

Investigation of the Fitness Landscapes and Multi-parent Crossover for Graph Bipartitioning

Yong-Hyuk Kim and Byung-Ro Moon

School of Computer Science & Engineering, Seoul National University
Shillim-dong, Kwanak-gu, Seoul, 151-742 Korea
{yhdfly, moon}@soar.snu.ac.kr

Abstract. An empirical study is performed on the local-optimum space of graph bipartitioning. We examine some statistical features of the fitness landscape. They include the cost-distance correlations and the properties around the central area of local optima. The study revealed some new notable results about the properties of the fitness landscape; e.g., the central area yielded fairly good quality in the local-optimum space. We performed an experiment on a spectrum of different exploitation strengths of the central areas. From the results, it seems attractive to exploit the central area, but excessive or insufficient exploitation is not desirable.

1 Introduction

An NP-hard problem such as graph partitioning problem or traveling salesman problem (TSP) has a finite solution set and each solution has a cost. Although finite, the problem space is intractably large even for a small but nontrivial problem. It is almost impossible to find an optimal solution for those problems by exhaustive or simple search methods. Thus, in case of NP-hard problems, heuristic algorithms are being used. Heuristic algorithms provide reasonable solutions in acceptable computing time but have no performance guarantee.

Consider a combinatorial problem $C = (\Omega, f)$ and a local optimization algorithm $L_c : \Omega \rightarrow \Omega$, where Ω is the solution space and f is the cost function. If a solution $s^* \in \Omega$ is in $L_c(\Omega)$, then s^* is called a local optimum with respect to the algorithm L_c . For each local optimum $s^* \in L_c(\Omega)$, we define the neighborhood set of s^* to be a set $N(s^*) \subset \Omega$ such that, for every s in $N(s^*)$, $L_c(s)$ is equal to s^* . That is, s^* is the attractor of the solutions in $N(s^*)$. We examine the space $L_c(\Omega)$ and hope to get some insight into the problem space. This is an alternative for examining the intractably huge whole problem space. Good insight into the problem space can provide a motivation for a good search algorithm.

A number of studies about the ruggedness and the properties of problem search spaces have been conducted. Sorkin [21] defined the fractalness of a solution space and proposed that simulated annealing [18] is efficient when the space is fractal. Jones and Forrest [14] introduced the fitness-distance correlation as a measure of search difficulty. Manderick *et al.* [19] measured the ruggedness of a

problem space by autocorrelation function and correlation length obtained from a time series of solutions. Weinberger [23] conjectured that, if all points on a fitness landscape are correlated relatively highly, the landscape is bowl shaped. Boese *et al.* [3] suggested that, through measuring cost-distance correlations for the TSP and the graph bisection problem, the cost surfaces are globally convex; from these results they proposed an adaptive multi-start heuristic and showed that the heuristic is efficient [3]. Kauffman [15] proposed the NK-landscape model that can control the ruggedness of a problem space.

In this paper, we present a number of experiments to analyze problem spaces more elaborately. We examine the cost-distance correlations and the properties around the central areas of local optima. Based on the empirical study, we perform an experiment on a spectrum of different exploitation strengths of the central areas under a genetic algorithm (GA) framework. We perform these experiments on the graph bipartitioning problem.

The remainder of this paper is organized as follows. In Section 2, we summarize the graph bipartitioning problem, the Fiduccia-Mattheyses algorithm (FM) which is used as a major local optimization algorithm in this paper, and test graphs. We perform various experiments and analyze fitness landscapes in Section 3. In Section 4, we propose a multi-parent crossover for graph bipartitioning. Finally, we make our conclusions in Section 5.

