SOLVING CAPACITY PROBLEMS AS ASYMMETRIC TRAVELLING SALESMAN PROBLEMS

Tad Gonsalves and Takafumi Shiozaki

Department of Information and Communication Sciences,
Faculty of Science & Technology, Sophia University, Tokyo, Japan

ABSTRACT

The railway capacity optimization problem deals with the maximization of the number of trains running on a given network per unit time. In this study, we frame this problem as a typical asymmetrical Travelling Salesman Problem (ATSP), with the ATSP nodes representing the train arrival /departure events and the ATSP total cost representing the total time-interval of the schedule. The application problem is then optimized using the standard Ant Colony Optimization (ACO) algorithm. The simulation experiments validate the formulation of the railway capacity problem as an ATSP and the ACO algorithm produces optimal solutions superior to those produced by the domain experts.

KEYWORDS

Capacity Problems, Travelling Salesman Problem, Ant Colony Optimization, Swarm Intelligence, Soft Computing.

1. INTRODUCTION

Rail capacity optimization is a prominent application problem in the transportation domain. Railway domain experts define the railway capacity as the maximum number or pair of trains with standard load passing by a fixed equipment in unit time on the basis of the given type of locomotives and vehicles [1]. The optimal capacity usually depends on the condition of the fixed equipment as well as the organization of the train operation. According to the European Community directives, the provision, maintenance and marketing of the railway track capacities should be separated from the operation of trains [2]. 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.

DOI : 10.5121/ijaia.2015.6204

The Travelling Salesman Problem (TSP) is one of the most studied classical combinatorial optimization techniques. We review the literature on TSP and its variants and on the various computing techniques evolved by the operations research community to find the TSP solutions in reasonable computational time. We model the rail capacity optimization problem as a form of an asymmetric TSP and use some of the soft–computing techniques to find the optimal solution. The TSP is known to be NP-hard problem. Apart from the exact methods [3], some of the conventional heuristic techniques designed to solve the TSP include branch and cut [4], dynamic programming [5], regression analysis [6], 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 [7-10].

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. When foraging for food, the 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 [11-14]. Among the better performing ACO algorithms are variants such as Ant Colony System (ACS) [15-16], MAX-MIN Ant System (MMAS) [17-19], rank-based Ant System [20] and the cunning Ant System (cAS) [21].

In some studies, the ACO algorithm is combined with the Genetic Algorithm to improve the optimization results of the TSP [22-25]. The Particle Swarm Optimization (PSO) and the Ant Colony Optimization (ACO) are two well-known paradigms of Swarm Intelligence, which is a rapidly developing branch of Computational Intelligence. PSO-ACO hybrid algorithms performing better than each the individual algorithms in the solution of the TSP benchmark problems are also reported in literature [26-28]. The artificial immune system algorithm based on the immunological model of the vertebrates [29] is also hybridized with ACO to improve the latter's performance [30].

In this study, the railway capacity optimization problem is cast in the form of an asymmetric 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 reviews the literature on TSP and its variations. Section 3 describes the TSP and the design of the ACO as its appropriate solution methodology. Section 4 describes the formulation of the railway capacity optimization problem (RCP) as the ATSP and its solution using the standard ACO algorithm. Section 5 presents the simulation optimization results and section 6 concludes the paper.

2. LITERATURE REVIEW

This section describes the different variations of the TSP and its applications to some real-life problems.

2.1 TSP Variations

The Travelling Salesman Problem (TSP) can be stated as: Given *N* number of cities in a round-tour, a salesman has to visit each city exactly once before returning to the starting city. What should be the *order* of visiting the cities so that the length of the route becomes minimal? Simply stated, the TSP deals with the determination of a permutation of *N* number of nodes so as to yield a minimum cost of traversing through them. The TSP is a well-known classical combinatorial optimization. It is proved to be NP-hard [31] and forms the corner stone as abenchmark problem in operations research and combinatorial optimization studies. Variations of the TSP, as described below, are also found in literature.

