detailed results to show the search behaviors of our algorithm, (iii) comparison of solution qualities with existing distance-based mating methods, and (iv) analysis on computation time of the proposed method and existing ones.

The remainder of this paper is divided into five parts. Previously suggested Hungarian mating schemes and their hybrid strategy are presented in Section 2. Motivation of this study is described in Section 3. Section 4 explains our adaptive hybrid mating scheme. In Section 5, we give experimental results and analyze the results. In Section 6, we draw conclusions.

## 2. Background

*2.1. Hungarian Method.* Assume a weighted complete bipartite graph with bipartition  $(X, Y)$ :  $X = x_1, x_2, \dots, x_{n/2}$ ,  $Y = y_1, y_2, \dots, y_{n/2}$ , and each edge  $(x_i, y_j) \in X \times Y$  has its weight  $w_{ij}$ . For optimal matching, we consider a problem of finding a maximum (or minimum) weight bijective matching as follows:

$$\max_{\sigma \in \Sigma_{n/2}} \left( \sum_{i=1}^{n/2} w_{i\sigma(i)} \right) \text{ or } \min_{\sigma \in \Sigma_{n/2}} \left( \sum_{i=1}^{n/2} w_{i\sigma(i)} \right), \quad (1)$$

where  $\sigma$  is a permutation of size  $n/2$ .

Optimum matching can be computed by the Hungarian method [14]. It can be computed in  $O(n^3)$  time [15]. Avis [16] has suggested an approximation algorithm of  $O(n^2)$  time. The Hungarian method has been utilized in various studies [17–20].

*2.2. Hungarian Mating Scheme.* Recently, we suggested Hungarian mating schemes [10]. We divided population into the same number of female and male individuals. The Hungarian method is applied to mating. In each problem, a suited

distance metric is adopted. It will be described in the next subsection.

Figure 1 describes the mating schemes. Distribution of an example population is illustrated in Figure 1(a). Thirty individuals are shown: fifteen plus symbols (+) are female solutions, and fifteen filled circles (•) are male ones. The Euclidean distance is adopted as the distance metric. Figure 1(b) shows a result of random mating scheme which is called “RAND.” The result of minimizing the sum of distances is displayed in Figure 1(c). The result is obtained from the following formula with the Hungarian method:

$$\min_{\sigma \in \Sigma_{n/2}} \sum_{i=1}^{n/2} d(m_i, f_{\sigma(i)}), \quad (2)$$

where  $m_i$ 's are male solutions,  $f_{\sigma(i)}$ 's are their mated female ones, and  $d$  is the Euclidean distance function. The scheme in Figure 1(c) will be called “NEAR” in this study. Figure 1(d) shows the result of maximizing the sum of distances. The result is obtained from the following formula with the Hungarian method:

$$\max_{\sigma \in \Sigma_{n/2}} \sum_{i=1}^{n/2} d(m_i, f_{\sigma(i)}). \quad (3)$$

This scheme is called “FAR.”

NEAR method extremely concentrates on exploitation, while FAR method extremely focuses on exploration. In bijective mating, NEAR method minimizes the sum of distances. It is an extreme strategy to decrease diversity and use inherent parts of solutions. Similarly, FAR method is an extreme method to preserve diversity.

*2.3. Test Problems and Distance Metric.* Our test problems are TSP and the graph bisection problem. In TSP, a complete undirected graph  $G$  is given. Each edge in  $G$  has a nonnegative weight. The objective of the problem is to find a minimum tour that passes through all the vertices (i.e., Hamiltonian cycle) of  $G$ .

Assume an undirected graph  $G = (V, E)$ :  $V$  is a vertex set and  $E$  is an edge set.  $K$ -way partitioning is defined as partitioning the vertex set  $V$  into  $K$  disjoint subsets  $C_1, C_2, \dots, C_K$ . A  $K$ -way partition is said to be balanced if the difference of cardinalities between the largest and the smallest subsets is zero or one. The cut size of a partition is the number of edges with endpoints in different subsets of the partition. The  $K$ -way partitioning problem is the problem of finding a  $K$ -way balanced partition with minimum cut size. In this paper, we set  $K$  to be equal to two and call this problem “graph bisection.”

The distance metric is the same as that used in our previous study [10]. In TSP, for the phenotype distance metric, the quotient swap distance [20] was used. The quotient swap distance is defined as the smallest one among swap distances. The swap distance between  $X$  and  $Y$  is the minimum number of swaps to make  $X$  be equal to  $Y$ . In other words, between  $X$  and every shifted  $Y$  the minimum value of the swap distance is the quotient swap distance between  $X$  and  $Y$ .

