

A new idea for train scheduling using ant colony optimization

K. Ghoseiri

*School of Railway Engineering,
Iran University of Science and Technology, Iran*

Abstract

This paper develops a new algorithm to the train scheduling problem using Ant Colony System (ACS) meta-heuristic. At first, a mathematical model for a kind of train scheduling problem is developed and then the algorithm based on the meta-heuristic is presented to solve the problem. The problem is considered as a traveling salesman problem (TSP) wherein cities represent the trains. ACS determines the sequence of trains dispatched on the graph of the TSP. Based on the sequence obtained and removing for collisions incurred, train scheduling is determined. Numerical examples in small and medium sizes are solved by using the algorithm. The solutions are compared to the exact optimum solutions to check for quality and accuracy. Comparison of the solutions shows that the algorithm results in good enough quality and time savings. A case study is presented to illustrate the solution.

Keywords: meta-heuristic, ant colony optimization, Ant Colony System, train scheduling problem, traveling salesman problem.

1 Introduction

Rail transport planning is a very complex task which is carried out based on the mutual reaction among a large number of impressed components. As it was mentioned in Ghoseiri et al [5] and Lindner [10], in respect of complexity of rail transport planning, this process is divided into several steps. These procedures include the demand analysis, line planning, train scheduling, rolling stock planning and crew management [5]. These problems are NP-hard problems and solving them with exact algorithms is very difficult and time consuming.



Although metaheuristic algorithms produce the solutions close to optimum, there is a little research in the rail transport planning problems. As an example, Huntley et al [8] developed a simulated annealing approach to train scheduling for CSX Transportation. Van Wezel et al [15] applied Genetic Algorithm to improve the timetable. Martinelli and Teng [11] used the Neural Network to routing in a railway. Nachtigall and Voget [12] applied the Genetic Algorithm to solve train scheduling problems. Gorman [6] used a combination of Genetic algorithm and Tabu Search for addressing the weekly routing and scheduling problem. Pacciarelli and Pranzo [14] used the Tabu Search to solve train scheduling problem. Kwan and Mistry [9] used a Co-evolutionary Algorithm to create a train timetable. Engelhardt-Funke and Kolonko [3] used an advanced evolutionary algorithm to solve train scheduling problems.

Fischetti et al [4], Gutin and Punnen [7] and Noon and Bean [13] showed that the train scheduling problem can be transformed to a travel salesman problem easily. Dorigo and Gambardella [2] showed that ACS algorithm has been more successful than the other metaheuristics in solving TSP. Therefore, considering the capability of transforming train scheduling problem to a TSP problem, good responses can be expected from solving it by ACS algorithm. In solving the train scheduling problems, ACO algorithms are not used so far.

In this article, by considering the good quality of ACS algorithm in solving TSP problem and also transforming capability of train scheduling problem to TSP it is decided to solve the train scheduling problem by this algorithm.

2 ACS compound model and train scheduling problem

Train scheduling is a combinatorial optimization problem. In this problem the aim is to determine the arrival and departure times from stations which train passes on them. This problem is known to be NP- hard type. Because of dimension and natural complexity in mathematical models, traditional optimization techniques are not useful for solving the problem and the accurate methods are just usable with examples in small sizes and for solving the problem with real dimension, the heuristic methods should be used. In this article, for solving the train scheduling problem ACS algorithm is chosen.

2.1 Ant colony system (ACS)

ACS was suggested as a new heuristic method to solve optimization problem by Dorigo and Gambardella [1]. This is the reformed form of AS algorithm and functions as follow:

Each ant generates a complete solution by choosing the nodes according to a probabilistic state transition rule.

$$s = \begin{cases} \arg [\text{Max}_{j \in N_i^k} \{[\tau_{ij}] [\eta_{ij}]^\beta\}] & \text{if } q \leq q_0 \\ S & \text{if } q > q_0 \end{cases} \quad (1)$$



$$p_{ij}^k = \frac{[\tau_{ij}][\eta_{ij}]^\beta}{\sum_{l \in N_i^k} [\tau_{ij}][\eta_{ij}]^\beta} \quad (2)$$

Where q is a random number uniformly distributed in $[0 \dots 1]$, q_0 is a parameter between 0 and 1 and S is a random variable selected according to the probability distribution given in Eq. (2), τ_{ij} is the amount of pheromone in edge ij ,

$\eta_{ij} = \frac{1}{\delta_{ij}}$ that δ_{ij} is the cost of edge ij , β determine the relative importance

of η versus τ and N_i^k is the remaining nodes set of ant k based on moving from node i to build an feasible solution.

