Journal of Heuristics, 10: 111–133, 2004

Investigation of the Fitness Landscapes in Graph Bipartitioning: An Empirical Study

YONG-HYUK KIM
BYUNG-RO MOON
School of Computer Science & Engineering, Seoul National University, Shillim-dong, Kwanak-gu, Seoul, 151-742, Korea
email: yhdfly@soar.snu.ac.kr
email: moon@soar.snu.ac.kr

Submitted in January 2003 and accepted by Edmund Burke in November 2003 after 1 revision

Abstract

An empirical study is performed on the local-optimum space in graph bipartitioning. We examine some statistical features of the fitness landscape and the local properties of the landscape. They include the distributions of local optima, their cost-distance correlations, their attraction powers, the properties around the central area of local optima, etc. The study reveals some new notable results about the properties of the fitness landscape. For example, the central area yielded good quality in local-optimum space, the local-optimum space had the self-similar structure of global convexity, local optima showed clusters in more than one place, etc. We also provide a simple experiment on whether it is worth to exploit the area around the Euclidean center of the problem space.

Key Words: cost surface, cost-distance correlation, central point, graph bipartitioning, heuristic algorithm

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 the 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 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 (Boese, Kahng, and Muddu, 1994).

A number of studies about the ruggedness and the properties of problem search spaces have been done. Sorkin (1991) defined the fractalness[1] of a solution space and proposed that simulated annealing (Kirkpatrick, Gelatt, and Vecchi, 1983) is efficient when the space is fractal. Jones and Forrest (1995) introduced fitness-distance correlation as a measure of search difficulty. Manderick, de Weger, and Spiessens (1991) measured the ruggedness of a problem space by autocorrelation function and correlation length obtained from a time series of solutions. Weinberger (1991) conjectured that, if all points on a fitness landscape are correlated relatively highly, the landscape is bowl shaped. Boese, Kahng, and Muddu (1994) suggested that, through measuring cost-distance correlation 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 (Boese, Kahng, and Muddu, 1994). Kauffman (1989) proposed the NK-landscape model that can control the ruggedness of a problem space.

In this paper, we propose a number of experiments to analyze problem spaces more elaborately. We examine the distributions of local optima, their cost-distance correlations, their attraction powers, the properties around the central areas of local optima, etc. 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, large-step Markov chain approach, and test graphs. We perform various experiments and analyze fitness landscapes in Section 3. In Section 4, we examine how strongly one needs to exploit the central area of a problem space. 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 (Ford and Fulkerson, 1962). In a roughly balanced bipartitioning problem, 10% of skewness is usually allowed (Saab, 1995). Since it is NP-hard for general graphs (Garey and Johnson, 1979), practically heuristic algorithms are used. These include FM algorithm (Fiduccia and Mattheyses, 1982), a representative linear time heuristic, PROP (Dutt and Deng, 1996) based on probabilistic notion, LG (Kim and Moon, 2004) based on lock gain, etc. In this paper, we consider only roughly balanced bipartitioning problem allowing 10% of skewness.

---

do {
    Compute gain g_v for each $v \in V$;
    Make gain lists of g_vs;
    $Q = \phi$;
    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 $k \in \{1, \ldots, |V| - 1\}$ that maximizes $\sum_{i=1}^{k} g_{v_i}$;
    Move all the vertices in the subset $\{v_1, ..., v_k\}$ to their opposite sides;
} **until** (there is no improvement)

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

2.2. *Fiduccia-Mattheyses algorithm (FM)*

Fiduccia and Mattheyses (1982) introduced a heuristic for roughly balanced bipartitioning

problem. The FM algorithm as well as the Kernighan-Lin algorithm (KL) (Kernighan and Lin, 1970) 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. *Large-step Markov chain (LSMC)*

Martin, Otto, and Felten (1991) proposed the large-step Markov chain (LSMC) method for the traveling salesman problem (TSP). It iteratively applies a local optimization algorithm, and then perturbs the resulting local optimum by a *kick move* to obtain the starting solution for the next local optimization. LSMC actually performs simulated annealing (SA) (Kirkpatrick, Gelatt, and Vecchi, 1983) over the set of local optima, with "kick move + local optimization" as its neighborhood operator. Like SA, it optimally uses the Boltzmann acceptance criterion in adopting a new local optimum. A number of studies about LSMC have been done (Hong, Kahng, and Moon, 1997; Fukunaga, Huang, and Kahng, 1996). A study about various kick moves for hypergraph partitioning was presented in Fukunaga, Huang, and Kahng (1996). In our experiments later, we use random perturbation for the kick move, and use zero-temperature LSMC as in Martin, Otto, and Felten (1991) and Fukunaga, Huang, and Kahng (1996) that is, a new local optimum is accepted only when it is better than its predecessor. LSMC will be used in Section 3.5 for the study of attraction powers around local optima and in Section 4 for comparing with our proposed method.

2.4. *Test beds*

We tested on a total of 17 graphs which consist of two groups of graphs. They are composed of 17 graphs from Johnson et al. (1989) (9 random graphs and 8 geometric graphs). The two classes were used in a number of other studies (Lim and Chee, 1991; Saab, 1995; Bui and Moon, 1996; Moon and Kim, 1997; Steenbeek, Marchiori, and Eiben, 1998; Battiti and Bertossi, 1998, 1999; Merz and Freisleben, 1998; Kim and Moon, 2004). The classes are briefly described below.

1. G*n.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. U$n.d$: 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 $d = n\pi t^2$ is the expected vertex degree.

3. Investigation of the problem space

In this section, we make new attempts to examine properties of the problem space. We first extend the experimentation of Boese, Kahng, and Muddu (1994) 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 and propose a supportive experiment to observe the self-similar fractal structure of global convexity in problem spaces. We then measure the distribution of local optima and their attraction powers. 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 $\mathfrak{a}, \mathfrak{b} \in U$, we define the *distance* between $\mathfrak{a}$ and $\mathfrak{b}$ as follows:[2]

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

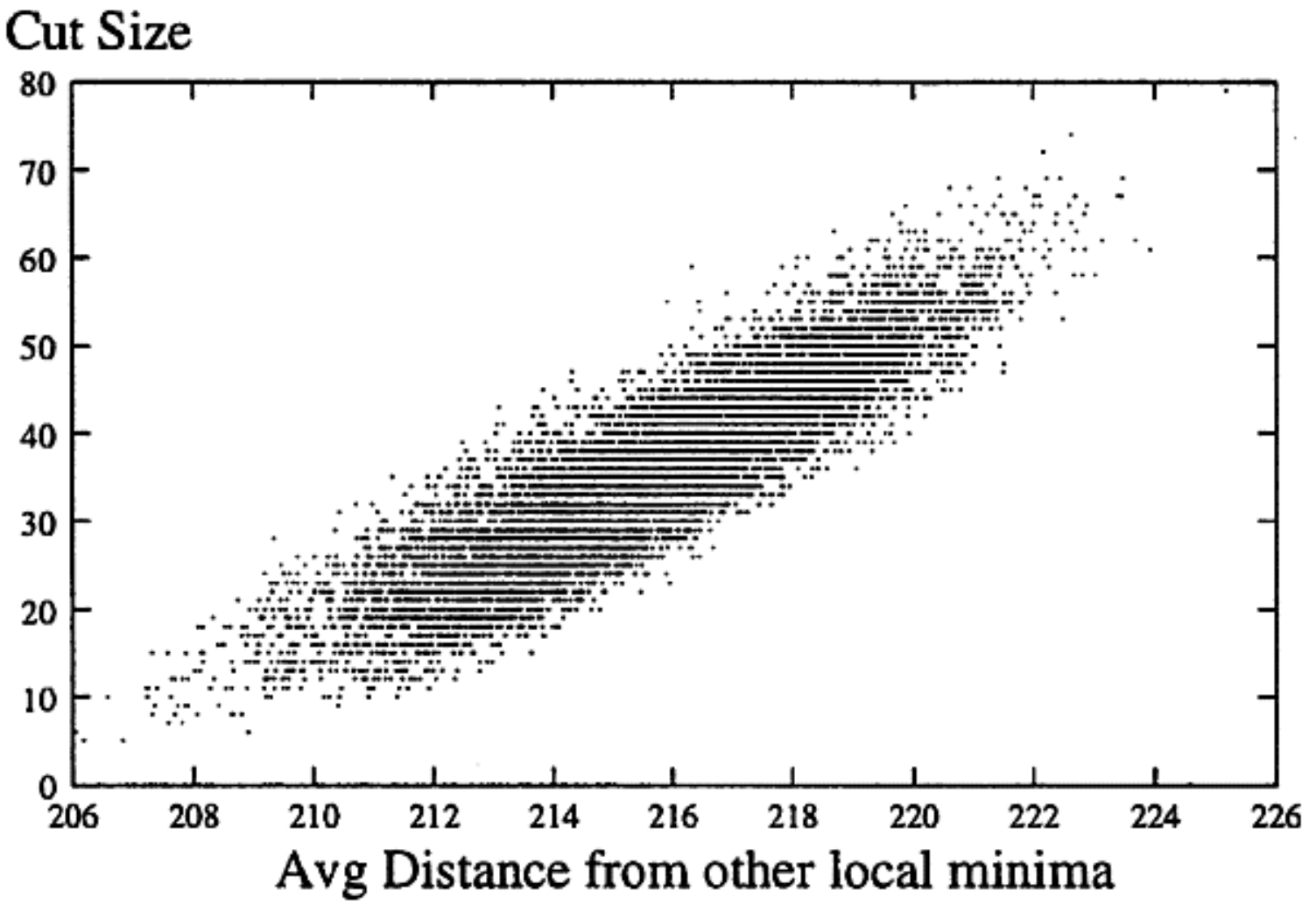
where $\mathfrak{H}$ is the Hamming distance.

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

3.1. *Cost-distance correlation*

Given a set of local minima, Boese, Kahng, and Muddu (1994) 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 provide some additional information and statistics.

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. Figures 2 and 3 show the plotting results for two graphs (U500.05 and 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"



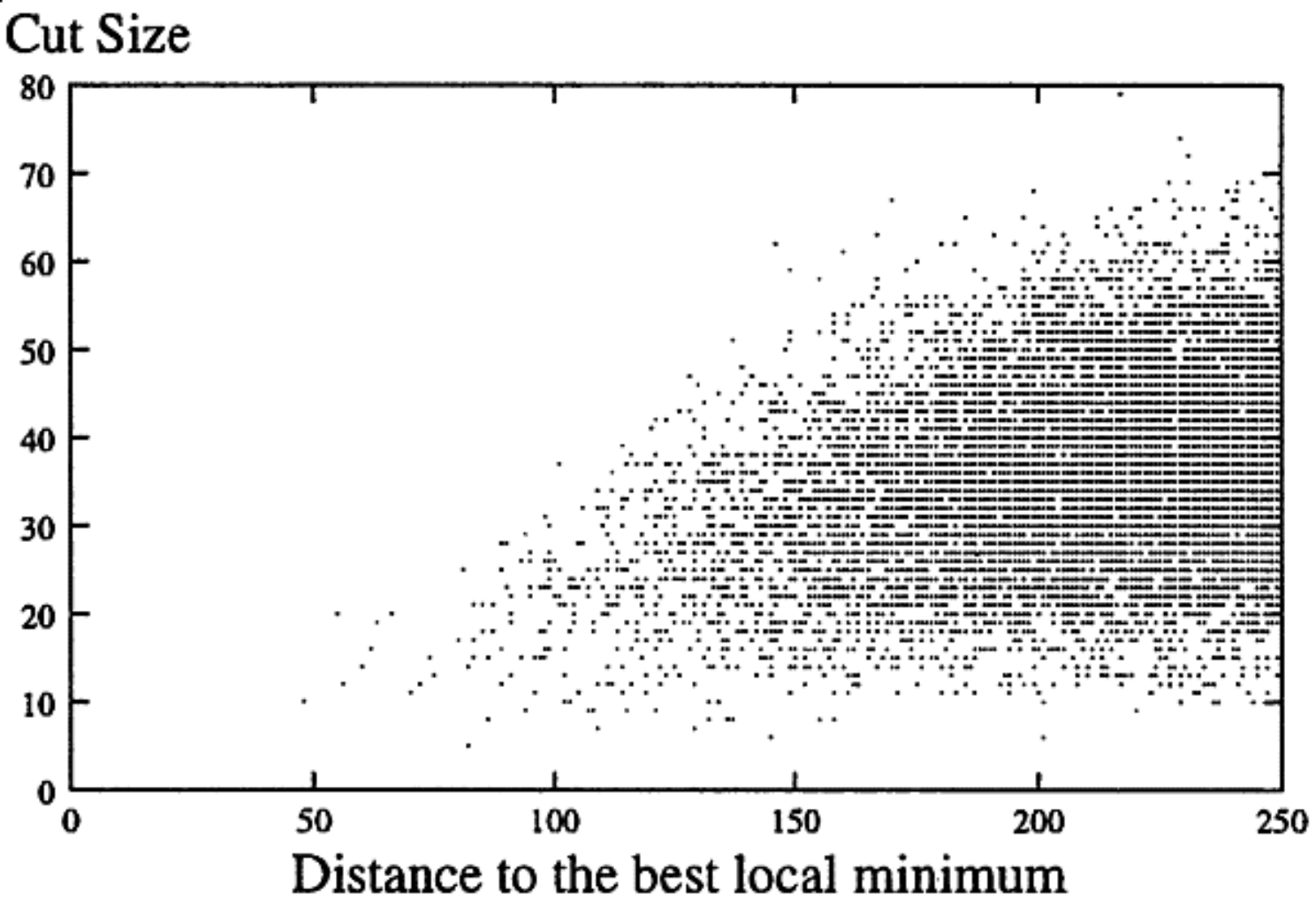


Figure 2. Relationship between cost and distance: U500.05 (see Table 1).

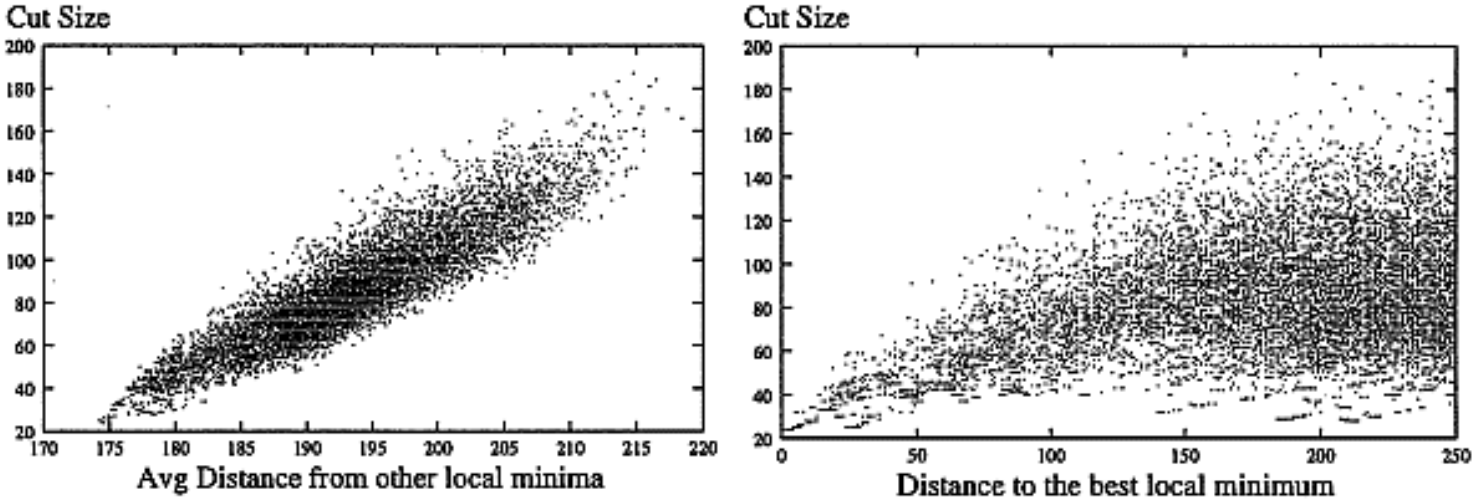


Figure 3. Relationship between cost and distance: U500.10 (see Table 1).

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.

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 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.

