American Journal of Applied Sciences 5 (11): 1543-1546, 2008
ISSN 1546-9239
© 2008 Science Publications

Traveling Salesman Approach for Solving Petrol Distribution Using Simulated Annealing

Zuhaimy Ismail and Wan Rohaizad Wan Ibrahim
Department of Mathematics, Faculty of Science, Universiti Teknologi Malaysia,
Skudai 81310, Johor, Malaysia

Abstract: This research presents an attempt to solve a logistic company's problem of delivering petrol to petrol station in the state of Johor. This delivery system is formulated as a travelling salesman problem (TSP). TSP involves finding an optimal route for visiting stations and returning to point of origin, where the inter-station distance is symmetric and known. This real world application is a deceptive simple combinatorial problem and our approach is to develop solutions based on the idea of local search and meta-heuristics. As a standard problem, we have chosen a solution is a deceptively simple combinatorial problem and we defined it simply as the time spends or distance travelled by salesman visiting n cities (or nodes) cyclically. In one tour the vehicle visits each station just once and finishes up where he started. As standard problems, we have chosen TSP with different stations visited once. This research presents the development of solution engine based on local search method known as Greedy Method and with the result generated as the initial solution, Simulated Annealing (SA) and Tabu Search (TS) further used to improve the search and provide the best solution. A user friendly optimization program developed using Microsoft C++ to solve the TSP and provides solutions to future TSP which may be classified into daily or advanced management and engineering problems.

Key words: Heuristic method, tabu search, simulated annealing, travelling salesman problem and greedy search

INTRODUCTION

The present society are built on infrastructure of information technology (IT) comprised of computers and systems of communication. Nowadays, societies worldwide are fed with terms such information society is referred to as the knowledge society. Achieving the level of information society as a knowledge society, further progress of IT infrastructure and software are needed to support higher level of demands (Bodin et. al, 1983). Malaysia is a developing country which is progressing towards developing an IT society. In such a society, the software development consists of the development of algorithms, which are targets of our study. In many real life problems we do not know their solutions yet and we have to develop algorithms to search for such solutions (Gaskell, T.J., 1967; Zuhaimy Ismail and Irhamah 2007). Area of algorithms is vast and in this research, we are interested in developing algorithms for solving the problems formulated as combinatorial optimization (or discrete optimization). One such combinatorial optimization problem is the distribution of petrol from the depot to various petrol stations. This type of distribution problem may be modeled as a Travelling Salesman Problems (TSP).

In TSP, the salesman (say) must make a complete tour of a given set of stations in the order to minimize the total distances travelled. Suppose that the delivery vehicle is to visit N stations and need to determine the shortest path that he must make to cover all the identified stations, passing each station only once and finishing his track at the station of origin. If there are N stations to visit, the salesman has N! numbers of order to visits. Figure 1 gives a sample of the possible combination of the journey of the salesman.

For example, the path 01234 in Fig. 1 indicates that the path goes from station 0 - station 1- station 2 - station 3 - station 4 - station 0. The aim of TSP is to determine the path that gives the shortest distance and it is a deceptively simple combinatorial problem and many real-world scheduling or sequencing problem can be formulated in a similar fashion. These problems are usually very difficult if we want to compute exact optimal solutions and we have to resort to approximate (or heuristic) algorithms to obtain good suboptimal solutions. In order to develop a satisfactory algorithm

Corresponding Author: Zuhaimy Ismail, Department of Mathematics, Faculty of Science, Universiti Teknologi Malaysia, Skudai 81310, Johor, Malaysia

for a given problem, it is necessary to exploit special mathematical structures of the problem and it requires a