2 Preliminaries

2.1 Graph Bipartitioning

Let $G = (V, E)$ be an unweighted undirected graph, where V is the set of vertices and E is the set of edges. A bipartition (A, B) consists of two subsets A and B of V such that $A \cup B = V$ and $A \cap B = \phi$. The *cut size* of a bipartition is defined to be the number of edges whose endpoints are in different subsets of the bipartition. The bipartitioning problem is the problem of finding a bipartition with minimum cut size. If the difference of cardinalities between two subsets is at most one, the problem is called *graph bisection* problem and if the difference does not exceed the fixed ratio of $|V|$, the problem is called *roughly balanced bipartitioning* problem. Without balance criterion, we can find the optimal solution in polynomial time by maxflow-mincut algorithm [10]. In a roughly balanced bipartitioning problem, 10% of skewness is usually allowed [20]. Since it is NP-hard for general graphs [11], heuristic algorithms are used practically. These include FM algorithm [9], a representative linear time heuristic, PROP [5] based on probabilistic notion, LG [17] based on lock gain, etc. In this paper, we consider only roughly balanced bipartitioning problem allowing 10% of skewness.

2.2 Fiduccia-Mattheyses Algorithm (FM)

Fiduccia and Mattheyses [9] introduced a heuristic for roughly balanced bipartitioning problem. The FM algorithm as well as the Kernighan-Lin algorithm

```

do {
  Compute gain  $g_v$  for each  $v \in V$ ;
  Make gain lists of  $g_v$ s;
   $Q = \emptyset$ ;
  for  $i = 1$  to  $|V| - 1$  {
    Choose  $v_i \in V - Q$  such that  $g_{v_i}$  is maximal and
      the move of  $v_i$  does not violate the balance criterion;
     $Q = Q \cup \{v_i\}$ ;
    for each  $v \in V - Q$  adjacent to  $v_i$ 
      Update its gain  $g_v$  and adjust the gain list;
  }
  Choose  $k \in \{1, \dots, |V| - 1\}$  that maximizes  $\sum_{i=1}^k g_{v_i}$ ;
  Move all the vertices in the subset  $\{v_1, \dots, v_k\}$  to their opposite sides;
} until (there is no improvement)

```

Fig. 1. The Fiduccia-Mattheyses algorithm (FM)

(KL) [16] is a traditional iterative improvement algorithm. The algorithm improves on an initial solution by single-node moves. The main difference between KL and FM lies in that a new partition in FM is derived by moving a single vertex, instead of KL's pair swap. The structure of the FM algorithm is given in Figure 1. FM proceeds in a series of *passes*. In each pass, all vertices are moved in chain and then the best bipartition during the pass is returned as a new solution. The algorithm terminates when one or a few passes fail to find a better solution. With an efficient data structure, each pass of FM runs in $\Theta(|E|)$ time.

2.3 Test Beds

We tested on a total of 17 graphs which consist of two groups of graphs. They are composed of 17 graphs from [13] (9 random graphs and 8 geometric graphs). The two classes were used in a number of other studies [20] [4] [1] [17]. The classes are briefly described below.

1. *Gn.d*: A random graph on n vertices, where an edge is placed between any two vertices with probability p independent of all other edges. The probability p is chosen so that the expected vertex degree, $p(n - 1)$, is d .
2. *Un.d*: A random geometric graph on n vertices that lie in the unit square and whose coordinates are chosen uniformly from the unit interval. There is an edge between two vertices if their Euclidean distance is t or less, where $d = \pi t^2$ is the expected vertex degree.

3 Investigation of the Problem Space

In this section, we first extend the experimentation of Boese *et al.* [3] to examine the local-optimum space. We denote by local-optimum space the space consisting of all local optima with respect to a local optimization algorithm. Next, we examine the area around the “central point” of local optima. In our experiments, we use a sufficiently large number of local optima. We do not care

about solutions other than local optima. The local optimizer in our experiments is the FM algorithm.

In the graph bipartitioning problem for a graph $G = (V, E)$, each solution (A, B) is represented by a $|V|$ -bits code. Each bit corresponds to a vertex in the graph. A bit has value zero if the vertex is in the set A , and has value one otherwise. In this encoding, a vertex move in the FM algorithm changes the solution by one bit. Thus, it is natural to define the distance between two solutions by the Hamming distance. However, if the Hamming distance between two solutions is $|V|$, they are symmetric and equal. We hence define the *distance* between two solutions as follows.