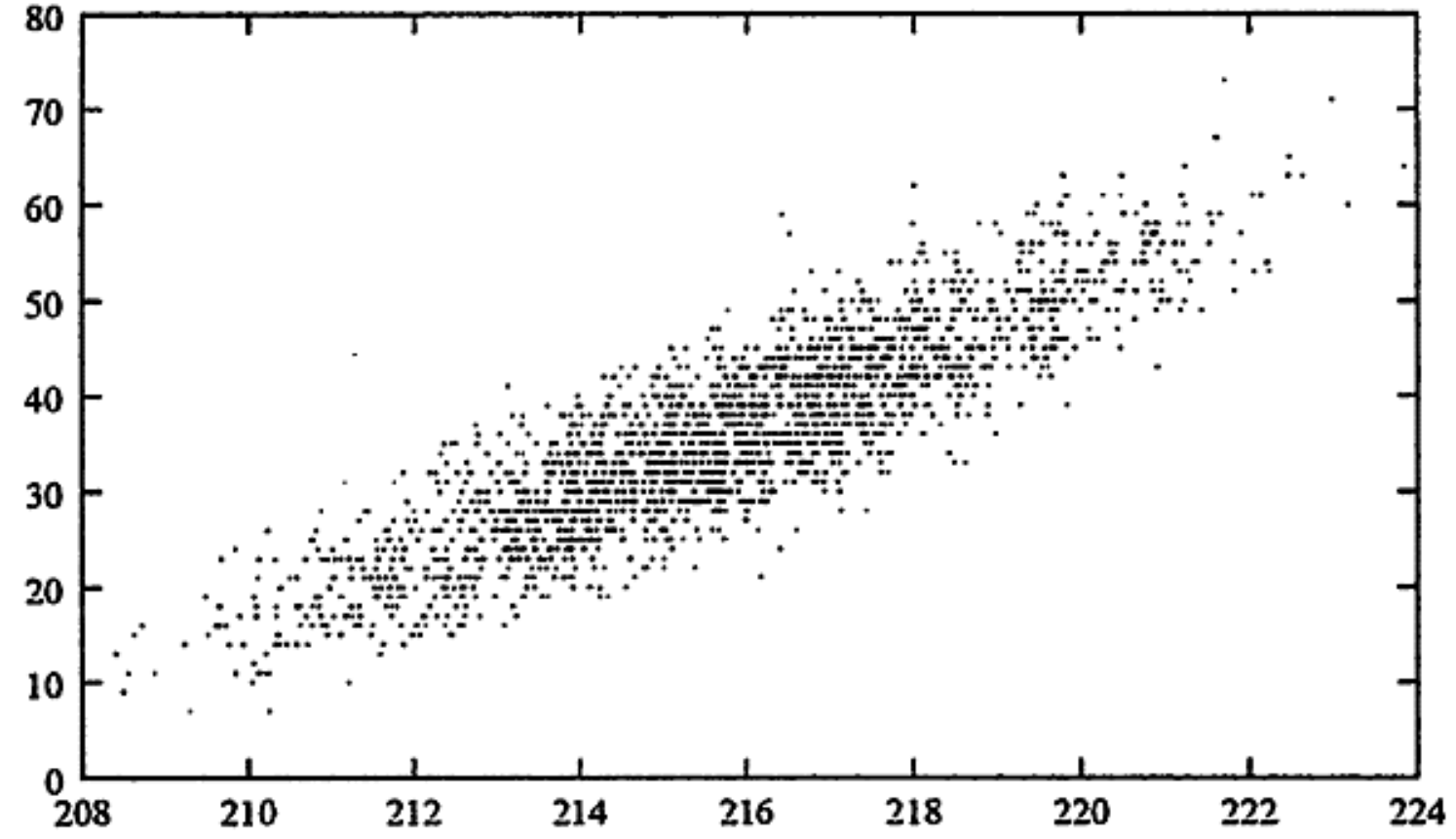
Figure 4, Tables 2 and 3 compare the data with different local optimizers. A greedy local optimizer which moves only vertices with positive gain was named GREEDY. Its principle is the same as that of the steepest descent algorithm in the differentiable cases. NONE means a set of random solutions without any local optimization. From the cut sizes in Tables 2 and 3, FM is clearly stronger than the GREEDY algorithm. The stronger the local optimizer, 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 4, 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.[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

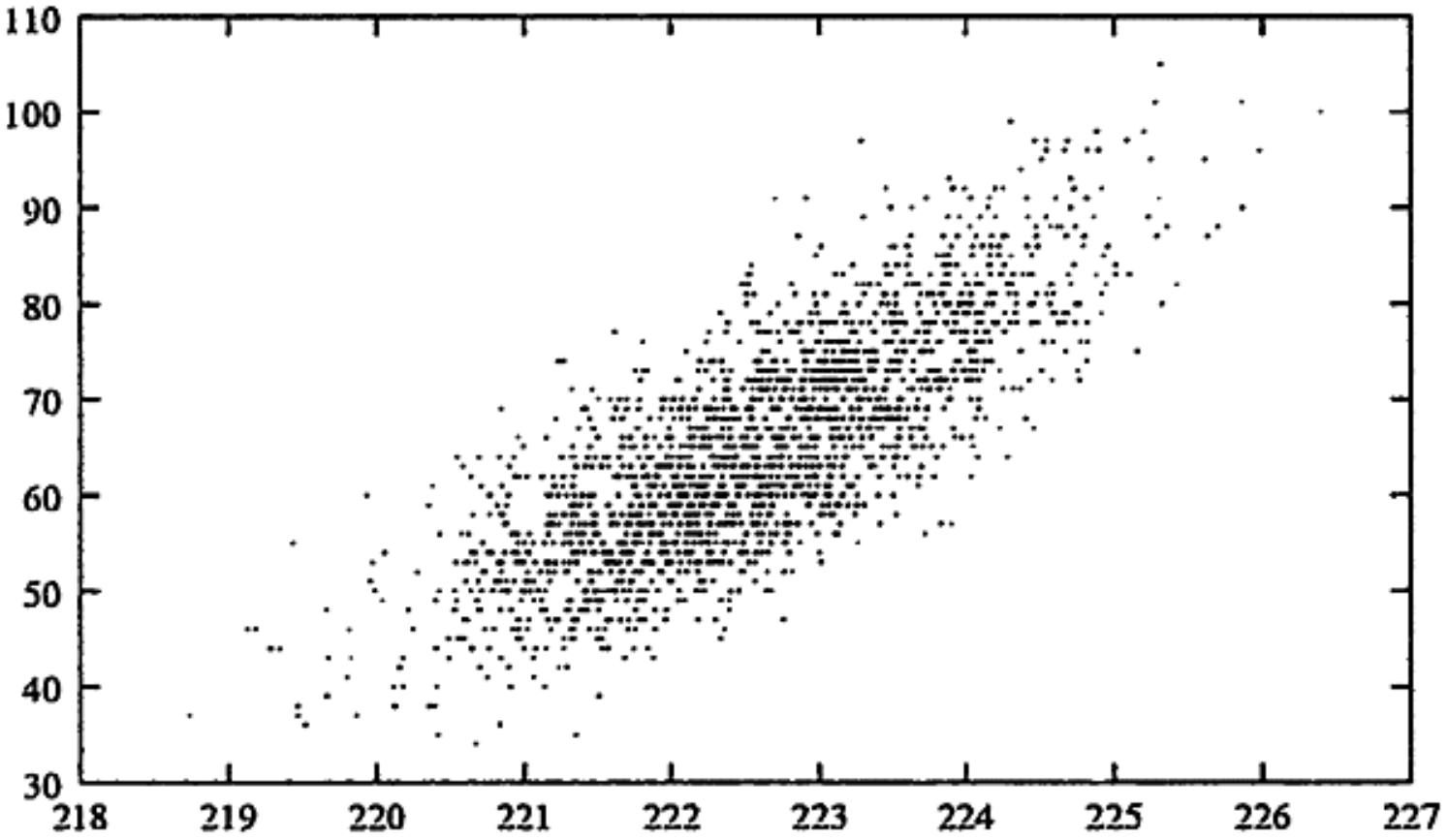
Cut Size



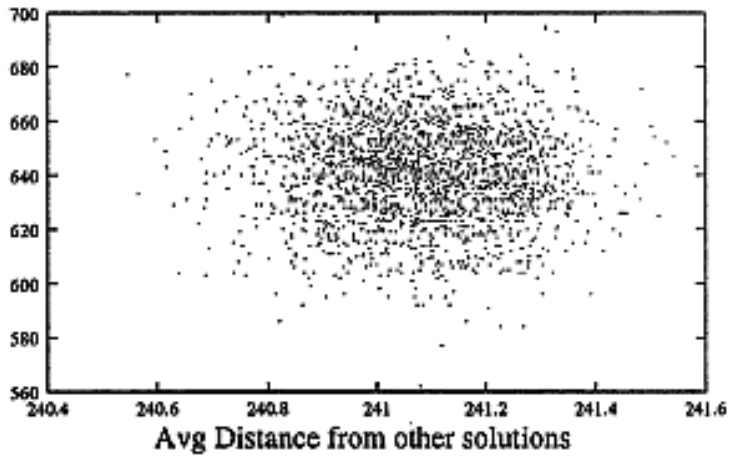
(a) FM

Cut Size

Cut Size



(b) GREEDY



(c) NONE

Figure 4. Relationship between cost and distance with different local optimizer in the graph U500.05 (see Table 3).

3.2. Approximate central point

The results of Boese, Kahng, and Muddu (1994) 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 point"[4] of

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

the subspace may be near the optimal solution. Hence, computing the "central point" not only supports the results of Boese et al. but also may be helpful for obtaining the optimal solution. In this subsection, we propose a heuristic method to find the central point of the local-optimum space in the graph bipartitioning problem.

Given a subspace Ω' of the whole solution space in the graph bipartitioning problem, the "approximate central point"[5] is computed as follows. First, since each solution has a pair of encodings, we make the set $S_{\Omega'}$ that contains only the encodings with zero in the first position, i.e., $S_{\Omega'} \subset \{0\} \times \{0, 1\}^{|V|-1}$. Next, let the encoding of the best solution in $S_{\Omega'}$ be p_{best}. For each a in $S_{\Omega'}$, if the Hamming distance between a and p_{best} is more than $\lfloor |V|/2 \rfloor$, take the mirror solution a' by reversing 0's and 1's. Then, the Hamming distance between a' and p_{best} is always not greater than $\lfloor |V|/2 \rfloor$. Let $S'_{\Omega'}$ be the set that consists of all such encodings. 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 center of Ω' than p_{best}.[6] The proof of this proposition is given in Appendix A.

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.

3.3. *Fractal structure of global convexity*

Here, a fractal structure means self-similar substructures. In a globally convex problem space, we experimentally check such similarity as in figure 5. Consider, for each solution


```
// Let the local-optimum space Ω and the positive constant ε be given.
for each solution s ∈ Ω
{
    Choose ε-ball K_ε(s) centered at s, where K_ε(s) = {α ∈ Ω : d(s,α) ≤ ε};
    If the solution s is the minimal solution in subspace K_ε(s) and |K_ε(s)| is "not too small"†,
        then make the same experiment as Section 3.1 for the subspace K_ε(s);
}
```

† We mean that the experimental data for subspace are creditable. We set the size to some integer larger than 30.

Figure 5. The test of self-similarity.

s in the problem space, a subspace centered at s. If the solution s is the minimum in the subspace, we call this subspace a *hole subspace*. Given a local-optimum space, we select all hole subspaces with a proper size. If the hole subspace is not too sparse, we perform the same experiment as Section 3.1 with the local optima in the hole subspace.

Roughly speaking, if each hole subspace also has a positive cost-distance correlation, it implies that the problem space has similar subspaces. In the experiment, it is important to choose "proper" neighborhood. The number of hole subspaces is too low if the radius is too large, and the density of hole subspaces is too small if the radius is too small.[7] For our experimental data, we use only hole subspaces that have more than 30 local optima. Generally speaking, given a space Ω with a cost function, if we show that M is globally convex for every hole subspace M, we may guess that Ω has the fractal structure of global convexity.

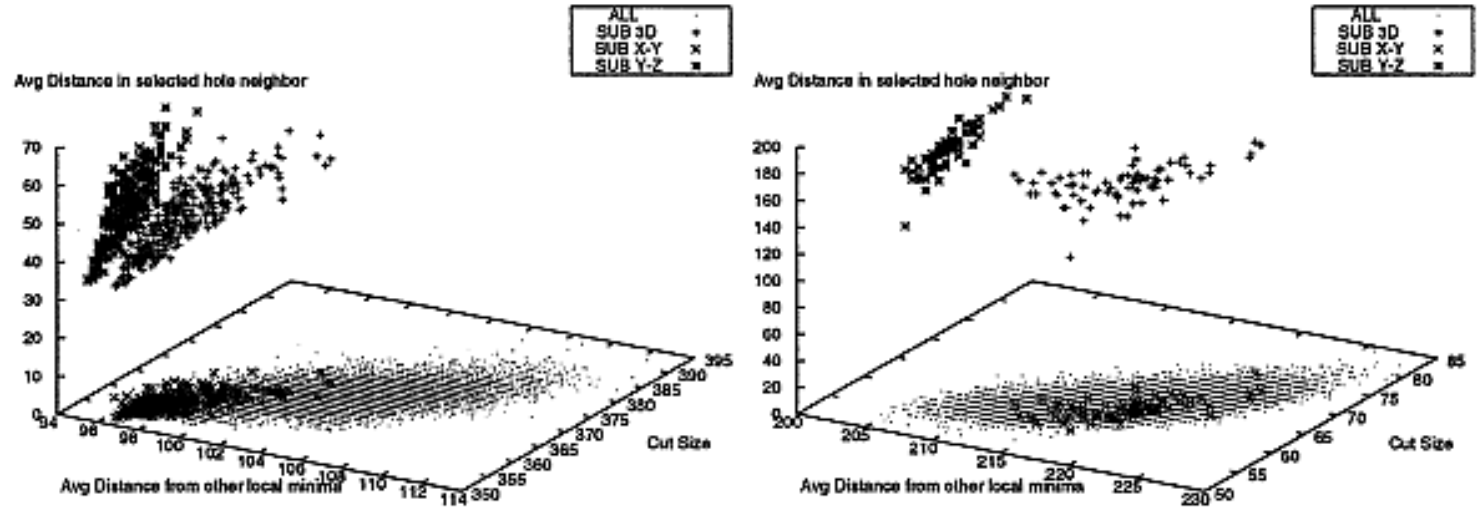
The experimental data are given in Table 4 and figure 6. In figure 6, we selected an arbitrary

hole subspace among all subspaces found by the method of figure 5 for each graph. First, the symbol '+' in the figure indicates each element of the hole subspace with the coordinate "⟨average distance from the other local minima in the whole solution space(X), cut size(Y), average distance from the other local minima in the hole subspace(Z)⟩." In the XY plane, the elements of the whole solution space are plotted by dot('·') and those of the hole subspace are indicated by the symbol '×' in order to compare the distributions of the whole space

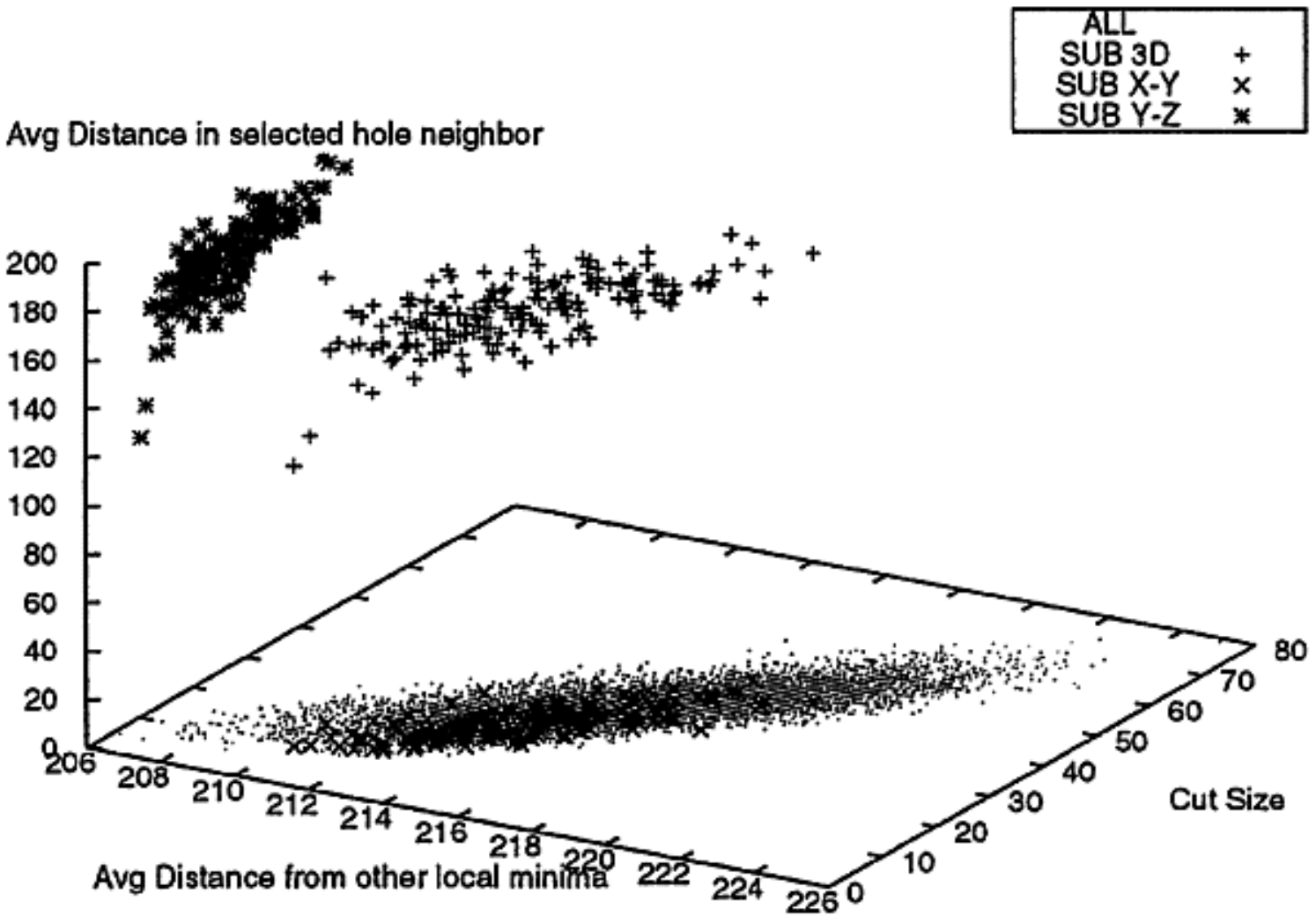
Table 4. The hole subspace for each graph.

Items	G250.10	G500.2.5	U500.05	U500.10
Population size	9877	10000	10000	9302
Cost-distance correlation	**0.77**	**0.78**	**0.89**	**0.91**
Radius	$0.188\|V\|$	$0.281\|V\|$	$0.266\|V\|$	$0.125\|V\|$
Number of holes	20	12	5	7
Ave. density	100.95	229.00	199.40	191.43
Ave. corr. in holes	**0.62**	**0.45**	**0.46**	**0.70**

Radius: the radius of ball, i.e., the maximum value of $d(\cdot, hole)$; Number of holes: the number of hole subspaces; Ave. density: the average number of local minima in a hole subspace; Ave. corr. in holes: the average correlation in hole subspaces.