The state transition rule given in Eq. (1) and Eq. (2) is called pseudo-random-proportional rule. In ACS, only globally best ant which has built the best solution, deposits pheromone in the graph. When all the ants built their solutions, global updating rule of pheromone is applied. This rule is as follow:

$$\tau_{ij} \leftarrow (1 - \rho)\tau_{ij} + \rho\Delta\tau_{ij} \quad (3)$$

where $0 < \rho < 1$ is pheromone decay parameter and $\Delta\tau_{ij}$ equals to:

$$\Delta\tau_{ij} = \begin{cases} \frac{1}{\cos t_{gb}} & \text{if } (i,j) \in \psi^{gb} \\ 0 & \text{if } (i,j) \notin \psi^{gb} \end{cases} \quad (4)$$

ψ^{gb} is the best solution which was built and $\cos t_{gb}$ is the cost of the best solution. While building a solution by using local updating rule of pheromone in this method, the pheromone level is reduced by each ant in visiting edges.

$$\tau_{ij} \leftarrow (1 - \xi)\tau_{ij} + \xi\tau_0 \quad (5)$$

τ_0 is a small fixed value and $0 < \xi < 1$ is the local evaporation coefficient of pheromone.

2.2 The proposed mathematical model of train scheduling

This model can be used for a single track and double track without any branches. In this model it is supposed that the trains are just dispatched from the first and last station. After preparation, the trains in the beginning or end stations should be dispatched immediately. In the case that the prepared trains to dispatch are stopped in the stations with unpermitted time stop, we undergo some cost. In this model, the speed and trip times in each track section for each train are supposed as a fixed value. In this model the train can travel in two directions, but they are not permitted to overtake.

2.2.1 Notation

R: the group of trains that should be dispatched from right station to left.

L: the group of trains that should be dispatched from left station to right.



T: the group of total trains. ($i, j \in R$ or L or T and $T = R \cup L$)

B: set of track sections. ($k \in B$)

Track section k is a section of track that connects two stations k and $k+1$.

Track sections and stations are indexed in numerical order from left to right.

S: set of stations. ($k \in S$)

D: the set of permitted stop time in the station. ($d_{ij} \in D$)

AD: the set of arrival and departure times from station. ($Xa(i, k), Xd(i, k) \in AD$)

2.2.2 Parameters

Trip time: the time that train i needs to pass track section k . (t_{ik})

Dwell time: this time indicates the permitted dwell time of train i in station k . (d_{ij})

Headway: minimum time interval between trains i and j to arrive/depart in/from track section k . (h_{ijk})

Train importance weight: (W_i)

2.2.3 Decision making variables

Binary variables:

$$a_{ij} = \begin{cases} 1 & \text{if train } j \in R \text{ enter the track section after train } i \in R \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

$$b_{ij} = \begin{cases} 1 & \text{if train } j \in L \text{ enter the track section after train } i \in L \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

$$c_{ijk} = \begin{cases} 1 & \text{if train } j \in L \text{ enter the track section } k \text{ after train } i \in R \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

Continuous variables:

$Xa(i, k)$: the arrival time of train i to station k .

$Xd(i, k)$: the departure time of train i from station k .

2.2.4 Objective function

Objective function in this model is to minimize the total train delays in the stations. The delay equals time difference of train stop time in a station with the permitted dwell time in the station.

$$\text{Min } \mathbf{z} = \sum_{i \in T} \sum_{k \in S} W_i (Xd(i, k) - Xa(i, k) - d_{ik}) \quad (9)$$