Asymmetric and symmetric TSP

The original TSP has two variants - the symmetric TSP (STSP) and the asymmetric TSP (ATSP) [32]. In the case of the STSP, the distance between two cities is the same in each opposite direction, so also the cost. In the ATSP, the distance between two cities is the different when travelled in the opposite direction; the cost may also be different; at times only one-way traffic is allowed on a path leading from one city to the next. Solving the TSP is comparatively easier than the ATSP, since the symmetry reduces the number of feasible solutions to half.

The Generalized Traveling Salesman Problem (GTSP)

The Generalized Traveling Salesman Problem (GTSP) is a variant of the standard Traveling Salesman Problem (TSP). As in the TSP, the graph considered consists of *n* nodes, and the cost between any two nodes is known. The GTSP differs from the TSP in that the node set is partitioned into m clusters. An optimal GTSP solution is a cycle of minimal cost that visits exactly one node from each cluster [33].

Travelling Purchaser Problem (TPP)

The Travelling Purchaser Problem (TPP) deals with a set of *m* markets, and a set of *p* products that the salesman must purchase. Each product is available, although with different quantities, in a subset of markets. However, the price of each of the products depends on the market where it is available. The demands for each product and the cost of traveling are the givens of the problem. In TPP, the objective is to purchase the whole demand of products, departing and returning to a depot, so as to minimize the sum of the costs of travelling and of the products purchased [34]. The applications of the TPP are mainly in the contexts of networking, routing and scheduling. Heuristics and approximate algorithms to solve the TTP are found in [35-36], while TPP applications and soft computing solutions techniques are found in [34], [37-38].

2.2 TSP Applications

Although a lot of research has been done in developing exact and approximate solution methods of the TSP, most of these studies have been restricted to testing the performance of the solution methods to the benchmark problems. In comparison, the applications of the TSP to real-life problems appear to be not too many. The typical TSP applications reported in literature are Vehicle Routing, Job Sequencing and Computer Wiring [39]. Other applications include warehouse order-picking with multiple stock locations, airport selection/routing for courier planes, and certain types of flexible manufacturing scheduling [40], clustering proteins from protein interaction network [41], mobile testing [38] and chip insertion in electronic board assembly [42].

The Railway Traveling Salesman Problem (RTSP) presented in [43-45] is an extension of the Generalized Asymmetric Traveling Salesman Problem (GATSP). In the RTSP, a salesman using the railway network wishes to visit a certain number of cities on business, starting and ending at the same city, and having as goal to minimize the overall time of the journey. The RSP uses the given railway network and train schedules and tries to minimize the overall journey time, including the sojourn time. The RCP introduced in this study is very different from the RTSP. Given a railway network, the RCP tries to determine the train schedule so as to maximize the capacity of a given network of trains. It is not a TSP. However, it can be formulated as a TSP by treating the train scheduling ***events*** as nodes of the TSP, as described in section 4.

3. 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.

3.1. 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

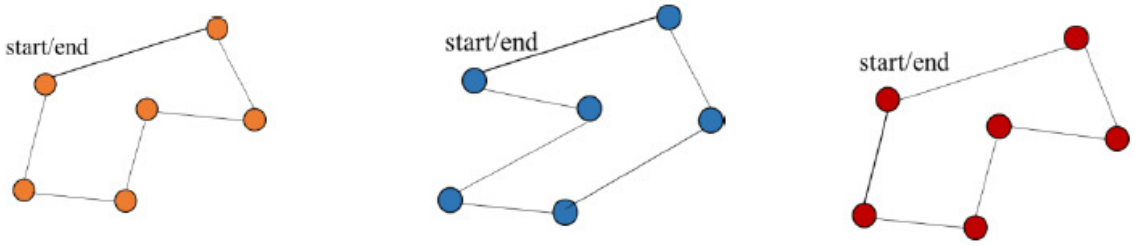

Figure 1. Three feasible routes of a 6-node TSP

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:

$$S_n = (n-1)!/2 \qquad (1)$$

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(n22n) 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 like the TSP.