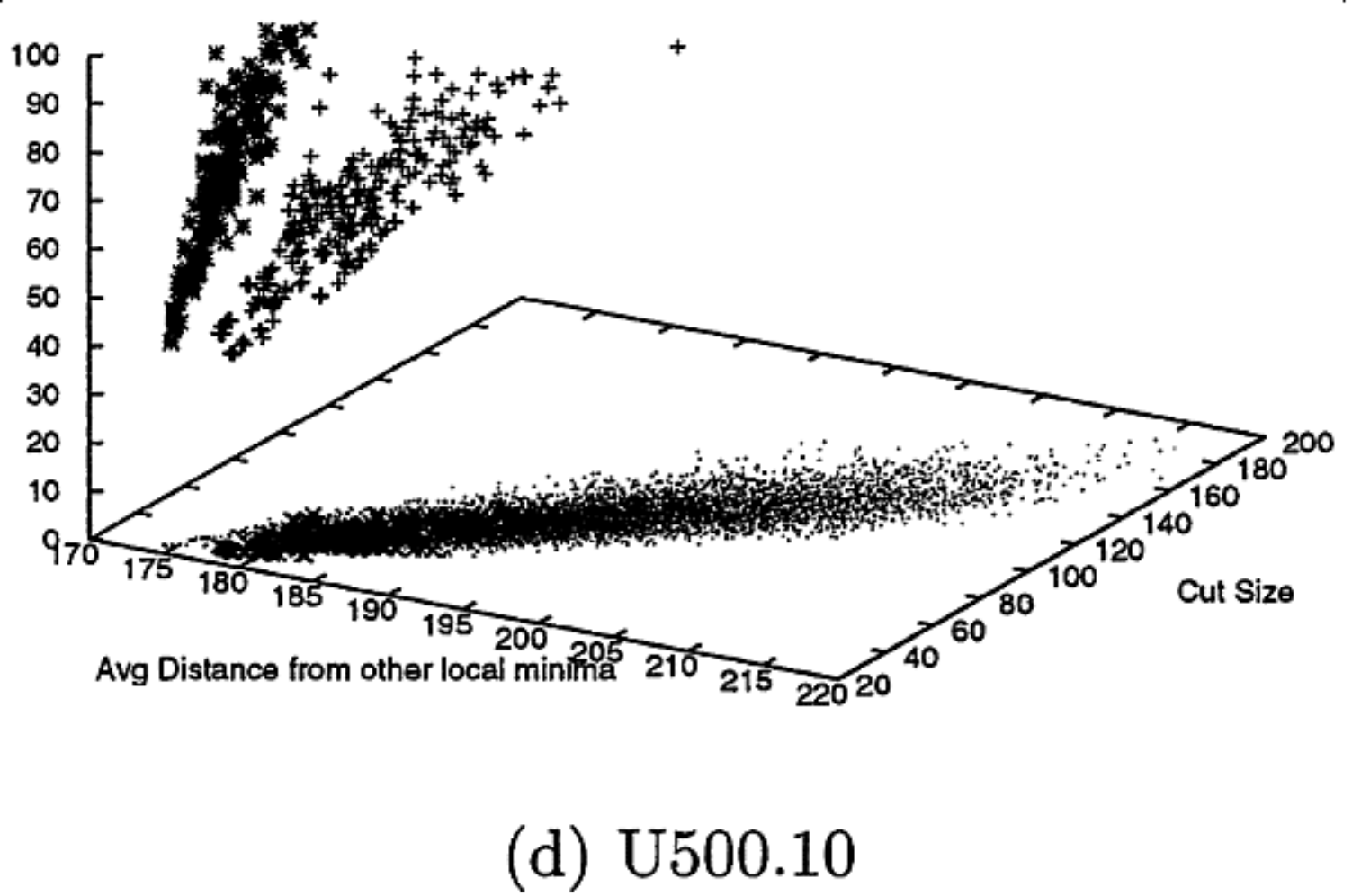


(a) G250.10



(c) U500.05

(b) G500.2.5



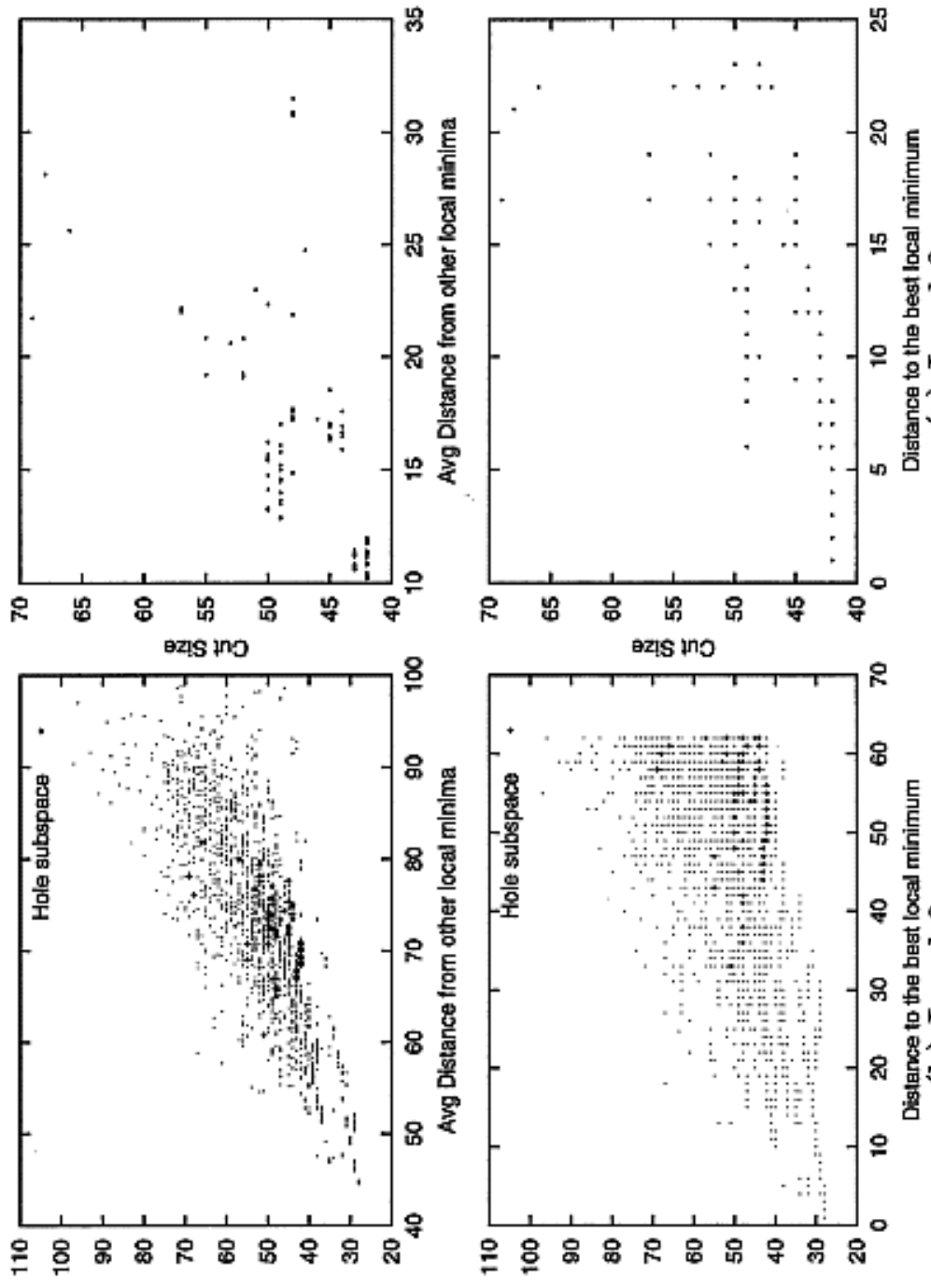
(d) U500.10

Figure 6. Relationship between cost and distance in a hole subspace (see Table 4).

and the hole subspace. That is, the local optima in the hole subspace are projected to the XY plane and indicated by the symbol '$\times$'. In the YZ plane, the solutions of the hole subspace are projected and indicated by the symbol '$*$'. This plotting in the YZ plane shows the cost-distance relationship of the solutions inside the hole subspace. The figures show that hole subspaces also have strong correlation similarly to the whole spaces although their correlations are not as strong as those of the whole spaces.

Table 4 shows some figures relevant to figure 6. The meaning of each item is as follows. "Radius" means the size of subspace. "Number of holes" means the number of hole subspaces found from the experimental data. "Ave. density" is the average number of local minima in a hole subspace. "Ave. corr. in holes" is the average cost-distance correlation in hole subspaces. The numerical data show that the overall correlations of subspaces are smaller than those of the whole spaces. They are also different depending on graphs. However, it is clear that hole subspaces still maintain strong positive correlations.

Our next experiment tests the consistency of convexity as we hierarchically extract smaller and smaller subspaces. Figure 7 shows an example with the graph U500.10. In figure 7(a), the



(b) Level 2

(c) Level 3

able 5).

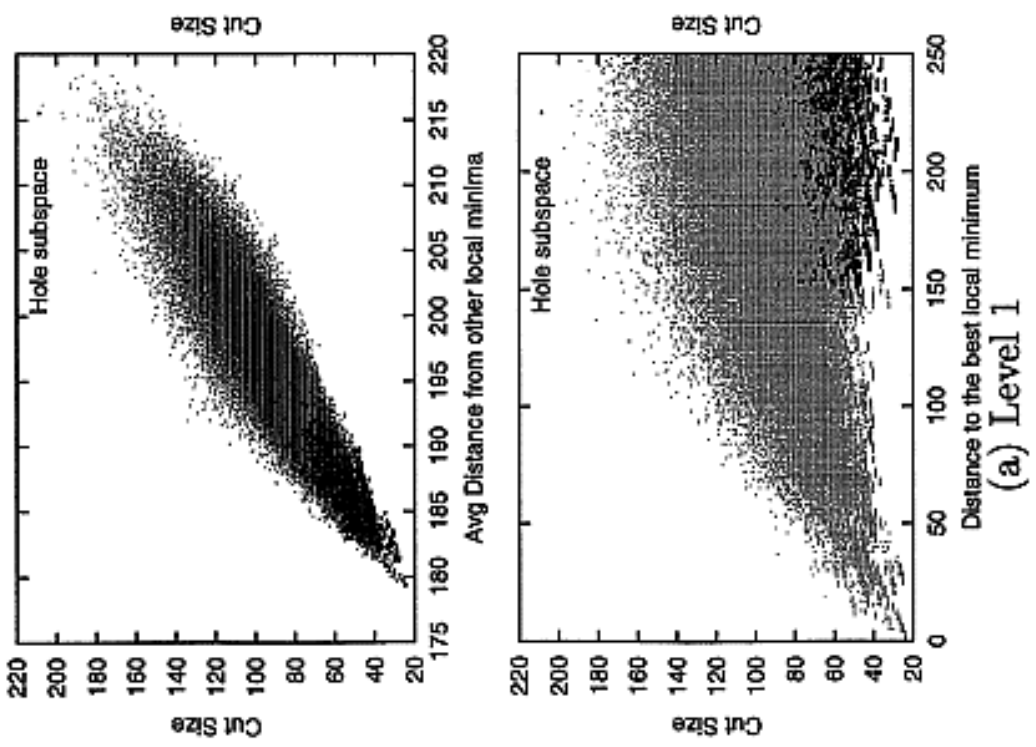


Figure 7. The self-similarity of the graph U500.10 (see T

Table 5. The results for each level in the graph U500.10.

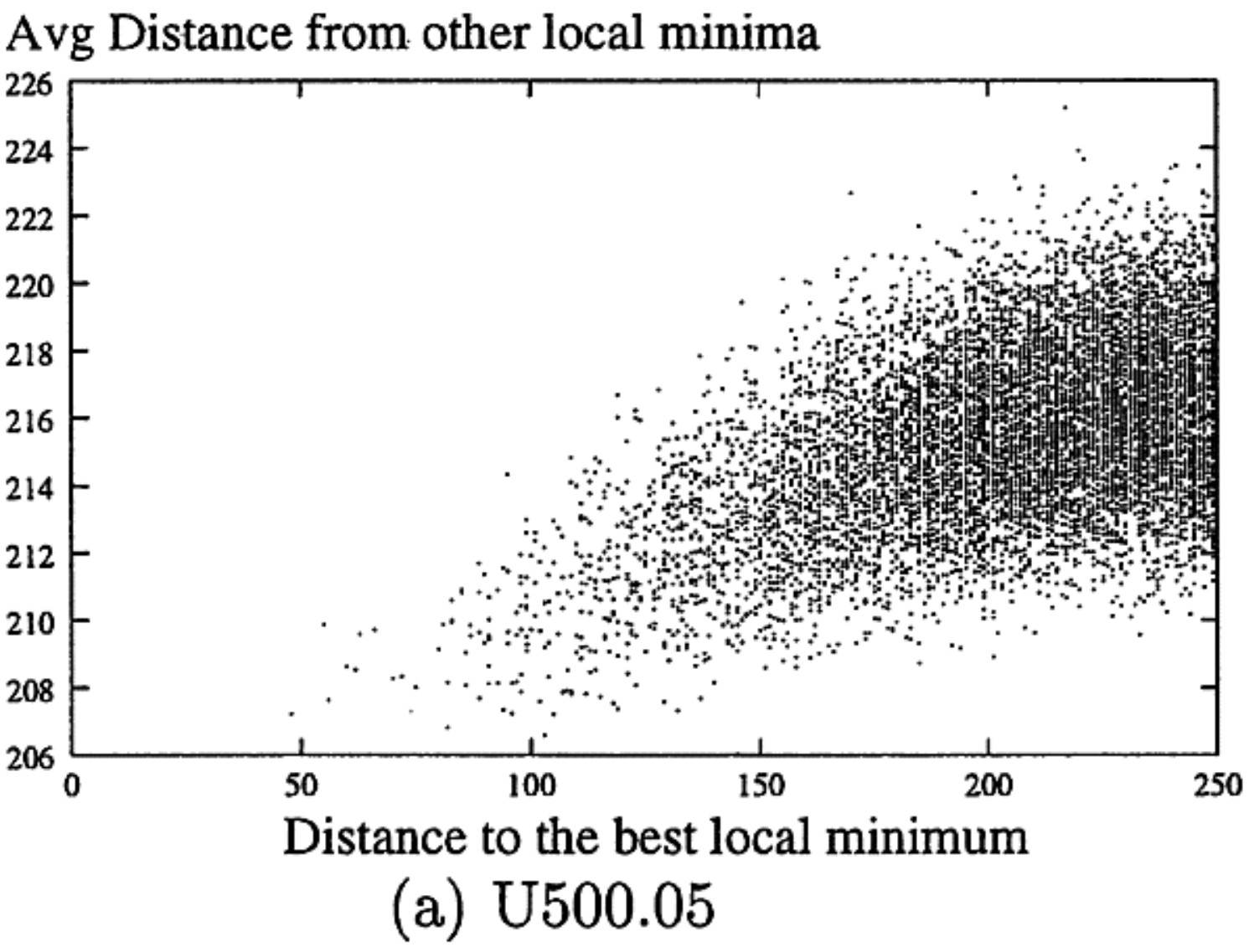
Items	Level 1	Level 2	Level 3
Population size	91229†	2260	116
Best cut	24	28	42
Average cut	87.79	50.65	46.33
Cost-distance correlation	**0.92**	**0.80**	**0.69**
Average distance	194.88	71.44	14.96
Radius	$0.500\|V\|$	$0.125\|V\|$	$0.047\|V\|$

†Data from removing duplicated 18,771 local optima among 110,000 local optima.

cost-distance relations for 91,229 local optima are plotted. Figure 7(b) shows the plottings for a hole subspace extracted from the space used in (a). Similarly, figure 7(c) shows those for a hole subspace extracted from the subspace in (b). The bold points in (a) represent the points extracted for (b); the bold points in (b) represent the points extracted for (c). One can see that each set has a strong positive correlation. The convexity remains consistent in each level of spaces. Table 5 shows the figures relevant to figure 7.