2.2.5 Constraints

Trip times constraints of dispatched trains from right station:

$$Xa(i, k) - Xd(i, k-1) = t_{ik} ; i \in R, k \in S \quad (10)$$

Trip times constraints of dispatched trains from left station:

$$Xa(i, k-1) - Xd(i, k) = t_{ik} ; i \in L, k \in S \quad (11)$$



Stop times constraints of dispatched trains from left and right stations:

$$Xd(i, k) - Xa(i, k) \geq d_{ik} ; i \in T, k \in S \quad (12)$$

Sequence constraints of dispatched trains from right station:

$$\begin{aligned} Xd(j, k-1) - Xa(i, k) &\geq h_{ijk} - M(1 - a_{ij}) \\ Xd(i, k-1) - Xa(j, k) &\geq h_{ijk} - Ma_{ij} \end{aligned} \quad (13)$$

$i, j \in R, k \in S$

Sequence constraints of dispatched trains from left station:

$$\begin{aligned} Xd(j, k) - Xa(i, k-1) &\geq h_{ijk} - M(1 - b_{ij}) \\ Xd(i, k) - Xa(j, k-1) &\geq h_{ijk} - Mb_{ij} \end{aligned} \quad (14)$$

$i, j \in L, k \in S$

Safety constraints that ensure no collision occur between two trains of opposite directions:

$$\begin{aligned} Xd(i, k) - Xa(j, k) &\geq h_{ijk} - Mc_{ijk} \\ Xd(j, k+1) - Xa(i, k+1) &\geq h_{ijk} - M(1 - c_{ijk}) \end{aligned} \quad (15)$$

$i \in R, j \in L, k \in S$

2.3 The solution method of the proposed model using ACS

In the proposed algorithm, it is supposed that the trains play the role of cities (nodes) in TSP. The dispatched trains from left to right and also dispatched trains from right to left form two independent sub-networks in TSP. According to the definition, selected path of each ant in the trains' network indicates the sequence of train dispatching.

For instance, in figure 1 which includes 7 trains (3 dispatching trains from right to left and 4 dispatching trains from left to right), if an ant chooses the path of Start node, train 1, train 3 and train 2 it mean the dispatching sequence of trains 1,3,2.

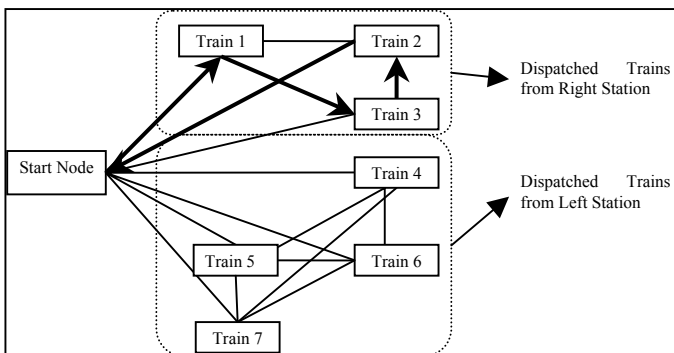


Figure 1: Problem's graph for the example of seven trains.

In this algorithm, a colony induces $2 \times n$ ants. Each group includes 2 ants that one of them is allocated to build the sequence in set of dispatched trains from right to left station and another one is allocated to build sequence in dispatched trains from left to right station.

At first both ants are placed in figurative node of zero (the start node). Then one of the ants is chosen from the first group randomly. The first chosen ant chooses a train in its train group by using pseudo-random-proportional rule. The arrival and departure times of this train from start station to the final station is calculated. Then another ant chooses its train. The arrival departure of this train from each station is identified in regard to collision with the opposite train. If the chosen train is the dispatched train from the left station, in the case that the obtained times are true in equation (16), the collision occurs:

$$\begin{aligned} Xa(i, k) + \text{headway} &> Xd(j, k) \\ &\& \\ Xd(i, K - 1) - \text{headway} &< Xd(j, k - 1) \end{aligned} \quad (16)$$

In this equation j is the selected train, i indicates chosen trains from the group of dispatched trains from right station and k is a track section in which the collision occurred. In this case for collision resolution between two trains, the departure time of chosen train from the related station is changed as follows:

$$Xd(j, k) = Xa(i, k) + \text{headway} \quad (17)$$

The arrival and departure times of this train to its last station is calculated based on this time.