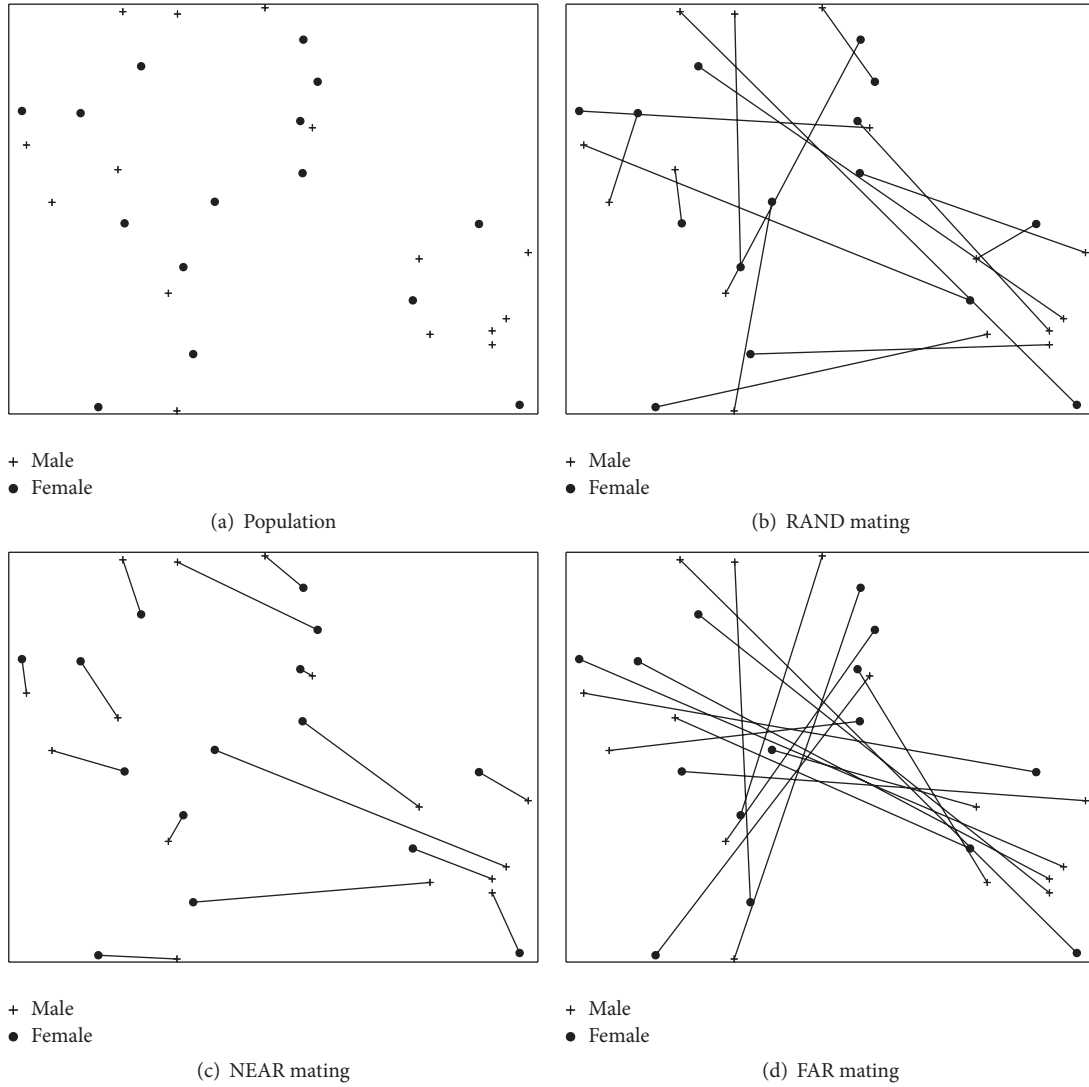


FIGURE 1: Hungarian mating schemes [10].

In graph bisection, for the phenotype distance metric, the quotient Hamming distance [20] was used. Similar to the quotient swap distance, the quotient Hamming distance is defined as the smallest one among Hamming distances. The Hamming distance between two strings is defined by the number of different positions at which the corresponding symbols are different. In other words, the quotient Hamming distances between  $X$  and  $Y$  are the smaller value of the Hamming distance between  $X$  and  $Y$  and that between  $X$  and  $\bar{Y}$ .

### 3. Motivation

We reported that the best Hungarian mating scheme varies according to problems and their sizes [10]. We proposed a simple hybrid scheme of changing the mating scheme from NEAR to RAND at the  $m$ th generation in TSP, where  $m$  is the number of cities. The hybrid scheme for graph

bisection changes mating scheme from RAND to FAR at the 100th generation. Parameters  $m$  and 100 are based on some empirical observation. But the hybrid scheme also has a weakness. The switching time before running of GAs was predetermined. So it is hard to apply the method to new problems or instances.

Galán et al. [8] reported that a self-adaptive mating scheme can be better than traditional random mating and their best-first mating and best-last mating. In the best-first mating, each solution pairs up with its nearest one in the order from the best solution to the worst one. In contrast, in the best-last mating, each solution pairs up with its farthest one in the order from the best solution to the worst one. The best-first mating resembles NEAR method as the best-last mating resembles FAR method. NEAR and FAR are extreme cases of mating. The ideal mating scheme may exist in some middle point of NEAR and FAR as Galán et al. [8] showed in function optimization.

Suggesting a new adaptive hybrid scheme of the Hungarian mating schemes is our main goal. A good scheme may (i) work irrespective of problems or instances, (ii) select its action as environment changes, and (iii) show better results than consistently applied pure scheme. A new scheme with these features will be proposed.

In this paper, we show the influence of the proposed mating scheme, not aiming to beat the state-of-the-art results of TSP and graph bisection. Local optimization may play an important role in making practical solutions. We do not use any local optimization. Instead, we concentrate on balance between exploration and exploitation with a new mating scheme.

## 4. Proposed Method

**4.1. Voting Rules.** We assume the same number of male and female solutions as Goh et al. did in [21]. In each generation, our method selects FAR, RAND, or NEAR for the next generation. Our method does not simulate three schemes as they are. Instead, the appropriate scheme is adaptively adopted. For that, a mating scheme for the next generation is selected with majority voting. Every matched pair has to vote. Our crossover operator generates two offspring, and their gender is randomly assigned. The voting is carried out after mutation. So our voting algorithm compares two parents and two offspring after mutation. The rules of voting are described by Algorithm 1. If one of the parents is the same as its son or daughter, this pair votes for FAR scheme. Otherwise, a ratio of distance between parents over the sum of the mother-daughter distance and the father-son distance is considered. In the case where the ratio is less than  $\alpha$ , this pair votes for FAR scheme. If the ratio is equal to or greater than  $\alpha$  and less than  $\beta$ , this pair votes for RAND scheme. The remaining case is where the ratio is equal to or greater than  $\beta$ . In this case, this pair votes for NEAR scheme. In the next generation, the scheme which gets the most votes is adopted.

**4.2. Parameter Setting.** We set  $\alpha$  to be 0.5 and  $\beta$  to be 1. Figure 2 describes the median of the ratio values according to generation for an instance of each test problem.  $x$ -axis represents generation and  $y$ -axis represents the ratio values.

We call the median of the ratio values after crossover (before mutation) (thin line) BM. The median of the ratio values after mutation (thick points) is called AM. After crossover (before mutation), most of BM values are close to 1. BM does not change much, while the diversity of population decreases. On the other hand, AM drops slowly as the diversity decreases.

A mutation operator moves an individual to nearby space. The distribution of moving distance by a mutation is independent of the distance between parents. The expected value of BM is one when we use a geometric crossover [22]. It will be proven in the next subsection. AM values over 1 appear frequently when the distance between parents is long enough. It means that we have sufficient diversity to

```

// input: two parents and two offspring
// output: FAR, NEAR, or RAND
// d(x, y): distance function between x and y
Function vote( $p_1, p_2, o_1, o_2$ )
{
  If  $d(p_1, p_2) = 0, d(p_1, o_1) = 0,$  or  $d(o_2, p_2) = 0$  then
    return FAR;
  end if
  ratio  $\leftarrow d(p_1, p_2)/(d(p_1, o_1) + d(o_2, p_2));$ 
  if ratio  $< \alpha$  then
    return FAR;
  end if
  if  $\alpha \leq$  ratio  $< \beta$  then
    return RAND;
  end if
  if ratio  $\geq \beta$  then
    return NEAR;
  end if
}

```

ALGORITHM 1: Voting rules.

consume. So a family votes for NEAR. Besides, the lower bound of BM is 0.5 when we use a geometric crossover. It will also be proven in the next subsection. AM values below 0.5 appear due to mutation effect. They are observed when the distance between parents is very close to 0. So a family votes for FAR. In other words, an influence of the mutation is estimated by the distance between parents. High influence of the mutation or a low AM value means that the matched parents are too close to each other to produce new solutions, while low influence of the mutation or a high AM value means that the parents are far from each other so we can match nearer solutions.

**4.3. Theoretical Support.** A binary crossover operator is geometric if all offspring are in a convex segment between parents. That is,  $d(p_1, p_2) = d(p_1, o) + d(o, p_2)$ , where  $d(p_1, p_2)$  is a distance between  $p_1$  and  $p_2$ ,  $p_i$ 's are parents, and  $o$  is an offspring obtained from a geometric crossover. Let  $D$  be the distance between both parents. We assume that  $D = d(p_1, p_2) \neq 0$ , crossover is geometric [22],  $p_1 \neq p_2$ ,  $p_1 \neq o_1$ , and  $p_2 \neq o_2$ . We remind the reader that our ratio value is defined as

$$\frac{d(p_1, p_2)}{d(p_1, o_1) + d(o_2, p_2)}, \quad (4)$$

where  $o_1$  and  $o_2$  are offspring obtained from a geometric crossover between  $p_1$  and  $p_2$ .