3.4. *Distribution of local optima*

As a result of the experiments of Boese, Kahng, and Muddu (1994), we agree with the conjecture about the global convexity of local-optimum space but it is difficult to obtain more detailed deduction. In this subsection, we conduct experiments to get more detailed induction. Figure 8 shows the relationship between the distance to the best local minimum and the average distance from the other local minima for each local minimum in the local-optimum space.[8] In the figure, there are considerably many solutions such that they are



(a) U500.05

Avg Distance from other local minima

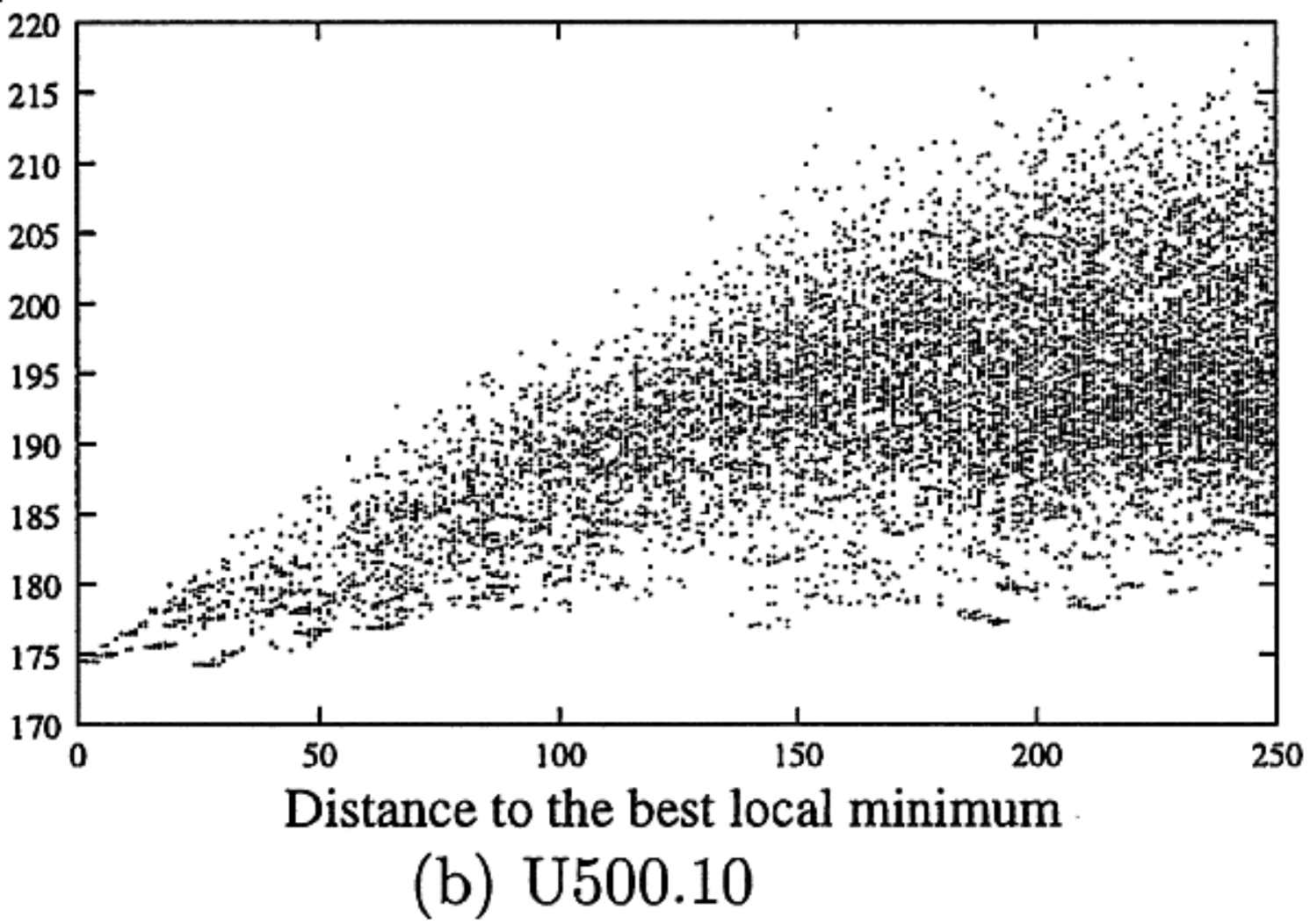


Figure 8. Relationship between distance to the best local optimum and average distance from the other local minima.


```
// Let the local-optimum space Ω and the positive constant ε be given.
for each solution s ∈ Ω
{
    Choose ε-ball K_ε(s);
    Compute its cardinality |K_ε(s)|;
    With the proper function p(s) that can identify the location of s in the whole space, plot (p(s), |K_ε(s)|);
}
```

Figure 9. Plot the distribution of local optima.

far from the best solution but their average distances are small. We suspect that this fact suggests solutions are clustered in more than one place. In this subsection, we intend to understand their distributions in more detail. The experimental method is given in figure 9.

For each solution s in the problem space, we choose a ball centered at s. Figure 10 plots the densities of the balls. Figure 10 shows that the density of local optima near the central point is remarkably high. One can see that the local optima are more sparsely distributed as they locate distant from the central point. One can also observe that local optima are clustered in more than one place. Although ubiquitous, U500.10 shows the strongest evidence among the graphs in figure 10. High-density balls are often observed in the areas far from the center. It suggests the existence of "medium valleys"[9] or "small valleys." It could not be explained by the experimental methods such as Boese, Kahng, and Muddu (1994). Table 6 shows the statistical data for the experimental result of figure 10. In the table, "Density" means the average number of elements in each ball.

3.5. Local optima as attractors

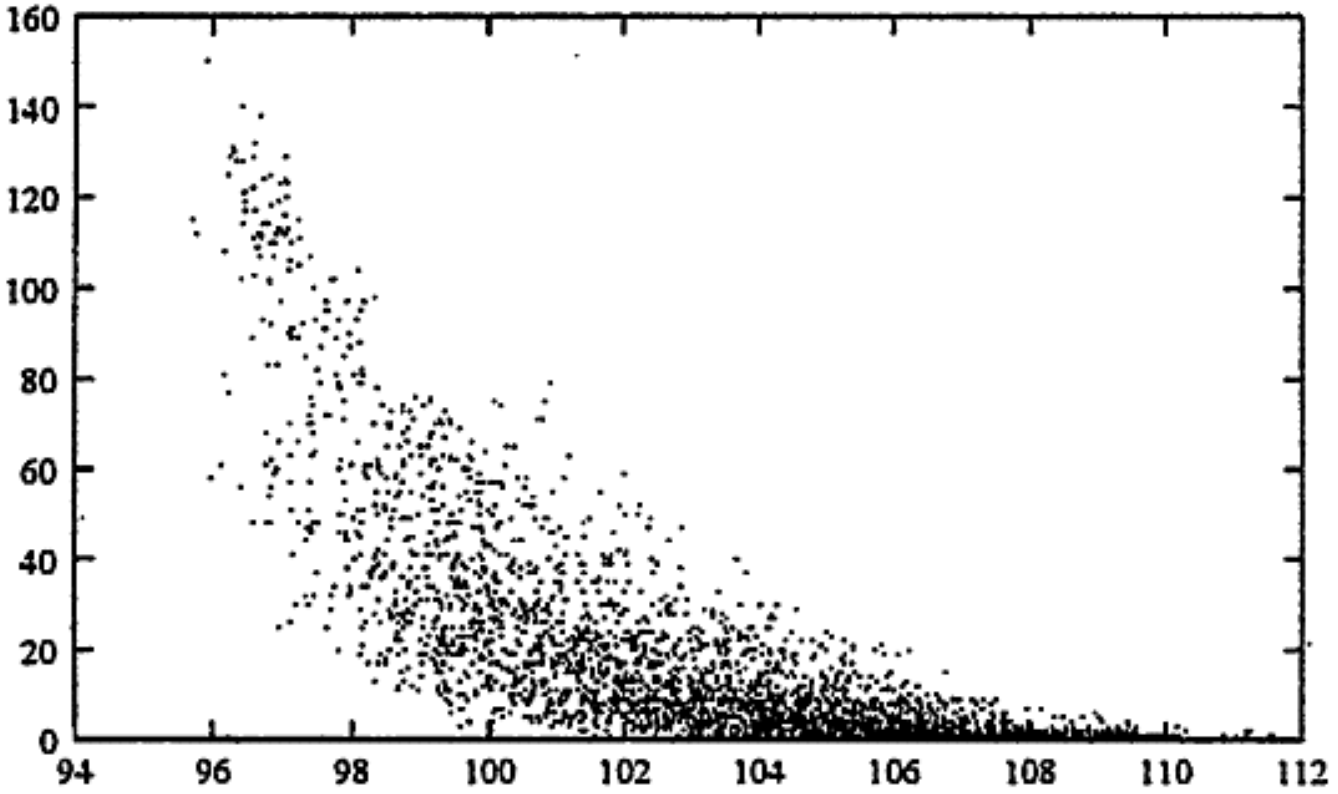
A local optimum is obtained by a local optimizer. Since each local optimum is a solution stabilized by the local optimizer in the whole solution space, it plays a role of *attractor* in its neighborhood. We conducted experiments to measure the attraction powers of local optima. The experimental method is given in figure 11. We perturb each local optimum by reversing i percent of the bits and apply the local optimizer to the perturbed solution to obtain a local optimum. Then, we compute the distance d between the new local optimum and the previous one, and plot the relation between the size of perturbation and the distance d. If there is no attraction power, the distance d is expected to remain the same as the size of perturbation.

Table 6. Statistical data of figure 10.

Items	G250.10	G500.2.5	U500.05	U500.10
Population size	2989	3000	3000	2902
Radius	$0.188\|V\|$	$0.281\|V\|$	$0.266\|V\|$	$0.125\|V\|$
Density	17.35	21.66	18.12	22.73
Average distance	102.97	217.00	215.68	191.98

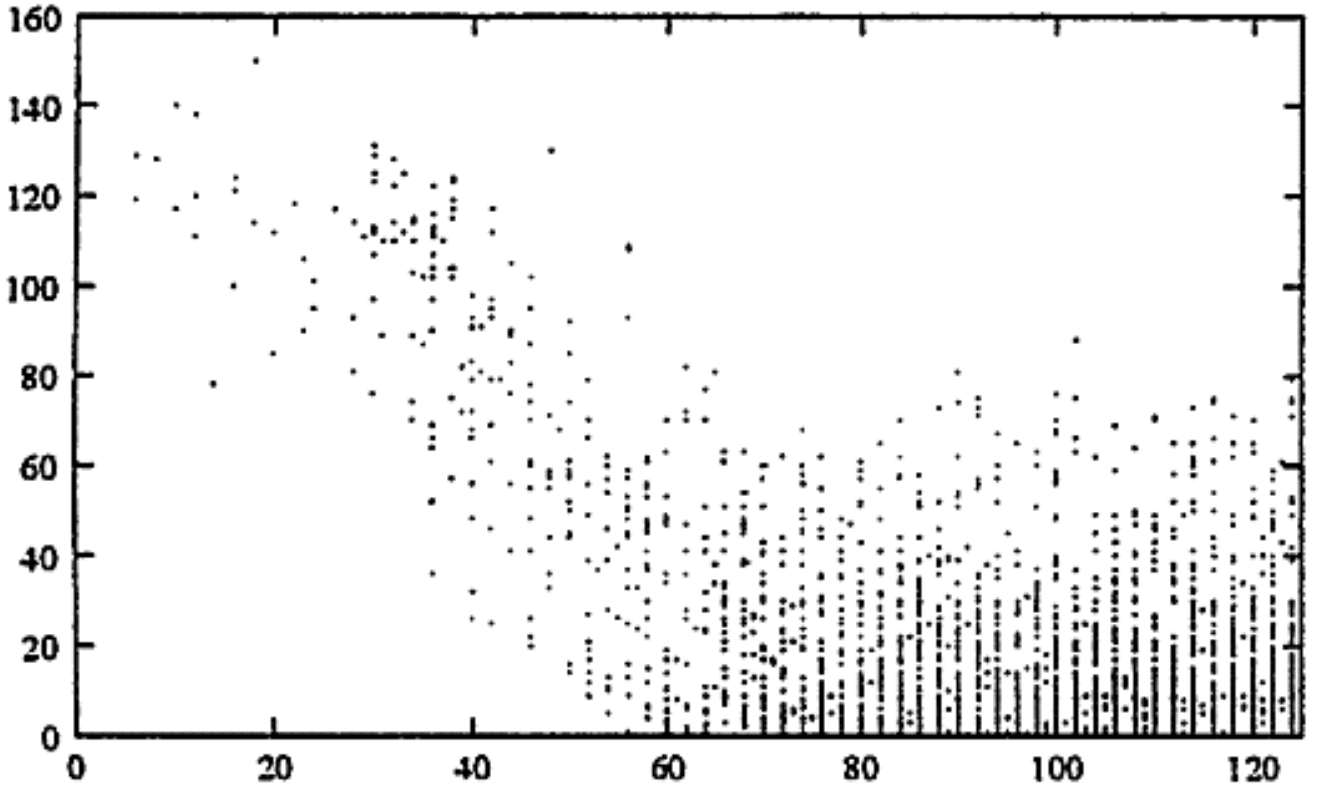
Density: the average number of local minima in a subspace.