Definition 1 Let the universal set U be $\{0, 1\}^{|V|}$. For $\mathbf{a}, \mathbf{b} \in U$, we define the distance between \mathbf{a} and \mathbf{b} as follows:¹

$$d(\mathbf{a}, \mathbf{b}) = \min(\mathfrak{H}(\mathbf{a}, \mathbf{b}), |V| - \mathfrak{H}(\mathbf{a}, \mathbf{b}))$$

where \mathfrak{H} is the Hamming distance.

By the definition, $0 \leq d(\mathbf{a}, \mathbf{b}) \leq \lfloor |V|/2 \rfloor$ while $0 \leq \mathfrak{H}(\mathbf{a}, \mathbf{b}) \leq |V|$.

3.1 Cost-Distance Correlation

Given a set of local minima, Boese *et al.* [3] plotted, for each local minimum, i) the relationship between the cost and the average distance from all the other local minima, and ii) the relationship between the cost and the distance to the best local minimum. They performed experiments for the graph bisection and the traveling salesman problem, and showed that both problems have strong positive correlations for both i) and ii) in the above. This fact hints that the best local optimum is located near the center of the local-optimum space. From their experiments, they conjectured that cost surfaces of both problems are globally convex. In this subsection, we repeat their experiments for other graphs and extend their study to get more insight.

The solution space for the experiment is selected as follows. First, we choose thousands of random solutions and obtain the corresponding set of local optima by locally optimizing them. Next, we remove the duplicated solutions in the set if any. Figure 2 shows the plotting results for the graph U500.10. It is consistent with Boese *et al.*'s results with strong cost-distance correlation. More statistics for a number of graphs are given in Table 1. The meaning of each item in the table is as follows. "Population size" means the number of local optima we used for each graph. "Best cut" is the cost of the best local optimum. "Average cut" is the average cost of the local optima. "Cost-distance correlation" is the correlation

¹ Given an element $\mathbf{a} \in U$, there is only one element such that it is different from \mathbf{a} and the distance d to \mathbf{a} is zero. If the distance between two elements is equal to zero, we define them to be in relation R . Then, the relation R is an equivalence relation. Suppose Q is the quotient set of U by relation R , it is easily verified that (Q, d) is a metric space.

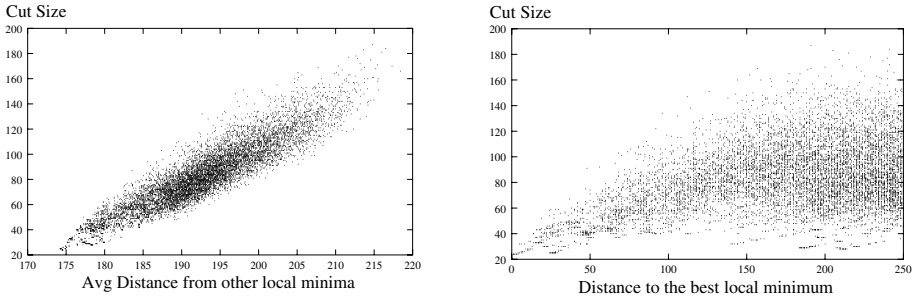


Fig. 2. Relationship between cost and distance: U500.10 (see Table 1)

Table 1. The results for each graph

Items	G250.10	G500.2.5	G1000.2.5	U500.05	U500.10	U1000.05
Population size	9877	10000	10000	10000	9302	10000
Best cut	352	52	103	5	24	16
Average cut	367.65	64.58	128.17	35.62	83.58	70.76
Cost-distance correlation	0.77	0.78	0.83	0.89	0.91	0.88
Central point cut (CP)	380	60	118	5	24	17
CP + FM	352	51	99	5	24	16
Average distance	102.94	217.11	453.44	215.63	192.83	448.09

Population size : the number of local optima
 Best cut : the minimum cost
 Average cut : the average cost
 Cost-distance correlation : correlation coefficient between cost and average distance from each local optimum to others
 Central point cut (CP) : the cost of the approximate central point in solution space
 CP + local opt : the cost after local optimization on the approximate central point
 Average distance : the average value of distances between a pair of local optima

coefficient between the costs of local optima and the average distances from the other local optima. “Central point cut (CP)” is the cost of the approximate central point of the local-optimum space (see Section 3.2 for the approximate central point). “CP + local opt” is the cost after local optimization on the approximate central point. Finally, “Average distance” means the average distance between a pair of local optima. Overall, each graph showed strong positive correlation. Depending on graphs, correlation coefficients were a bit different. Geometric graphs showed larger correlation coefficients than random graphs. In the statistical data of Table 1, each population was obtained from 10,000 random initial solutions. Among the six graphs, four graphs had no duplications and the other two graphs had 123 and 698 duplications, respectively. It is surprising that there were no duplications in the first 10,000 attractors for four of them. It seems to suggest that the number of all possible local optima with respect to FM is immeasurably large.