**Proposition 1.** *Under these assumptions, the expected value of one's ratio is 1. That is,*

$$E \left[ \frac{d(p_1, p_2)}{d(p_1, o_1) + d(o_2, p_2)} \right] = 1. \quad (5)$$

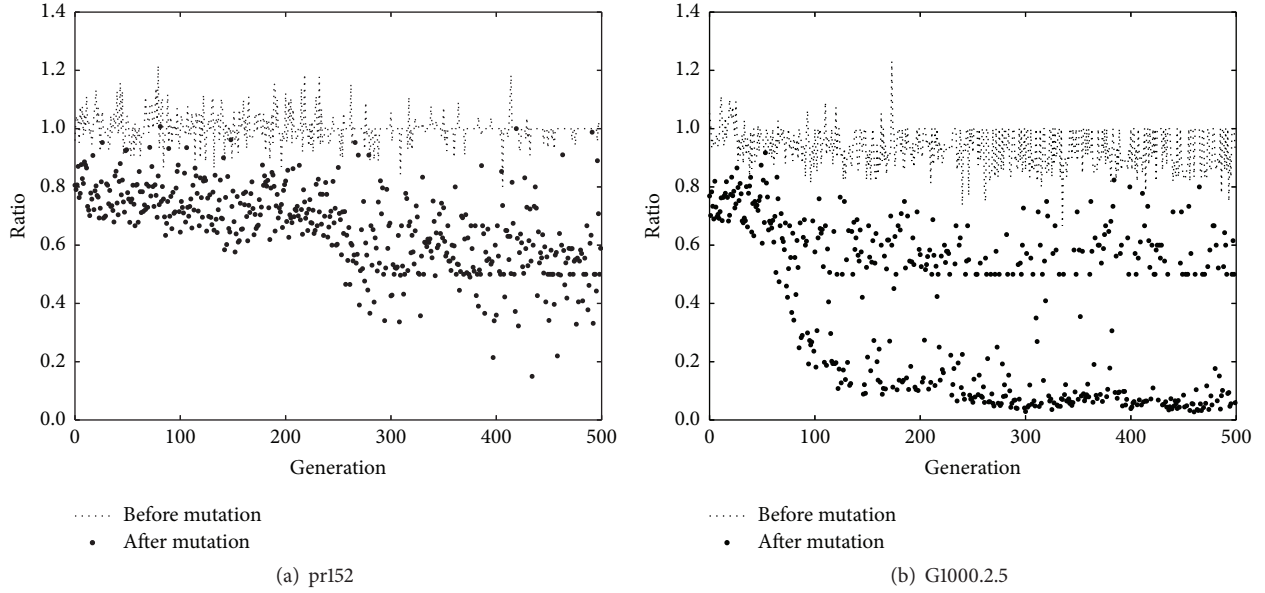


FIGURE 2: Median of ratio values according to generation.

*Proof.* It is enough to show that

$$\begin{aligned}
 E[d(p_1, o_1) + d(o_2, p_2)] &= D, \\
 E[d(p_1, o_1) + d(o_2, p_2)] \\
 &= E[d(p_1, o_1)] + E[d(o_2, p_2)] \\
 &\quad (\because E[\cdot] \text{ is linear}) \\
 &= E[d(p_1, o_1)] + E[d(p_1, p_2) - d(p_1, o_2)] \\
 &\quad (\because \text{Crossover is geometric}) \quad (6) \\
 &= E[d(p_1, o_1)] + E[D - d(p_1, o_2)] \\
 &= E[d(p_1, o_1)] + D - E[d(p_1, o_2)] \\
 &\quad (\because E[\cdot] \text{ is linear}) \\
 &= D \quad (\because E[d(p_1, o_1)] = E[d(p_1, o_2)]).
 \end{aligned}$$

□

**Proposition 2.** *Under the same assumptions, the lower bound of one's ratio value is 0.5. That is,*

$$\frac{d(p_1, p_2)}{d(p_1, o_1) + d(o_2, p_2)} \geq \frac{1}{2}. \quad (7)$$

*Proof.* By the assumption of geometric crossover,

$$\begin{aligned}
 d(p_1, p_2) &\geq d(p_1, o_1), \\
 d(p_1, p_2) &\geq d(o_2, p_2).
 \end{aligned} \quad (8)$$

By summing the above inequalities,  $2d(p_1, p_2) \geq d(p_1, o_1) + d(o_2, p_2)$ . Hence, we obtain

$$\frac{d(p_1, p_2)}{d(p_1, o_1) + d(o_2, p_2)} \geq \frac{1}{2}. \quad (9)$$

□

## 5. Experiments

**5.1. Tested GA.** We use a generational GA for test. Each solution is bijectively matched with opposite gender. A couple of individuals produce one female solution and one male solution. The genders are randomly assigned. We used 50 female and male individuals. We applied elitism [23] as a replacement strategy in both genders. Fifty best solutions remained for the next generation among previous 50 parents and new 50 offspring. Table 1 gives the other genetic operator settings.

**5.2. Traveling Salesman Problem.** From TSPLIB [24], four Euclidean instances are selected: berlin52, kroA100, bier127, and pr152. In each instance, the number of cities is represented in the right part of the name.

Figure 3 displays the fitness of the best individual according to generation. The average of the best fitness values (Avg) and the standard deviation (Std) per 200 generations are shown in Table 2. Method “single best” denotes the best single result among RAND, NEAR, and FAR in each generation. Method “simple hybrid” is the strategy that changes mating scheme once to a proper one. It was introduced in our previous work [10]. The results of all figures and tables are the average values over 1,000 runs.

TABLE 1: Genetic parameter settings.

	TSP	Graph bisection problem
Representation	Order-based encoding	Assignment of one gene for each vertex (zero or one)
Recombination	Partially matched crossover [11]	One-point crossover
Mutation	Double-bridge kick move [12] (50%)	Random swap of some pairs of genes (50%)
Repair	—	Random repair until partition is feasible
Stopping criterion	1,000 generations	500 generations

TABLE 2: Results of TSP.

