A Parallel Tabu Search Algorithm Based on Partitioning Principle for TSPs

Yi HE[1], *Guangyuan LIU*[2] *and* *Yuhui QIU*[2]

[1]School of Economics and Management, Chongqing Normal University, Chongqing 400047, P.R. China
[2]Faculty of Computer & Information Science, Southwest-China University, Chongqing 400715, P.R. China

Summary
We presented a parallel tabu search (PTS) algorithm for the traveling salesman problem (TSP), which is NP-hard. To parallelize tabu search (TS) algorithm efficiently, the search space decomposition based on partition principle was used to balance the computing load, while exploitation in subspace had been boosted by an adaptive search strategy of intensification and diversification. Numerical results illustrated this algorithm was efficient and easy to implement.
Key words:
Parallel Tabu Search, Meta-heuristic, TSP

1. Introduction

Many heuristic methods currently used in combination optimization are inspired by adaptive natural behaviors or natural systems, such as genetic algorithms (GA), simulated annealing (SA), neural networks, etc. tabu search (TS) belongs to this class of biologically inspired heuristics and it is a simulation of intelligence process. The basic idea is to avoid the explored space or prohibit the performed operation.

Traveling salesman problem (TSP) is one of the hardest among the NP-hard combinatorial optimization problems, whose task is to find a route through a given set of cities with shortest possible length, and there is no known deterministic optimal search algorithm that runs in polynomial time for TSP [1].

There are several approximate algorithms to solve the combinatorial problems, such as nearest neighbor, greedy, branch and bound, and nested partitions method, etc [2-5]. Furthermore, in recent decades, some new meta-heuristics, such as neural network, SA, GA, tabu search, and ant colony optimization (ACO), has also been applied to solve this problem [6-9], and satisfactory results are obtained. With the growth of the problem size, parallelization of these meta-heuristics is one interesting topic. Moreover, parallel processing is implemented with low cost and very practical since MPUs or Clusters are available at low price and high performance, and message passing interface (MPI) come up to the standard. Consequently, our aim is to develop a Parallel model based on partition for large scale TSPs.

Manuscript received August 5, 2006.
Manuscript revised August 25, 2006.

The paper is organized as follows. First, we will introduce tabu search algorithm in Section 2. Our Parallel TS algorithm for TSP will be detailed in Section 3. In Section 4, we gave the results of experiments for two standard instances from the TSP-library and discussions follow in Section 5.

2. Tabu Search

Tabu search is a meta-heuristic method that can be used to solve combinatorial optimization problems [7-8]. Its flexible control framework and several significant successes in solving NP-hard problems, e.g. TSP, aroused rapid growth of its application. The method of neighborhood exploration and the use of short-term and long-term memories distinguish tabu search from local search and other heuristic search methods, and result in lower computational cost and better space exploration [10].

Tabu search involves a lot of techniques and strategies, but it mainly comes from the use of short-term memories (tabu list) that keep track of recently examined solutions, intending to avoid cycling in the solution exploration. In addition to the tabu list (TL), a long-term memories and other prior information about the solution can used to improve the intensification and diversification of the search. It can be confirmed that the strategy of intensification and diversification is very important at most time. Therefore, we proposed an adaptive search strategy of intensification and diversification to improve the efficiency of TS, its main idea was to adjust dynamically the numbers of intensification elements and diversification elements in candidate set respectively by interactive cooperation between neighborhood and candidate set. It will be detailed in later section.

With the requirement of solving the large-scale combinatorial optimization problems, PTS (Parallel Tabu Search) was involuntarily proposed and applied successfully to many fields [12-16]. In this paper, a coarse-grained PTS approach based on partition principle was proposed and it will be described in detail in next.

3. A PTS Algorithm for TSP

3.1 Parallel Model

The parallel model was an asynchronous master/slave paradigm. The master process executes the solution space partitioning and the task assigning, and the slave process executes the search procedure.

