

The ant colony algorithm supported optimum search in the electronic freight and warehouse exchanges

Gábor Kovács

Received 2010-10-27

Abstract

The basic function of electronic freight and warehouse exchanges is to establish connection between free freight and storage capacities and tasks [6]. In the database of such online fairs there is high number of freight and storage capacity offers and tasks, which provides good optimization opportunity for those with free capacity. Based on all these this essay describes two optimum search problems emerging in electronic freight and warehouse exchanges, then presents, in details, the ant colony algorithms for their solution, which then compares with other known solution possibilities.

Keywords

electronic freight and warehouse exchange · optimization · ant colony algorithm

Acknowledgement

This work is connected to the scientific program of the "Development of quality-oriented and harmonized R+D+I strategy and functional model at BME" project. This project is supported by the New Széchenyi Plan (Project ID: TÁMOP-4.2.1/B-09/1/KMR-2010-0002).

Gábor Kovács

Department of Transport Technology, BME, 1111 Budapest, Bertalan Lajos u 2., Hungary
e-mail: kovacs@kku.bme.hu

1 Optimum search problems in the electronic freight and warehouse exchanges

In the freight exchange the optimum search task may be formulated on the basis of the following objective function: those having free freight capacity wish to establish routes providing optimal profit from the freight tasks appearing in the freight exchange. Many freight tasks may be included into the route, but a new freight task may be commenced only after the completion of the previous one. At the calculation of profit the costs of getting to the departure point of the freight task (k_i^{getting}), the costs of actual transport of goods ($k_i^{\text{transport}}$), the costs of the vehicle's return way to the depot (k_l^{return}) and the remuneration for the completion of the freight task shall be considered (FD). The objective function is to reach the maximum profit (2) (l: the number of completed freight tasks):

$$K_l = \sum_{i=1}^l k_i^{\text{getting}} + \sum_{i=1}^l k_i^{\text{transport}} + k_l^{\text{return}} \quad (1)$$

$$H_l = \sum_{i=1}^l \{FD_i\} - K_l \rightarrow \text{MAX!} \quad (2)$$

In the warehouse exchange those having free storage capacity wish to choose storage tasks from the available storage tasks by setting the goal of ideal exploitation of capacity.

2 The developed ant colony (ACO) algorithms

The ACO (ant colony optimization) is an optimizing algorithm, a method developed by Marco Dorigo based on the modelling of the ants' social behaviour [1, 3–5]. In nature ants search for food by chance, then if they find some, on their way back to the ant-hill they mark the way with pheromone. Other ants – due to the pheromone sign – choose the marked way with higher probability instead of accidental wandering. Shorter ways may be completed quicker, thus on these ways more pheromone will be present than on longer ones. After a while the amount of pheromone drops (evaporation), by this preventing sticking to local optimum.

In the electronic freight exchange similar problem emerges as the ants' search for food: the target is the performance of

freight tasks offering the higher route level profit departing from the depot of the vehicle, with taking into account the limiting conditions. The problem, therefore, is twofold: on the one hand, the freight tasks to be performed shall be selected, and, on the other hand, their order shall be defined (FB_ACO, Fig. 1).

In the electronic warehouse exchange the task is simpler but is still similar to the food search: storage tasks shall be selected by taking into account storage capacity, possibly with the best possible exploitation of capacity (RB_ACO, Fig. 1).

The ant colony algorithm usable in electronic freight exchanges (FB_ACO) operates upon the following large-scale procedure:

1 Definition of starting data:

- starting point of optimum search (the vehicle location);
- narrowing down search space (local search): the selection of performable freight tasks depending on the distance compared to the depot of the vehicle;
- stating the distance matrix of freight tasks (what distance shall be taken from the depot of the vehicle or the completion of a task till the commencement of a new task);
- collection of the main features of the freight tasks (transport cost, carriage);
- establishment of pheromone matrix (the strength of the order of freight tasks following each other and the vehicle's depot, initially contains only 1).