Density of Subspace



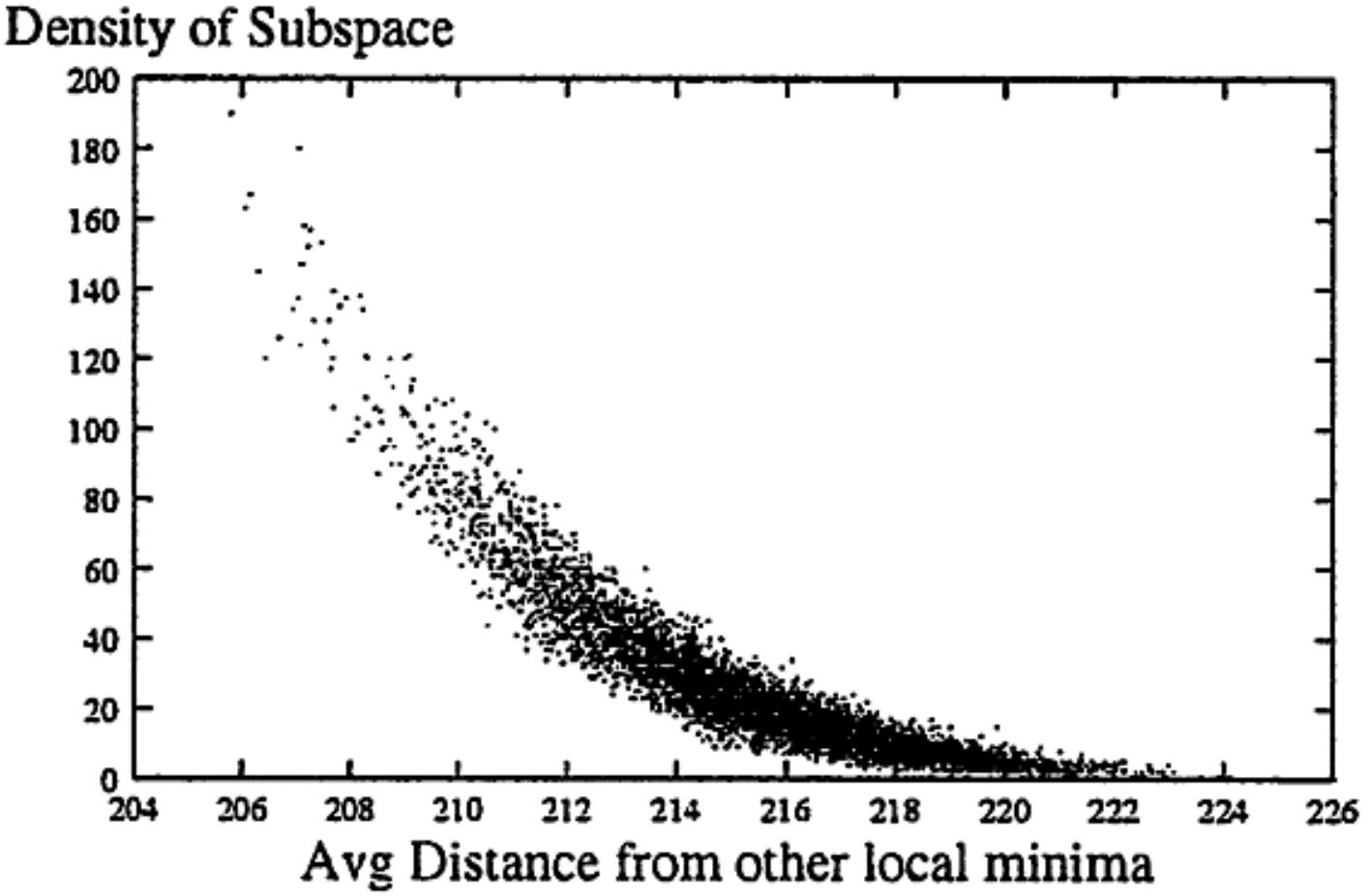
Avg Distance from other local minima

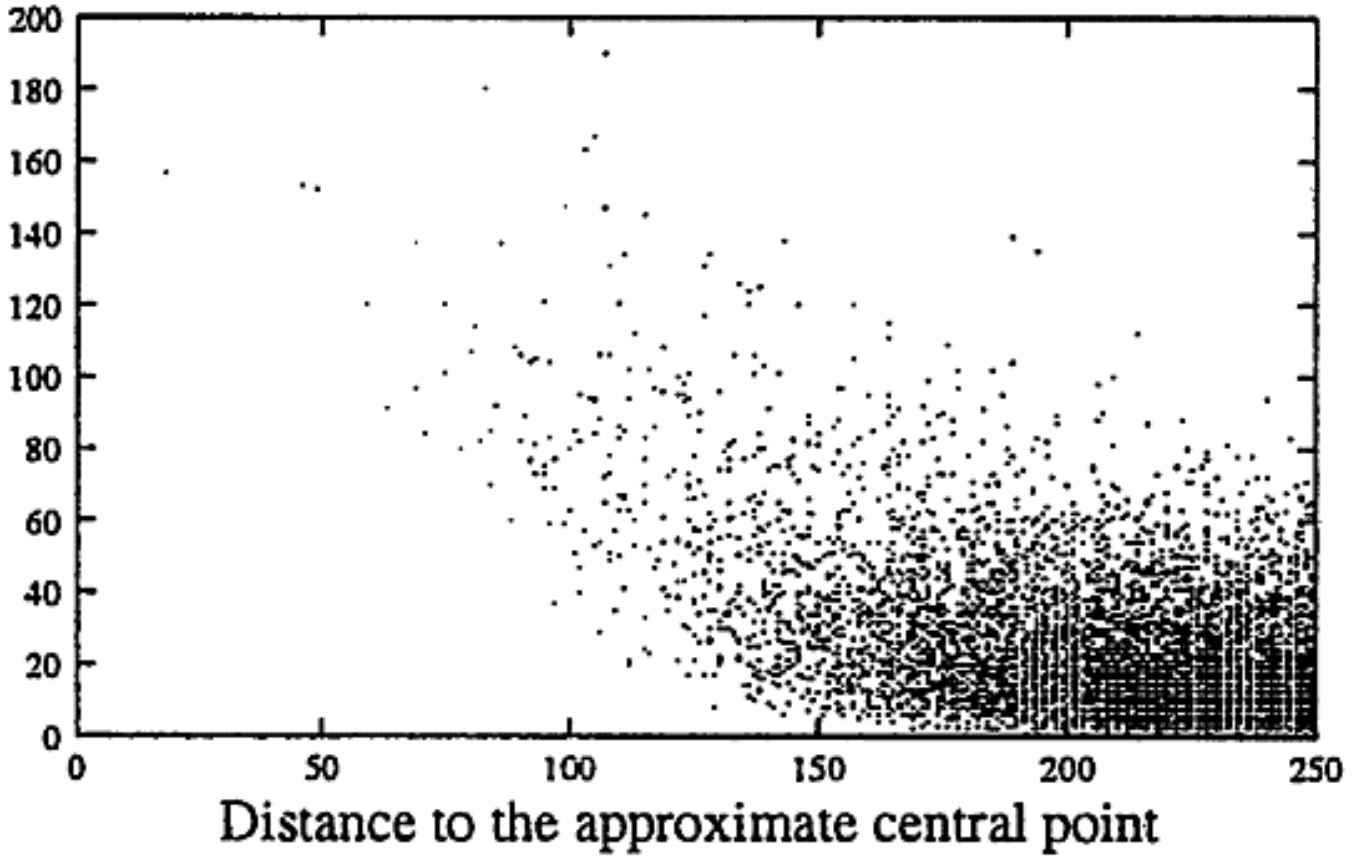
Density of Subspace



Distance to the approximate central point

(a) G250.10

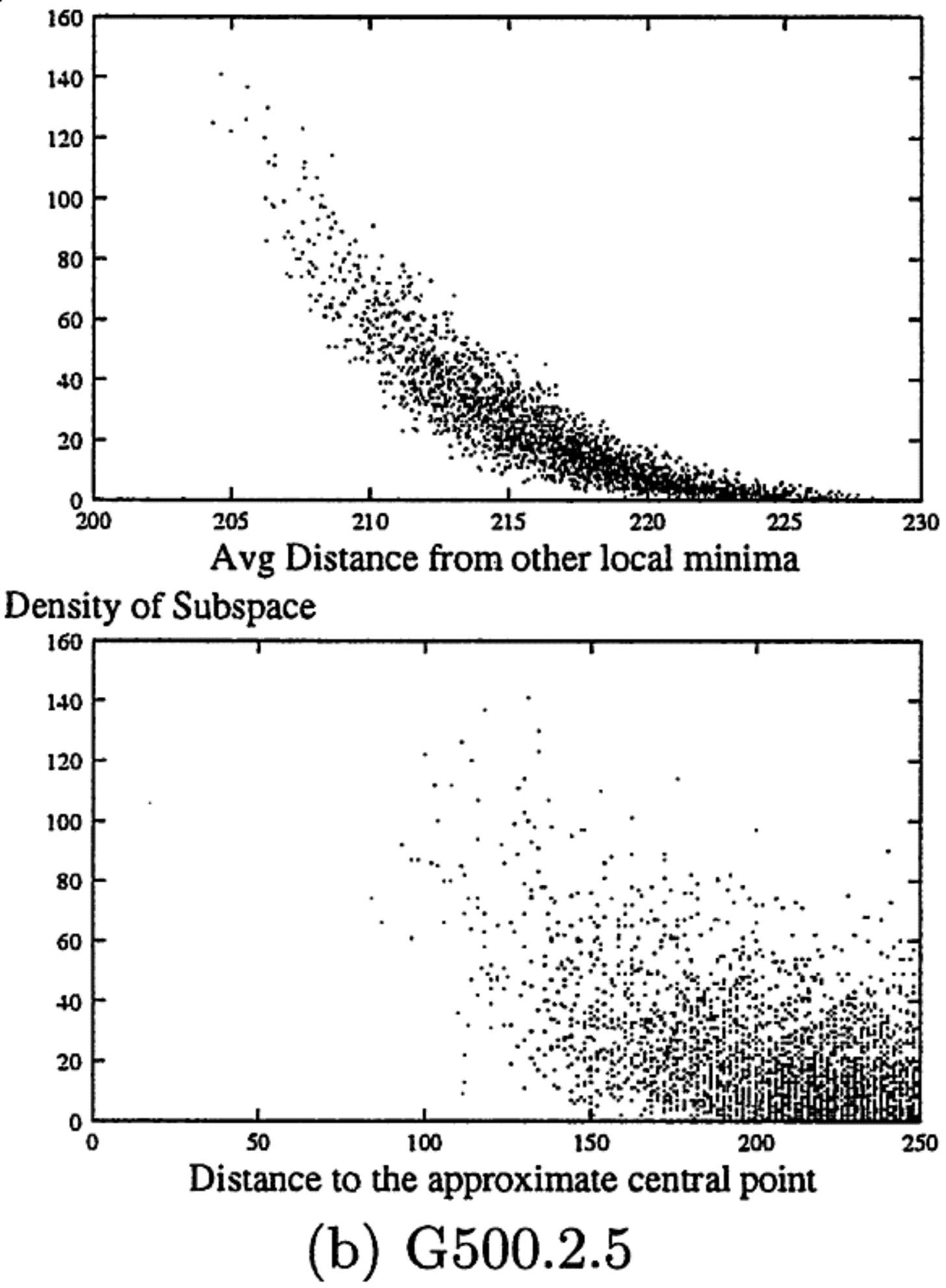




(c) U500.05

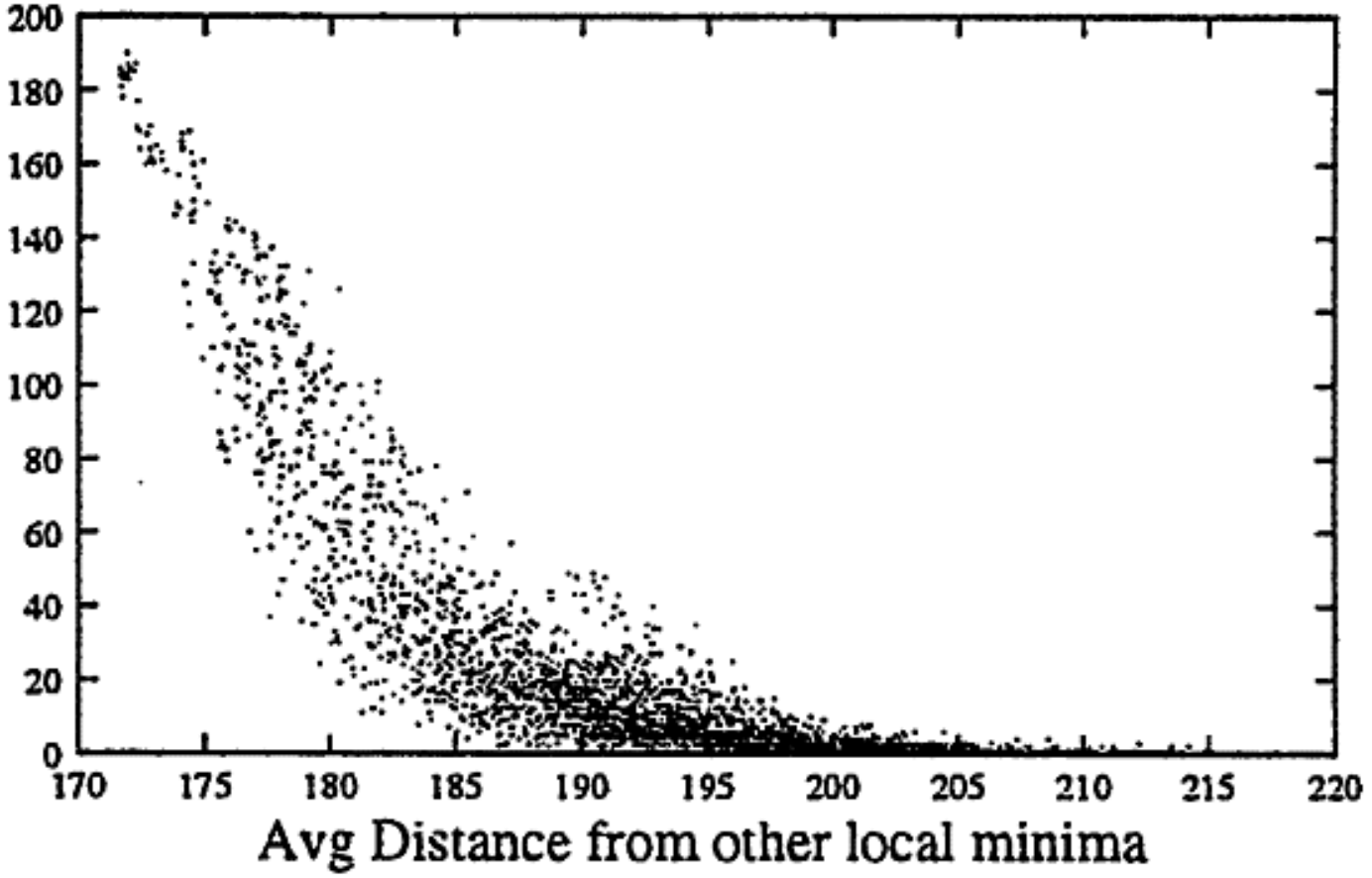
Figure 10. Distribution of local minima (see Table 6).

Density of Subspace

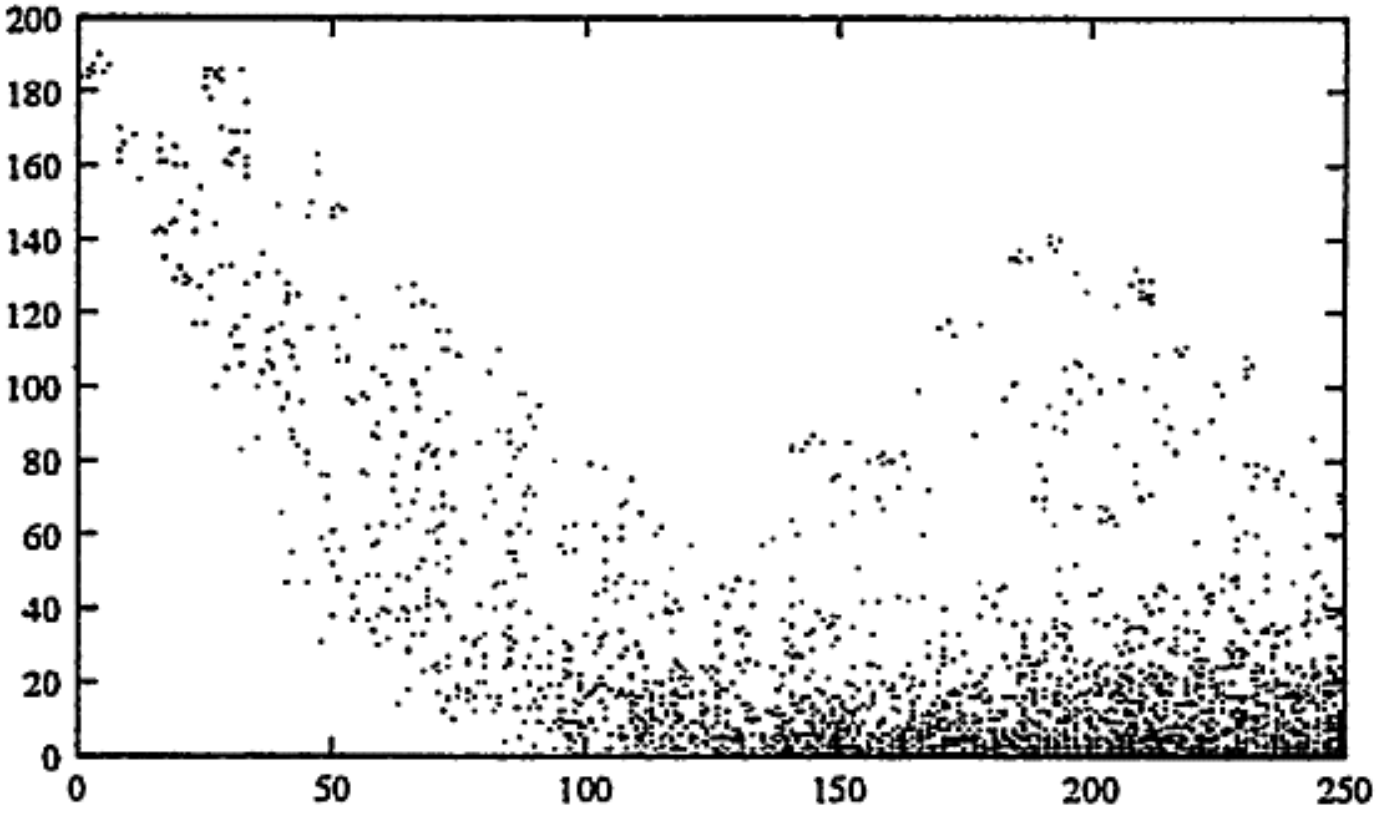

160
140
120
100
80
60
40
20
0
200
205
210
215
220
225
230
Avg Distance from other local minima
Density of Subspace
160
140
120
100
80
60
40
20
0
0
50
100
150
200
250
Distance to the approximate central point

(b) G500.2.5

Density of Subspace



Density of Subspace

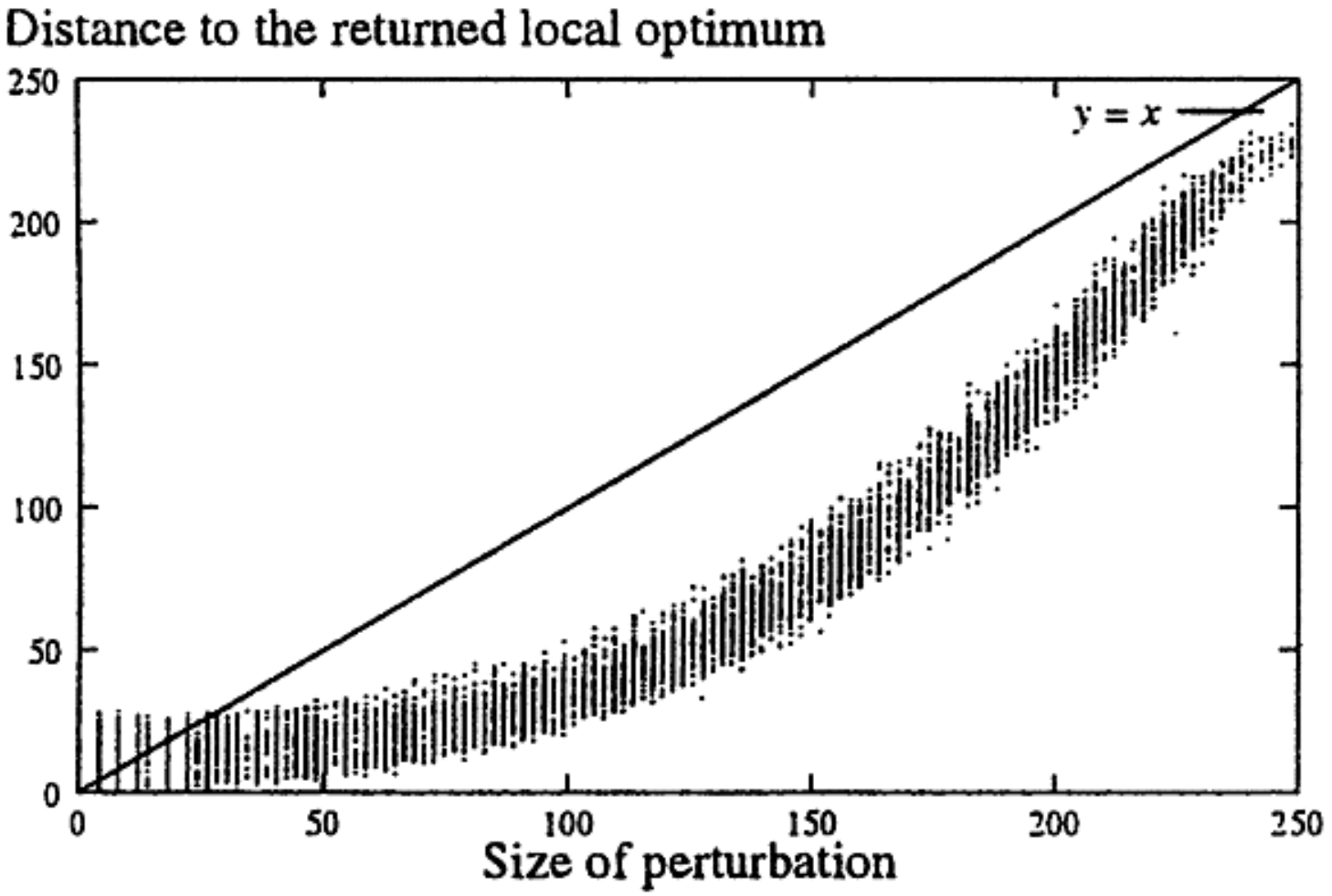


(d) U500.10


```
// Assume that the positive integer m is given.
for each local minimum s
    for i = 1 to 50 // perturbation rate: 1% ~ 50%
    {
        for k = 1 to m
        {
            s1 ← i%_random_perturbation(s);
            s2 ← local_opt(s1);
            D_{k,i} ← d(s, s2);
        }
        Plot ( |V| · i/100, ( Σ_{k=1}^{m} D_{k,i} ) /m) ;
    }
```

