ANT COLONY OPTIMIZATION FOR CAPACITY PROBLEMS

Tad Gonsalves and TakafumiShiozaki

Department of Information and Communication Sciences,
Faculty of Science & Technology, Sophia University, Tokyo, Japan
tad-gonsal@sophia.jp, zakishio814@gmail.com

ABSTRACT

This paper deals with the optimization of the capacity of a terminal railway station using the Ant Colony Optimization algorithm. The capacity of the terminal station is defined as the number of trains that depart from the station in unit interval of time. The railway capacity optimization problem is framed as a typical symmetrical Travelling Salesman Problem (TSP), with the TSP nodes representing the train arrival /departure events and the TSP total cost representing the total time-interval of the schedule. The application problem is then optimized using the ACO algorithm. The simulation experiments validate the formulation of the railway capacity problem as a TSP and the ACO algorithm produces optimal solutions superior to those produced by the domain experts.

KEYWORDS

Travelling Salesman Problem, Ant Colony Optimization, Capacity Problems, Meta-heuristic Optimization, Soft Computing.

1. INTRODUCTION

This study focuses on the simulation optimization of rail capacity, a prominent application problem in the transportation domain. Zhu [1] defines the railway capacity as the maximum number or pair of trains with standard load passing by a fixed equipment in unit time (usually one day) on the basis of the given type of locomotives and vehicles. It usually depends on the condition of the fixed equipment as well as the organization of the train operation. According to the European Community directives [2], the provision, maintenance and marketing of the railway track capacities should be separated from the operation of trains. This would imply a separation of the management of the railway infrastructure from the management of the railway operation. With this in mind, we frame the aim of this study as: the optimization of rail capacity by managing the train operation, given a fixed railway network and equipment infrastructure. In particular, we focus on the capacity of a terminal station, i.e., the number of trains departing from the terminal station in unit time. The problem basically boils down to constructing an optimal schedule of the passenger trains so as to maximize the terminal station capacity. However, owing to the multiple decision variables, the problem becomes a typical combinatorial optimization problem which cannot be solved using the conventional optimization algorithms.

David C. Wyld et al. (Eds) : ITCS, CST, JSE, SIP, ARIA, DMS - 2015
pp. 157–164, 2015. © CS & IT-CSCP 2015 DOI : 10.5121/csit.2015.50116

In order to solve the combinatorial optimization problem, we first cast it in the form of a Travelling Salesman Problem and use some of the soft–computing techniques to find the optimal solution. Although the TSP has applications in practical problems like Vehicle Routing, Job Sequencing, Computer Wiring, etc. [3], it is known to be NP hard. Since brute force approach is an infeasible option, heuristics approach can be fairly relied upon to solve thesetypes of problems since heuristics approach utilizes much less computing power. Some of the conventional heuristic techniques designed to solve the TSP include branch and cut [4], dynamic programming [5], regression analysis [6], exact methods [7], etc. Recently many meta-heuristic algorithms (i.e. heuristics that do not depend on the domain knowledge of the problem) are successfully employed to search for the optimal TSP solution. The Genetic Algorithm (GA) based on the Darwinian theory of natural selection and its variants are reported to be successful in finding the optimal solutions to the benchmark TSP problems in a reasonable amount of computing time [8-11]. In some studies, the Genetic Algorithm is combined with other meta-heuristic optimization algorithms to improve the optimization results [12].

However, the most successful soft computing algorithm to obtain the optimal solution of the TSP is the Ant Colony Optimization (ACO) algorithm. The development of the ACO algorithm has been inspired by the foraging behaviour of some ant species. These ants deposit pheromone on the ground in order to mark some favourable path that should be followed by other members of the colony. The ACO algorithm exploits a similar mechanism for solving optimization problems [13-19]. From the early nineties, when the first ACO algorithm was proposed, it has attracted the attention of an increasing numbers of researchers and it has been extended to many successful applications.

In this study, the railway capacity optimization problem is cast in the form of a TSP. The arrival/departure *events* in the schedule are treated as *nodes* which need to be ordered under the given scheduling constraints so as to minimize the entire schedule time. Some of the other constraints are imposed by the track-changing hardware equipment. The time between two events is considered to be the *distance* between two TSP edges and the train operation schedule is considered to be the *tour length* of the TSP. The standard ACO application to this problem yields an optimal schedule, under the given infrastructure and operational constraints.

This paper is organized as follows: Section 2 describes the TSP and ACO. Section 3 describes the formulation of the railway capacity optimization problem (RCP) as the TSP and its solution using the standard ACO algorithm. Section 4 presents the simulation optimization results and section 5 concludes the paper.

2. TSP AND ACO

In this section, we introduce the Travelling Salesman Problem and the Ant Colony Optimization algorithm. We show how the Ant Colony Optimization algorithm is designed to solve the Travelling Salesman Problem.

