COOPERATING OF LOCAL SEARCHES BASED HYPERHEURISTIC APPROACH FOR SOLVING TRAVELING SALESMAN PROBLEM

Montazeri Mitra[1], Abbas Bahrololoum[2], Hossein Nezamabadi-pour[3], Mahdieh Soleymani Baghshah[2] and Mahdieh Montazeri[4]

[1] Member of Young Researchers Society, Computer Engineering Department, Shahid Bahonar University, Kerman, Iran
[2] Computer Engineering Department, Shahid Bahonar University, Kerman, Iran
[3] Electronic Engineering Department, Shahid Bahonar University, Kerman, Iran
[4] Kerman Medical University, Kerman, Iran

Keywords: Meta-heuristic algorithm, Hyper-heuristic algorithm, Traveling Salesman Problem.

Abstract: Until now various heuristic optimization methods have been developed for solving NP-Hard problems. These methods by trading off between exploration and exploitation attempt to find an optimum solution. In this paper, we introduce a new optimization algorithm based on hyper-heuristic for solving TSP. A hyper-heuristic approach has two layers. In low level, we have six local searches and in high level we use Genetic Algorithm. Genetic Algorithm corporate local searches efficiency. The proposed method has high ability to searching in solution space and explores and exploit appropriately. This method exploits space depended on characteristics of the region of the solution space that is currently under exploration and also the performance history of local searches. The proposed method is used to solve TSP and compared with well-known methods. The experimental results confirm the efficiency of the proposed method.

1 INTRODUCTION

The traveling salesman problem, TSP, which is known as NP-hard in the field of combinatorial optimization has been studied by many researchers since it appeared in the 1930. In this problem, we have *n* cities and for each pair of cities a connection cost is defined. The problem is concerned with finding an optimum path which starts from one city and passes all other cities exactly once and finally returns to the starting city. The exact approaches such as dynamic programming (Neapolitan and Naimpour, 2004), can find exact solution but if the number of cities is large, the problem will be intractable. Therefore heuristic approaches been attended in the last decades. This class of methods has been achieved acceptable solutions within reasonable time and is relatively efficient for dealing NP-hard problems (Ozaglam and Cunkas, 2008). However, since a heuristic operator always seeks to find immediate improvement, the heuristic method is at the risk of quickly trapping a local optimum. In such cases, it is often desirable to guide the heuristic search by employing a strategy known as a meta-heuristic. This strategy encourages the discovery of better solutions in the search space by tightening a focus on good solutions and improving upon them (intensification), and to encourage the exploration of the solution space by broadening the focus of the search into new areas (diversification) (O'Brien, 2007). Till now, many studies on TSP are performed using the meta-heuristic algorithms. Dorigo and Gambardella (Dorigo and Gambardella, 2008) used ants of the artificial colony to generate successively shorter feasible tours by using information accumulated in the form of a pheromone trail deposited on the edges of the TSP graph and their results in comparison with previous methods were satisfying.

Some drawbacks cause that meta-heuristic algorithms couldn't be suitable for these problems. One of these drawbacks is that meta-heuristic algorithms often require domain knowledge and so parameters are needed to be tuned expertly. For example, in GA, which is known as a meta-heuristic algorithm, the chromosome of a genetic algorithm is either the solution of the target problem or a structure of the solution. This means that problem specific knowledge is essential in the design of chromosomes. No Free Lunch Theorem introduced by Wolpert and MacReady (Wolpert and MacReady, 1997) states no individual algorithm is best and each algorithm has its own advantages and drawbacks. Therefore, we need algorithms which automatically combine the strength and compensate the weakness of the known heuristics.

In meta-heuristics, there is no further focus on the exploitation aspect when a potential region is identified (Ang and Tan, 2010). Thus, Memetic Algorithms (MA) which incorporate local improvement search into meta-heuristics, were proposed (Ong and Keane, 2004). Experimental studies have been shown that a hybrid of a meta-heuristic and a local search is capable of more efficient search capabilities (Merz and Freisleben, 1999). MA uses only one local search to explore the whole solution space. As shown in Ref. (Ong and Keane, 2004), the choice of the local search has an important impact on the search performance of MA. Therefore, inappropriate use of local search may result in memetic algorithms performing poorer than standard GA (Ong and Keane, 2004). Different local search methods have different biases. These biases may be suitable for some classes of problems but not for others. Therefore, we need to have multiple local searches to achieve improved search performance and reduce the probability of utilizing an inappropriate local method. However, we need to have a supervisor managing the choice of local searches that should be applied at any time. Therefore, for solving the above mentioned problems, new approach known as hyper-heuristic is proposed. In this paper, we propose a novel algorithm that is based on hyper-heuristic approach for solving TSP. The proposed method cooperates local searches appropriately based on hyper-heuristic approaches.

The rest of this article is organized as follows: Section 2 describes hyper-heuristic in details. In Section 3, we explain the proposed method. Finally, experimental results and conclusion are presented in Section 4 and 5, respectively.