First of all, the high quality initial solution was generated by greedy method [11] on master, and then it was partitioned into m about equal sub-sequences, so that the size of sub-sequence would be reduced to about 1/m times of primary size. Consider the size of TSP is n, the solution space complexity is O(n!). As a result, the subspace complexity would be reduced to O((n/m)!) for each slave. These m sub-sequences were constructed a job queues. A job would be sent to a slave process as soon as it was idle when the master process detected it. When a salve process finished a job, it sent the result back to the master and waited for next job. The above procedure continued until all the sub-sequences were finished and their results were sent back to the master. When the master received all the results, it merged all the sub-solution into an integrated solution. To heel, the integrated solution was partitioned into m parts and sent to slave processes again and repeated the above procedure until the terminal condition was satisfied. In order to maintain the diversity of exploitation solution space, the divided points of the TSP ring route must be chosen at random. For each sub- sequence, the start point city and the end point city were fixed during running on slave processes so that the master easily merged them into a legal and integrated solution. After certain partition times and searching for subspace, the master broadcasted the terminal message to each slave process and outputted the global best solution, then halt.

3.2 TS Algorithm Executed on Slave Process

Described as the above, a serial TS program was run on slave process. In order to enhance the performance of TS, an adaptive search strategy of intensification and diversification was proposed. Its details were as follows:

3.2.1 Tabu Objective

The tour length was taken as tabu objective, and it was also treated as evaluation value.

3.2.2 An Adaptive Search Strategy of Intensification and Diversification

In TS, intensification strategies are used to encourage TS to search more thoroughly the neighborhood of elite solutions in order to obtain the global optimal, and diversification strategies are used to encourage TS to search solution space as widely as possible, especially the unvisited regions before. Diversification strategies may be more important than intensification strategies when searching falls into a local optimal, for it can change the search direction and makes it possible to jump out of the local optimal. Therefore, intensification strategies and diversification strategies are very important and the coordinatively important, unfortunately, they are often conflicting in many cases, because it is usually hard to determine what time for intensification search and what time for diversification search. To eliminate the conflicting, an adaptive search strategy of intensification and diversification was proposed. This strategy achieved the goal by adjusting dynamically the number of intensification elements and diversification elements (detailed explanation in the next paragraph) in candidate set.

Demonstrated with Fig.1, we divided the elements in neighborhood as well as candidate set into two parts. The former part is called intensification elements, used for intensification search, and the later part is called diversification elements, used for diversification search. The intensification elements were high quality and were generated by nearest swap method. That is, for each sub-sequence, except the start city and the end city, each city was swapped its position with the next city, viz. the nearest city.

The diversification elements were generated by the randomization swap method, which choose randomly two cities from the current solution and exchange their position except the start city and the end city for each sub-sequence.

The same as neighborhood, the elements of candidate set were also divided into two parts, named intensification elements and diversification elements respectively. The intensification elements were composed of the best solutions picked out from the intensification elements in neighborhood. However, the diversification elements were composed of the solutions randomly picked out from the diversification elements in neighborhood. Why randomly pick out these elements? It is to maintain the diversity of these elements. The number of intensification elements and diversification elements in candidate set will be adjusted dynamically according to the solution quality during the search procedure. The detailed operating method is as follows:

Let *CL* denote the size of candidate set, i.e. the number of all the elements in candidate set (including intensification and diversification). Let *DL* denote the lengths of the divided point, that is, the elements from 1 to *DL* are intensification elements, and the elements from *DL*+1 to *CL* are diversification elements (demonstrated with Fig.2). Before the iteration search, first set *DL* = *CL*/2, viz. the number of intensification and diversification elements are exactly half respectively. During the iteration search, *DL* was adjusted dynamically according to the following rules:

If the current solution (*x_current*) obtained at the current step is better than that (*x_previous*) obtained at preceding step, *DL* = *DL*+1; otherwise, *DL* = *DL*–1. To ensure intensification search, *DL* is no longer subtracted when *DL* = 1; to ensure diversification search, *DL* is no longer added when *DL* = *CL*–1. That is to say, in the entire procedure, candidate set includes intensification elements and diversification elements simultaneously; moreover, the number of them is not less than 1 respectively.