Figure 3, Table 2, and Table 3 compare the data with different local optimizers. A greedy local optimizer which moves only vertices with positive gain was named GREEDY. Its principle is the same as that of the steepest descent algorithm in the differentiable cases. NONE means a set of random solutions without any local optimization. From the cut sizes in Table 2 and Table 3, FM is clearly stronger than the GREEDY algorithm. The stronger the local opti-

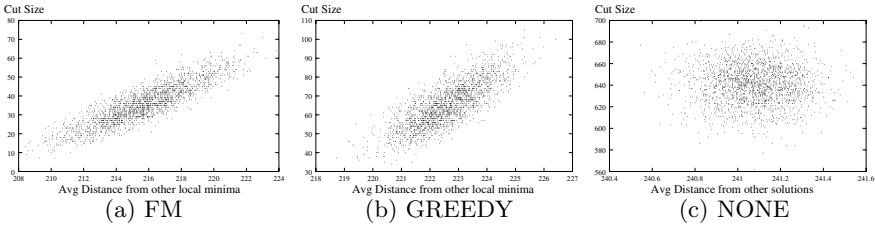


Fig. 3. Relationship between cost and distance with different local optimizer in the graph U500.05 (see Table 3)

Table 2. The data comparison with different local optimizer in the graph G500.10

Local opt	FM	GREEDY	NONE
Population size	2000	2000	2000
Best cut	623	666	1101
Average cut	648.60	706.26	1178.00
Cost-distance correlation	0.77	0.81	-0.02
Central point cut (CP)	659	670	1138
CP + local opt	623	643	-
Average distance	218.58	229.71	241.09

Table 3. The data comparison with different local optimizer in the graph U500.05

Local opt	FM	GREEDY	NONE
Population size	2000	2000	2000
Best cut	7	34	562
Average cut	35.86	65.16	640.89
Cost-distance correlation	0.88	0.79	-0.02
Central point cut (CP)	5	33	581
CP + local opt	5	30	-
Average distance	215.71	222.58	241.08

mizer, the smaller the average distance between two local optima and the more sharing among local optima. However, from Tables 1–3, it is surprising that, differently from our expectation, the average distance between two arbitrary local optima is nearly 80% ~ 90% of the possible maximum distance $\lfloor |V|/2 \rfloor$. This is an evidence of the huge diversity of local optima. In Figure 3, a stronger local optimization shows stronger cost-distance correlation. Since the average distances in graphs are various, these values may have some potential to be used as measures of the problem difficulty with respect to a local optimizer.²

3.2 Approximate Central Point

The results of Boese *et al.* [3] for the TSP and the graph bisection problem suggest that the best solution is located near the center of the local-optimum space. As a result of this, given a subspace of local optima for a problem, the “central

² This is not a simple issue, though.

point”³ of the subspace may be near the optimal solution. Hence, computing the “central point” not only supports the results of Boese *et al.* but may also be helpful for obtaining the optimal solution.

Given a subspace Ω' of the whole solution space in the graph bipartitioning problem, the “approximate central point”⁴ is computed as follows. Let one of the two encodings of the best solution in Ω' be p_{best} . First, since each solution has a pair of encodings, we make a set $S_{\Omega'}$ that contains only one encoding e for each solution in Ω' so that the Hamming distance between e and p_{best} is not greater than $\lfloor |V|/2 \rfloor$. Next, for each position, count the number of 0's and that of 1's for all elements of $S_{\Omega'}$. Make the approximate central point c so that each position of c has the more-frequently-appeared bit. Then, the approximate central point c is closer to the center⁵ of Ω' than p_{best} .⁶ That is, we have the following proposition.