2 HYPER-HEURISTIC

Hyper-heuristic is an approach that was proposed in 2000 by Cowling et al (Cowling et al., 2001). It is a heuristic approach which selects heuristics and is a higher abstract level over meta-heuristics.

General framework of hyper-heuristic has two levels. At the low level, there is a set of local searches, which also known as Low Level Heuristics (LLH)s, to modify solution locally in an attempt to return an improved solution. At the high level, there is a black-box choice function. It manages the selection of LLHs. These selections depend upon the characteristics of the region of the solution space currently under exploration and the performance history of the local searches.

3 THE PROPOSED METHOD

In the proposed method, GA is used as the high level hyper-heuristic. This GA is not a direct GA, in fact each individual in GA's population consist of a sequence of integer numbers. Each number is an LLH choice which tells us which LLH must be applied and each individual tells in which order to apply LLHs. In fact, the key note behind this method is its LLHs cooperation which is act of operating together. Optimization problems in science and engineering commonly have large search spaces, which contain numerous local landscapes of diverse forms. The joint operation of diverse LLHs to cope with the large search space is facilitated via the problem decomposition or diversity in the LLH selection (Ong and Keane, 2004). Fig. 1 shows the flowchart of the proposed method. In the next sub-sections, we introduce the proposed method.

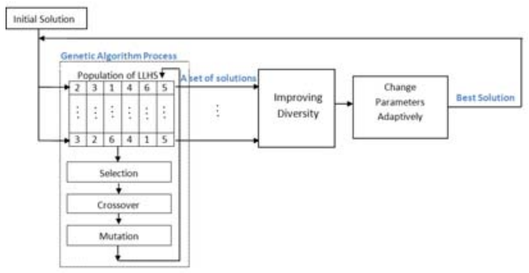

Figure 1: Flowchart of the proposed method.

3.1 Genetic Algorithm Process

3.1.1 Population of Low Level Heuristic

Hyper-heuristic concludes from a set of LLHs created by human expert. In the proposed method, six LLHs are used which make small change in current solution: First LLH is 2-OPT algorithm. It basically removes two edges from the tour, and reconnects the two paths created. There is only one

Table 1: The best, worst and average solutions that achieve by the proposed method over 100 runs.

Database	Number of city (n)	Best known solution	Average solutions	Minimum solutions	Maximum solutions
C20	20	62575	62575	62575	62575
S21	21	60000	60000	60000	60000
C30	30	62716	62716	62716	62716
F32	32	84180	85645	84180	92123
C40	40	62768	62768	62768	62768
F41	41	68168	68738	68168	91774

Table 2: Comparison average solutions of the proposed method with average solutions of well known methods on Ozcan database on 100 runs.

Database	Number of city (*n*)	Best known solution	The Proposed method	Results obtained from literature				
				PSO (Çunka and Özsalam, 2009)	GA (Çunka and Özsalam, 2009)	TGMA-HC (Ozcan and Erenturk, 2004)	SSMA-HC (Ozcan and Erenturk, 2004)	IDGA (Lau and Xiao, 2008)
C20	20	62575	**62575**	63276	63188	134497	107493	116367
S21	21	60000	**60000**	60786	60648	120276	93626	121439
C30	30	62716	**62716**	63625	63356	165795	116633	120025
F32	32	84180	85645	85535	**85392**	146317	108048	124640
C40	40	62768	**62768**	64212	63753	197829	128117	125721
F41	41	68168	**68738**	69995	69702	158461	115860	125461

way to reconnect the two paths so that the resulted tour will be valid (Nilsson, 2003). Second LLH is 2-CHANGE turns a tour into a slightly different tour. It randomly select two edge *(a, b), (c, d)* where *a#d, b#c* and also replaces them with *(a, c), (b, d)* (Keller and Poli, 2008). Third LLH is IF 3-CHANGE that randomly selects edges as arguments for 3-CHANGE, and then IF 3-CHANGE makes betters the cycle for the given arguments. It actually executes 3-CHANGE (Keller and Poli, 2008). Forth LLH is IF 2-CHANGE. This LLH executes 2-CHANGE if this will generates shorter the tour under construction (Keller and Poli, 2008). Fifth LLH does if first LLH will be shorter (Nilsson, 2003). Sixth LLH is as a 3- CHANGE. It deletes three mutually disjoint edges from a given tour, and reconnects the obtained three paths so that a different tour results (Keller and Poli, 2008).

These LLHs decode in population of GA.

3.1.2 Genetic Operators

We use the roulette wheel selection operator, one-point crossover and a mutation operator that selects some positions in one individual randomly and mutates genes at these positions to other values ranging from 1 to 6 (Davis, 1991).

3.2 Improving Diversity

Using intensive LLHs causes to miss diversity of search space. In fact, in our GA process, we increase exploitation and missing a few explorations due to increasing the selection pressure. Therefore after the LLHs apply to current solution, we add a special mutation step. This step is used for the purpose of maintaining the population diversity. The special mutation which is being used for TSP is swap mutation which selects randomly two positions and swaps them (Banzhaf, 1990).