2.1. Travelling Salesman Problem (TSP)

The Travelling Salesman Problem (TSP) is a classic problem in computer science which may be stated as follows: Given a list of cities and their pairwise distances, the task is to find the

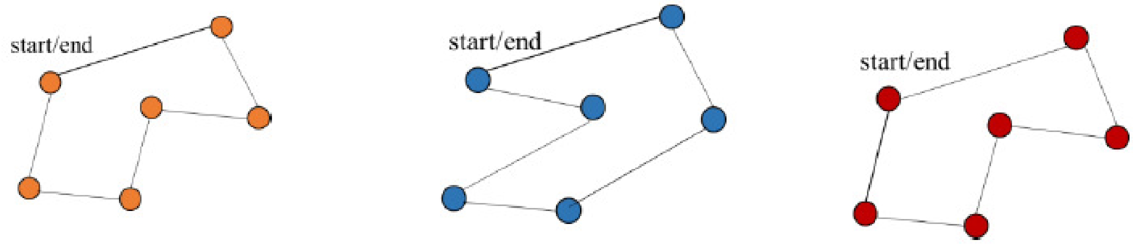


Figure 1. Three feasible routes of a 6-node TSP

shortest possible route that visits each city exactly once and then return to the original city. If n is the number of cities to be visited, the total number of possible routes covering all cities, S_n is given by:

$$S_n = (n-1)!/2 \tag{1}$$

A naive solution solves the problem in O(n!) time, simply by checking all possible routes, and selecting the shortest one. A more efficient dynamic programming approach yields a solution in $O(n^2 2^n)$ time [3]. The TSP is proved to be NP-hard and various Operation Research (OR) solution techniques have been proposed, yielding varying degrees of success [4-7]. The Ant Colony Optimization, described in the following sub-section is a novel soft computing algorithm developed to tackle combinatorial optimization problems.

2.2. Ant Colony Optimization CO

The Ant Colony Optimization (ACO) which is based on the foraging behaviour of ants was first proposed by Dorigo [13].

1 Initialize parameters and solutions
2 While the termination criterion is not met
3 Evaluate solutions
4 Update pheromone
5 Construct new solutions
6 End
7 Output the optimum solution

Figure 2. The ACO algorithm

A generic ACO algorithm is shown in Figure. 2. In step 1, the algorithm parameters are initialized and all the artificial ants (random solutions) are generated. The loop from lines 2 through 6 is repeated until the termination condition is met. The steps inside the loop consist of evaluating the solutions, updating the pheromones and constructing new solutions from the previous solutions. The two main steps inside the loop are further described below.

<u>Solution construction</u>

Ant *k* on node *i* selects node *j*, based on the probability, p_{ij}, given by:

$$p_{ij}^{k} = \begin{cases} \dfrac{[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{j\in\mathcal{N}_i^k}[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}} & \text{if } j \in \mathcal{N}_i^k, \\ 0 & \text{otherwise,} \end{cases} \tag{2}$$

where $\mathcal{N}_i^k$ denotes the set of candidate sub-solutions; τ_{ij} and η_{ij} denote, respectively, the pheromone value and the heuristic value associated with e_{ij}.

Updating the pheromone

The pheromone update operator employed for updating the pheromone value of each edge $e_{ij}$is defined as

$$\tau_{ij} = (1-\rho)\tau_{ij} + \rho\sum_{k=1}^{m}\Delta\tau_{ij}^{k} \tag{3}$$

$$\Delta\tau_{ij}^{k} = \frac{1}{L^k} \tag{4}$$

Where $L_k$denotes the quality of the solution created by ant k; $\rho \in (0,1]$denotes the evaporation rate.

3. CAPACITY PROBLEM AS TSP

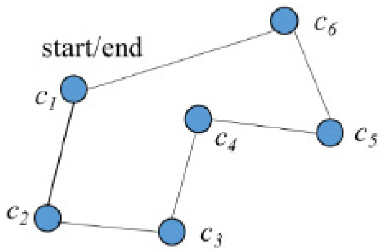
This section describes in detail the railway capacity problem to be optimized. It explains the optimization constraints, the framing of the railway capacity problem as a typical TSP and finally the solution process by using the standard ACO algorithm.

3.1. Capacity Problem

When dealing with the railway capacity problem, the railway management has to consider the different types of capacities in the railway domain. Some of the relevant capacities, for instance, are: (1) the capacity of the platform to hold passengers, (2) the capacity of the carriages to hold passengers, (3) the rail network capacity to hold the number of trains at a given time, and (4) the capacity of the railway station structure to schedule the maximum number of trains per unit time. Dealing with all these types of capacities simultaneously is a complex problem. This study is dedicated to the maximization of only the type 4 railway capacity, i.e., maximization of the number of trains that can be scheduled at a railway station per unit time. The type 4 rail capacity optimization in turn leads to optimization of the royalties and alleviation of the congestion problem during rush hours.