2 Calculation of way selection probability:

- the probability that "r" freight task (or the vehicle location) will be followed by "s" freight task (α and β are based on a lot of runs, pheromone is more important than the heuristic information – distance) (3):

$$p_{r,s} = \frac{\varphi_{r,s}^\alpha \times \left(\frac{1}{d_{r,s}}\right)^\beta}{\sum_{t=1}^L \left[\varphi_{r,t}^\alpha \times \left(\frac{1}{d_{r,t}}\right)^\beta\right]} \quad (3)$$

$\varphi_{r,s}$: the amount of pheromone on edge r, s

$d_{r,s}$: the desirability (1/distance) of edge r, s

L : the number of optional freight tasks ($t = 1 \dots s \dots L$)

α : a parameter to control the influence of $\varphi_{r,s}$ ($\alpha = 2$)

β : a parameter to control the influence of heuristic information (distance) $\beta = 1/3$

- a matrix may be formed from the above mentioned probabilities (probability matrix).

3 Establishment of solution possibilities:

- establishment of random numbers, then selection of freight tasks upon probability matrix, till the realization of the limiting condition (distance);
- definition of the main features of the route (freight tasks, sequence, profit, distance);

- execution of the above mentioned tasks in accordance with the number of ant colonies (for example 50 ants = 50 versions).

4 Evaluation of the results of the iteration step:

- settling of profit matrix (how much profit the order of freight tasks following each other and the depot will bring from the aspect of the full route);
- filling in the profit matrix: inserting the highest route level profit reached in the iteration step into the certain relations. In this matrix, however, only those relations will show values which were touched by at least one ant; in case of more than one ant the highest value reached during the step will be included;
- updating the maximum profit (H_{max}) reached in the iteration steps, if improvement was realized;
- updating the pheromone matrix (4) (the 5/36 multiplier ensures balance between conservative and explorer search, H_{Max} results strong elitism):

$$\varphi_{r,s} = \varphi_{r,s} + \frac{5}{36} \cdot \varphi_{r,s} \cdot \frac{H_{r,s}}{H_{Max}} \quad (4)$$

$H_{r,s}$: the best profit if "r" freight task will be followed by "s" freight task

H_{Max} : the best profit during the iteration

- pheromone abrasion (5) (pheromone shall be replaced only on routes visited during the given iteration, pheromone amount is between 0,5 and 2):

$$\varphi_{r,s} = \varphi_{r,s} \times (1 - \rho) \quad (5)$$

ρ : the rate of pheromone abrasion ($\rho = 0, 1$).

5 Making new and new iteration steps (step 2, 3 and 4) as long as further improvement cannot be reached (H_{max}), or after certain number of steps.

In case of ant colony algorithm usable in electronic warehouse exchanges (RB_ACO, Fig. 1) only significant differences compared to the above mentioned will be presented. The first step is the definition of starting data and the establishment of the pheromone vector (in this case there are no task relations, thus instead of matrix the formation of a vector is enough). In case of selection of the storage task the task-related quantity to be stored will appear as heuristic information, thus the task selection probability – elements of probability matrix (6) – (the probability that the "r" storage task will be selected):

$$p_r = \frac{\varphi_r^\alpha \times m_r^\beta}{\sum_{t=1}^L \left[\varphi_t^\alpha \times m_t^\beta\right]} \quad (6)$$