Problem instance	Method	Gen 200	Gen 400	Gen 600	Gen 800	Final (= 1000)
		Avg (Std)	Avg (Std)	Avg (Std)	Avg (Std)	Avg (Std)
berlin52	Single best	1.11e4 (5.59e2)	9.71e3 (4.23e2)	9.34e3 (3.26e2)	9.71e3 (2.85e2)	9.07e3 (3.08e2)
	Simple hybrid	1.08e4 (4.34e2)	9.72e3 (3.75e2)	9.39e3 (3.48e2)	9.22e3 (3.37e2)	9.10e3 (3.18e2)
	New method	<b>1.07e4 (4.68e2)</b>	<b>9.63e3 (3.69e2)</b>	<b>9.30e4 (3.45e2)</b>	<b>9.13e3 (3.23e2)</b>	<b>9.02e3 (3.09e2)</b>
kroA100	Single best	6.87e4 (4.10e3)	4.84e4 (2.50e3)	4.13e4 (1.71e3)	3.85e4 (1.80e3)	3.65e4 (9.03e2)
	Simple hybrid	<b>6.62e4 (3.20e3)</b>	<b>4.76e4 (2.22e3)</b>	4.10e4 (1.74e3)	3.78e4 (1.58e3)	3.60e4 (1.41e3)
	New method	6.78e4 (4.09e3)	4.79e4 (2.40e3)	<b>4.04e4 (1.76e3)</b>	<b>3.71e4 (1.56e3)</b>	<b>3.51e4 (1.48e3)</b>
bier127	Single best	3.36e5 (9.09e3)	2.52e5 (9.44e3)	2.18e5 (7.03e3)	2.00e5 (6.76e3)	1.90e5 (2.25e3)
	Simple hybrid	<b>3.27e5 (1.09e4)</b>	<b>2.49e5 (9.06e3)</b>	<b>2.16e5 (7.22e3)</b>	1.99e5 (6.70e3)	1.89e5 (6.53e3)
	New method	3.38e5 (1.38e4)	2.52e5 (9.79e3)	2.57e5 (7.83e3)	<b>1.97e5 (6.80e3)</b>	<b>1.87e5 (6.20e3)</b>
pr152	Single best	5.07e5 (1.66e4)	3.28e5 (3.90e4)	2.47e5 (1.36e4)	2.05e5 (1.13e4)	1.81e5 (8.36e3)
	Simple hybrid	<b>4.78e5 (1.75e4)</b>	<b>3.14e5 (1.59e4)</b>	<b>2.40e5 (1.32e4)</b>	2.01e5 (1.12e4)	1.79e5 (9.39e3)
	New method	5.09e5 (2.65e4)	3.30e5 (2.21e4)	2.45e5 (1.60e4)	<b>2.01e5 (1.23e4)</b>	<b>1.76e5 (1.00e4)</b>

CPU: Intel Xeon E5530 2.40 GHz. Average from 1,000 runs.  
Avg: average (the smaller, the better); Std: standard deviation.

TABLE 3: Statistical test of TSP.

Problem instance	Compared method	$t$ -test	$p$ value
berlin52	Single best	+	1.33e - 03
	Simple hybrid	+	5.71e - 09
kroA100	Single best	+	1.0e - 103
	Simple hybrid	+	4.01e - 42
bier127	Single best	+	2.41e - 42
	Simple hybrid	+	9.72e - 18
pr152	Single best	+	4.46e - 26
	Simple hybrid	+	2.01e - 10

$p$  value: the smaller, the more significant.  
+ : significantly better under level  $1.00e - 02$ .

In early stages of each run, the simple hybrid method showed the best fitness. But at the end of each run, our method outperformed the others for all instances.

Table 3 shows the statistical test result of significance for Table 2. We used Welch's  $t$ -test [25].  $t$ -value of  $A - B$  in Table 3 is computed as follows:

$$t = \frac{\overline{X}_A - \overline{X}_B}{\sqrt{S_A^2/n_A + S_B^2/n_B}}, \quad (10)$$

where  $\overline{X}_A$  is the average of  $A$ ,  $S_A$  is the standard deviation of  $A$ , and  $n_A$  is the test number of  $A$ . The more significant result

causes the lower  $p$  value. In most cases,  $p$  values are very close to zero. A plus mark (+) denotes that our scheme has passed  $t$ -test under significance level, 0.01. For all instances, our scheme is significantly better than the others.

Figure 4 shows the average voting rate of three schemes. The graph shows the average over 1,000 runs. In the early stage, RAND and NEAR get higher chance to be elected. NEAR is rarely selected in the early stage. As the diversity decreases, supporters of FAR increase. At the end of each run, almost all families vote for FAR. When we compare four instances in Figure 4, we can conclude that our method is adaptive. Consuming a diversity in a small space is faster than that in a large space. So our algorithm changes the mating scheme from RAND (or rarely NEAR) to FAR. The speed of changing scheme for instance pr152 was slower than that for instance berlin52.

**5.3. Graph Bisection Problem.** We used four popular instances with 1,000 vertices [26]. The difference of the instances is edge density. The right part of each name (such as 05) represents the average vertex degree.

Figure 5 illustrates the fitness of the best individual over all generations. The results of all figures and tables are written with the average values over 1,000 runs as in TSP. The average of the best fitness values (Avg) and the standard deviation (Std) per 100 generations are listed in

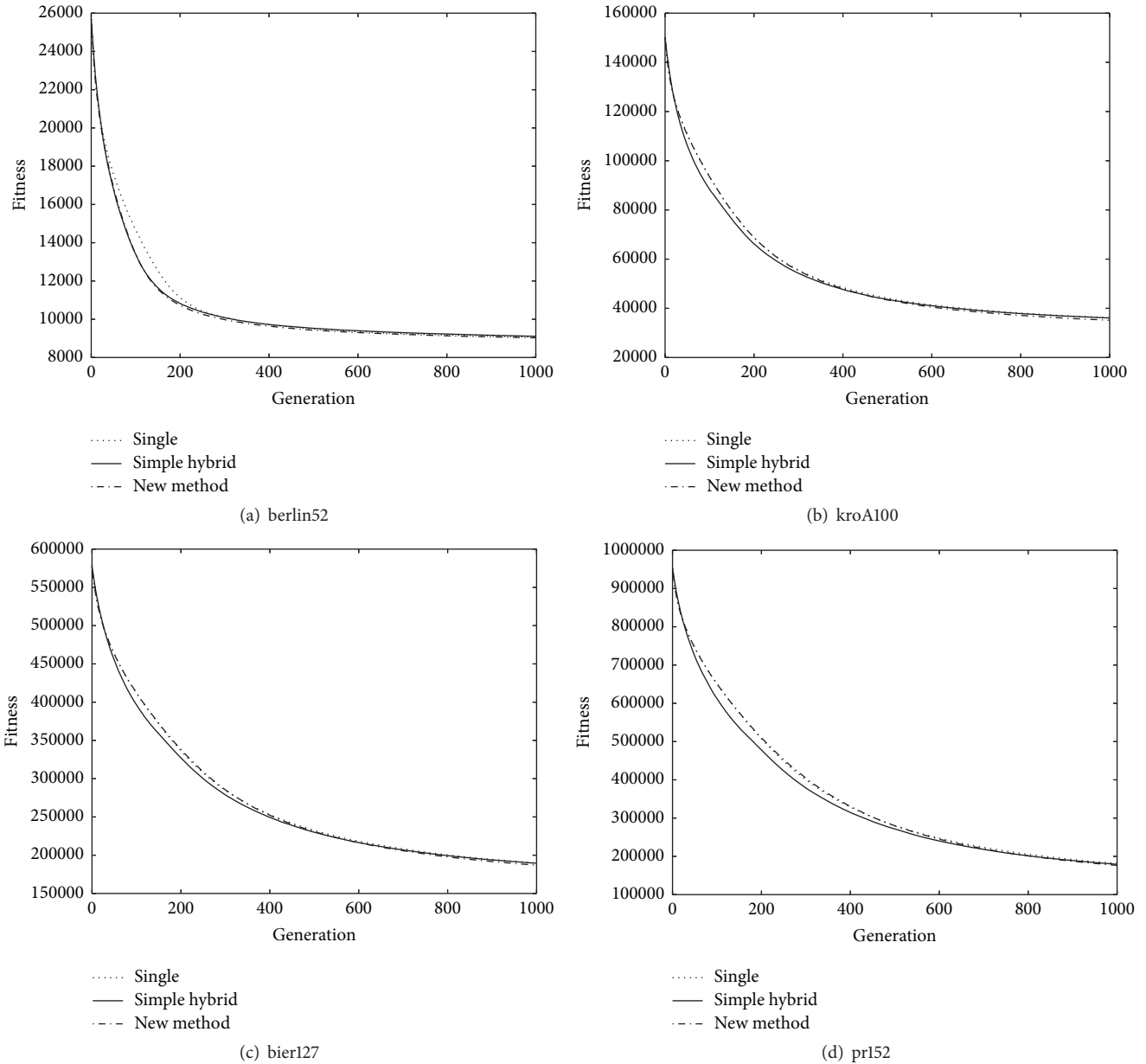


FIGURE 3: Fitness of mating schemes in TSP (the smaller, the better).