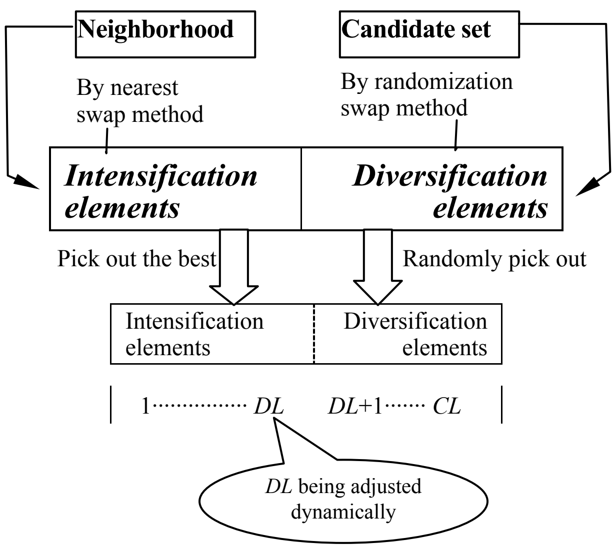

Fig. 1 The adaptive search strategy of intensification and diversification

When the solution quality is improved, the number of intensification elements will increase, therefore, the probability of intensification search will increase too; whereas the probability of diversification search will decrease. On the contrary, when the solution quality is not improved, the probability of diversification search will increase, and it will expand the search space around the current solution (*x_current*) in order to escape the local optimal. These alternate intensification and diversification processes continue until the termination condition is satisfied. Doing like this, TS can automatically determine what time for intensification search and what time for diversification search. Owing to the adaptivity of this strategy, we called this strategy-based TS as an adaptive Tabu Search.

4. Simulation Experiments

To examine the efficiency of PTS, two TSP instances were selected from TSPLIB [17], namely, pcb442 and att532. Let x denotes the numbers of cities in a sub-consequence, which was computed on slave process. The parameters, including tabu tenure, size of neighborhood and candidate set, partition times, max iterated steps and etc. were listed in Tab.I.

Our proposed PTS was run on a workstation composed of two CPUs (Pentium IV, 2.0 GHz) and five processes, including one master process and four slave processes, implemented by MatlabMPI [18]. For each TSP instance, the PTS ran 20 times and all the results were shown on Tab.II. The best tour routes obtained by PTS were shown on Fig.2 and Fig.3 respectively.

Table 1: TS algorithm parameters setting

TSP instance	*pcb442*	*att532*
tabu tenure	x	x
neighborhood size	$3x$	$3x$
candidate set size	10	10
max iterated steps of Serial TS	40	50
numbers of sub-sequence	20	30

Table 2: Experiment results of PTS

TSP instance	Known optimum	Experiment results		Gap (%)	
		best	average	best	average
pcb442	50778	50837	51601.7	0.12	1.62
att532	27686	28753	29394.3	3.85	6.17

Gap: the percentage difference between the solution and the known optimum

Table 3: Comparison with other algorithms

TSP instance	Known optimum	Gap of different algorithms (%)		
		Genetic Algorithm [19]	Parallel 2-opt algorithm [20]	Parallel Tabu Search
pcb442	50778	12.76	—	0.12
att532	27686	16.18	6.5	3.85