3.2. Capacity problem as TSP

In the generalized form of the TSP, the cities are represented as *nodes* (Figure 3a). The task is then finding the shortest route, starting from a given node and visiting each node in the network exactly once before returning to the starting node. In the Railway Capacity Optimization (RCP) problem, the arrival/departure *events* (Figure 3b) in the schedule are treated as *nodes* which need to be ordered under the given scheduling constraints so as to minimize the entire schedule time.



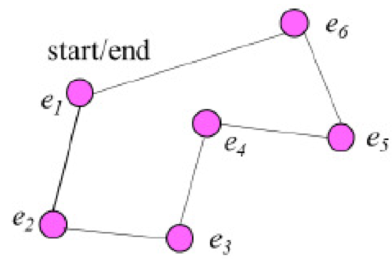


Figure 3a.The TSP nodes (cities) Figure 3b. The RCP nodes (events)

3.3. Structure constraints

In this study, we consider a railway terminal station with four railroads, each with an attached platform. The trains can arrive at the terminal and leave the terminal via any of these four railroads. There are five train services, namely, S55, S5, L7, S2 and S1 and the railroad access constraints are given in Table 1.

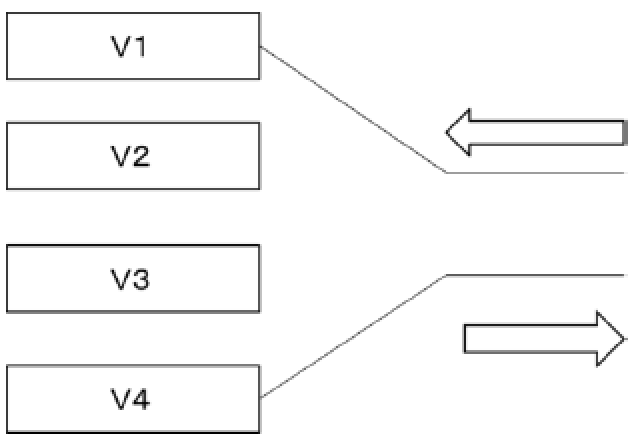


Figure 4. The platforms in the terminal station

.

Table 1. Given parameters of the train capacity problem

Trains		Offset		Time betweew arrivals		Stop Time(seconds)		platform	
Trains	N°t/h	OffsetMin	OffsetMax	Cmin	Cmax	Dmin	Dmax	arrival	departure
L7	2	0	0	0	700	110	300	v2	v2
S1	1	0	0	0	700	110	300	v4	v4
S2	1	0	0	0	700	110	300	v3	v3
S5	1	0	0	0	700	110	300	v1	v1
S55	1	0	0	0	700	110	300	v1	v1

4. OPTIMIZATION RESULTS

The aim of the simulation experiments using the ACO algorithm is to maximize the number of trains leaving the terminal station in an hour. However, to reduce the calculation load, we divide the hourly interval into 5 equal intervals, each being of 720 seconds (12 minutes) duration. The assumption here is that the train schedule is periodic. The same period can then be stretched over an hour. In Table 3, the final capacity of the terminal station is calculated by using the following formula:

$$C = \frac{3600}{T} * 6 \tag{5}$$

where, T is the total time for the entire schedule covering a period of 720 seconds.

The minimum time for the entire schedule over a period of 720 seconds is found to be 555 seconds and correspondingly the maximum capacity is 38.9 trains/hour

We conducted several experiments by varying the α, β and ρ parameters of the ACO algorithm. Some of the optimal results obtained by these tuned parameters are shown in Table 4. Another important parameter that needs an empirical tuning is the population size of the agents, N. Table 5 shows the results obtained by varying this number. As expected, the larger the population size, the better the results are, although this increases the computational overhead.

Table 2. Varying the population size of the ACO agents

N	α	β	ρ	T seconds (average)
100	1	5	0.7	558.4
200	1	5	0.7	558
300	1	5	0.7	557

5. CONCLUSIONS

The Ant Colony Optimization soft computing algorithm is apt for solving combinatorial optimization problems like the classical NP-hard Travelling Salesman Problem. Basing the search on the stigmery of the food foraging real-life ant colony, the algorithm explores the huge search space of the NP-hard problems to find the optimal solution. In this study, the authors have applied the ACO algorithm to optimize the capacity of a terminal railway station. The capacity optimization problem is cast into the form of a TSP-like problem, where the arrival and departure events of the trains are considered to be the nodes and the schedule length as the TSP total route. The standard ACO optimizes the schedule length subject to the infrastructure and operational constraints. The simulation experiments validate the formulation of the railway capacity problem as a TSP. The optimal solutions obtained by the soft-computing technique is superior to those produced by the domain experts.