Proposition 1 $\forall p_{best} \in S_{\Omega'}$, let $S_{\Omega'} = \{s_1, s_2, \dots, s_n\}$. Then,

$$\sum_{i=1}^n d(p_{best}, s_i) \geq \sum_{i=1}^n d(c, s_i).$$

Proof: Let $B_j(x)$ be the j^{th} value of x .

$$\begin{aligned} \sum_{i=1}^n d(p_{best}, s_i) &= \sum_{i=1}^n \mathfrak{H}(p_{best}, s_i) \\ &= \sum_{i=1}^n \sum_{j=1}^{|V|} |B_j(p_{best}) - B_j(s_i)| \\ &= \sum_{j=1}^{|V|} \sum_{i=1}^n |B_j(p_{best}) - B_j(s_i)| \\ &= \sum_{j=1}^{|V|} |\{s \in S_{\Omega'} : B_j(s) \neq B_j(p_{best})\}| \end{aligned}$$

³ We define “central point” to be the nearest solution to the center of local-optimum space.

⁴ In this problem, it is not easy to find the exact central point by a simple computation. Each solution has two different encodings. In order to get the distance to other solution, we select one to which the Hamming distance is smaller than the other. The more the solutions, the more complex the whole phase about which encoding is used to calculate the distance.

⁵ Here, the center of Ω' is defined to be the point that has the minimum average distance from the other solutions in Ω' .

⁶ Since the approximate central point obtained in this way can violate balance criterion, adjustment is required. Although not mentioned, the experimental data showed that most of adjusted approximate central points were closer to the center of Ω' than p_{best} .

$$\begin{aligned}
&\geq \sum_{j=1}^{|V|} |\{s \in S_{\Omega'} : B_j(s) \neq B_j(c)\}| \\
&= \sum_{i=1}^n \mathfrak{H}(c, s_i) \\
&\geq \sum_{i=1}^n d(c, s_i).
\end{aligned}$$

Q.E.D.

Although the approximate central points are calculated through a simple computation, it turned out that the costs of the approximate central points are quite attractive (see Tables 1–3). It is amazing that the cut size of the approximate central point without any fine-tuning was sometimes equal to or better than that of the best solution (see the cases of U500.05 and U500.10 in Table 1). In order to check the local optimum near the center, we applied local optimization to the approximate central point. The results are in the row “CP + local opt” of Tables 1–3. In all of the ten cases, the costs of the local optima near the approximate central points were at least as good as those of the best solutions; surprisingly enough, they were better than those of the best solutions in five cases of them. This shows the attractiveness of the central area of the local-optimum space, and provides a motivation for intensive search around the central area.

4 Exploiting Approximate Central Points

We observed in Section 3.2 that the approximate central points obtained by simple computation are quite attractive. In this section, we propose a pseudo-GA that exploits the areas around the approximate central points. Based on the GA, we perform an experiment on how strong exploitation of the central area is desirable.

4.1 A Pseudo-GA That Exploits the Central Areas

Multi-parent crossover is a generalization of the traditional two parent recombination. It was first introduced by Eiben *et al.* [8] and has been extensively studied in the past [7] [6] [22]. But, it is not adequate for problems with multiple representations for a solution like the graph partitioning problem. Unlike previous works, our multi-parent crossover is designed based on the statistical features of problem spaces. The offspring of our multi-parent crossover is exactly the approximate central point of the solutions in the population. Formally, the process of our multi-parent crossover is as follows. Consider the distance measure d defined in Section 3, the parent set P , and a parent p_k in P .⁷ For each parent $a \in P$, if

⁷ Assume that the distance d between any two elements in P is larger than zero.

