International Journal of Fuzzy Logic and Intelligent Systems, vol. 1, no. 1, June 2001 pp. 6-12

A Hybrid of Evolutionary Search and Local Heuristic Search for Combinatorial Optimization Problems

Lae-Jeong Park and Cheol Hoon Park*

School of Electronics Engineering and Information Technology, Kangnung National University
123 Chibyung-Dong, Kangnung 205-702, Korea
*Dept. of Electrical Engineering and Computer Science, KAIST
373-1 Kusong-Dong Yusong-Gu, Taejon 305-701, Korea

Abstract

Evolutionary algorithms(EAs) have been successfully applied to many combinatorial optimization problems of various engineering fields. Recently, some comparative studies of EAs with other stochastic search algorithms have, however, shown that they are similar to, or even are not comparable to other heuristic search. In this paper, a new hybrid evolutionary algorithm utilizing a new local heuristic search, for combinatorial optimization problems, is presented. The new intelligent local heuristic search is described, and the behavior of the hybrid search algorithm is investigated on two well-known problems: traveling salesman problems (TSPs), and quadratic assignment problems(QAPs). The results indicate that the proposed hybrid is able to produce solutions of high quality compared with some of evolutionary algorithms and simulated annealing.

Key Words : Evolutionary Algorithms, Heuristic Search, Combinatorial Optimization

I. Introduction

As have been attracting much attention as a new class of global optimization techniques since they have shown good performance in many combinatorial optimization problems[1]. Even though literatures have offered many applications showing the effectiveness of EAs, recent empirical and theoretical researches have raised questions about the effectiveness of evolutionary algorithms[2,3,4]. Moreover, several empirical evidences on real-world applications have revealed no explicit superiority or even inferiority of EAs to other stochastic search techniques such as simulated annealing(SA).

Much of comparative research has suggested some reasons of the relative weakness of the evolutionary search over other stochastic search methods[3,5]. It has been, in common, reported that, the weakness of EAs are :

(a) they make ``premature" decisions of promising regions of the search space due to the competition among individuals of the finite population. Individuals compete each other without exploring local search regions through sufficient variation of each individual. Competitive selection under the premature evaluation of local regions often misleads to a poor local region in the case of small population size and/or highly multimodal landscapes.

(b) EAs do not have an explicit mechanism of escaping from local optima after all individuals become similar. Even though, when the mutation rate is not zero, new individuals on uphill paths can be generated, it is extremely rare that the new ones survive the subsequent competition and thus provide a chance of finding individuals in better local basins.

Manuscript received March 15, 2001; revised May 1, 2001. This paper was supported in part by the Brain Korea 21 Project in 2001.

Consequently, attempts have been made to solve these problems and to enhance the performance of the evolutionary search by combining or hybridizing it with other local search algorithms, like hill-climbing and simulated annealing. The general idea behind hybrid evolutionary algorithms(HEAs) is to combine the advantages of evolutionary search that globally identifies promising regions with local search capable of quickly finding good solutions in a local region. Several cascade hybrids of evolutionary algorithm and simulated annealing have been proposed[6,7,8]. In the hybrids, for each generation, genetic operations are followed by a full-schedule of low-temperature simulated annealing as a local search. However, choosing an appropriate initial temperature of simulated annealing is difficult. Instead of stochastic local search, exact local neighborhood search or local hill-climbing can be incorporated into the evolutionary search[9]. Some of the hybrid evolutionary algorithms are similar to so-called memetic algorithms(MAs)[10]. By applying local search after each of the genetic operators, MAs can search in the space of locally optimal solutions rather than the entire search space of all candidate solutions. However, the exact or near-exact local neighborhood search spends much of computation time to search for the locally best solution around the local region.

In this paper, a new hybrid that incorporates an intelligent local heuristic search into the evolutionary search, for combinatorial optimization problems, is presented. The new local heuristic search is a probabilistic greedy search, and its control parameters of the greediness are chosen in consideration of the landscape characteristic of the local region. It has a merit of controlling the exploitation and thus may reduce time taken to find locally optimal solutions, compared with simulated annealing and local neighborhood search. Also, the hybrid uses a distance-based restriction competition to prevent the premature competition of individuals and restart scheme for further exploration.