3.3 Change Parameters Adaptively

In this step, the mutation rate and crossover rate are adapted in each generation according to the change in fitness (Cowling et al., 2002).

4 EXPERIMENTAL RESULT

The proposed method has been tested on datasets suggested by Ozcan et al. in 2004 (Ozcan and Erenturk, 2004).

In all cases, population size is set to 80; number of generation is 10000, crossover rate 0.7 and 0.5 for mutation rate.

Table 1 shows the best, worst and average solutions achieved by the proposed method over 100 runs on different datasets. As seen in these tables, in most cases, the proposed method finds global optimum over 100 runs. This issue is considerable.

Table 2 shows the comparison of the proposed method with literature results. In each row the best solution is bold. As shown in this table, in most cases, the proposed method finds the routes better than other methods. The results of the proposed method have been compared with those of PSO and GA implemented in Ref. (Çunka and Özsalam, 2009). The proposed MA in Ref. (Ozcan and Erenturk, 2004) was introduced as Steady State Memetic Algorithm with Hill Climbing (SSMA-HC) and a Trans-Generational Memetic Algorithm with Hill Climbing (TGMA-HC).

Finally, we compare our proposed method with Iterative Deepening Genetic Annealing Algorithm (IDGA) method to show that our method is more efficient than both the previous methods and also a proper hybrid of them. In Ref. (Lau and Xiao, 2008), it was verified that IDGA is more appropriate than SA and GA alone or hybrid for solving TSP.

5 CONCLUSIONS

In this paper, a new optimization algorithm based on hyper-heuristic approach was introduced for solving TSP. Proposed method searches the solution space appropriately in which depended upon the characteristics of the region of the solution space currently under exploration and the performance history of local search. Our method used GA to select local search. In which local searches were act of operating together, our method cooperated local searches. The proposed method also remained robust to increasing the number of dimension which is a key element in the development of any evolutionary algorithm. Our method had an excellent convergence rate. In fact, finding the global optimum in high speed is the salient property of our method. This method was used to solve TSP and compared with different well-known methods. Experimental results confirmed the superior performance of it.

REFERENCES

Neapolitan, R., Naimpour, K., 2004. Foundation of Application Using C++ Pseudo Code, third Edition, *Jones and Bartlett Publishers.*

Ozaglam, M. Y., and Cunkas, M., 2008. Particle swarm optimization algorithm for solving optimization problems. *Polytechnic* 11: 193–198.

Dorigo M., Gambardella L. M., 2008. Ant colony system: A cooperative learning approach to the traveling salesman problem, *IEEE Transaction on Evolutionary Computation*, 1, 53–66.

Wolpert, D. and MacReady W. G. 1997. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1:67 82.

Ang, J. H., Tan K. C., A. A. Mamun 2010. An evolutionary memetic algorithm for rule extraction Expert Systems with Applications 37 1302–1315.

Ong Y. S., Keane A. J. 2004. Meta-Lamarckian learning in memetic algorithms. *IEEE Transactions on Evolutionary Computation*, 8(2), 99–110.

Merz, P., Freisleben, B. 1999. A comparison of memetic algorithms, Tabu search, and ant colonies for the quadratic assignment problem. *In Proceedings of the congress on evolutionary computation* (Vol. 1, pp. 2063–2070).

Cowling P., Kendall G. Soubeiga E. 2001. A Parameter-Free Hyperheuristic for Scheduling a Sales Summit. *In proceedings of 4th Metahuristics International Conference* (MIC 2001), Porto Portugal, 16-20, pp 127-131.

Keller R. E. Poli R., 2008. Self-adaptive Hyperheuristic and Greedy Search, *IEEE computer and information science*.

Nilsson Ch., 2003. Heuristics for the Traveling Salesman Problem, *International conference on heuristic*.

Davis L., 1991. Handbook of Genetic Algorithms. Van Nostrand Reinhold, New York.

Banzhaf W., 1990. The molecular traveling salesman, Biological Cybernetics, vol. 64, pp. 7–14.

Cowling P., Kendall G., Han L., 2002. An Investigation of a Hyperheuristic Genetic Algorithm Applied to a Trainer Scheduling Problem. *In Proceedings of the 2002 Congress on Evolutionary Computation* (CEC 2002), Pages 1185-1190, Hilton Hawaiian Vilage Hotel, Honolulu, Hawaii, 12-17.

Ozcan E., Erenturk M., 2004. A brief review of memetic algorithms for solving Euclidean 2D traveling salesrep problem. *Proc. of the 13th Turkish Symposium on Artificial Intelligence and Neural Networks* 99–108.

Çunka, M., Özsalam M. Y., 2009. A comparative study on particle swarm optimization and genetic algorithm for traveling salesmen problem, *Taylor & Francis*, Cybernetics and Systems.

Lau, H. C., Xiao, F., 2008. The oil drilling model and iterative deepening genetic annealing algorithm for the TSP, In A. Fink and F. Rothlauf (eds), *Advanced in Computational Intelligence in Transportation and Logistics*, Studies in Computational Intelligence. Springer, 169-184.