$\mathfrak{H}(p_k, a) > \lfloor |V|/2 \rfloor$, make a transition that interchanges 0 and 1 at every gene position of a . Let the new set resulted from the transitions be $P' = \{p_1, p_2, \dots, p_n\}$. Then, for each $i = 1, 2, \dots, n$ ($p_i \neq p_k$), $0 < \mathfrak{H}(p_k, p_i) \leq \lfloor |V|/2 \rfloor$. Now, generate an offspring c such that for each $j = 1, 2, \dots, |V|$

$$B_j(c) = \begin{cases} 1, & \text{if } |\{p \in P' : B_j(p) = 1\}| > \lfloor n/2 \rfloor \\ 0, & \text{otherwise} \end{cases}$$

where $B_j(x)$ is the j^{th} gene value of x .

Figure 4 shows the template of pseudo-GA that is designed to exploit the central areas. It is a type of hybrid steady-state genetic algorithm using the multi-parent crossover described above.

- *Initialization and genetic operators:* The GA first creates K local optima at random. We set the population size K to be from 10 to 100. Simply, the total population is selected as the parent set and the GA performs K -parent crossover on them. Then the mutation perturbs the offspring by R percent. Mutation is important in this model since the crossover strongly drives the offspring to the central area; thus an appropriate degree of perturbation is needed.
- *Local optimization:* One of the most common local optimization heuristic for graph partitioning is the FM algorithm. We apply it to the offspring after mutation.
- *Replacement and stopping condition:* After generating an offspring and applying a local optimization on it, the GA replaces the most similar member of the population with the offspring. Maintaining population diversity, a randomly generated local optimum replaces the most similar member of the population per generation. It stops after a fixed number, $(M - K)/2$, of generations.

4.2 Experimental Results

We tested the GA with a number of different population sizes. The population size is denoted by K . K is also the number of parents for crossover; in other words, it is the degree of exploitation around the central area. The values of K represent a spectrum of the exploitation strengths of the central area. If K is equal to M , since it just generates initial population without genetic search, the heuristic equals the multi-start heuristic. The multi-start heuristic returns the best local optimum among a considerable number of local optima fine-tuned from random initial solutions. Although the multi-start heuristic is simple, it has been useful in a number of studies [12] [2].

The experimental results are given in Table 4. We used the FM algorithm as the local optimizer. We set M and R to 1,000 and 20 respectively in all cases and performed 100 runs for each case.

Overall, one can observe that it is helpful to exploit the central areas to some extent. Figure 5 shows two sample plottings, which have roughly bitonic

```

MPGA( $M, K, R$ )
//  $M$ : running time budget,  $K$ : population size, and  $R$ : perturbation rate
{
  for each  $i = 1, 2, \dots, K$  // Generate initial population with size  $K$ 
  {
    Generate a random individual  $P_i$ ;
     $P_i \leftarrow local\_opt(P_i)$ ;
  }
   $B \leftarrow$  the best among population;
  do {
    Make an offspring  $C$  using  $K$ -parent crossover from population;
     $C \leftarrow R\%\_random\_mutation(C)$ ;
     $C^* \leftarrow local\_opt(C)$ ;
    Replace the most similar individual from population with  $C^*$ ;
    Generate a random individual  $T$ ;
     $T \leftarrow local\_opt(T)$ ;
    Replace the most similar individual from population with  $T$ ;
     $B \leftarrow$  the best among  $B, C^*$ , and  $T$ ;
  } until (the number of generations is  $(M - K)/2$ )
  return  $B$ ;
}

```

Fig. 4. A simple genetic algorithm using multi-parent crossover

spectra of performance. The results of Table 4 and Figure 5 show that it is useful to exploit the central area, but that excessive or insufficient exploitation is not desirable.

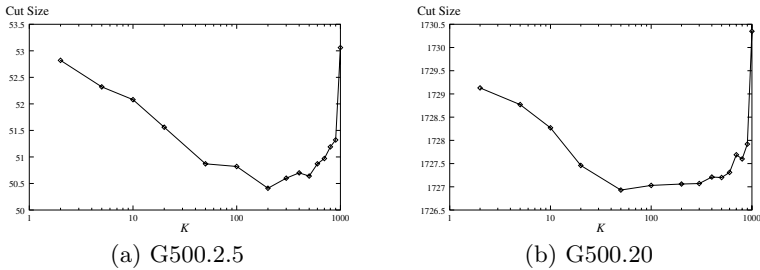


Fig. 5. Two sample spectra extended from Table 4

5 Conclusions

The fitness landscape of the problem space is an important factor to indicate the problem difficulty, and the analysis of the fitness landscape helps efficient search in the problem space. In this paper, we made a number of experiments and got some new insights into the global structure of the graph-partitioning problem space. We extended previous works and observed that the central area of multiple local optima is quite attractive.