If the chosen train is the dispatched train from right station, in this case the obtained times are true in equation (18), therefore, the collision is occurred.

$$\begin{aligned} Xa(i, k - 1) + \text{headway} &> Xd(j, k - 1) \\ &\& \\ Xd(i, k) - \text{headway} &< Xd(j, k) \end{aligned} \quad (18)$$

In this equation, j is the chosen train, i indicates the chosen trains from the group of dispatched trains from left station and k is a track section in which the collision is occurred. In this case for resolution of collision between two trains, departure time of above chosen train from related station changes as follows:

$$Xd(j, k - 1) = Xa(i, k - 1) + \text{headway} \quad (19)$$

The chosen trains are omitted from the set of trains. Then randomly an ant is selected again. This ant chooses a train from its group. The arrival and departure times of this train is identified with its chosen sequence in its group. When the arrival and departure time from a section was identified, the collision condition of chosen train with dispatched chosen train in opposite direction is checked. In the case of collision, it is removed. This operation continues in the same way so that all the arrival and departure times from all stations are identified and there are not any collisions in the sections. Then the next train is chosen by other ants. This procedure continues until ants choose the all trains of their own group.

3 The result analysis of the model

To analysis the solution results obtained from ACS-TS are compared with those of exact optimization method of the train scheduling model. For this purpose, computations are carried out for 28 problems. Figure 2 shows the run times of solving the problems using LINGO software in comparison with the proposed ACS algorithm.

It was considerable that in 28 solved problems the overall delay amount in dispatching trains from both methods was almost equal. Although proposed ACS time solving method has shown considerable time saving in comparison to the exact solving method.

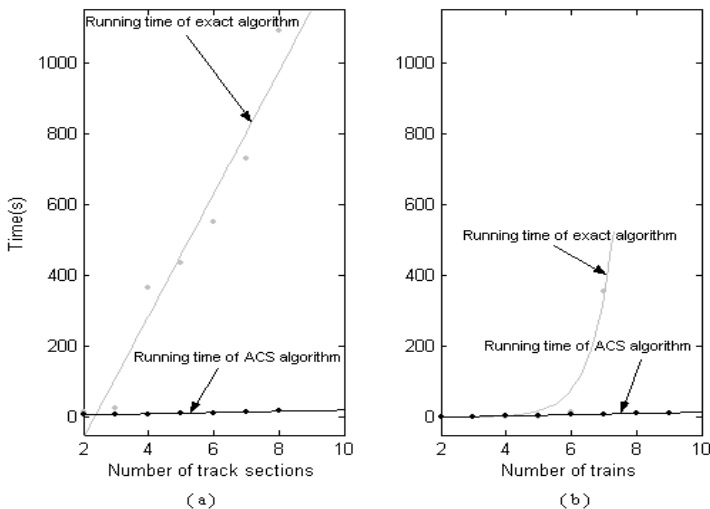


Figure 2: Run times of solving the train scheduling problems.

4 The case study

In this section to clarify the proposed algorithm much more, a problem with 30 trains and 4 sections is solved.

At first, according to Dorigo and Gambardella [1] the primary values for parameters are set to the following values:

$\rho = 0.1$, $\xi = 0.1$, Pheromone primary amount in edges = 0.000005, and $q_0 = 0.9$.

Also according to definition of problem, the number of ants in this colony is considered as equal to 30 and the fix primary amount $\tau_0 = 0.012$ that is obtained by $\tau_0 = \frac{1}{(n.L_{nn})}$ where n is the number of trains and L_{nn} is the solution cost



produced by a heuristic method (For further study refer to Dorigo and Gambardella [1]. Furthermore by considering this fact that in proposed algorithm, the length (cost) of arcs doesn't have a meaning therefore by supposing $\beta = 0$, the length effect of edges is omitted in ACS.

Then the parameters best values were experimentally determined. For this purpose the parameter values changed and then algorithm was run ten times. After this step the best value was chosen and then the problem was solved with these parameters.

$$q_0 = 0.9, \rho = 0.35, \xi = 0.2, \tau_0 = 0.$$

After determining parameters, the proposed algorithm for a problem with 30 trains was run. At the end, the amount of delay in this state was equal to 2492.1. Figure 3 is the indicator of convergence in improving the solutions in each cycle of running the algorithm.