Table 4. Method “single best” and method “simple hybrid” are the same as in TSP. In almost all generations of all the instances, our method outperformed the others. Table 5 shows  $t$ -test for the results in Table 4. It is conducted in the same way as that used in TSP. In all instances except one our scheme was significantly superior to the others.

Figure 6 shows the average voting rate of three schemes. While TSP showed different speed of changing schemes according to each instance, the figures of four graph bisection instances are almost the same as each other. Previously we reported that NEAR showed very poor results in this problem [10]. With our new method, NEAR is naturally abandoned because FAR increases very fast.

5.4. *Comparison with Existing Methods.* We compared our method with existing distance-based mating ones. We implemented variants of Ishibuchi and Shibata’s [13] and Galán et al.’s [8] methods with two same-sized genders. Ishibuchi and Shibata’s method [13] selects one parent that is the farthest individual from the average among the results of repeated tournament selections of  $\alpha$  times. Their method selects the other parent that is the nearest individual from the first parent among the results of repeated tournament selections of  $\beta$  times. We set  $\alpha$  and  $\beta$  to be 9 as in [13]. The transformed variant selects the first parent from the female solutions and selects the second parent from the male solutions. It is repeated until all solutions are one-to-one matched. Galán et al.’s method [8] selects one parent that is the best. As the other

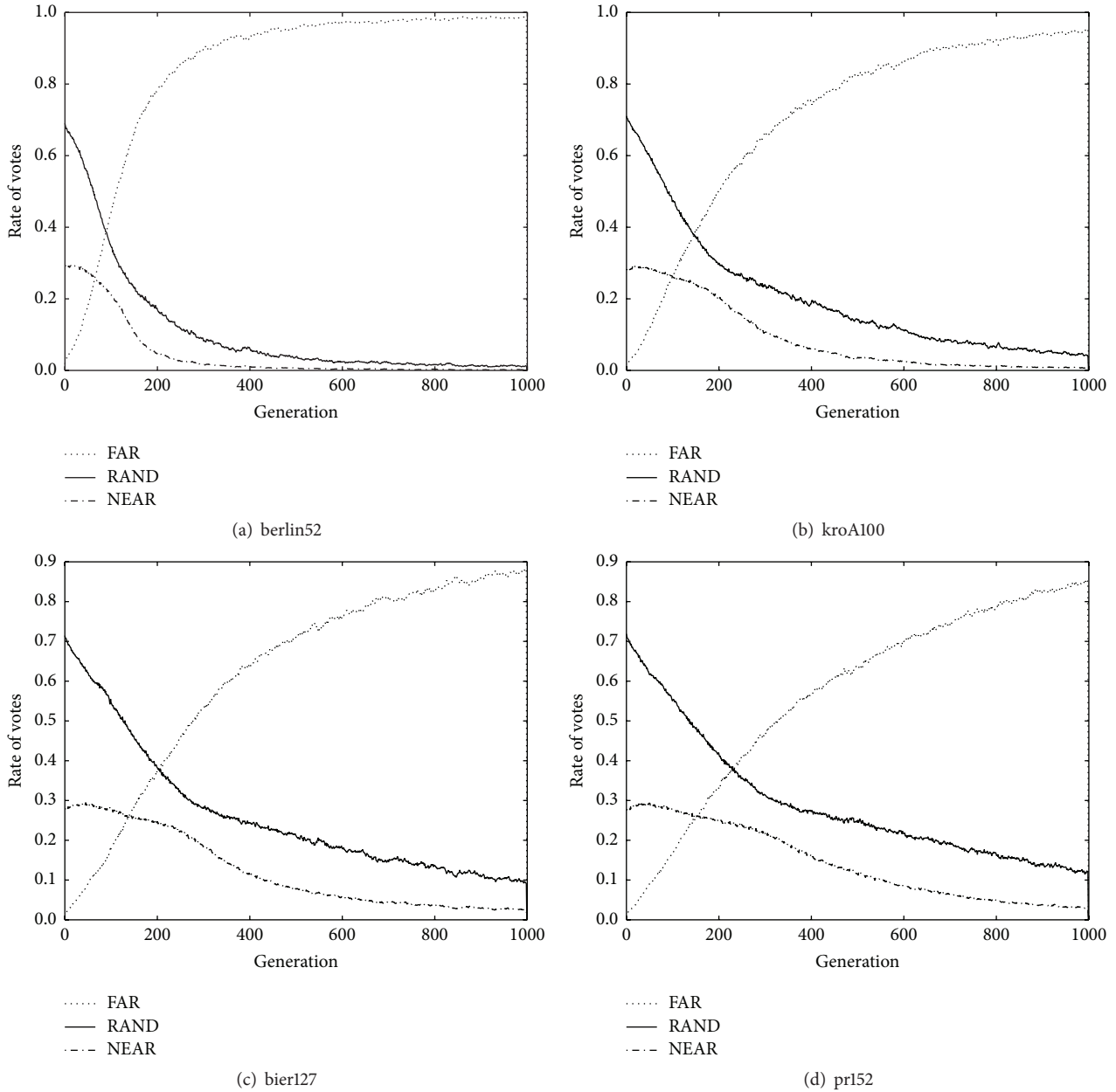


FIGURE 4: Voting rates of schemes in TSP.

parent, their method selects the  $(\gamma - 1)$ th nearest individual, where  $\gamma$  is the mating preference of the first parent. The mating preference is inherited in crossover, and it increases by 1 with probability 0.25 or decreases by 1 with probability 0.25, in mutation. The same as the variant of Ishibuchi and Shibata's method [13], we made this method select the first parent from the female solutions and the second parent from the male solutions. It is repeated until all solutions are one-to-one matched. All the conditions and settings excluding mating are the same as those in the experiments of the previous sections.

Table 6 compares the solution qualities of these two existing methods and ours. For all instances of two test problems, our method significantly outperformed the others. Table 7

compares the computation times with respect to mating. Each value in Table 7 except mating proportion is measured in seconds. Our method took more time than Galán's method. But our method was faster than Ishibuchi's. Galán's method repeats finding the  $(\gamma - 1)$ th nearest individual, whereas our method maximizes (or minimizes) the sum of distances. For graph bisection problem, computation times of instances of our method are similar to each other because the instances have the same number of nodes. In TSP, as the solution space grows, the proportion of mating time decreases, because the mating time of our method is mainly bounded by population size. As distance scale grows, mating time increases. It can be resolved by approximating the scale of distance values.