This paper is organized as follows. Section 2 describes, in detail, the hybrid stochastic search: the intelligent local heuristic search, the competition scheme, and the restart method. In section 3, results of the proposed hybrid search for two combinatorial optimization problems, i.e., traveling salesman problems(TSPs), and quadratic assignment problems(QAPs) are described. Also, a comparison with canonical evolutionary algorithms and simulated annealing is made, including discussion on their search behavior in terms of the problem size and structure. Conclusion is made in section 4.

II. Hybrid of Evolutionary Search and Intelligent Local Heuristic Search

The proposed hybrid is a population-based heuristic local search and can be regarded as a case of MAs with an elaborate local heuristic search. The general procedure of the hybrid is shown in Fig. 1.

```
begin
 initialize population P;
 repeat
   for each individual i belonging to P
       do i := local-heuristic-search(i);
   // select good individuals from P
   Ps := binary-competition(P);
   // generate new individuals for restart
   Pnew := generation(Ps);
   P := Ps + Pnew;
until (termination=true)
end
```

Fig. 1. Pseudo code of the hybrid algorithm.

Initial individuals are chosen either randomly or generated according to a particular initialization. The local search is performed on each of individuals independently until it reaches a local optimum. Then, individuals compete with others in the population when all individuals have reached a certain development. Fitter local optima, Ps survive the competition probabilistically. Then, new individuals Pnew are generated, and their restart states are chosen to continue searching for better local optima in unexplored regions. The search behavior of the hybrid is characterized by the three components: local heuristic search, types of competition and restart schemes.

2.1 Intelligent Local Heuristic Search

The intelligent local heuristic search(ILHS) is the most important component of the proposed hybrid algorithm, the procedure of which is shown in Fig. 2. To be specific, two inputs are an initial solution s(0) and an initial value of the control parameter, threshold T(0). A probabilistic transition function, p(T(i),D) is used during a predetermined number of iterations, so-called transition epoch L, and if no improvement during the transition epoch, the ILHS stops. Otherwise, it continues to perform the probabilistic search after adjustment of the threshold T(i). During the transition epoch, a neighbor for the next transition is selected in a sequential order.

```
begin
 s(0) and T(0) are given
 repeat i = 1,2,3, ...
   cost(prev) := C(s(i));
   repeat
      // generate a neighbor
      s' := N(s(i));
      D := C(s') - C(s(i));
      // minimization problem
      if(rand( ) < p(T(i),D)) s(i) := s';
      if (escape=true) T(i) = reduce(T(i));
   until (transition epoch)
   if(C(s(i)) = cost(prev)) stop
   else s(i+1) := s(i);
   T(i+1) = reduce(T(i));
 end repeat
end
```

Fig. 2. Pseudo code of the ILHS.

The p(T(i),D), shown in Eq. (1), decides whether or not to accept the move associated with the cost change D. The acceptance probability is determined by the threshold T. When T is 0, the ILHS results in a greedy algorithm, and on the other hand, it allows many uphill moves for large T. It has been observed from a lot of experiments that the ILHS with the value of T slightly greater than 0 is more powerful than those with 0 or high values. It should be noted that p(T(i),D) is slightly different from the Metropolis criterion[11] in that moves that would make the current solution much worse is not accepted, whereas in the Metropolis criterion acceptance probability of the worse solution is given by the exponential form. The choice of acceptance criterion depends on the behavior of the cost function, which will be discussed later.

$$p(T(i),D)=\begin{cases}\min\left\{1,1-\frac{D}{T(i)}\right\}, & \text{if } D\le T(i)\\ 0, & \text{if } D> T(i)\end{cases} \quad (1)$$

Initially, T0 is set to Tmin that is a non-negative value close to zero so that the ILHS performs greedy-like search. After all the individuals converge to local optima and then, their survival is determined based on their cost values. In the next iteration, the ILHS starts from these local optimal solutions or randomly chosen solutions. When the ILHS begins from one of local optima found, T0 is set to Tesc that is larger than Tmin. The reduction of Tesc is needed for the ILHS to converge, which is executed by the reduce() function. Two modes for threshold reduction from Tesc to Tmin are offered: a rapid mode and slow mode. The first reduce() function is used for the rapid mode and the second is used for both of them. The idea of the rapid mode is that Tesc is expected to be switched to Tmin after the ILHS gets out of the basin of the local optimum. Heuristic decision criteria of the time when Tesc switches to Tmin are: (1) when a better solution than the local optimum is found during the search with Tesc; (2) when the ILHS has little chance to return to the same local optimum even though Tesc is reduced to Tmin. Because of complexity, a decision of switching is made in the course of search by examining the degree of returning to the local optimum based on a ratio, G of the number of down moves accepted on the search trajectory and the transition epoch. On the other hand, in the slow mode, the threshold is gradually lowered like the temperature of SA, as shown in Eq. (2).

$$T(i+1) = \max(T_{\min}, \alpha T(i)) \tag{2}$$

where α is a decreasing rate between 0 and 1.