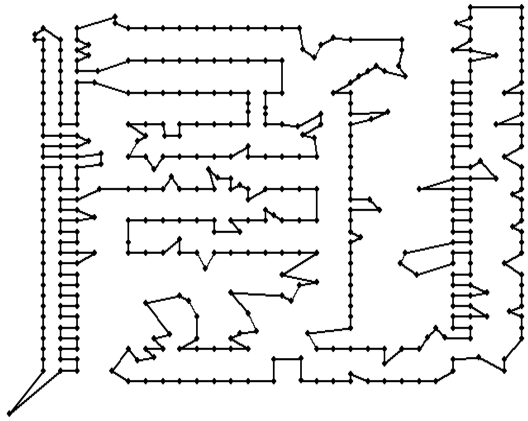

Fig.2 tour route of *pcb442*

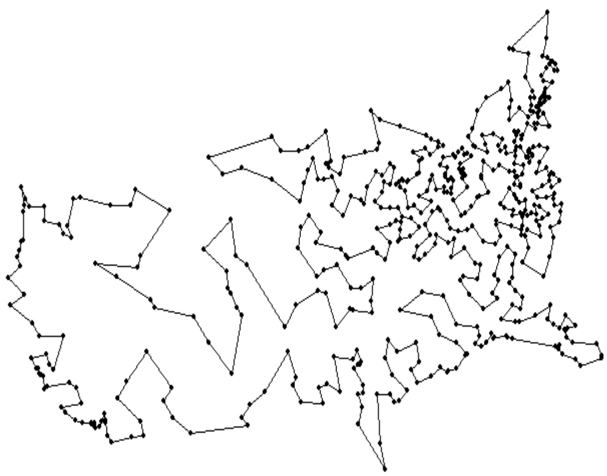

Fig.3 tour route of *att532*

5. Conclusions

In literature [19], three new crossover operators for GAs were proposed in solving the TSP. The best experience results of *pcb442* and *att532* were retrieved from Tab.10 and Tab.11 of the reference [19] respectively. In addition, a scalable parallel local search algorithm based on data parallelism was proposed for TSPs, detailed in reference [20]. The best result of *att532* was retrieved from the Tab.1 of the paper. The comparison of best results between these two algorithms and our PTS were described in Tab3.

From Tab.3, we can find out that the best or average gap of our proposed PTS are less than other two algorithms., and the results are satisfied. Therefore, a conclusion can be drawn that our proposed PTS is feasible and effective. Furthermore, the adaptive search strategy of intensification and diversification needs no particular information about the problem. It can be extensively applied to other combinatorial optimization problems, such as QAP (Quadratic Assignment Problem), KP (Knapsack Problem) and VRP (Vehicle Routing Problem), etc. This is our next work and we will develop more efficient partition strategy to further improve the efficiency of the PTS.

Acknowledgments

This work was partly supported by key Project of Chinese Ministry of Education (104262) and fund project of Chongqing Education Commission (KJ060801), China.

References

[1] Christos H. Papadimitriou, The Euclidean Traveling Salesman Problem is NP-complete, Theoretical Computer Science, Vol.4, No.3, pp. 237-244, 1977.

[2] Moon, Chiung; Kim, Jongsoo; Choi, Gyunghyun; Seo, Yoonho, An efficient genetic algorithm for the traveling salesman problem with precedence constraints, European Journal of Operational Research, Vol.140, No.3, pp. 606-617, 2002.

[3] Renaud, Jacques; Boctor, Fayez F., An efficient composite heuristic for the symmetric generalized traveling salesman problem, European Journal of Operational Research, Vol.108.No.3, pp. 571-584, 1998.

[4] Martín-Valdivia M., Ruiz-Sepúlveda A., Triguero-Ruiz F., Improving local minima of Hopfield networks with augmented Lagrange multipliers for large scale TSPs, Neural Networks, Vol.13, No.10, pp.283-285,2000.

[5] Pedro M.Talaván and Javier Yáñez, Parameter setting of the Hopfield network applied to TSP, Neural Networks, Vol.15, No.3, pp.363-373, 2002.

[6] Leyuan Shi, Sigurdur Olafsson and Ning Sun, New parallel randomized algorithms for the traveling salesman problem, Computers & Operation Research, Vol.26, No.4, pp.371-394, 1999.

[7] Glover F. and Laguna M., Tabu Search, Boston, Kluwer Academic Publishers, 1997.