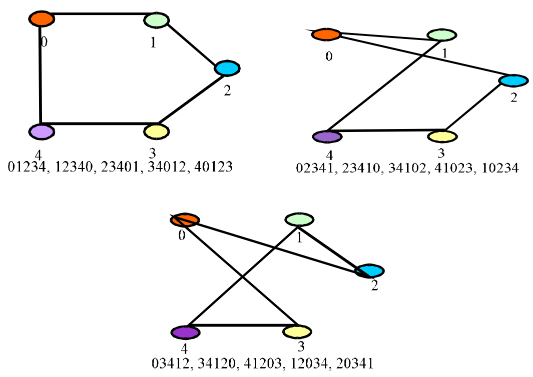

Fig. 1: Travelling salesman problem

large amount of time and man power. Furthermore, it is often the case that one algorithm specially tailored to a particular problem is no longer useful if some additional constraints are added or the objective function is slightly modified. To cope with such situation, it would be very useful if we could realize a general solver that works for all types of combinatorial optimization problems to give reasonably good suboptimal solutions. However, after considering the implication of complexity theory, we conclude that no single engine can do such job, but we need multiple search engines developed for appropriate standard problems. In this research we proposed the development of a petrol delivery system as a TSP and solved it using Simulated Annealing and Tabu Search. Our sets of solution engines appear successfully when applied to real-world industrial problems. However we have to admit that it is still far from being complete and further efforts are necessary to make them more effective.

TSP AS NP-HARDNESS PROBLEM

An important theoretical achievement in complexity theory is the concept of NP-completeness and NP-hardness. The NP is a class of problems that includes most of the combinatorial problems encountered in many applications. Some problems in NP class can be shown to be NP-complete or NP-hard. (We are not going to expand the differences in the definitions of NP-completeness and NP-hardness, but we use only terminology NP-hard hereafter for simplicity sake.) NP-hard problems are computationally intractable (not solvable in polynomial time). Therefore, it is not possible to build a general solver that works efficiently for all problems. (Gendreau, M. *et. al.* 1994). An approach to overcome this difficulty may be using an efficient approximate algorithm for an appropriate NP-hard problem. As good approximate solutions are sufficient for most of the practical purposes, this approach is quite appealing.

As described earlier that TSP is deceptively simple combinatorial problem, it is in fact an NP-hard problem or NP-completeness. This has been shown to be the case for all the different cases described earlier and certainly this is also true for a wider TSP is also NP-complete. Here we briefly recall the TSP and we introduce the notation used to describe our heuristic. Let V be the set of all stations to be visited and n = |V |, the number of stations in this set. Let also d (with m = |d|) be the distance travel between two selected stations. With each such feature, the station vector may be described as V = (v1 , ... , vm), where vr is the coordinate of each station in the set. TSP consists of finding a minimum cost of travelling such that all the station is visited once and must be visited twice. The objective function considered in this research is the ordinary Euclidean distance between two stations as $d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$ i,j=0,1,2,3,…,n where (x_I,y_i) is the coordinate for station number one and (x_I,y_i) is the coordinate for another station and *n* is the total number of the stations in the system Fig. 2. The number of possible solution for Euclidean TSP with fix starting and ending point is given by $\frac{(n-1)!}{2}$.

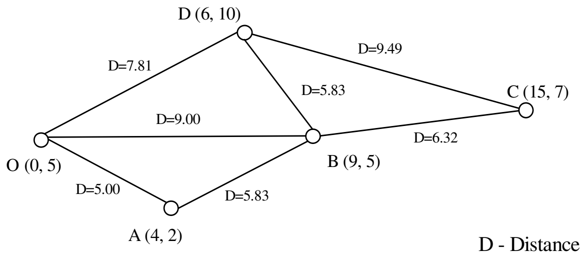

Fig. 2: TSP with Euclidean distance as a cost

LOCAL SEARCH AND METAHEURISTICS

A powerful tool to obtain good approximate solutions is local search (LS), which works as follows. Starting from an initial solution x, it repeats replacing it with a better solution in its neighborhood $N(x)$ until no better solution is found in $N(x)$, where $N(x)$ is a set of solutions obtainable from x by slight perturbations. The resulting solution x, which cannot be improved by any solution in $N(x)$, is called locally optimal (with respect to N). In designing an LS algorithm, we first fixed the search space, which is the set of solutions potentially visited during search. Specify the generation of an initial solution *x*. In many cases, it is generated randomly, but more sophisticated approaches such as greedy search heuristics are also possible. In greedy search, it reads the coordinate of each station and set the initial solution. Following this, it creates a new solution by randomly swapping two paths and compares the new solution with the previous one. If the new solution is better than the previous one, accept the new solution or else reject the solution and restore the previous solution. The search ends once the stopping criteria or maximum number of iteration reached and set the current solution as the final solution.