Choosing appropriate values of Tmin and Tesc is important because Tmin determines the degree of exploitation and a high value of Tesc may upset search efforts so far to find the promising region. Indeed, their optimal values rely on the given problem. Therefore, the estimation of Tmin is made by averaging costs of uphill trials during a set of independent greedy moves to multiple local optima, as shown in Eqs. (3)-(4),

$$\widetilde{T_{\min}}^{\,i} = \frac{1}{f_i} \sum_{j=1}^{f_i} \delta_j^{\min +} \tag{3}$$

$$T_{\min} = \frac{1}{M} \sum_{i=1}^{M} \widetilde{T_{\min}}^{\,i} \tag{4}$$

where $\delta_j^{\min +}$ is the cost change of the smallest uphill trials generated during the j-th transition epoch, f_i represents the number of transition epoch to a local optimum in the i-th run of the greedy algorithm, and M represents the number of independent runs. The heuristic seems to be reasonable than random wandering because information of uphill moves during the trajectory along the paths to local optima is meaningful for the ILHS.

Tesc is used to enable the ILHS to escape from the converged local optimum and to continue to search for another local optimum. The value of Tesc is determined by Eq. (5)

$$T_{esc} = T_{\min} \cdot rand(B_l, B_u) \tag{5}$$

where rand(l,u) generates a random number between l and u, and B_l and $B_u (= B_l + \Delta B)$ represent the lower and upper bounds of the random number, respectively, which are chosen by trials-and-errors or some preliminary experiments.

2.2 Restricted Competition

After all individuals converge to local optima, a binary competition occurs between individuals based on their cost values as well as their locations in the search space. For each individual, the competing one is chosen randomly without replacement and competition is made only when the distance between them is less than the predetermined value, so-called competition radius, Rc. The distance can be measured either by calculating how many transitions are required to reach another individual or by calculating some quantities based on problem-specific knowledge. The strategy restricts competition between two individuals that are located far apart in the search space to avoid premature exclusion of possibly promising local regions that otherwise could be identified in the succceding generations. In minimization problems, the binary competition of individuals A and B is performed according to the probability that A survives:

$$\Pr\{A\} = \frac{C_b - C_{\min}}{(C_a - C_{\min}) + (C_b - C_{\min})} \tag{6}$$

where C_a and C_b represent the cost values of A and B, respectively, and $C_{\min}$ represents the minimum of costs of all individuals in the current population.

2.3 Restart Schemes

This section presents how to restart to search so as to find new local optima in the next local search phase. The winners restart to search from the converged local optima with a high value of T, Tesc, because their survival implies that there may exist more promising regions near their locations. There are several ways of choosing new states of the losers. They can be either chosen at random, regardless of local optima found previously, or restarted from the local optima found by winners. The former chooses exploration by extending the search to regions far away from the local optima found. On the other hand, the latter adopts exploitation by concentrating the search on identified local regions.

In the restart phase, these two methods are used to set states of the losers: (1) the concentration(exploitation) mode in which the loser restarts from the local optimum that the competing winner found. In the case, the loser has Tesc to escape from the local optimum. (2) scattering(exploration) mode in which the loser restarts from a new state far away from all of the previously found local optima to find another better local optimum in the unexplored search region. In the scattering mode, a list of the local optima found is maintained to select the restart states. To do this, a population of states are created randomly, and then the best state among them is chosen only if it is away from local optima in the list by a prescribed distance.

III. Experimental Results

Experimental results of the proposed hybrid on two combinatorial optimization problems, traveling salesman problems(TSP), and quadratic assignment problems(QAP) are presented, together with comparisons with simulated annealing and canonical evolutionary algorithms.