Table 4. The comparison of cut sizes

Graphs	$K = 2$		$K = 5$		$K = 10$		$K = 20$			
	Ave [†]	CPU [‡]	Ave [†]	CPU [‡]	Ave [†]	CPU [‡]	Ave [†]	CPU [‡]		
G500.2.5	52.82	1.56	52.32	1.51	52.08	1.56	51.56	1.63		
G500.05	216.09	2.13	215.10	2.09	214.54	2.13	214.70	2.19		
G500.10	622.28	3.41	621.46	3.36	620.64	3.41	619.96	3.45		
G500.20	1729.13	6.77	1728.77	7.53	1728.27	7.62	1727.46	7.20		
G1000.2.5	105.51	3.51	104.28	3.46	103.79	3.53	102.72	3.66		
G1000.05	445.68	4.81	454.79	4.76	452.82	4.87	451.53	5.02		
G1000.10	1363.76	8.63	1361.27	10.02	1359.44	9.65	1358.09	8.59		
G1000.20	3357.62	27.04	3353.72	28.13	3350.87	26.45	3349.23	28.37		
U500.05	7.64	1.95	7.22	1.83	7.06	1.92	6.67	1.98		
U1000.05	24.60	4.84	25.73	4.51	24.35	5.30	22.81	4.87		
U1000.10	41.84	7.18	41.68	6.78	40.64	9.84	40.31	7.35		

Graphs	$K = 50$		$K = 100$		$K = 200$		$K = 500$		Multi-Start [§]	
	Ave [†]	CPU [‡]	Ave [†]	CPU [‡]	Ave [†]	CPU [‡]	Ave [†]	CPU [‡]	Ave [†]	CPU [‡]
G500.2.5	50.87	1.92	50.82	2.40	50.41	3.22	50.64	4.64	53.06	1.63
G500.05	214.12	2.45	213.95	2.92	213.92	3.75	213.78	5.19	217.23	2.20
G500.10	619.55	3.71	619.38	4.24	619.38	5.06	619.94	6.58	623.07	3.54
G500.20	1726.93	9.11	1727.03	8.38	1727.06	8.45	1727.20	11.60	1730.35	8.46
G1000.2.5	100.19	4.25	99.20	5.19	98.38	7.04	97.95	10.30	105.99	3.64
G1000.05	449.27	5.69	448.86	6.59	447.58	8.45	448.15	11.62	457.34	4.95
G1000.10	1356.30	10.10	1354.35	11.20	1354.69	12.17	1355.46	18.19	1365.14	10.21
G1000.20	3346.51	27.57	3345.77	27.56	3344.67	30.15	3345.18	33.71	3363.30	26.80
U500.05	5.60	2.24	5.04	2.71	4.98	3.55	4.87	5.06	7.98	2.07
U1000.05	20.12	5.40	17.28	6.38	14.72	8.28	12.74	11.50	24.09	5.16
U1000.10	40.13	7.74	40.55	8.48	40.86	10.51	40.97	13.58	42.89	8.43

§ $K = M (= 1000)$.

† Average over 100 runs.

‡ CPU seconds on Pentium III 750 MHz.

For the other geometric graphs (U500.10, U500.20, U500.40, U1000.20, and U1000.40) not shown here, all the methods always found the best known.

It seems clear that there are high-quality solutions clustered near the central area of local optima. Hence, it is attractive to exploit the central area. Too much exploitation of the central area perhaps makes the search diversity low. It seems desirable to exploit the central area avoiding excessive or insufficient exploitation. We showed that the performance of search could be improved by a multi-parent crossover based on the exploitation of the central area. The results presented in this paper can also be good supporting data for the previous studies on multi-parent crossover [8] [7] [6] [22]. More theoretical arguments for our empirical results are left for future study.