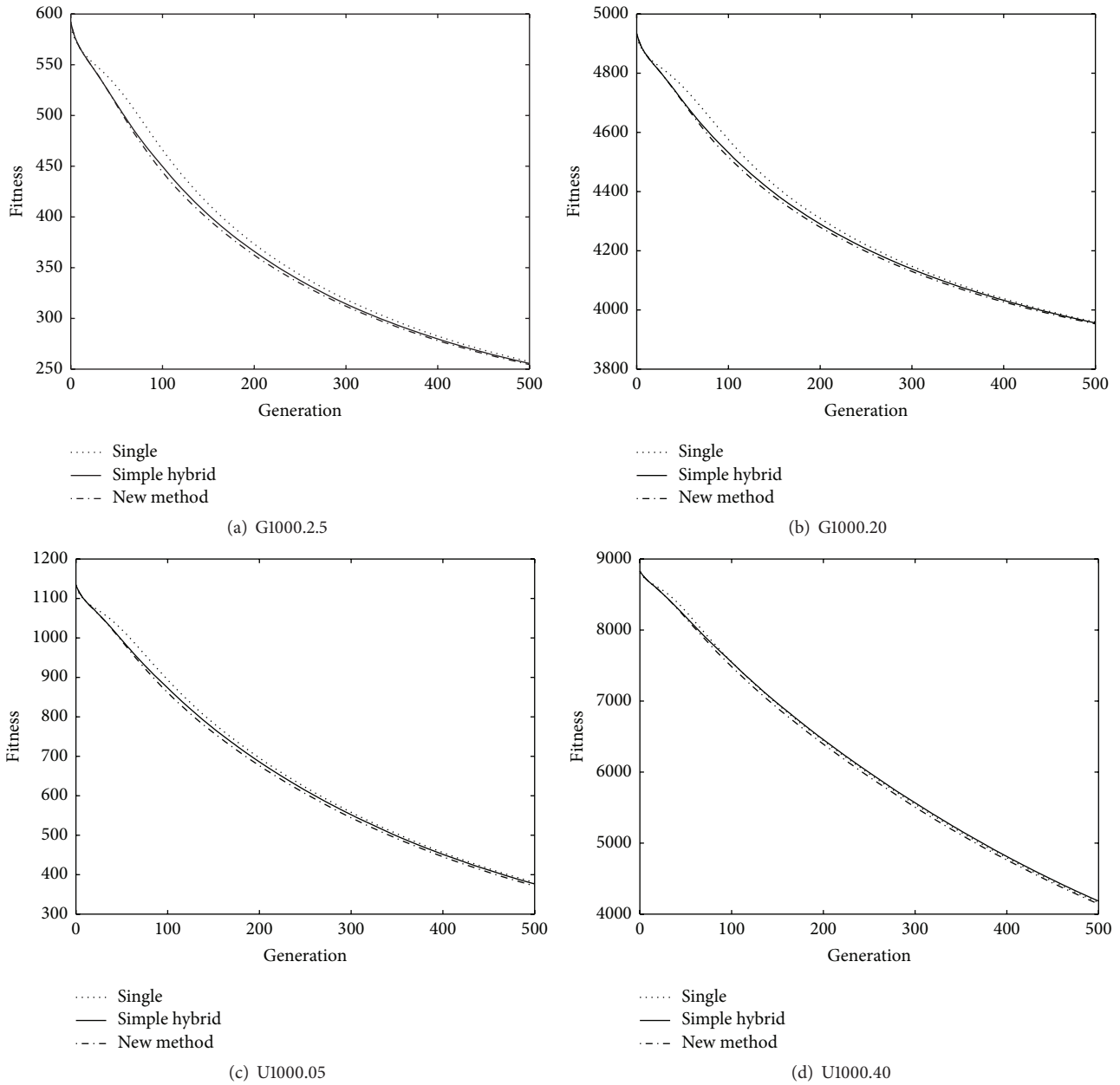


FIGURE 5: Fitness of mating schemes in graph bisection (the smaller, the better).

Our mating method did not overburden the entire GA, and also we expect reducing time burden through some improved implementation.

### 6. Concluding Remarks

We analyzed the proposed adaptive hybrid mating scheme for TSP and graph bisection. FAR scheme is biased on exploitation, while NEAR scheme is biased on exploitation. Our mating scheme assesses the distance of the matched parents with their offspring. With this assessment, our adaptive scheme tries to find a balanced point between exploration

and exploitation in each generation. We also compared the proposed method with two existing distance-based methods. The proposed method showed better performance than the two existing methods.

We set the threshold parameters as 0.5 and 1.0 with some observation and the values are theoretically justified. But we expect that the method of dynamically adjusting these values may produce better results. Real-coded problems may have different features from combinatorial optimization. With our scheme, more various problems such as function optimization can be tested. There are still opportunities for further enhancements and we will study the presented method

TABLE 4: Results of graph bisection.

Problem instance	Method	Gen 100	Gen 200	Gen 300	Gen 400	Final (= 500)
		Avg (Std)	Avg (Std)	Avg (Std)	Avg (Std)	Avg (Std)
G1000.2.5	Single best	4.48e2 (1.10e1)	3.71e2 (9.72e0)	3.19e2 (1.17e1)	2.83e2 (1.02e1)	2.57e2 (9.27e0)
	Simple hybrid	4.50e2 (1.17e1)	3.66e2 (9.36e0)	3.14e2 (9.03e0)	2.80e2 (8.82e0)	2.55e2 (1.02e1)
	New method	<b>4.44e2 (1.16e1)</b>	<b>3.62e2 (9.55e0)</b>	<b>3.11e2 (9.18e0)</b>	<b>2.78e2 (8.77e0)</b>	<b>2.54e2 (8.68e0)</b>
G1000.20	Single best	4.58e3 (3.11e1)	4.30e3 (2.78e1)	4.15e3 (2.70e1)	4.04e3 (3.16e1)	3.96e3 (3.00e1)
	Simple hybrid	4.53e3 (3.11e1)	4.29e3 (2.83e1)	4.13e3 (2.78e1)	4.03e3 (2.72e1)	3.96e3 (2.71e1)
	New method	<b>4.51e3 (3.43e1)</b>	<b>4.28e3 (2.90e1)</b>	<b>4.13e3 (2.72e1)</b>	<b>4.02e3 (2.64e1)</b>	<b>3.95e3 (2.61e1)</b>
U1000.05	Single best	8.71e2 (2.14e1)	6.96e2 (3.11e1)	5.58e2 (2.77e1)	4.56e2 (2.55e1)	3.80e2 (2.41e1)
	Simple hybrid	8.75e2 (2.20e1)	6.86e2 (2.25e1)	5.52e2 (2.26e1)	4.51e2 (2.26e1)	3.76e2 (2.28e1)
	New method	<b>8.64e2 (2.26e1)</b>	<b>6.76e2 (2.24e1)</b>	<b>5.44e2 (2.23e1)</b>	<b>4.45e2 (2.24e1)</b>	<b>3.71e2 (2.20e1)</b>
U1000.40	Single best	7.53e3 (1.32e2)	6.44e3 (1.77e2)	5.55e3 (2.06e2)	4.79e3 (2.39e2)	4.16e3 (2.64e2)
	Simple hybrid	7.55e3 (1.32e1)	6.46e3 (1.77e2)	5.57e3 (2.14e2)	4.81e3 (2.48e2)	4.18e3 (2.74e2)
	New method	<b>7.48e3 (1.34e2)</b>	<b>6.40e3 (1.27e2)</b>	<b>5.50e3 (2.03e2)</b>	<b>4.76e3 (2.35e2)</b>	<b>4.14e3 (2.65e2)</b>

CPU: Intel Xeon E5530 2.40 GHz. Average from 1,000 runs.  
Avg: average (the smaller, the better); Std: standard deviation.