3.1 TSP and QAP Instances

TSPs and QAPs are well-known NP-complete problems whose cost functions depend on the permutation of elements. The goal of TSP is to minimize the length of a tour starting from any city, visiting each of the N cities once and only once, and returning to the departure city. The TSPs in which the cities are placed in [0,1]2 and the distances are symmetric and Euclidean were considered in the study. Two types of city distributions, uniformly distributed cities and cities on a lattice with perturbation (50% of distance between neighboring cities) were used to generate TSPs with different number of cities (N = 50, 100, 200, 300).

The QAP is a facilities location problem, the task of which is to assign N facilities (with given flows between them) to N or more locations (with given distances between them) in such a way that the sum of the product between flows and distances is minimized. QAPs with different sizes, N = 30, 72, 81, were generated in a manner that the distance matrices are rectilinear and two sets of flows, high- flow and low-flow dominance, are included.

3.2. Results and Discussions

The performance of the hybrid was evaluated compared with those of simulated annealing and canonical evolutionary algorithms. For transition operators, inversion, which reverses a part of the tour, was chosen for the TSPs since it is more effective than transposition, which swaps two randomly chosen cities[12]. Based on preliminary results, the transposition was used for the QAPs. The experimental conditions of the three algorithms are as follows.

The SA described in [13] was adopted where geometric annealing scheme is used and initial temperature values are determined by the degree, Pa to which uphill moves during a certain number of random transitions are allowed. The chain length of the SA was chosen roughly according to the number of neighbors $\binom{N}{2}$, and Pa of 0.8 and cooling rate of 0.987 were used.

EA with q-tournament selection[1], a class of EAs, was adopted in the experiments because of its effectiveness over other EAs such as evolution strategies and genetic algorithms on many optimization problems[3,5]. Several population sizes were tested in proportion to the problem size, e.g., from 100 to 400 in such a way that the EA converges to a local optimum within the given number of evaluations. Crossover or recombination operators were not used because it seemed to offer neither substantial benefits in convergence time nor good performance without incorporation of problem-specific knowledge.

The transition procedure of the hybrid for permutation problems during the transition epoch is as follows. For each element, another element is randomly selected and then both elements are exchanged. The resulting new neighbor is accepted probabilistically, as shown in Eq. (1). It is repeated until a new neighbor is accepted or the iteration reaches the predetermined number of times, ηN. Parameters of the hybrid are given by table 1, and there values were determined by the trial-and-errors.

Table 1. Parameter values of the hybrid for TSPs and QAPs.

Parameters	TSPs	QAPs
$(B_l, \Delta B)$	(5.0, 5.0)	(5.0, 5.0)
G	0.1	0.3
Rc	0.4	0.9
η	0.15	0.15

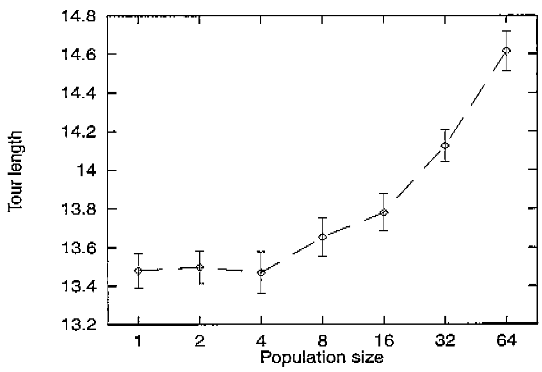


Fig. 3. The performance of the hybrid for a TSP of 300 cities with respect to the number of individuals. Total number of evaluations is 107.

All experiments of the hybrid were performed with population size of 8 except for large-sized problems like experiments of 200-, 300-cities TSPs and 72-, 81-facilities QAPs where 4 individuals are used. It is a question that, given a fixed number of evaluations, which one is a better strategy, a large number of individuals each performing the local search for a small number of generations or a small number of individuals for a large number of generations. As shown in Fig. 3, It was observed in the hybrid that, the smaller the number of individuals is (the longer each local search performs), the better performance the hybrid gives. Similar results were obtained for the QAPs.