[8] Glover F., Laguna M., Tabu search, in: C.R. Reeves (Ed.), Modern Heuristic Techniques for Combinatorial Problems, Blackwell Scientific Publications, Oxford, pp.70-150, 1992.

[9] Cheng-Fa Tsai, Chun-Wei Tsai, A new approach for solving large traveling salesman problem using evolutionary ant rules, in Proceeding of the 2002 International Joint Conference on Neural Networks (IJCNN'02), Vol. 2, pp.1540 – 1545, 2002.

[10] Habib Youssef, Sadiq M. Sait and Hakim Adiche, Evolutionary algorithms, simulated annealing and tabu search: a comparative study, Engineering

Applications of Artificial Intelligence, Vol.14, No.2, pp.167-181, 2001.
[11] Guangyuan Liu, Yi He, Yuhui Qiu and Juebang Yu, Research on Influence of Solving Quality Based on Different Initializing Solution Algorithm in Tabu Search, Proceedings of IEEE International Conference on Communications Circuits and Systems and West Sino Expositions, Chengdu, China, pp.1141-1145, 2002.
[12] E.G. Talbi, Z. Hafidi, J-M. Geib, A parallel adaptive tabu search approach, Parallel Computing, Vol.24, No.14, pp.2003-2019, 1998.
[13] A. Bortfeldt, H. Gehring, D. Mack, A parallel tabu search algorithm for solving the container loading problem, Parallel Computing, Vol.29, No.5, pp. 641-662, 2003.
[14] Hiroyuki Mori and Yoshihiro Ogita, A Parallel Tabu Search Based Approach to Optimal Network Reconfigurations for Service Restoration in Distribution Systems, Proceedings of the 2002 IEEE International Conference on Control Applications, Sept.18-20, , pp814-819. Glasgow, Scotland, UK, 2002.
[15] A. Augugliaro, L. Dusonchet, E.R. Sanseverino, An Evolutionary Parallel Tabu Search approach for distribution systems reinforcement planning, Advanced Engineering Informatics, Vol.16, No.3, pp.205–215, 2002,
[16] De Falco, R. Del Balio, E. Tarantino, and R. Vaccaro, Improving search by incorporating evolution principles in parallel tabu search, 1994 IEEE Conference on Evolutionary Computation, Vol.2, pp823-828, 1994.
[17] Gerhard R. TSPLIB, http://www.iwr.uni-heidelberg.de/groups/c-omopt/software/TSPLIB95/, 2001.
[18] Jeremy Kepner, Parallel Programming with MatlabMPI, http://www.ll.mit.edu/MatlabMPI.
[19] Chun-Hung Cheng, Wing-Kin Lee, and Kam-Fai Wong, A Genetic Algorithm-Based Clustering Approach for Database Partitioning, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS — PART C: APPLICATIONS AND REVIEWS, Vol. 32, No. 3, pp.215-230, 2002.
[20] M.G.A. Verhoeven, E.H.L. Aarts and P.C.J. Swinkels, A parallel 2-opt algorithm for the Traveling Salsman Problem, Future Generation Computer Systems, Vol.11. No.2, pp.175-182, 1995.

Yi HE received M.S. degree in Electrical Engineering from Southwest-China Normal University and PH.D degree in Computer Science from Southwest-China University in 2003 and 2006, respectively, and now has been an associate professor in computer science at Chongqing Normal University since 2004. His research interest includes neural network, intelligence optimization algorithms and its applications.

Guangyuan LIU received M.S. degree and PH.D degree in circuit and system from University of Electronic Science and Technology of China in 1995 and 1999, respectively, and now has been a professor in computer science at Southwest-China University since 1999. His research interest includes neural network, fuzzy system and computational intelligence.

Yuhui QIU has been a professor in computer science at Southwest-China University since 1992 and a senior member of IEEE. His research interest includes artificial intelligence, multi-agent system and machine learning.