3.2. Ant Colony Optimization

The Ant Colony Optimization (ACO) which is based on the foraging behaviour of ants was first proposed by Dorigo [11]. 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 steps inside the loop consist of evaluating the solutions, updating the pheromones and constructing new solutions from the previous solutions. This loop is repeated until the termination condition is met. The two main steps inside the loop are further described below.

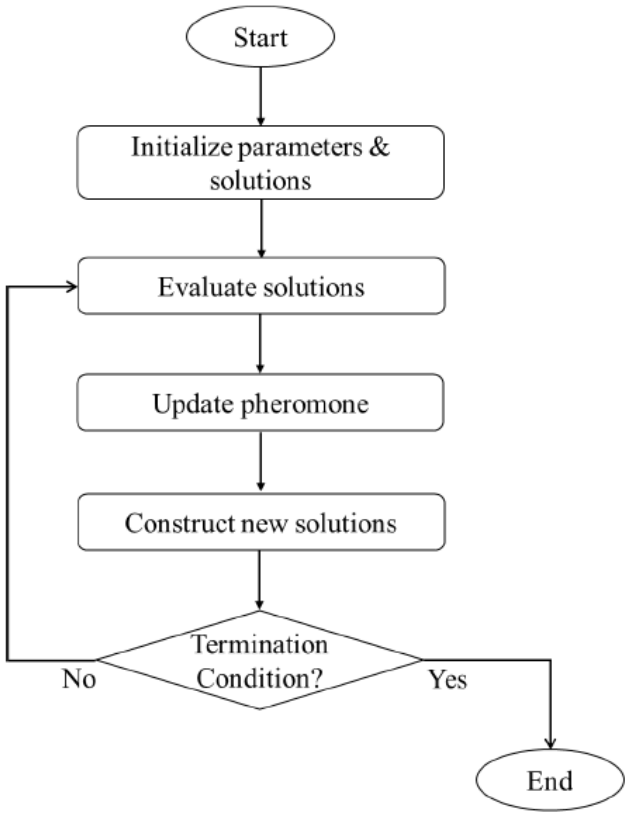

Figure 2. The ACO algorithm

Solution construction

Ant *k* on node *i* selects node *j,* based on the probability, *pij,* given by:

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

where $\mathcal{N}_i^k$ denotes the set of candidate sub-solutions; τ_{ij} and ηij 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:

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

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

4. 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 asymmetric TSP and finally the solution process by using the standard ACO algorithm.

4.1. 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.

4.2. 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. Since the time interval between any two events *ei* and *ej* is not necessarily the same as the time interval between *ej*, and *ei* the problem is clearly asymmetric.

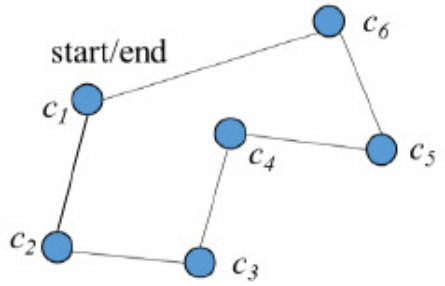

Figure 3a. The TSP nodes (cities)

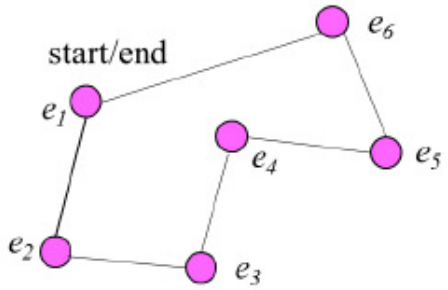

Figure 3b. The RCP nodes (events)

4.3. 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.