As shown in table 2, the hybrid provides a significant improvement over the EA regardless of the types of TSP instances. Both of the hybrids with the rapid and slow modes gave better results than the SA for TSPs of perturbated lattice-type cities, regardless of problem size, but only the hybrid with the slow mode outperforms the SA for TSPs of uniformly distributed cities. Comparing the results for the uniform and the lattice TSPs, the lattice TSPs seem to be more amenable to the hybrid compared with the SA than uniform TSPs. It is conjectured that the phenomenon is due to different characteristics of heights of uphill barriers on the landscape, which is determined by locations of cities in the case of the Euclidean TSPs. For QAPs, the SA is apparently superior to the EA consistently irrespective of problem size and structure, as in TSPs. The hybrids also outperform the EA for two types of QAPs. The hybrid with the slow mode is slightly better than one with the rapid mode regardless of the QAP type, and the performance difference becomes larger as the problem size increases. However, the hybrids produce slightly poor solutions than those of the SA, unlike in TSP. These results suggest that the performances of the hybrid and the SA are apparently different depending on the given problem structure.

Table 2. The experimental results of SA, EA, and the hybrid for two types of TSPs. The values in ( ) indicate the standard deviation of tour lengths.

Lattice type	Size			
	50	100	200	300
SA	5.244 (0.031)	8.208 (0.062)	11.134 (0.064)	14.049 (0.075)
EA	5.275 (0.065)	8.258 (0.084)	11.509 (0.108)	14.510 (0.151)
Hybrid I	5.235 (0.012)	8.184 (0.042)	11.069 (0.082)	13.989 (0.064)
Hybrid II	5.242 (0.035)	8.230 (0.037)	11.080 (0.078)	13.966 (0.095)
Uniform type	Size			
	50	100	200	300
SA	5.887 (0.027)	8.020 (0.042)	11.168 (0.097)	13.518 (0.094)
EA	5.907 (0.066)	8.098 (0.078)	11.366 (0.119)	13.930 (0.167)
Hybrid I	5.845 (0.001)	8.014 (0.058)	11.207 (0.094)	13.660 (0.155)
Hybrid II	5.851 (0.079)	8.001 (0.046)	11.167 (0.094)	13.468 (0.110)

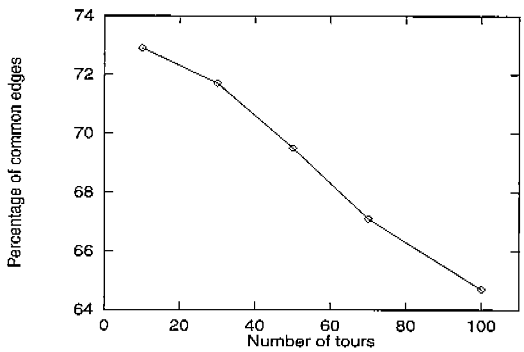
In this sense, TSPs seem to be a problem belonging to a class of combinatorial optimization problems that can be successfully attacked with the hybrid. To be specific, many edges between cities in the tour are commonly

Table 3. The experimental results of SA, EA, and the hybrid for two types of QAPs.

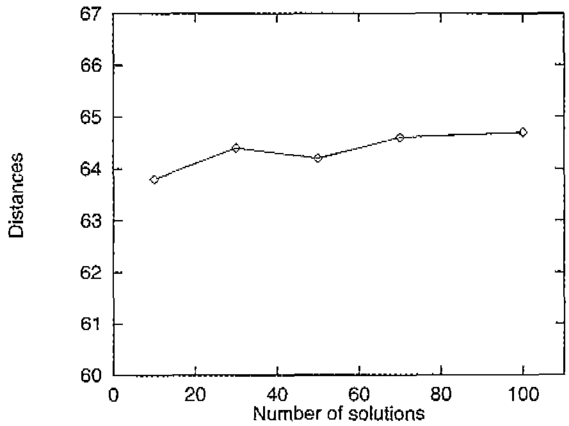
Dense type	Size		
	30	72	81
SA	29050 (26)	252521 (151)	342080 (143)
EA	29111 (88)	253814 (333)	343790 (636)
Hybrid I	29023 (50)	252700 (240)	342678 (340)
Hybrid II	29010 (23)	252670 (125)	342370 (270)
Sparse type	Size		
	30	72	81
SA	9123 (58)	111823 (223)	154598 (195)
EA	9177 (71)	113286 (557)	156593 (692)
Hybrid I	9066 (47)	112025 (238)	154835 (197)
Hybrid II	9112 (34)	112012 (87)	154783 (169)

local optima are located closely each other on the search space, which can be explained by Fig. 4(a). The property of TSPs is supported by the comparison experiments between the concentration and the scattering modes of the ILHS. The poor performance of the scattering mode is likely to be caused by less concentration on looking for new local optima around the promising region previously found. On the other hand, QAPs do not seem to have such a property of the distribution of local optima in the search space. Unlike TSPs, distances between local optima, as shown in Fig. 4(b), do not have a certain correlation of local optima. Because there is no relationship between local optima in QAPs, it may be difficult for the evolutionary search like the hybrid to be effective even though the local heuristic search is employed. In QAPs, it is difficult to look for promising regions by exploiting the correlation structure of local optima. In this sense, QAPs seem to be one of the problems where the evolutionary search is less effective than the SA.