REFERENCES

[1] Xiaoning Zhu, “Computer-based simulation analysis of railway carrying capacity utilization”, Proceedings of the International Conferences on Info-tech and Info-net, ICII2001, Beijing, 2001, vol.4, pp.107-112.

[2] Kuckelberg, A., “Component based system architecture for railway capacity management systems”, Proceedings of the Fourth International Conference on Quality Software, QSIC 2004., pp.189-196.

[3] Rajesh Matai, Surya Singh and Murari Lai Mittal (2010). Traveling Salesman Problem: an Overview of Applications, Formulations, and Solution Approaches, Traveling Salesman Problem, Theory and Applications, Donald Davendra (Ed.), pp. 1-24.

[4] Sarubbi, J.; Miranda, G.; Luna, H.P.; Mateus, G., “A Cut-and-Branch algorithm for the Multicommodity Traveling Salesman Problem,” IEEE International Conference on ,Service Operations and Logistics, and Informatics, IEEE/SOLI 2008, vol.2, pp.1806-1811.

[5] Jellouli, O., “Intelligent dynamic programming for the generalised travelling salesman problem,” 2001, IEEE International Conference on Systems, Man, and Cybernetics, 2001, vol.4, pp.2765-2768.

[6] Shut, V., Prozherin, I., “A solution of travelling salesman problem by a method of correlative-regression analysis,” Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, International Workshop on, 2001, pp.267-269.

[7] Woeginger, G.J. (2003), “Exact Algorithms for NP-Hard Problems: A Survey”, Combinatorial Optimization – Eureka, You Shrink! Lecture notes in computer science, vol. 2570, Springer, pp. 185–207.

[8] Pullan, W., “Adapting the genetic algorithm to the travelling salesman problem,” The 2003 Congress on Evolutionary Computation, CEC '03, 2003, vol.2, pp.1029-1035.

[9] FatihTasgetiren, M.; Suganthan, P.N.; Quan-ke Pan; Yun-Chia Liang, “A genetic algorithm for the generalized traveling salesman problem,” IEEE Congress on Evolutionary Computation, CEC 2007, 2007, pp.2382-2389.

[10] Geetha, R.R., Bouvanasilan, N., Seenuvasan, V., “A perspective view on Travelling Salesman Problem using genetic algorithm,” World Congress on Nature & Biologically Inspired Computing, NaBIC 2009, 2009, pp.356-361.

[11] Mudaliar, D.N., Modi, N.K., “Unraveling Travelling Salesman Problem by genetic algorithm using m-crossover operator,” International Conference on Signal Processing Image Processing & Pattern Recognition (ICSIPR), 2013, pp.127-130.

[12] Chen, S.M., &Chien, C.Y. (2011). Solving the traveling salesman problem based on the genetic simulated annealing ant colony system with particle swarm optimization techniques. Expert Systems with Applications, vol. 38, pp. 14439-14450.

[13] Dorigo, M. (1992). Optimization, learning and natural algorithms.Politecnico di Milano, Italy: Ph.D. Thesis.

[14] Dorigo, M., & Caro, D. G. (1999). Ant colony optimization: a new meta-heuristic (vol. 2). Proceeding of the 1999 Congress on Evolutionary Computation.
[15] Dorigo, M., Maniezzo, V., &Colorni, A. (1996). Any System: Optimization by a colony of cooperating agents. IEEE Trans Syst Man Cybernet Part B.
[16] Doringo, M., & Gambardella, L. M. (1997). A cooperative learninig approach to the traveling salesman problem.IEEE Transactions on Evolutionary Computation.
[17] Bullnheimer, B., Hartl, R., & Strauss, C. (1999). A new rank-based version of the Ant System: A computational study. Central European J Operations Res Econom.
[18] Blum, C., &Dorigo, M. (2004). The hyper-cube framework for ant colony optimization. IEEE Trans Syst Man Cybernet Part B, 34(2), 1161-1172.
[19] M. Dorigo and T. Stützle.Ant Colony Optimization.MIT Press, Cambridge, MA, 2004.

AUTHORS

Dr. Tad Gonsalvesobtained the BS degree in theoretical Physics and the MS degree in Astrophysics from the Pune University. He obtained the PhD in Systems Engineering from Sophia University, Tokyo, Japan. Currently he is an Assistant Professor in the Department of Information and Communication Sciences, Faculty of Science & Technology in the same university. His research interests include design of Expert Systems, Evolutionary Algorithms, Machine Learning and Parallel Programming.

TakafumiShiozakiis an under-graduate student is the Department of Information and Communication Sciences, Faculty of Science & Technology, Sophia University, Tokyo, Japan. His research is on the application of the Evolutionary Algorithms to diverse real-world problems.