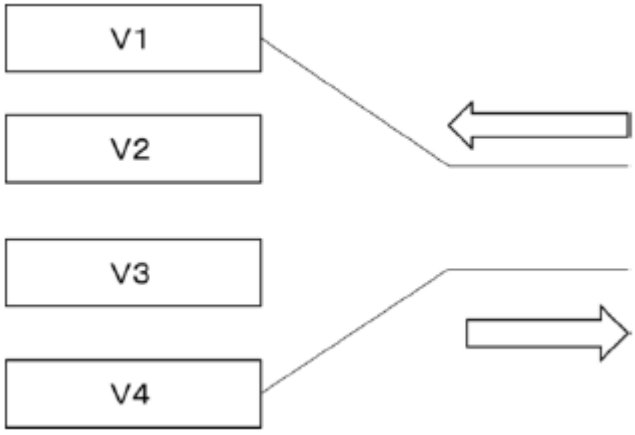

Figure 4. The platforms in the terminal station

Operational Constraints

a) For a given train service, there cannot be a departure event before the arrival event.
b) After arriving on a platform, the train must stop there for a minimum period of 110 seconds.
c) All train services must be scheduled as exactly one train per hour, except for L7 which may be scheduled at most two trains per hour.
d) At the beginning of the train schedule (at time t = 0) all trains are being stopped inside the terminal station. This implies that the first event for each train is the departure event from the terminal station.
e) There is a certain amount of "track recovery time" between two consecutive arrival-departure events.

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

5. 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 repeated over an hour's interval. In Table 2, the final capacity of the terminal station is calculated by using the following formula:

$$C = \frac{3600}{T} * 6 \qquad (5)$$

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

The optimal schedule produced by the ACO algorithm is shown in Table 2. The column to the left indicates the train service arriving at or departing from the platform shown in the adjoining column. The time column indicates the time at which the arrival or the departure event takes place. 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.

Table 2. The optimal schedule

Train	Platform	Event	Time (sec)	Capacity
S55	v1	departure	0	1
S2	v3	departure	54	2
S5	v1	arrival	107	2
L7	v2	departure	108	3
S1	v4	departure	162	4
L7	v2	arrival	214	4
S5	v1	departure	217	5
S2	v3	arrival	358	5
L7	v2	departure	358	6
S55	v1	arrival	428	6
L7	v2	arrival	493	6
S1	v4	arrival	555	6
Capacity = 38.9 trains/hour				

Table 3. Varying the algorithm parameters

α	β	ρ	1	2	3	4	5	6	7	8	9	10	Average
1	1	0.9	556	557	564	556	557	557	556	563	567	557	559
1	1	0.7	563	556	555	556	556	566	557	557	555	558	557.9
1	1	0.5	556	557	558	563	557	555	563	557	556	563	558.5
1	1	0.3	557	556	563	555	557	557	563	560	577	558	560.3
1	5	0.5	555	556	563	556	556	563	555	556	557	556	557.3
1	3	0.5	557	556	562	557	557	556	556	557	557	557	557.2
5	1	0.5	557	556	555	557	556	557	557	556	555	564	557
3	1	0.5	557	556	556	555	555	563	558	555	555	556	556.6
1	5	0.7	563	558	558	556	557	557	555	556	563	557	558
1	3	0.7	557	556	555	556	557	563	557	563	563	556	558.3
5	1	0.7	557	557	557	557	556	555	557	558	557	557	556.8
3	1	0.7	557	556	557	556	556	556	563	555	556	557	556.9
1	5	0.9	557	558	557	563	555	556	556	557	564	557	558
1	3	0.9	556	557	560	555	557	556	556	556	557	556	556.6
5	1	0.9	563	556	556	557	563	557	556	558	556	557	557.9
3	1	0.9	556	563	557	555	556	556	555	558	557	557	557

We conducted several experiments by varying the □, □ and □ parameters of the ACO algorithm. Some of the optimal results obtained by these tuned parameters are shown in Table 3. Another important parameter that needs an empirical tuning is the population size of the ant colony, N. Table 4 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 4. 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

6. 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 an asymmetric 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 an asymmetric TSP. The optimal solutions obtained by the soft-computing technique are 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] 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.