Experiments were performed with smaller number of evaluations (the total number of evaluations) to check whether or not performance dependency on class of problems is consistent. The results indicate that the behaviors of the hybrid and the SA are not influenced by the given number of evaluations.



(a)



(b)

Fig. 4. Closeness/distance between local optima with respect to the number of local optima. 100 local optima was found by 100 runs of the hybrid. The point for 30 on the horizontal axis denotes the average common edges (distance for QAPs) of the best 30 local optima among 100 optima. (a) TSP of random 100 cities (b) QAP of 72 facilities.

IV. Conclusions

In this paper, a new hybrid evolutionary algorithm utilizing a local heuristic search, for combinatorial optimization problems, was presented. The components of the hybrid, competition schemes and restart methods were described, and in particular, a new sophisticated local heuristic search was introduced. The performance of the proposed hybrid was investigated on two well-known problems of different sizes: traveling salesman problems (TSPs), and quadratic assignment problems(QAPs) and compared with the performance of simulated annealing and canonical evolutionary algorithms. The experimental results have shown that the hybrid was clearly superior to canonical evolutionary algorithms and was significantly better than simulated annealing for TSPs, but not for QAPs. The their behaviors seem to be caused by the different characteristics of the search space of the two problems. Although it is necessary to test the hybrid on many classes of problems, compared with other currently good search heuristics, we believe that there exists a class of problems where the hybrid is able to achieve better performance than simulated annealing.

References

[1] T. Bäck and H.-P. Schwefel, "Evolutionary computation: an overview," *Proc. of ICEC*, pp. 20-29, 1996.
[2] K. Park, "A comparative study of genetic search," *Proc. of ICGA*, pp. 512-519, 1995.
[3] L.-J. Park, D.-K. Nam, et. al., "An empirical comparison of simulated annealing and genetic algorithms on NK fitness landscape," *Proc. of ICEC*, pp. 147-151, 1997.
[4] R. Salomon, "Raising theoretical questions about the utility of genetic algorithms," *Proc. of Conf. on Evolutionary Programming*, pp. 275-284, 1998.
[5] V. Maniezzo, M. Dorigo, and A. Colomi, "An experimental comparison of eight evolutionary heuristics applied to the quadratic assignment problem," *European Journal of Oper. Res.*, vol. 81, pp. 188-204, 1995.
[6] F.-T. Lin, C.-Y. Kao, and C.-C. Hsu, "Applying the genetic approach to simulated annealing in solving some NP-hard problem," *IEEE Trans. on SMC*, vol. 23, pp. 1752-1767, 1993.
[7] L.-J. Park and C. H. Park, "A fusion technique of GA and SA and its application to QAPs," *Proc. of ICONIP*, pp. 838-842, 1996.
[8] B. Li and W. Jiang, "A novel stochastic search algorithm", *IEEE Trans. on SMC*, vol. 30, pp. 193-198, 2000.
[9] P. Merz and B. Freisleben, "Genetic local search for the TSP: New results," *Proc. of ICEC*, pp. 159-164, 1997.
[10] P. Moscato and M. G. Norman, "A memetic approach for the TSP implementation for combinatorial optimization on message-passing systems," *Parallel Computing and Transputer Applications*, pp. 177-186, 1992.
[11] N. Metropolis, A. W. Rosenbluth, et. al. "Equation of state calculation by fast computing machines," *J. of Chem. Phys.*, vol. 21, pp. 1087-1092, 1953.
[12] P. F. Stadler and W. Schnabl, "The landscape of the traveling salesman problem," *Phy. Letters A*, vol. 161, pp. 337-344, 1992.
[13] D. S. Johnson, C. R. Aragon, et. al., "Optimization by simulated annealing: An experimental evaluation, Part I, Graph partitioning," *Oper. Res.*, vol. 37, pp. 865-892, 1989.