Our results were achieved in a specific problem, the graph partitioning problem. However, we expect that many other hard combinatorial optimization problems have similar properties. For example, in case of cost-distance correlation, TSP showed similar property to the graph partitioning problem [3]. We hope this study provides a good motivation for the investigation of problem spaces and the design of more effective search algorithms.

Acknowledgments. This work was partly supported by Optus Inc. and Brain Korea 21 Project. The RIACT at Seoul National University provided research facilities for this study.

References

1. R. Battiti and A. Bertossi. Greedy, prohibition, and reactive heuristics for graph partitioning. *IEEE Trans. on Computers*, 48(4):361–385, 1999.
2. K. D. Boese and A. B. Kahng. Best-so-far vs. where-you-are: Implications for optimal finite-time annealing. *Systems and Control Letters*, 22(1):71–78, January 1994.
3. K. D. Boese, A. B. Kahng, and S. Muddu. A new adaptive multi-start technique for combinatorial global optimizations. *Operations Research Letters*, 15:101–113, 1994.
4. T. N. Bui and B. R. Moon. Genetic algorithm and graph partitioning. *IEEE Trans. on Computers*, 45(7):841–855, 1996.
5. S. Dutt and W. Deng. A probability-based approach to VLSI circuit partitioning. In *Design Automation Conference*, pages 100–105, June 1996.
6. A. E. Eiben. Multi-parent recombination, 1997. *Handbook of Evolutionary Algorithms* (pp. 25–33). IOP Publishing Ltd. and Oxford University Press.
7. A. E. Eiben and T. Back. An empirical investigation of multiparent recombination operators in evolution strategies. *Evolutionary Computation*, 5(3):347–365, 1997.
8. A. E. Eiben, P.-E. Raué, and Zs. Ruttkay. Genetic algorithms with multi-parent recombination. In *Parallel Problem Solving from Nature – PPSN III*, pages 78–87, 1994.
9. C. Fiduccia and R. Mattheyses. A linear time heuristics for improving network partitions. In *19th ACM/IEEE Design Automation Conference*, pages 175–181, 1982.
10. L. R. Jr. Ford and D. R. Fulkerson. *Flows in Networks*. Princeton University Press, 1962.
11. M. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, San Francisco, 1979.
12. D. S. Johnson. Local optimization and the traveling salesman problem. In *17th Colloquium on Automata, Languages, and Programming*, pages 446–461. Springer-Verlag, 1990.
13. D. S. Johnson, C. Aragon, L. McGeoch, and C. Schevon. Optimization by simulated annealing: An experimental evaluation, Part 1, graph partitioning. *Operations Research*, 37:865–892, 1989.
14. T. Jones and S. Forrest. Fitness distance correlation as a measure of problem difficulty for genetic algorithms. In *Sixth International Conference on Genetic Algorithms*, pages 184–192, 1995.
15. S. Kauffman. Adaptation on rugged fitness landscapes. *Lectures in the Science of Complexity*, pages 527–618, 1989.
16. B. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *Bell Systems Technical Journal*, 49:291–307, Feb. 1970.
17. Y. H. Kim and B. R. Moon. A hybrid genetic search for graph partitioning based on lock gain. In *Genetic and Evolutionary Computation Conference*, pages 167–174, 2000.
18. S. Kirkpatrick, C. D. Jr. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, May 1983.
19. B. Manderick, M. de Weger, and P. Spiessens. The genetic algorithm and the structure of the fitness landscape. In *International Conference on Genetic Algorithms*, pages 143–150, 1991.

20. Y. G. Saab. A fast and robust network bisection algorithm. *IEEE Trans. on Computers*, 44(7):903–913, 1995.
21. G. B. Sorkin. Efficient simulated annealing on fractal landscapes. *Algorithmica*, 6:367–418, 1991.
22. I. G. Sprinkhuizen-Kuyper, C. A. Schippers, and A. E. Eiben. On the real arity of multiparent recombination. In *Proceedings of the Congress on Evolutionary Computation*, volume 1, pages 680–686, 6-9 1999.
23. E. D. Weinberger. Fourier and Taylor series on fitness landscapes. *Biological Cybernetics*, 65:321–330, 1991.