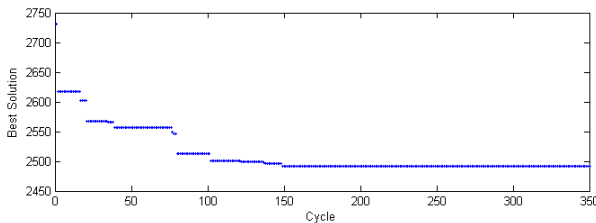


Figure 3: Convergence indicator in improving the solutions.

5 Conclusion

In this article, an algorithm was introduced for train scheduling problem by using ant colony optimization. At first a mathematical model was developed for a type of train scheduling problem and then for solving this model an algorithm was suggested based on ACS algorithm. According to the algorithm, the model transferred into a constructive graph in which each train was considered as a city in TSP graph. ACS algorithm identified the sequence of trains traveling in each direction on this graph and then based on this sequence the collision of the trains is removed and at the end the time-distance graph of trains movement was identified (either arrival or departure times). To check for quality of the solutions, numerical examples in small and average sizes were solved. The obtained results were compared with their exact optimum solutions. By comparing the results of exact solving of problems and solving them by the proposed algorithm, the time saving and good solution quality were concluded. At the end, a case study was presented for describing the computation manner.

References

- [1] Dorigo, M., & Gambardella, L.M., Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. *IEEE Transactions on Evolutionary Computation*, 1(1), pp. 53–66, 1997.



- [2] Dorigo, M., & Gambardella, L.M., Ant colonies for the traveling salesman problem. *BioSystems*, 43, pp. 73–81, 1997.
- [3] Engelhardt-Funke, O., & Kolonko, M., Analysing stability and investments in railway networks using advanced evolutionary algorithms. *International Transactions in Operational Research*, 11, pp. 381–394, 2004.
- [4] Fischetti, M., Salazar-Gonzales, J., & Toth, P., The Generalized Traveling Salesman Problem and Orienteering Problem. *Traveling Salesman Problem and its Variations*, ed. G. Gutin and A. P. Punnen, Kluwer Academic Publishers, pp. 609-663, 2002.
- [5] Ghoseiri, K., Szidarovszky, F., & Asgharpour, M.J., A Multi-Objective Train Scheduling: Model and Solution. *Transportation Research part B*, 38, pp. 927-952, 2004.
- [6] Gorman, M.F., An Application of Genetic and Tabu Searches to the Freight Railroad Operating Plan Problem. *Annals of Operations Research*, 78, pp. 51-69, 1998.
- [7] Gutin, G., & Punnen A. P., *The Traveling Salesman Problem and its Variations*, Kluwer Academic Publishers, 2003.
- [8] Huntley, C.L., Brown, D.E., Sappington, D.E., & Markowicz, B.P., Freight Routing and Scheduling at CSX. *Transportation Interfaces*, 25(3), pp. 58-71, 1995.
- [9] Kwan, R.S.K., & Mistry, P., *A Co-evolutionary Algorithm for Train Timetabling*, Research Report Series No. 2003.13, University of Leeds, School of Computing, 2003.
- [10] Lindner, T., *Train Schedule Optimization in Public Rail Transport*, Ph.D. Thesis, Technische Universität Braunschweig, 2000.
- [11] Martinelli, R.D., & Teng, H., Optimization of Railway Operations Using Neural Networks. *Transportation Research part C*, 4, pp. 33-49, 1996.
- [12] Nachtigall, K., & Voget, S., A Genetic Algorithm Approach to Periodic Railway. *Synchronization Computers and Operations Research*, 23, pp. 453-463, 1996.
- [13] Noon, C., & Bean, J., A Lagrangian Based Approach for The Asymmetric Generalized Traveling Salesman Problem. *Operations Research*, 39, pp. 623-632, 1991.
- [14] Pacciarelli, D., & Pranzo, M., A Tabu Search algorithm for the railway scheduling problem. *Proc. of the 4th Metaheuristic International Conference*, Porto (Portugal), pp. 16-20, 2001.
- [15] Van Wezel, M.C., Kok, J.N., Van den Berg, J., & Van Kampen, W., Genetic improvement of railway timetables. *Computer Science*, 866, pp. 566-574, 1994.