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

[8] Fatih Tasgetiren, 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.
[9] 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.
[10] 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.
[11] Dorigo, M. (1992). Optimization, learning and natural algorithms. Politecnico di Milano, Italy: Ph.D. Thesis.
[12] Dorigo, M., & Caro, D. G. (1999). Ant colony optimization: a new meta-heuristic (vol. 2). Proceeding of the 1999 Congress on Evolutionary Computation.
[13] Dorigo, M., and Stutzle, T., Ant Colony Optimization. Cambridge, MA: MIT Press, 2004.
[14] Blum, C., & Dorigo, M. (2004). The hyper-cube framework for ant colony optimization. IEEE Trans. Syst. Man Cybernet. Part B, 34(2), 1161-1172.
[15] Dorigo, M., and Gambardella, L.M., Ant Colony System: A cooperative learning approach to the traveling salesman problem, IEEE Transactions on Evolutionary Computation, 1997, 1(1): 53–66,.
[16] Dorigo, M., Maniezzo, V., & Colorni, A. (1996). Any System: Optimization by a colony of cooperating agents. IEEE Trans. Syst. Man Cybernet Part B.
[17] Maur, M., López-Ibáñez, M., Stützle, T., Pre-scheduled and adaptive parameter variation in MAX-MIN Ant System, IEEE Congress on Evolutionary Computation (CEC), 2010, pp.1-8.
[18] Imani, M., Pakizeh, E., Pedram, M.M., Arabnia, H.R., Improving MAX-MIN ant system performance with the aid of ART2-based Twin Removal method, 9th IEEE International Conference on Cognitive Informatics (ICCI), 2010, pp.186-193.
[19] Stutzle, T., and Hoos, H.H., MAX-MIN Ant System, Future Generation Computer Systems, 2000, 16(8): 889-914.
[20] 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.
[21] Tsutsui, S., cAS: Ant colony optimization with cunning ants, Proc. of the 9th Int. Conf. on Parallel Problem Solving from Nature (PPSN IX), 2006, pp. 162-171.
[22] Li, K., Kang, L., Zhang, W., Li, B., Comparative Analysis of Genetic Algorithm and Ant Colony Algorithm on Solving Traveling Salesman Problem, IEEE International Workshop on Semantic Computing and Systems, WSCS '08, 2008, pp.72,75.
[23] Takahashi, R., A Hybrid Method of Genetic Algorithms and Ant Colony Optimization to Solve the Traveling Salesman Problem, International Conference on Machine Learning and Applications, ICMLA '09, 2009, pp.81-88.
[24] Kang, L., Cao, W., An Improved Genetic & Ant Colony Optimization Algorithm for Travelling Salesman Problem, International Symposium on Information Science and Engineering (ISISE), 2010, pp.498-502.
[25] Wang, C., Guo, X., A hybrid algorithm based on genetic algorithm and ant colony optimization for Traveling Salesman Problems, 2nd International Conference on Information Science and Engineering (ICISE), 2010, pp.4257-4260.
[26] Gomez-Cabrero, D., Armero, C., Nalin Ranasinghe, D., The Travelling Salesman's Problem: A self-adapting PSO-ACS algorithm, International Conference on Industrial and Information Systems, ICIIS 2007,2007, pp.479-484.
[27] Elloumi, W., Baklouti, N., Abraham, A., Alimi, A.M., Hybridization of Fuzzy PSO and Fuzzy ACO applied to TSP, Hybrid Intelligent Systems (HIS), 2013.
[28] Rokbani, N., Abraham, A., Alimil, A.M., Fuzzy Ant Supervised by PSO and simplified ant supervised PSO applied to TSP, 13th International Conference on Hybrid Intelligent Systems (HIS), 2013, pp.251-255.