Metaheuristics are a framework considered for this purpose. It includes the following well-known approaches as special cases: iterated local search (ILS), tabu search (TS), simulated annealing (SA), genetic algorithm (GA) and others. All of these algorithms repeat the local search in an attempt of finding better solutions than those obtained in the previous rounds of local search (Bodin, *et al* 1983; Canen, A.G. and N.D. Pizzolato 1994). Algorithms in metaheuristics repeat the processes of generating an initial solution and its improvement by LS in the following manner. Generate an initial solution *x*; Improve *x* by applying (generalized) LS then if the stopping criterion holds, halt after outputting the best solution found so far. Otherwise, return to the beginning and generate new initial solution. To generate initial solutions, it is common that the computational history by then is taken into consideration. For example, a certain number of good solutions are maintained during computation and initial solutions are generated by combining them in some manner.

In SA, the probability is controlled by a parameter called temperature to diversify the search in the initial phase and then concentrate the search to the promising area found in the initial phase. In TS, the move in the search for a better solution is always done to the best solution in *N*(*x*) even if it is worse than x. In this case, to prevent cycling of solutions, a tabu list of solutions is prepared and the moves to tabu solutions are prohibited, where tabu list usually contains a certain number of most recently visited solutions or a set of features of such solutions. The stopping criterion in the final scan be very simple, e.g., it stops if a specified time limit of computation is over. Some description on metaheuristics can be found for example in (Zuhaimy and Irhamah, 2007). We now describe some details of the engines for TSP together with some computational results.

COMPUTATIONAL EXPERIMENTS

We have tested our heuristics on a set of data provided by our local logistic company for delivery of petrol to several stations. We took three sets of possible number of stations namely 25 stations, 50 stations and 100 stations. For analysis purposes, we only present a case with 25 stations. All experiments were performed on a 2.13 GHz Intel Core 2 Duo computer and, unless otherwise indicated, ten minutes of CPU time were allotted to the solution of each instance.

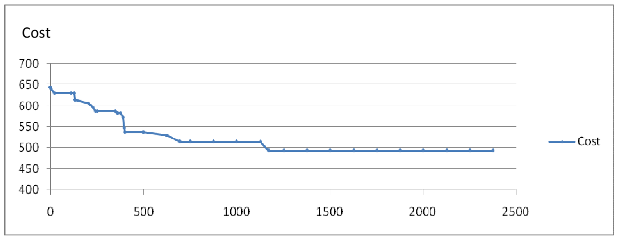

Fig. 3: Greedy Method…

The experiment begins with the presentation of the outcome of a single run to search for the simplest solution. The initial solution generated will be used for all different runs. Using the initial solution provided in Greedy Method, the Simulated Annealing has generated a better solution with the cost of 469.344. It has improved by 1.49% from the solution generated by Greedy method. Further runs using the same initial solution will give the same result. It also shows that four out of five runs does give the same solution (Fig. 4).

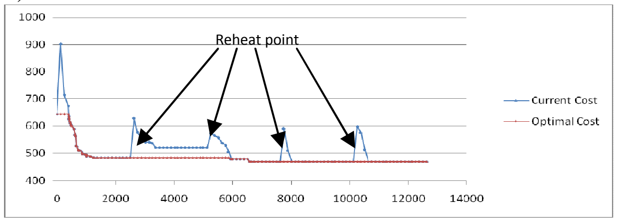

Fig 4: Reheating points

The temperature parameter in SA plays an important role in the search for an optimum solution. It is a parameter that determines the acceptability of a move. Its gradual reduction of temperature gives the best solution. It is found that after a prolong temperature reduction; it will eventually come to the same position as a greedy method where upon reaching the local optimum, no further improvement can be made. We introduced a new technique where heat is induced and the temperature is increased after a certain number of iteration. Result shows that it does give a better solution.Another heuristic method known as Tabu Search was introduced. Using the initial solution provided in Greedy Method, the Tabu Search has generated a better solution with the cost of 467.09. It has improved by 1.49% from the solution generated by Greedy method. Further runs using the same initial solution will give the same result. The solution generated using Tabu Search is the same as the solution obtained using simulated annealing.