TABLE 5: Statistical test of graph bisection.

Problem instance	Compared method	$t$ -test	$p$ value
G1000.2.5	Single best	+	5.66e - 10
	Simple hybrid	+	2.44e - 03
G1000.20	Single best	+	2.69e - 08
	Simple hybrid	+	5.30e - 03
U1000.05	Single best	+	2.42e - 16
	Simple hybrid	+	1.32e - 07
U1000.40	Single best	~	6.23e - 02
	Simple hybrid	+	2.62e - 04

$p$  value: the smaller, the more significant.  
+: significantly better under level  $1.00e - 02$ .  
~: not significantly different under level  $1.00e - 02$ .

TABLE 6: Comparison of results on two test problems.

Problem instance	Our method		Galán et al. [8]				Ishibuchi and Shibata [13]			
	Avg	Std	Avg	Std	$t$ -test	$p$ value	Avg	Std	$t$ -test	$p$ value
berlin52	9.02e3	3.09e2	9.27e3	3.32e2	+	1.6e - 59	9.24e3	3.38e2	+	4.4e - 45
kroA100	3.51e4	1.48e3	3.81e4	1.60e3	+	3.5e - 228	3.77e4	1.41e3	+	9.2e - 213
bier127	1.87e5	6.20e3	1.96e5	6.78e3	+	1.2e - 158	1.97e5	6.46e3	+	4.4e - 188
pr152	1.76e5	1.00e4	1.96e5	1.12e4	+	1.6e - 227	2.00e5	1.07e4	+	3.5e - 290
G1000.2.5	2.54e2	8.68e0	3.01e2	3.13e1	+	5.2e - 248	3.04e2	1.15e1	+	0*
G1000.20	3.95e3	2.61e1	4.09e3	9.47e1	+	1.0e - 245	4.09e3	3.21e1	+	0*
U1000.05	3.71e2	2.20e1	4.96e2	8.40e1	+	9.7e - 246	5.18e2	2.64e1	+	0*
U1000.40	4.14e3	2.65e2	4.99e3	5.94e2	+	5.9e - 220	5.31e3	2.23e2	+	0*

CPU: Intel Xeon E5530 2.40 GHz. Average from 1,000 runs.  
Avg: average (the smaller, the better); Std: standard deviation.  
 $p$  value: the smaller, the more significant.  
+: significantly better under level  $1.00e - 02$ .  
\*: it means that this value is less than  $1.0e - 300$ .

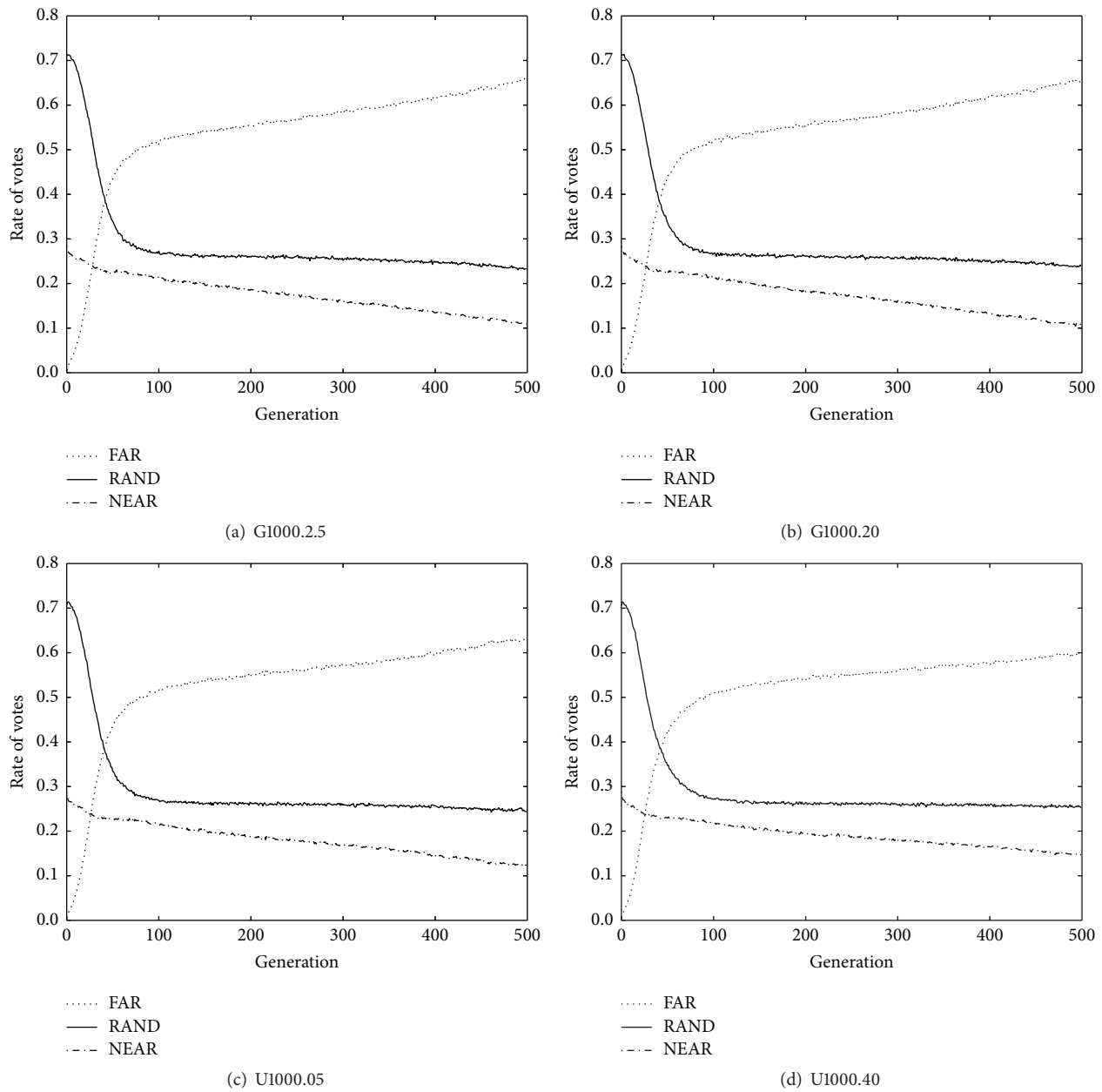


FIGURE 6: Voting rates of schemes in graph bisection.

with various environments such as various crossover, mutation rates, replacement, and local optimization for future work.

**Disclosure**

A preliminary version of this paper appeared in Proceedings of the Genetic and Evolutionary Computation Conference, 2015 (pp. 1413-1414).