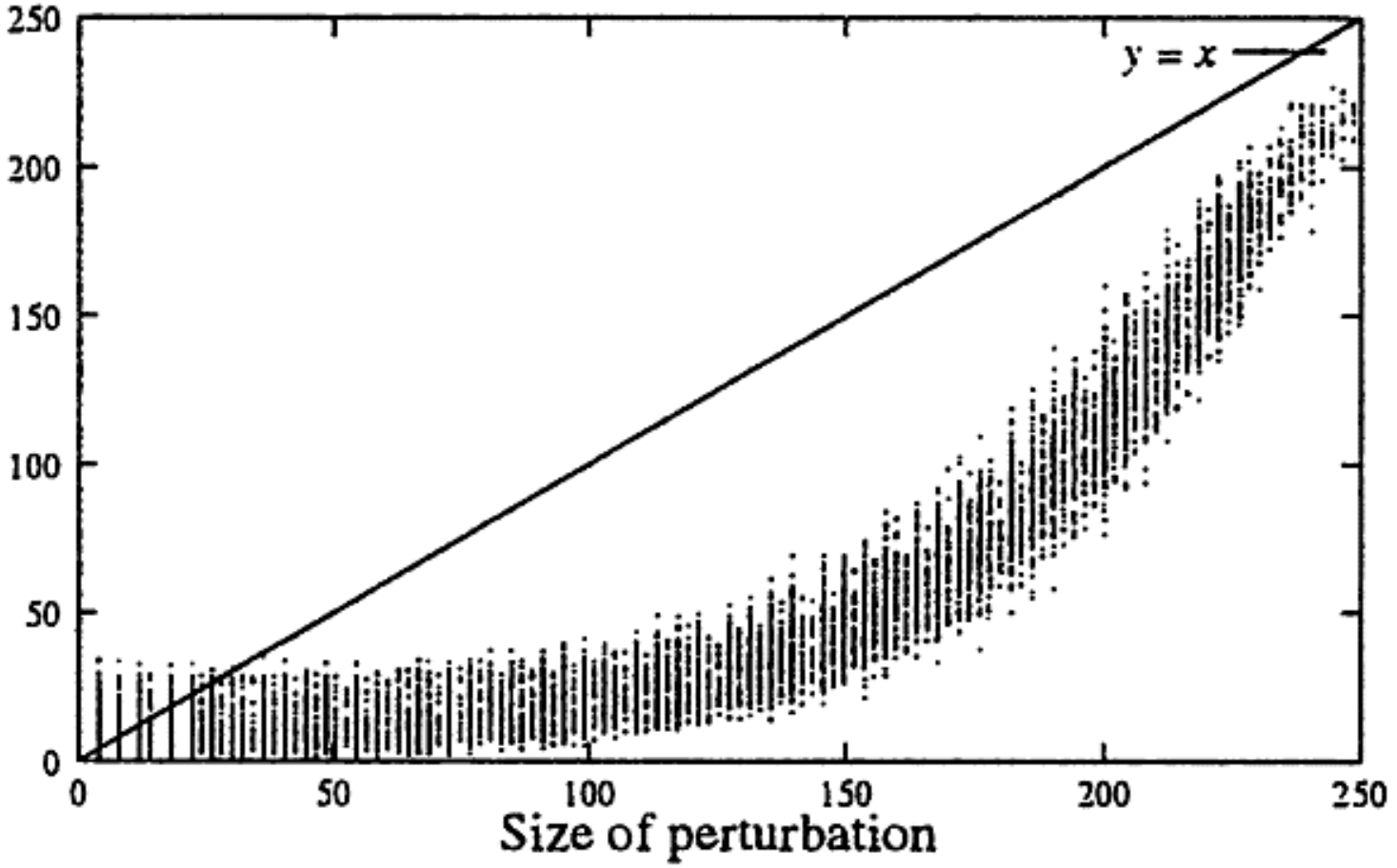
Figure 11. The test of attraction power.

Figure 12 shows the experimental results with $m = 20$ from 200 local minima. From the results, the following explanation for the graphs U500.05 and U500.10 is possible. The local optima have strong attraction powers with respect to both local optimizers. The attraction power is stronger with FM than with GREEDY.



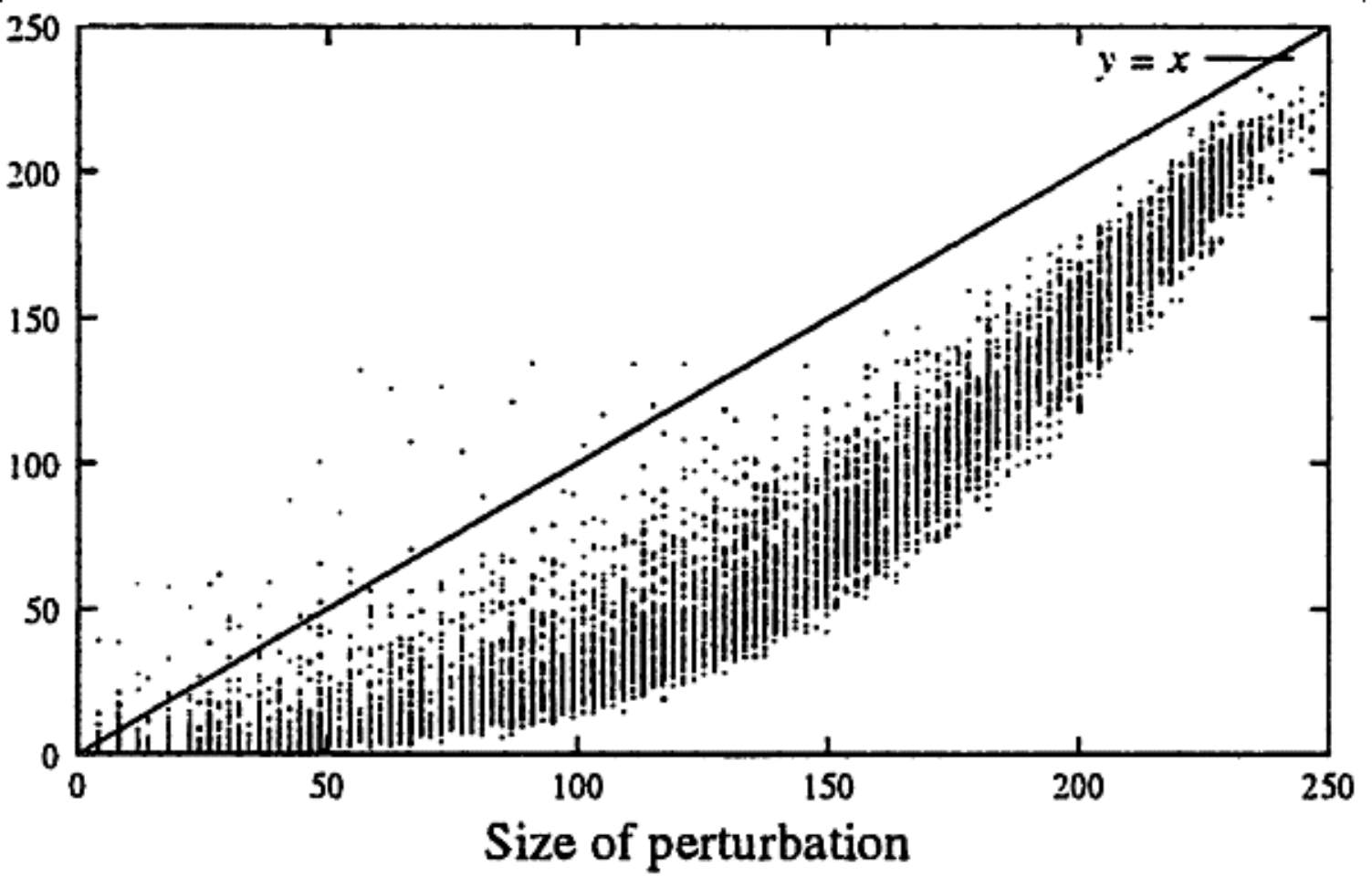
(a) U500.05 with GREEDY

Distance to the returned local optimum



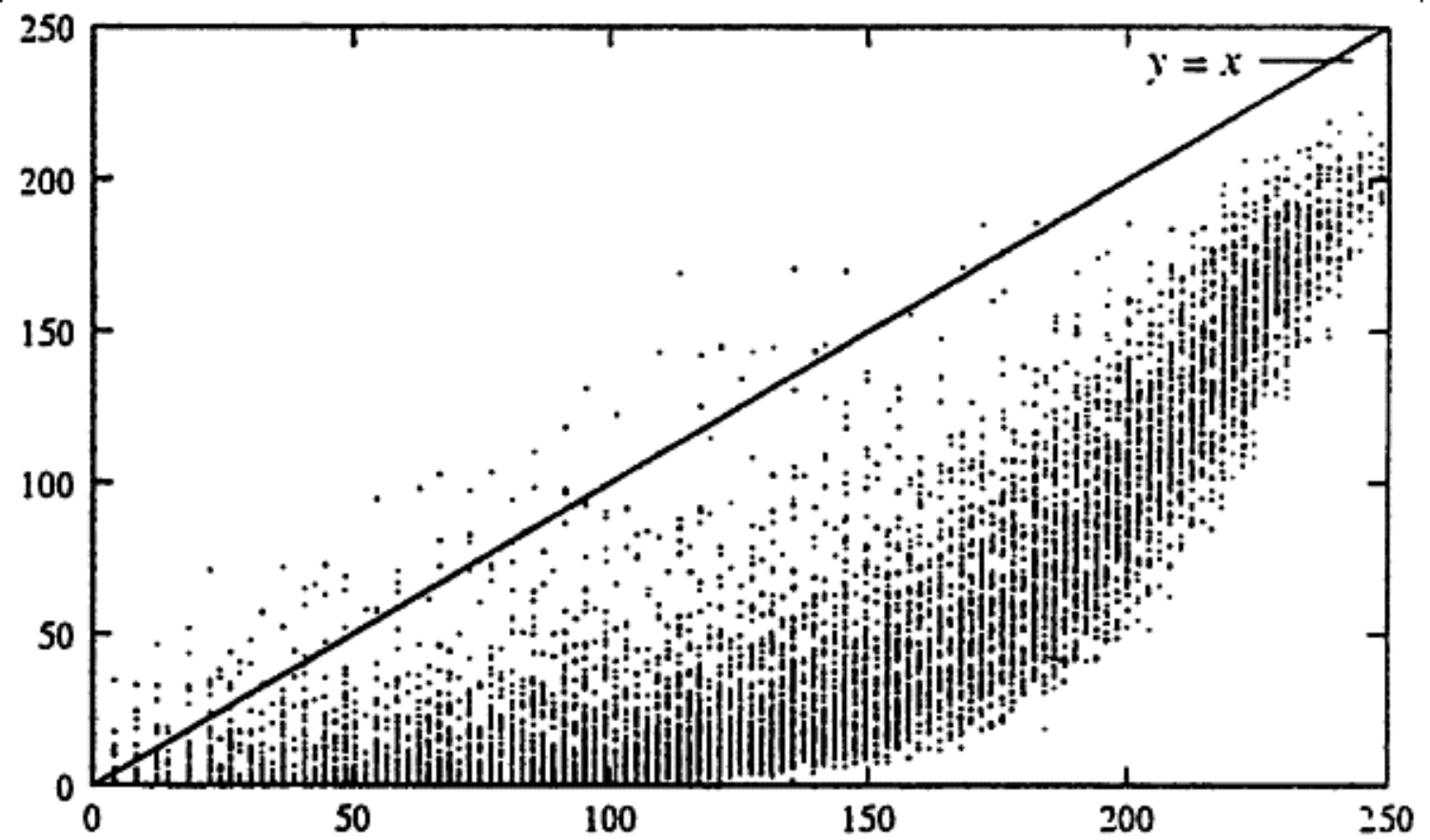
(c) U500.10 with GREEDY

Figure 12. Local optima as attractors.
Distance to the returned local optimum



(b) U500.05 with FM

Distance to the returned local optimum

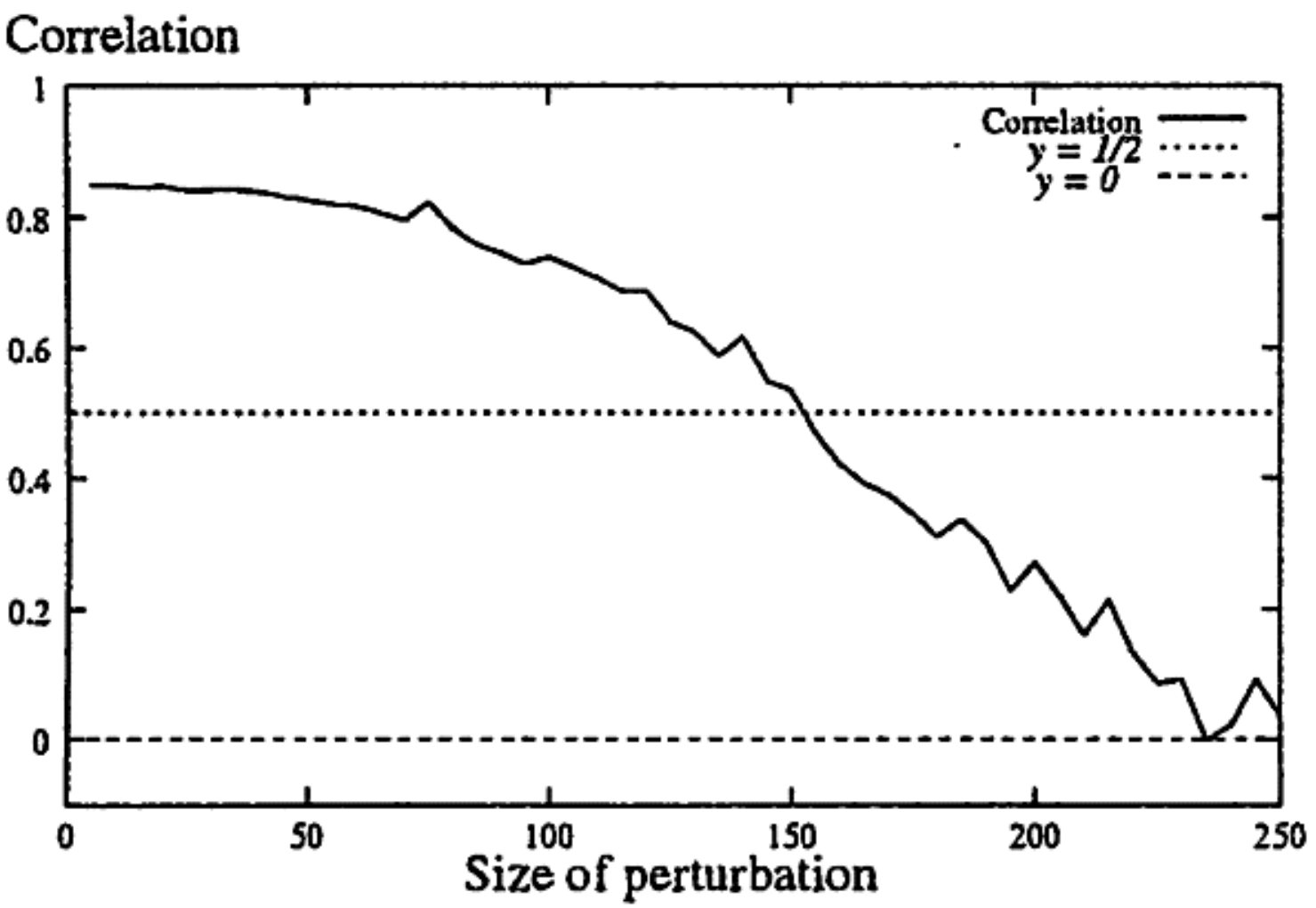


(d) U500.10 with FM

In the case of GREEDY with perturbation size 50 (10%) in U500.05, the perturbed local optima returned local optima of average distance 15.4 from the initial local optima; in the case of FM, the average distance was 9.5. This data seems to be an explanation for the observation of Hong, Kahng, and Moon (1997) that stronger local optimization heuristics need stronger perturbation to get good solutions in LSMC. One can observe wider basins of attraction in U500.10 than in U500.05. In the case of U500.10 with the same perturbation size 50, the average distances were 11.0 and 4.6 in GREEDY and FM, respectively. We conjecture that a small average distance implies the low ruggedness of the corresponding instance. However, we leave tighter investigation for this issue for future studies. One can also observe that FM rapidly loses the returning power when the perturbation size exceeds some point.