Tabu Search is a search procedure with the properties of preventing the move from returning the previous moves. The previous moves are listed in the tabu list. In some cases, these forbidden moves are allowed using an aspiration function where it is used with the hope of getting a better solution. In our study, it is found that the search keep accepting a non-improving move. To avoid unending search, we introduce the restore procedure of the optimal solution after a certain number of iteration. This is a new heuristic that we developed that is suitable for reaching the optimum solution. The result obtained using Tabu Search is 469.344 the same as the result using Simulated Annealing. Further experiments were conducted. The experiment was conducted several times and we called it a multiple run experiment. In this experiment, we run the search and upon reaching a solution, we continue to run the system using the final solution as the initial solution for the second and third runs. Table 1 above shows that Greedy Method is incapable of improving the final solution after several runs. Tabu Search and Simulated Annealing managed to improve the optimal solution after the second runs. Among the three heuristic searches, Simulated Annealing gives the best result. The final schedule route is as follows: DEPOT - 13 - 12 - 14 - 22 - 7 - 16 - 15 - 10 - 23 - 18 - 4 - 1 - 11 - 25 - 3 - 24 - 17 - 6 - 8 - 5 - 21 - 9 - 19 - 2 - 20 – DEPOT

There is a huge improvement in term of cost and number of iterations from the greedy method. It reduces the number of iteration by 59.6% for Simulated Annealing and 32.4% for Tabu Search. This experiment demonstrates that both Tabu Search and Simulated Annealing are capable of escaping from the local optimum and arrived at the global optimum.

Several experiments carried out to test the feasibility of three methods for solving TSP. The experiment with 25 cities shows that Tabu Search and Simulated Annealing produced the best solution. This demonstrates the supremacy of SA when compared to other search methods. The optimal solution to TSP is 469.344. The optimal solution and the optimal route as in Fig. 5.

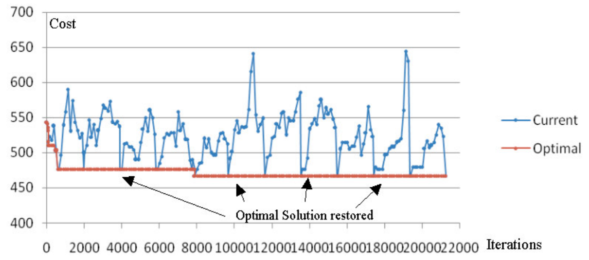

Fig. 5: Graph of current and optimum solution with number of iterations (Tabu Search)

CONCLUSION

We presented some of our developments of problem solving for solving petrol delivery system as a TSP and may be applied to other combinatorial optimization problems, all of which are based on meta-heuristics. Each of these approaches can handle various TSP problems within its scope and is reasonably efficient as evidenced by the computational results. Putting these heuristic approaches together, we found that quite a wide range of problems of practical importance can now be accessed and useful solutions can be obtained.

ACKNOWLEDGEMENT

We gratefully acknowledge the Ministry of Science, Technology and Innovation (MOSTI) of Malaysia for the IRPA RMK-8 grant funding (09-02-06-144 EAR) and UTM for their financial and administration support.

REFERENCES

1. Bodin, L., B. Golden, A. Assad and Bull, D., 1983. Routing and scheduling of vehicles and scheduling of vehicles and crews: The State of Art. Comput. Operat. Res., 10: 63-111.
2. Canen, A.G. and N.D. Pizzolato, 1994. The vehicle routeing problem. Logistics Inform. Manag., 7 (1): 11-13.
3. Gaskell, T.J., 1967. Bases for vehicle fleet scheduling. Operat. Res. Quaterly 18: 218.
4. Gendreau, M., A. Hertz and G. Laporte, 1994. A tabu search heuristic for the vehicle routing problem. Manag. Sci. 40: 1276-1290.
5. Golden, B.L., T.L. Magnanti and H.Q. Nguyen, 1977. Implementing vehicle routing algorithms. Networks 7: 113-148.
6. Zuhaimy Ismail and Irhamah, 2007. Vehicle Routing Problem in Optimizing Waste Collection. Proceeding AFSS2007, 77-84.
7. Dang Vu Tung, Anulark Pinnoi, 2000. Vehicle routing - scheduling for waste collection in Hanoi, pp: 449-468