**Competing Interests**

The authors declare that they have no competing interests.

**Acknowledgments**

This work was supported by the Engineering Research Center of Excellence Program of Republic of Korea Ministry of Science, ICT\& Future Planning(MSIP)/National Research Foundation of Korea (NRF) (Grant no. NRF-0421-20120031). This research was also supported by Basic Science Research

TABLE 7: Results of computation time.

Problem instance	Our method			Galán et al. [8]			Ishibuchi and Shibata [13]		
	Mating	Total	Proportion of mating (%)	Mating	Total	Proportion of mating (%)	Mating	Total	Proportion of mating (%)
berlin52	7.68	81.49	9.4	0.63	70.31	0.9	68.31	140.07	48.5
kroA100	14.60	283.97	5.1	0.68	252.33	0.3	244.85	505.28	48.4
bier127	20.04	451.89	4.4	0.62	405.57	0.2	388.40	796.24	48.7
pr152	25.24	634.28	3.9	0.65	570.15	0.1	546.99	1119.84	48.8
G1000.2.5	8.26	54.81	15.0	0.60	45.94	1.3	9.06	55.74	16.2
G1000.20	7.76	61.91	12.5	0.56	52.74	1.1	9.10	63.39	14.3
U1000.05	8.98	56.45	15.9	0.57	47.02	1.2	8.86	55.22	16.0
U1000.40	8.15	66.63	12.2	0.50	57.95	1.0	9.02	68.41	13.1

Average CPU seconds from 1,000 runs on Intel Xeon E5530 2.40 GHz.

Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (no. 2015R1D1A1A01060105).

## References

- [1] G. Ochoa, C. Mädler-Kron, R. Rodriguez, and K. Jaffe, "Assortative mating in genetic algorithms for dynamic problems," in *Applications of Evolutionary Computing*, pp. 617–622, Springer, Berlin, Germany, 2005.
- [2] L. B. Booker, "Improving the performance of genetic algorithms in classifier systems," in *Proceedings of the 1st International Conference on Genetic Algorithms*, pp. 80–92, Lawrence Erlbaum Associates, 1985.
- [3] C. M. Fonseca and P. J. Fleming, "Genetic algorithms for multiobjective optimization: formulation discussion and generalization," in *Proceedings of the 5th International Conference on Genetic Algorithms*, S. Forrest, Ed., vol. 1, pp. 416–423, July 1993.
- [4] F. Ramezani and S. Lotfi, "IAMGA: intimate-based assortative mating genetic algorithm," in *Proceedings of the Swarm Evolutionary and Memetic Computing Conference*, pp. 240–247, 2011.
- [5] C. Fernandes, R. Tavares, C. Munteanu, and A. Rosa, "Using assortative mating in genetic algorithms for vector quantization problems," in *Proceedings of the ACM Symposium on Applied Computing (SAC '01)*, pp. 361–365, ACM, 2001.
- [6] H. Ishibuchi, K. Narukawa, N. Tsukamoto, and Y. Nojima, "An empirical study on similarity-based mating for evolutionary multiobjective combinatorial optimization," *European Journal of Operational Research*, vol. 188, no. 1, pp. 57–75, 2008.
- [7] H. Ishibuchi and Y. Shibata, "Mating scheme for controlling the diversity-convergence balance for multiobjective optimization," in *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 1259–1271, Seattle, Wash, USA, June 2004.
- [8] S. F. Galán, O. J. Mengshoel, and R. Pinter, "A novel mating approach for genetic algorithms," *Evolutionary Computation*, vol. 21, no. 2, pp. 197–229, 2013.
- [9] C. Jung, Y.-H. Kim, Y. Yoon, and B.-R. Moon, "An adaptive method of Hungarian mating schemes in genetic algorithms," in *Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation*, pp. 1413–1414, Madrid, Spain, 2015.
- [10] C. Jung, Y.-H. Kim, Y. Yoon, and B.-R. Moon, "Investigation of Hungarian mating schemes for genetic algorithms," in *Proceedings of the 29th Annual ACM Symposium on Applied Computing (SAC '14)*, pp. 140–147, ACM, Gyeongju, Republic of Korea, March 2014.
- [11] D. E. Goldberg and R. Lingle, "Alleles, loci, and the traveling salesman problem," in *Proceedings of the International Conference on Genetic Algorithms and Their Applications*, pp. 154–159, 1985.
- [12] O. Martin, E. W. F. Steve, and W. Otto, "Large-step Markov chains for the traveling salesman problem," *Complex Systems*, vol. 5, no. 3, pp. 299–251, 1991.
- [13] H. Ishibuchi and Y. Shibata, "A similarity-based mating scheme for evolutionary multiobjective optimization," in *Genetic and Evolutionary Computation—GECCO 2003*, E. Cantú-Paz, J. A. Foster, K. Deb et al., Eds., vol. 2723 of *Lecture Notes in Computer Science*, pp. 1065–1076, 2003.
- [14] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, pp. 83–97, 1955.
- [15] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, Prentice Hall, 1982.
- [16] D. Avis, "A survey of heuristics for the weighted matching problem," *Networks*, vol. 13, no. 4, pp. 475–493, 1983.
- [17] P. Hahn, T. Grant, and N. Hall, "A branch-and-bound algorithm for the quadratic assignment problem based on the Hungarian method," *European Journal of Operational Research*, vol. 108, no. 3, pp. 629–640, 1998.
- [18] A. Moraglio, Y.-H. Kim, Y. Yoon, and B.-R. Moon, "Geometric crossovers for multiway graph partitioning," *Evolutionary Computation*, vol. 15, no. 4, pp. 445–474, 2007.
- [19] Y. Yoon, Y.-H. Kim, and B.-R. Moon, "Feasibility-preserving crossover for maximum k-coverage problem," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '08)*, pp. 593–598, Atlanta, Ga, USA, July 2008.
- [20] Y. Yoon, Y.-H. Kim, A. Moraglio, and B.-R. Moon, "Quotient geometric crossovers and redundant encodings," *Theoretical Computer Science*, vol. 425, pp. 4–16, 2012.
- [21] K. S. Goh, A. Lim, and B. Rodrigues, "Sexual selection for genetic algorithms," *Artificial Intelligence Review*, vol. 19, no. 2, pp. 123–152, 2003.
- [22] A. Moraglio and R. Poli, "Topological interpretation of crossover," in *Proceedings of the Genetic and Evolutionary*

*Computation Conference (GECCO '04)*, vol. 3102, pp. 1377–1388, Seattle, Wash, USA, June 2004.

- [23] K. A. De Jong, *An analysis of the behavior of a class of genetic adaptive systems [Ph.D. dissertation]*, University of Michigan, Ann Arbor, Mich, USA, 1975.
- [24] G. Reinelt, “TSPLIB—a traveling salesman problem library,” *INFORMS Journal on Computing*, vol. 3, no. 4, pp. 376–384, 1991.
- [25] B. L. Welch, “The generalization of “Student’s” problem when several different population variances are involved,” *Biometrika*, vol. 34, no. 1-2, pp. 28–35, 1947.
- [26] Y.-H. Kim and B.-R. Moon, “Lock-gain based graph partitioning,” *Journal of Heuristics*, vol. 10, no. 1, pp. 37–57, 2004.



# Hindawi

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