[29] Nunes de Castro, L., Von Zuben, F. J., The Clonal Selection Algorithm with Engineering Applications, In Workshop Proceedings of GECCO, pp. 36-37, Workshop on Artificial Immune Systems and Their Applications, Las Vegas, USA, July 2000.
[30] Liu Y., Liu S., A Hybrid Model for Solving TSP Based on Artificial Immune and Ant Colony, International Conference on Information Engineering and Computer Science, 2009. ICIECS 2009, pp.1-5.
[31] Gregory, G., Anders, Y., Zverovich, A., Traveling salesman should not be greedy: domination analysis of greedy-type heuristics for the TSP, Discrete Applied Mathematics, 2002, 117(1-3): 81-86.
[32] Charikar, M., Goemans, M.X., Karloff, H., On the integrality ratio for asymmetric TSP, Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science, 2004, pp.101-107.
[33] Silberholz, J., and Golden, B., The Generalized Traveling Salesman Problem: A New Genetic Algorithm Approach, in, Extending the Horizons: Advances in Computing, Optimization, and Decision Technologies Operations Research/Computer Science Interfaces Series Volume 37, Springer, 2007, pp. 165-181.
[34]Almeida, C.P., Gonçalves, R.A., Delgado, M.R., Goldbarg, E.F., Goldbarg, M.C., A Transgenetic Algorithm for the bi-objective traveling purchaser problem, IEEE Congress on Evolutionary Computation (CEC), 2010, pp.1-8.
[35] de Assumpcao Drummond, L.M., Vianna, L.S., da Silva, M.B., Ochi, L.S., Distributed parallel metaheuristics based on GRASP and VNS for solving the traveling purchaser problem, Proceedings of the Ninth International Conference on Parallel and Distributed Systems, 2002., pp.257-263.
[36] Riera-Ledesma, J., Salazar-González, J.J., A heuristic approach for the Travelling Purchaser Problem, European Journal of Operational Research, 2005, 162(19):142-152. Logistics: From Theory to Application.
[37] Ravi, R., and F. Sibel, S., Approximation algorithms for the traveling purchaser problem and its variants in network design. Algorithms-ESA'99. Springer: Berlin Heidelberg, 1999. 29-40.
[38] Miyasawa, Y., Ueno, M., Mobile testing for authentic assessment in the field: Evaluation from actual performances, IEEE Humanitarian Technology Conference (R10-HTC), 2013 IEEE Region 10, 2013, pp.232-237.
[39] Matai, R., Singh, S., and Lai Mittal, M., (2010). Traveling Salesman Problem: an Overview of Applications, Formulations, and Solution Approaches, Traveling Salesman Problem, Theory and Applications, Donald Davendra (Ed.), pp. 1-24.
[40] Noon, C.E., and Bean, J.C., An Efficient transformation of the Generalized Travelling Salesman Problem, Technical Report 91-26, Ann Arbor, 1991.
[41] Sallim, J., Abdullah, R., Khader, A.T., ACOPIN: An ACO Algorithm with TSP Approach for Clustering Proteins from Protein Interaction Network, European Symposium on Computer Modeling and Simulation, EMS '08, 2008, pp.203-208.
[42] Chan, D. and Mercier, D., IC insertion: an application of the travelling salesman problem, International Journal of Production Research, 1989, 27(10): 1837-1841.
[43] Hadjicharalambous, G., Pop, P., Pyrga, E., Tsaggouris, G., and Zaroliagis, C., The railway traveling salesman problem. Algorithmic Methods for Railway Optimization, 4359:264-275, 2007.
[44] Pop, P.C., Pintea, C. M., and Sitar, C. P, An ant based heuristic for the railway traveling salesman problem. EvoWorkshops, Springer, 2007, 4448: 702-711.
[45] Bin, H., Raidl, G.R., Solving the Railway Traveling Salesman Problem via a Transformation into the Classical Traveling Salesman Problem, Eighth International Conference on Hybrid Intelligent Systems, HIS '08., 2008, pp.73-77.

AUTHORS

Dr. Tad Gonsalves obtained 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.

Takafumi Shiozaki is 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 problem