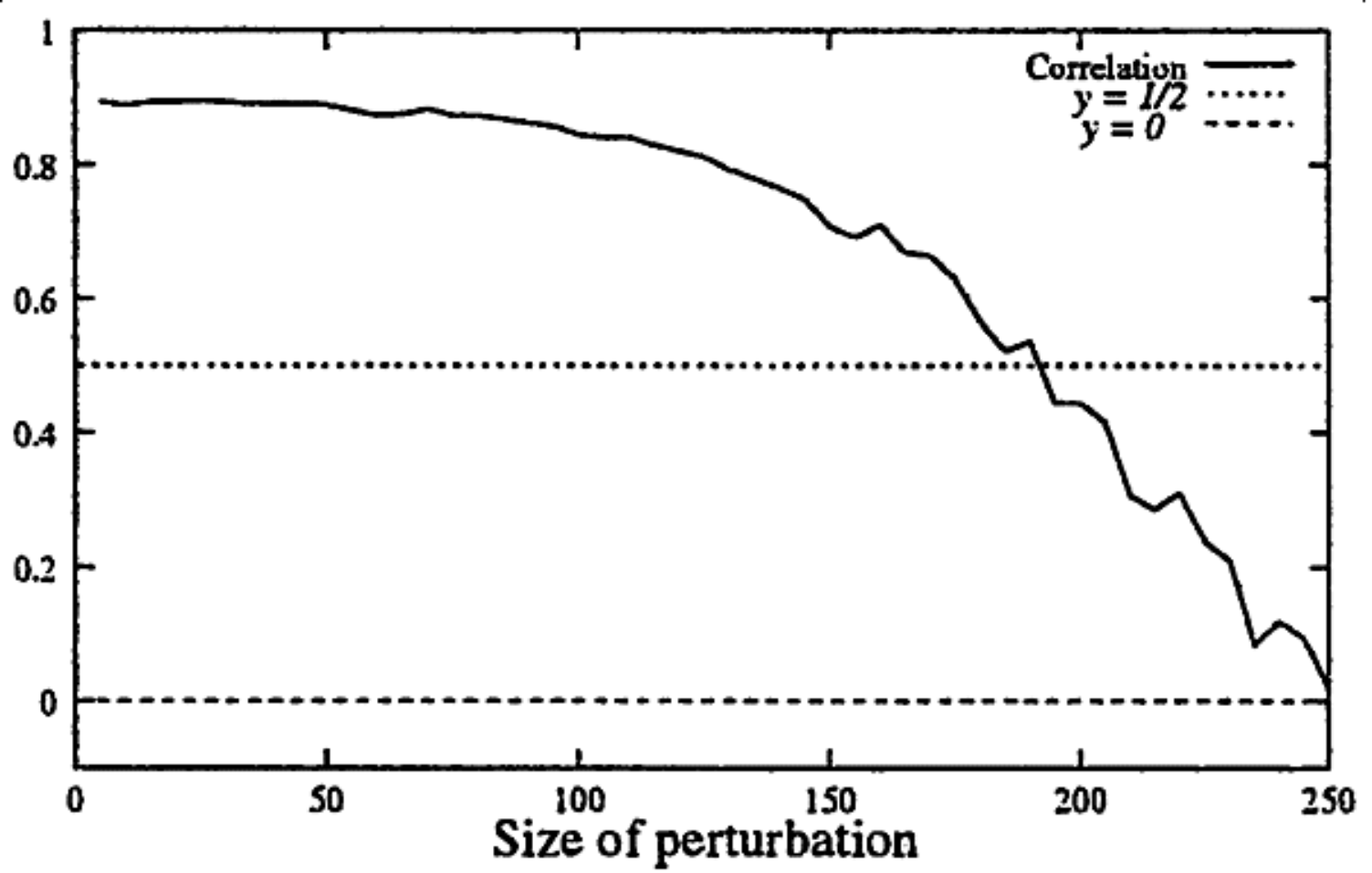
The results also show that the correlation between the costs of initial local optima and those of perturbation-induced local optima greatly depends on the perturbation rates. Figure 13 shows the spectrum of correlations with respect to perturbation sizes. The results

were obtained from 1,000 local optima and those after "perturbation + local optimization" on them. Table 7 provides the LSMC performance of the graph U500.10 over a number of perturbation rates. The best perturbation rate was around 30%. The perturbation rates corresponding to correlation 0.5 in figure 13 roughly corresponded to the best perturbation



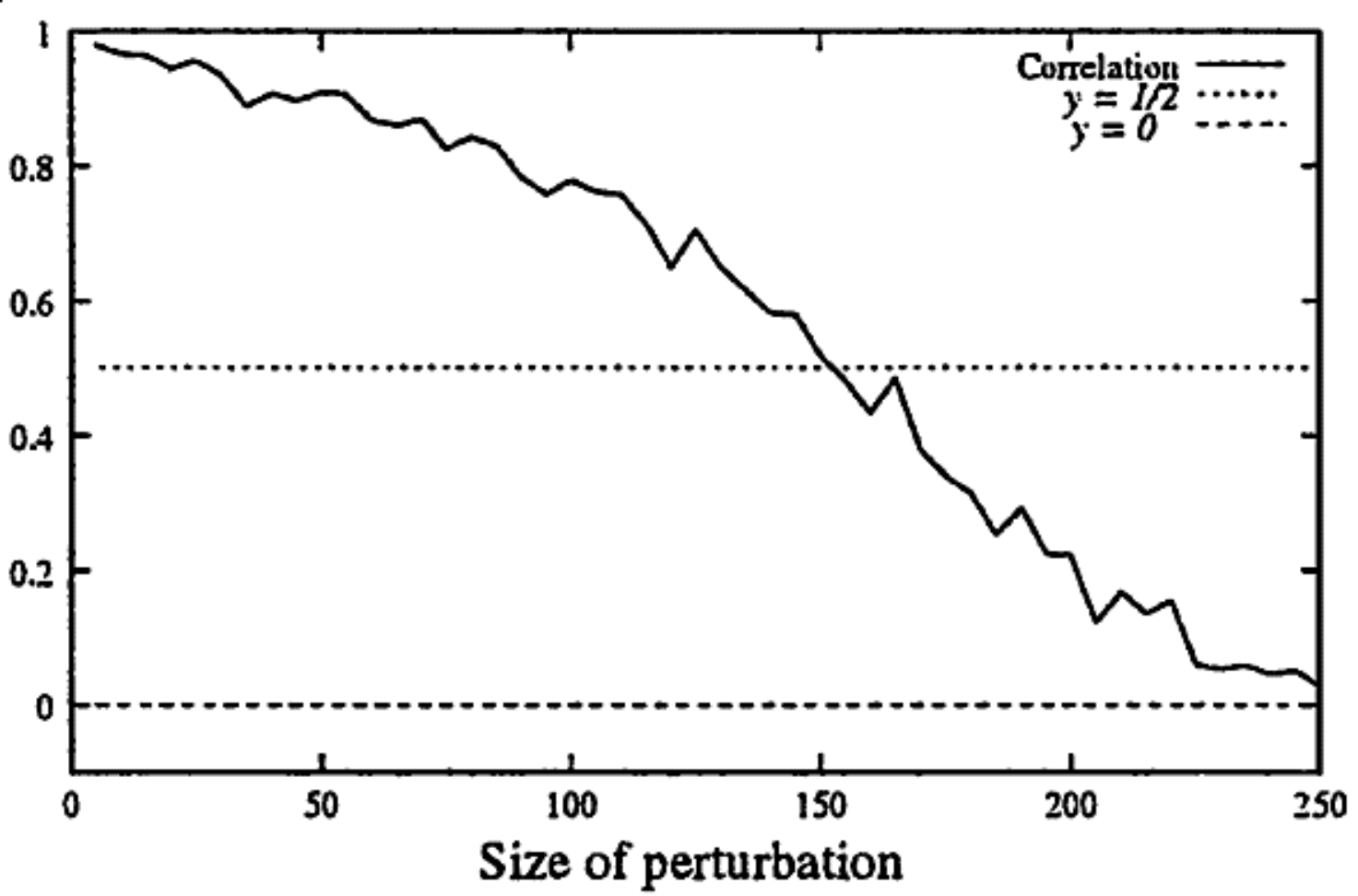
(a) U500.05 with GREEDY

Correlation



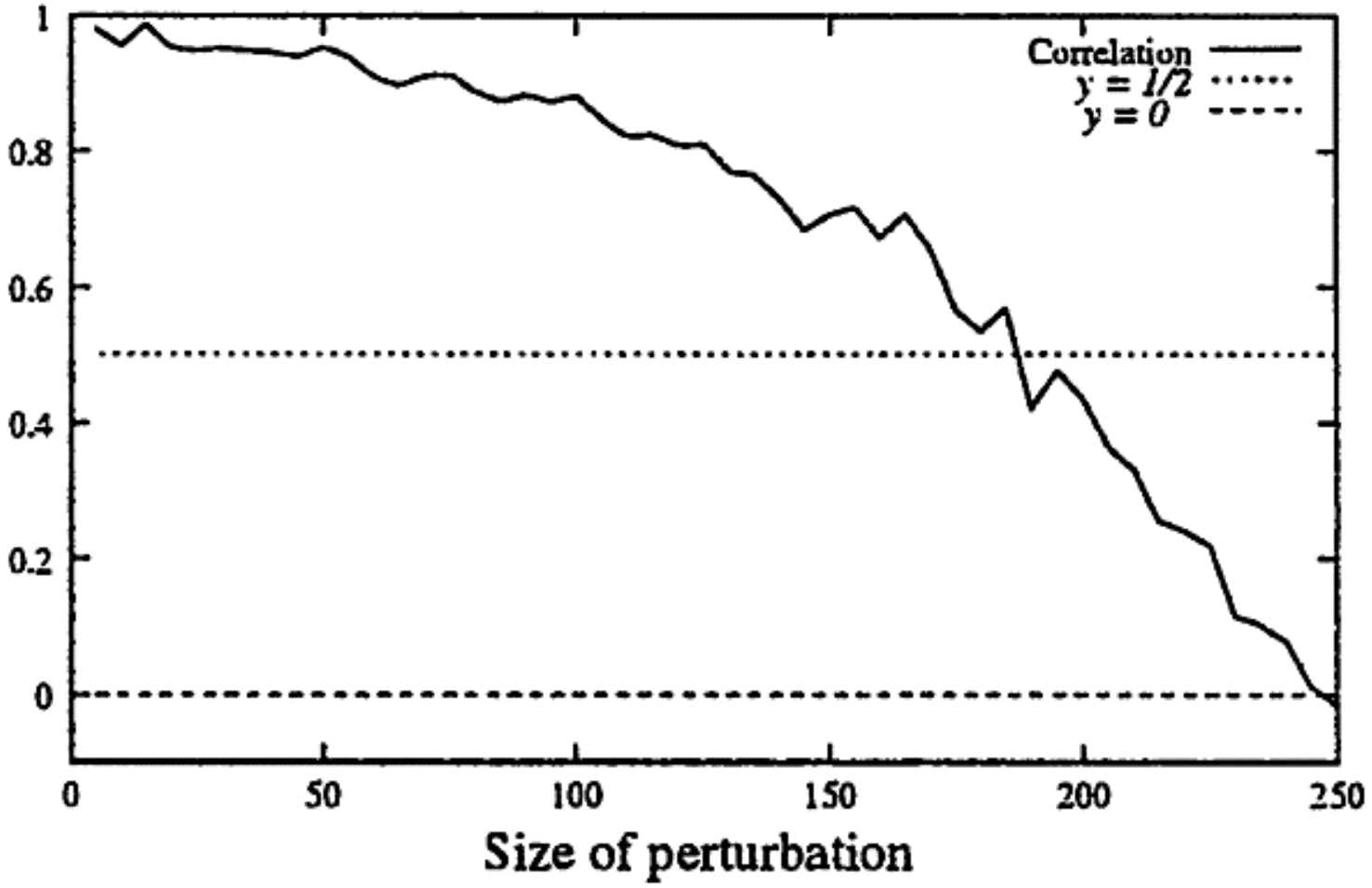
(c) U500.10 with GREEDY

Correlation



(b) U500.05 with FM

Correlation



(d) U500.10 with FM

Figure 13. Correlation between random perturbation rate and cut size.

Table 7. LSMC performance of the graph U500.05.

	Perturbation rate									
M	5%	10%	15%	20%	25%	30%	35%	40%	45%	50%
100	22.77	22.14	17.16	13.20	11.42	**10.07**	10.34	11.02	12.19	12.72
500	21.60	15.67	11.18	8.07	7.42	**6.39**	6.50	7.33	8.02	9.20
1000	22.59	12.99	8.91	8.21	6.70	6.06	**5.51**	5.85	7.03	7.96

Average results from 100 runs.
M is the number of iterations in LSMC.

rate. Manderick, de Weger, and Spiessens (1991) used the correlation length τ as a measure

of problem hardness. It is the degree of difference of two solutions with autocorrelation 0.5. It is interesting that autocorrelation 0.5 was related to the most attractive perturbation rate.

4. Exploiting approximate central points

We observed in Section 3.2 that the approximate central points obtained by simple computation are quite attractive. In Section 3.3, we showed that these properties are fractally maintained in local-optimum subspaces. In this section, we examine the advantage of exploiting the area around the approximate central point.

A search template for exploiting the approximate central points is given in figure 14. We call it *Central Search*. It is a variation of multi-start heuristics. A multi-start heuristic returns the best local optimum among several local optima fine-tuned from random initial solutions. Although multi-start heuristics are simple, they have been useful in a number of studies (Johnson, 1990; Boese and Kahng, 1994). The proposed heuristic proceeds in two threads. One thread makes a series of solutions by "random solution + local optimization" like the multi-start approach. Simultaneously, the other makes a series of solutions by "approximate central point of recent K local optima + random perturbation + local optimization." It adds the part of exploiting the central area of recent K solutions to the typical multi-start heuristic. K was used to control the degree of exploitation around the central area. For example, let M be 100. If K is 10, it exploits the central area 45 times out of 100 trials and, if K is 40, 30 times out of 100 trials. If K is equal to M, the heuristic equals the multi-start heuristic. The values of K represent a spectrum of the exploitation strength of the central area. Since the heuristic makes total M local optima, it has almost the same running time as multi-start heuristic with M iterations.

The experimental results of multi-start heuristic and the proposed heuristic are given in Table 8. We use the FM algorithm as the local optimizer. We set M and R to 100 and 20 respectively in all cases and performed 1,000 runs in the experiments.

Overall, exploiting the central area showed improvement over the multi-start heuristic in all test graphs. Figure 15 shows two sample spectra. The results show that it is useful to exploit the central area, but that excessive or insufficient exploitation is not desirable. This experiment is only a simple example to utilize the central area and is not the major output of our study.

MultiStart(M)
  for $i = 1$ **to** M

```
    {
        Generate a random solution P_i;
        P_i ← local_opt(P_i);
    }
    return the best among P_1, P_2, ..., P_M;

CentralSearch(M, K, R)
    for each i = 1, ..., K
    {
        Generate a random solution P_i;
        P_i ← local_opt(P_i);
    }
    B ← the best among P_1, P_2, ..., P_K;
    for i = 1 to (M − K)/2
    {
        Generate the approximate central point C of P_1, P_2, ..., P_K;
        C ← R%_random_perturbation(C);
        C* ← local_opt(C);
        Generate a random solution T;
        P_((i−1) mod K)+1 ← local_opt(T);
        B ← the best among B, C*, and P_((i−1) mod K)+1;
    }
    return B;
```

Figure 14. A simple variation of multi-start heuristic.

Cut Size

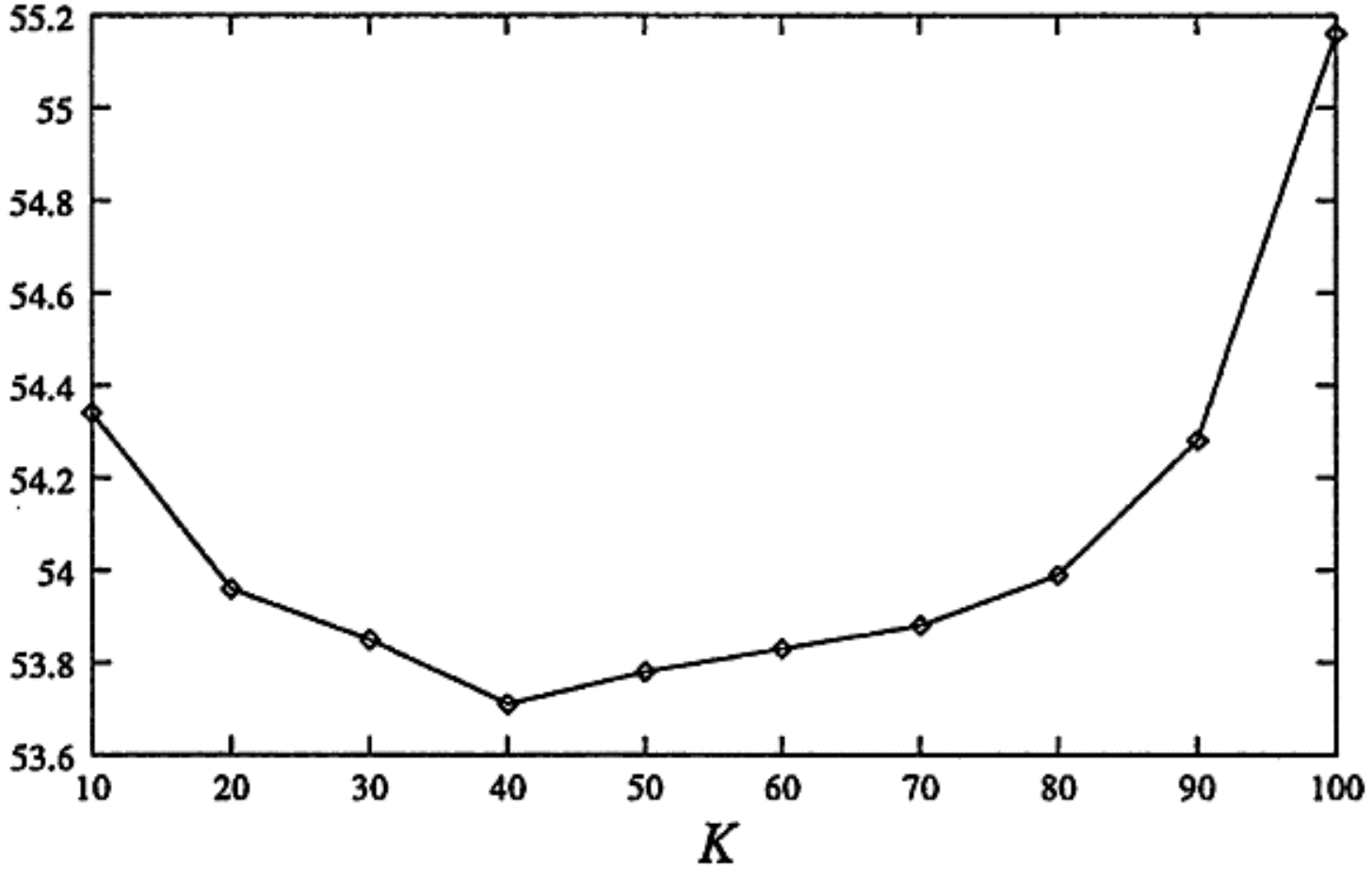
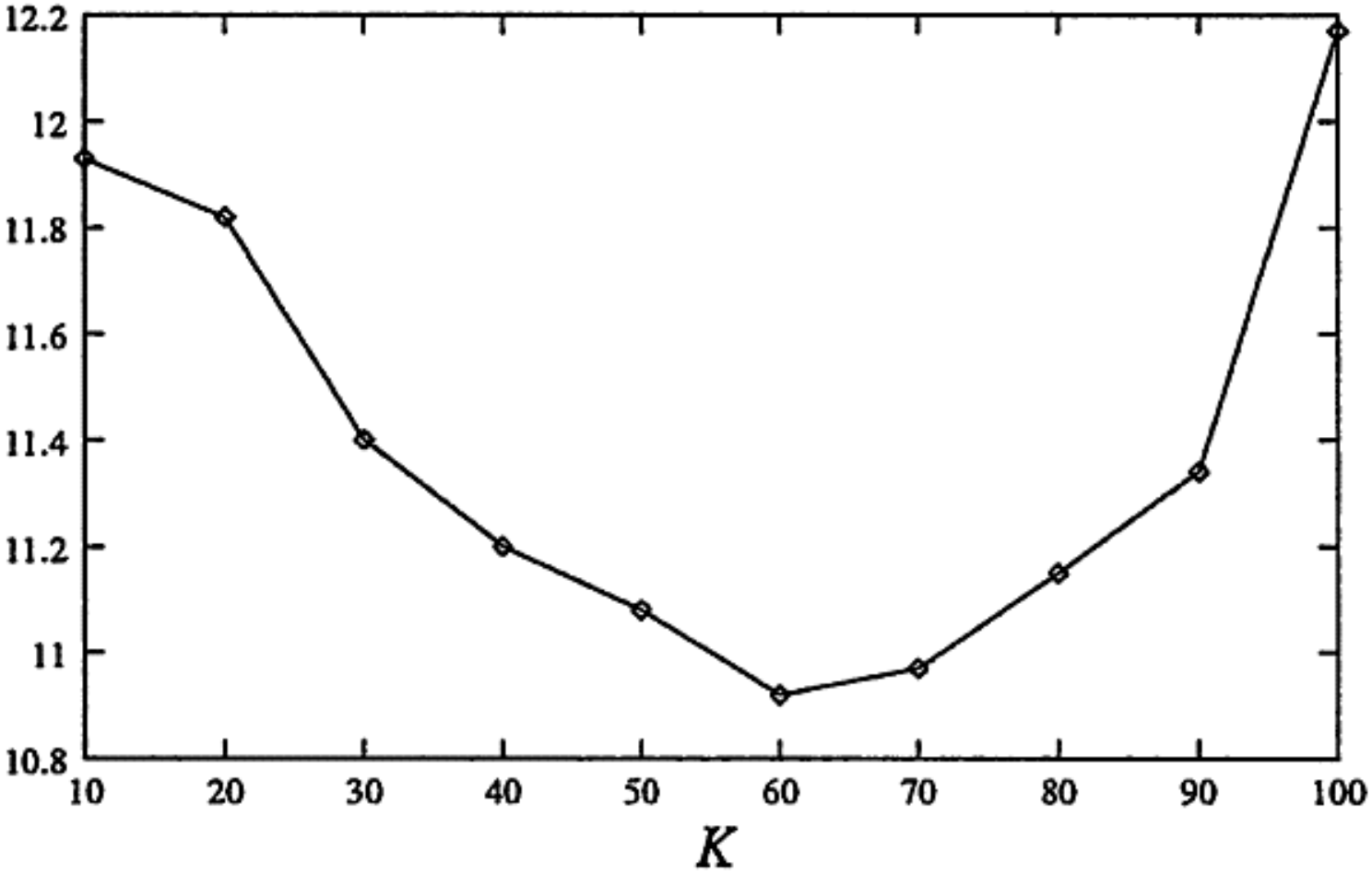


Figure 15. Two sample spectra from Table 8.
Cut Size



(B) U500.05

We also compare our proposed method with LSMC. As mentioned in Section 2.3, LSMC is an iterated "perturbation + local optimization." We use zero-temperature LSMC as in most literature (Martin, Otto, and Felten, 1991; Fukunaga, Huang, and Kahng, 1996; Hong, Kahng, and Moon, 1997); i.e., it chooses the best solution found so far as the solution

Table 8. The comparison of cut size ($M = 100$ and $R = 20$).

	Multi-Start[§]		$K = 10$		$K = 20$		$K = 30$		$K = 40$	
Graphs	Ave[†]	CPU[‡]	Ave[†]	CPU[‡]	Ave[†]	CPU[‡]	Ave[†]	CPU[‡]	Ave[†]	CPU[‡]
G500.2.5			55.16		0.28		54.34		0.26	
G500.05			222.35		0.38		219.39		0.36	

G500.10	627.90	0.67	625.42	0.68
G500.20	1736.51	2.26	1733.22	2.30
G1000.2.5	110.37	0.62	108.22	0.67
G1000.05	465.37	1.06	460.75	1.23
G1000.10	1374.44	2.68	1368.07	2.84
G1000.20	3379.56	6.43	3365.39	6.79
U500.05	12.17	0.36	11.93	0.37
U500.10	25.59	0.66	**25.04**	0.66
U1000.05	32.28	0.90	33.12	1.40
U1000.10	52.26	2.28	48.61	2.53
	$K = 50$		$K = 60$	
G500.2.5	53.78	0.30	53.83	0.30
G500.05	218.88	0.40	219.20	0.40
G500.10	624.60	0.69	624.73	0.75
G500.20	1732.43	2.04	1732.99	2.16
G1000.2.5	106.41	0.73	106.25	0.78

G1000.05	**458.49**	1.25	458.58	1.22	
G1000.10	1365.64	2.83	1365.79	2.65	
G1000.20	3362.20	6.37	**3361.58**	6.37	
U500.05	11.08	0.36	**10.92**	0.37	
U500.10	25.05	0.62	25.16	0.62	
U1000.05	30.67	1.12	30.20	1.25	
U1000.10	47.97	2.20	48.44	2.06	
53.96	0.27	53.85	0.28	**53.71**	0.29
219.02	0.37	218.94	0.39	**218.81**	0.39
624.82	0.69	**624.49**	0.70	624.55	0.73
1732.76	2.16	1732.41	2.14	**1732.30**	2.17
107.42	0.62	106.89	0.77	106.37	0.74
459.33	1.10	458.74	1.24	458.71	1.16
1366.33	2.80	1365.69	2.62	**1365.53**	2.78
3363.17	6.64	3362.47	6.29	3361.99	6.63
11.82	0.33	11.40	0.35	11.20	0.40
25.19	0.50	25.20	0.62	25.23	0.60
32.27	0.99	31.72	1.17	31.25	1.11

48.32	2.08	48.51	2.11	**47.80**	2.22
$K = 70$		$K = 80$		$K = 90$	
53.88	0.29	53.99	0.29	54.28	0.28
219.37	0.39	219.69	0.39	220.21	0.38
625.20	0.74	625.59	0.75	626.23	0.76
1733.10	2.11	1733.85	2.08	1734.52	2.11
106.07	0.78	106.40	0.65	107.30	0.64
459.06	1.21	460.02	1.15	461.25	1.24
1366.59	2.78	1367.61	2.67	1369.32	2.78
3361.63	6.47	3363.04	6.43	3366.46	6.59
10.97	0.42	11.15	0.36	11.34	0.36
25.25	0.69	25.14	0.58	25.14	0.59
30.12	1.23	30.76	1.21	30.57	1.15
48.44	2.22	48.67	2.25	49.02	2.35

[§] $K = M(=100)$.
[†] Average over 1,000 runs.
[‡] CPU seconds on Pentium celeron 466 MHz.
For four dense geometric graphs (U500.20, U500.40, U1000.20, and U1000.40), all the methods always found the best known.

for the next perturbation step. Table 9 shows the performance of Central Search and two

LSMCs. We performed 1,000 runs in each experiment. In each run, 1,000 new solutions were generated, which consequently required 1,000 runs of local optimization, for both Central Search and LSMC. As parameters for our method, we set M, K, and R to 1,000, 500, and 20 respectively. LSMC1 has 20% as the perturbation rate–the same perturbation

Table 9. The comparison of cut size (1,000 iterations).

	CentralSearch[1]	LSMC1[2]	LSMC2[3]
Graphs	Ave†	Ave†	Ave†
G500.2.5			**50.64**
G500.05			**213.78**
G500.10			**619.76**
G500.20			**1727.20**
G1000.2.5			**97.59**
G1000.05			448.01
G1000.10			1354.97

G1000.20	**3345.07**
U500.05	**4.77**
U500.10	**24.00**
U1000.05	**12.86**
U1000.10	**40.97**
51.71	51.25
221.04	215.07
627.28	621.04
1737.98	1729.60

100.27	100.19
453.31	**447.84**
1366.97	**1352.88**
3388.03	3357.90
7.57	5.67
33.70	25.36
16.73	17.68
59.27	43.39

[1]Our method with $(M, K, R) = (1000, 500, 20)$.
[2]LSMC with 1,000 iterations and perturbation rate 20%.
[3]LSMC with 1,000 iterations and the best perturbation rate 35%.

†Average over 1,000 runs.

rate as our method. LSMC2 has 35% as the perturbation rate where it shows the best results over a number of perturbation rates. Central Search overall outperformed the LSMCs.

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 several experiments and got some new insights into the global structure of the graph-partitioning problem space. We extended previous works and observed that good solutions are located near other good solutions and the central area is quite attractive. We supported the hypothesis that graph-partitioning problem spaces have self-similar fractal structures. We also showed that the best perturbation rate of LSMC accords with the point beyond which the local optima sharply lose the attraction power.

Boese, Kahng, and Muddu (1994) conjectured that the graph-partitioning problem space is globally convex; i.e., it has the "big valley." In our experiments, we found a strong evidence of the existence of "medium valleys" as a result of investigating the distribution of local optima. That is, some graph-partitioning problem spaces are not very globally convex but seem to have more than one convex subspace. It could not be explained by previous empirical methods.

It looks clear that there is a huge cluster of high-quality solutions near the central area of local optima. Hence, it is attractive to exploit the central area. Too much exploitation of

the central area makes the search diversity low. It is desirable to exploit the central area avoiding excessive or insufficient exploitation. We showed that the performance of search could be improved by a simple heuristic based on the exploitation of the central area. This provides a motivation to contrive more sophisticated heuristics exploiting the central area. More theoretical arguments for our empirical results and more analysis of various subspaces are left for future study.

Our results were achieved in a specific problem: graph partitioning. However, we expect that many other combinatorial optimization problems have similar properties. For example, in the case of cost-distance correlation of Section 3.1, TSP showed similar property to the graph partitioning problem (Boese, Kahng, and Muddu, 1994). We hope that this study provides a good motivation for the investigation of problem spaces and the design of more effective search algorithms.

Appendix A: Proof about approximate central point

Let the universal set U be $\{0, 1\}^{|V|}$. Consider the distance measure d defined in Section 3, a subset S in U, and an element s_k in S.[10] For each $a \in S$, if $\mathfrak{H}(s_k, a) > \lfloor |V|/2 \rfloor$, make a transition that interchanges 0 and 1 at every position of a. Let the new set resulted from the transitions be $S' = \{s_1, s_2, \ldots, s_n\} \subset U$. Then, for each $i = 1, 2, \ldots, n$ $(i \neq k)$, $0 < \mathfrak{H}(s_k, s_i) \leq \lfloor |V|/2 \rfloor$. Now, generate a new element c such that for each $j = 1, 2, \ldots, |V|$

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

where $B_j(x)$ is the jth element of x $(x \in U)$. Then, we have the following proposition.

Proposition 1.

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

Proof:

$$\begin{aligned} \sum_{i=1}^{n} d(s_k, s_i) &= \sum_{i=1}^{n} \mathfrak{H}(s_k, s_i) \\ &= \sum_{i=1}^{n} \sum_{j=1}^{|V|} |B_j(s_k) - B_j(s_i)| \\ &= \sum_{j=1}^{|V|} \sum_{i=1}^{n} |B_j(s_k) - B_j(s_i)| \end{aligned}$$

$$= \sum_{j=1}^{|V|} |\{s \in S' : B_j(s) \neq B_j(s_k)\}|$$

$$\geq \sum_{j=1}^{|V|} |\{s \in S' : B_j(s) \neq B_j(c)\}|$$

$$= \sum_{i=1}^{n} \mathfrak{H}(c, s_i)$$

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

Acknowledgments

KIM AND MOON

□

This work was supported by Brain Korea 21 Project and a grant of the International Mobile Telecommunications 2000 R&D Project, Ministry of Information & Communication, Korea. The ICT at Seoul National University provided research facilities for this study. We also thank anonymous referees for their helpful comments.

Notes

1. Sorkin defines "fractal" as follows: A cost surface over solution space Ω with distance d is defined to be *fractal* if for random solutions $s_1, s_2 \in \Omega$, $E[(f(s_1) - f(s_2))^2] \propto d(s_1, s_2)^{2H}$, where d is simply the graph distance induced by the neighborhood topology and H is the parameter with $0 \leq H \leq 1$.
2. 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 that Q is the quotient set of U by relation R, it is easily verified that (Q, d) is a *metric space*.
3. This is not a simple issue, though.
4. We define "central point" to be the nearest solution to the center of local-optimum space.
5. 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.

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}.
7. Consider the space F having finite elements distributed uniformly in an n-dimensional Euclidean space. Given a subspace Ω_1 of F, if we choose the subspace Ω_2 of F such that the radius of Ω_2 is a half that of Ω_1, the cardinality of Ω_2 is about $1/2^n$ times that of Ω_1.
8. The data of this figure are from the same solutions as those in figures 2 and 3.
9. The relative notion to big valleys mentioned by Boese, Kahng, and Muddu (1994).
10. Assume that the distance d between any two elements in S is larger than zero.

References

Battiti, R. and A. Bertossi. (1998). "Differential Greedy for the 0–1 Equicut Problem." In D.Z. Du and P.M. Pardalos (eds.), *Network Design: Connectivity and Facilities Location*. American Mathematical Society. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science* 40, 3–21.

Battiti, R. and A. Bertossi. (1999). "Greedy, Prohibition, and Reactive Heuristics for Graph Partitioning." *IEEE Trans. on Computers* 48(4), 361–385.

Boese, K.D. and A.B. Kahng. (1994). "Best-so-far vs. Where-you-are: Implications for Optimal Finite-Time Annealing." *Systems and Control Letters* 22(1), 71–78.

Boese, K.D., A.B. Kahng, and S. Muddu. (1994). "A New Adaptive Multi-Start Technique for Combinatorial Global Optimizations." *Operations Research Letters* 15, 101–113.

Bui, T.N. and B.R. Moon. (1996). "Genetic Algorithm and Graph Partitioning." *IEEE Trans. on Computers* 45(7), 841–855.

Dutt, S. and W. Deng. (1996). "A Probability-Based Approach to VLSI Circuit Partitioning." In *Design Automation Conference*, pp. 100–105.

Fiduccia, C. and R. Mattheyses. (1982). "A Linear Time Heuristics for Improving Network Partitions." In *19th ACM/IEEE Design Automation Conference*, pp. 175–181.

Ford, L.R. Jr. and D.R. Fulkerson. (1962). *Flows in Networks*. Princeton University Press.

Fukunaga, A.S., J.H. Huang, and A.B. Kahng. (1996). "On Clustered Kick Moves for Iterated-Descent Netlist Partitioning." In *IEEE Int'l Symp. on Circuits and Systems*, vol. 4, pp. 496–499.

Garey, M. and D.S. Johnson. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco: Freeman.

Hong, I., A.B. Kahng, and B.R. Moon. (1997). "Improved Large-Step Markov Chain Variants for the Symmetric TSP." *Journal of Heuristics* 3(1), 63–81.

Johnson, D.S. (1990). "Local Optimization and the Traveling Salesman Problem." In *17th Colloquium on Automata, Languages, and Programming*. Springer-Verlag, pp. 446–461.

Johnson, D.S., C. Aragon, L. McGeoch, and C. Schevon. (1989). "Optimization by Simulated Annealing: An Experimental Evaluation, Part 1, Graph Partitioning." *Operations Research* 37, 865–892.

Jones, T. and S. Forrest. (1995). "Fitness Distance Correlation as a Measure of Problem Difficulty for Genetic Algorithms." In *Sixth International Conference on Genetic Algorithms*, pp. 184–192.

Kauffman, S. (1989). "Adaptation on Rugged Fitness Landscapes." *Lectures in the Science of Complexity*, pp. 527–618.

Kernighan, B. and S. Lin. (1970). "An Efficient Heuristic Procedure for Partitioning Graphs." *Bell Systems Technical Journal* 49, 291–307.

Kim, Y.H. and B.R. Moon. (2004). "Lock-Gain Based Graph Partitioning." *Journal of Heuristics* 10(1), 37–57.

Kirkpatrick, S., C.D. Gelatt Jr., and M.P. Vecchi. (1983). "Optimization by Simulated Annealing." *Science* 220(4598), 671–680.

Lim, A. and Y.M. Chee. (1991). "Graph Partitioning Using Tabu Search." In *IEEE Int'l Symp. Circuits and Systems*, pp. 1164–1167.

Manderick, B., M. de Weger, and P. Spiessens. (1991). "The Genetic Algorithm and the Structure of the Fitness Landscape." In *International Conference on Genetic Algorithms*, pp. 143–150.

Martin, O.C., S.W. Otto, and E.W. Felten. (1991). "Large-Step Markov Chain for the Traveling Salesman Problem." *Complex Systems* 5(3), 299–326.

Merz, P. and B. Freisleben. (1998). "Memetic Algorithms and the Fitness Landscape of the Graph Bi-Partitioning Problem." In *Proceedings of the 5th International Conference on Parallel Problem Solving From Nature*, Springer-Verlag. *Lecture Notes in Computer Science*, 1498, 765–774.

Moon, B.R. and C.K. Kim. (1997). "A Two-Dimensional Embedding of Graphs for Genetic Algorithms." In *International Conference on Genetic Algorithms*, pp. 204–211.

Saab, Y.G. (1995)."A Fast and Robust Network Bisection Algorithm." *IEEE Trans. on Computers* 44(7), 903–913.

Sorkin, G.B. (1991). "Efficient Simulated Annealing on Fractal Landscapes." *Algorithmica* 6, 367–418.

Steenbeek, A.G., E. Marchiori, and A.E. Eiben. (1998). "Finding Balanced Graph Bi-Partitions Using a Hybrid Genetic Algorithm." In *IEEE International Conference on Evolutionary Computation*, pp. 90–95.

Weinberger, E.D. (1991). "Fourier and Taylor Series on Fitness Landscapes." *Biological Cybernetics* 65, 321–330.