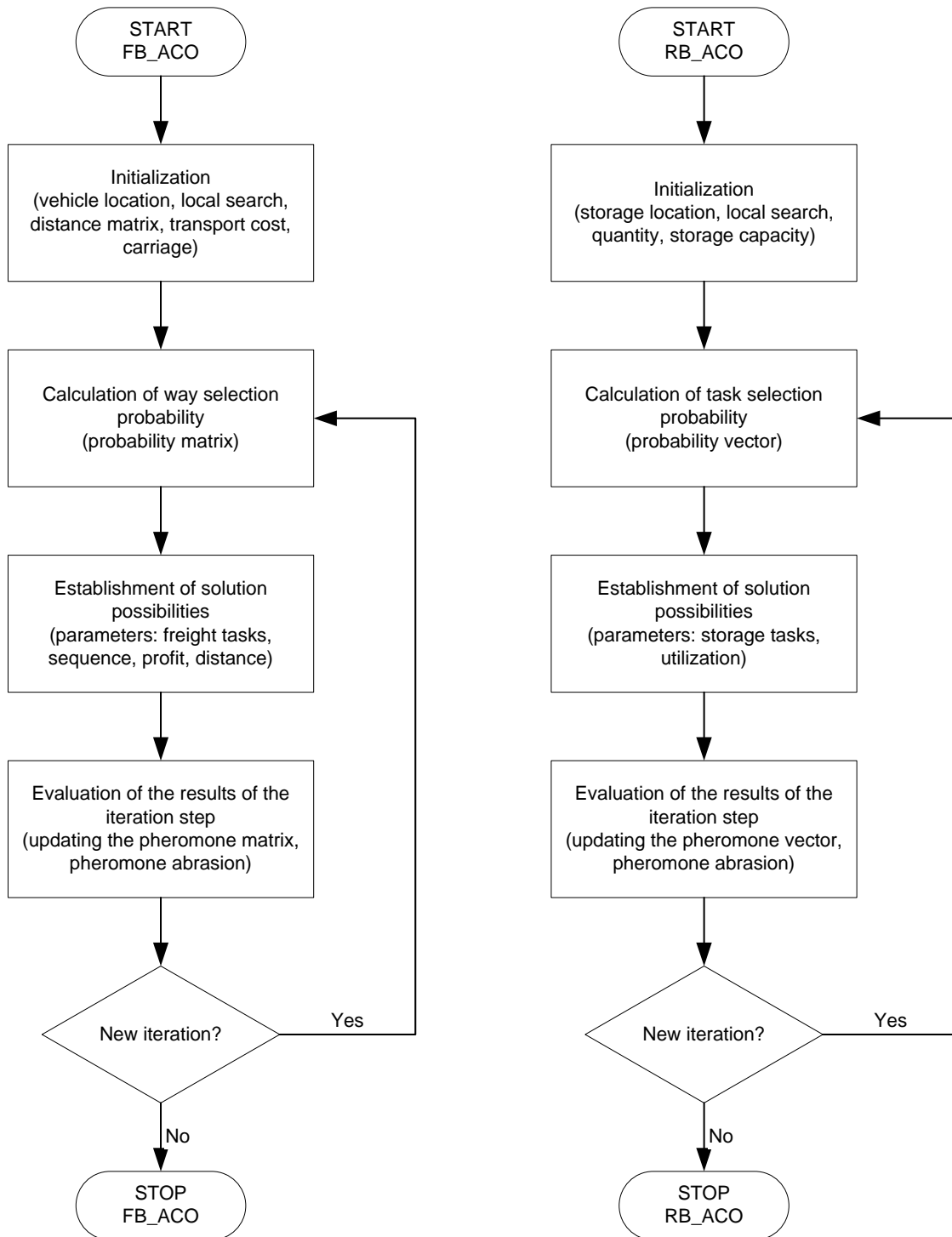


Fig. 1. The developed ant colony algorithms (FB_ACO and RB_ACO)

- φ_r : the amount of pheromone on edge "r"
- m_r : store quantity ("r" storage task)
- L: number of storage tasks ($t = 1 \dots r \dots L$)
- α : a parameter to control the influence of $\varphi_r (\alpha = 2)$
- β : a parameter to control the influence of heuristic information (quantity) $\beta = 1/3$

The selection of storage tasks and the storage of the main features of solution possibilities happen upon the probability vector (storage tasks, storage utilization). The pheromone update

Eq. (7) (and the pheromone abrasion) is carried out in the same way as in case of the previous algorithm:

$$\varphi_r = \varphi_r + \frac{5}{36} \cdot \varphi_r \cdot \frac{F_r}{F_{Max}} \quad (7)$$

F_r : the best utilization, if "r" storage task is included

F_{Max} : the best utilization during the iteration

It is worth mentioning that the algorithm developed for the optimization of routes (FB_ACO) is perfectly suitable for the optimization of storage use, because in this case not pheromone

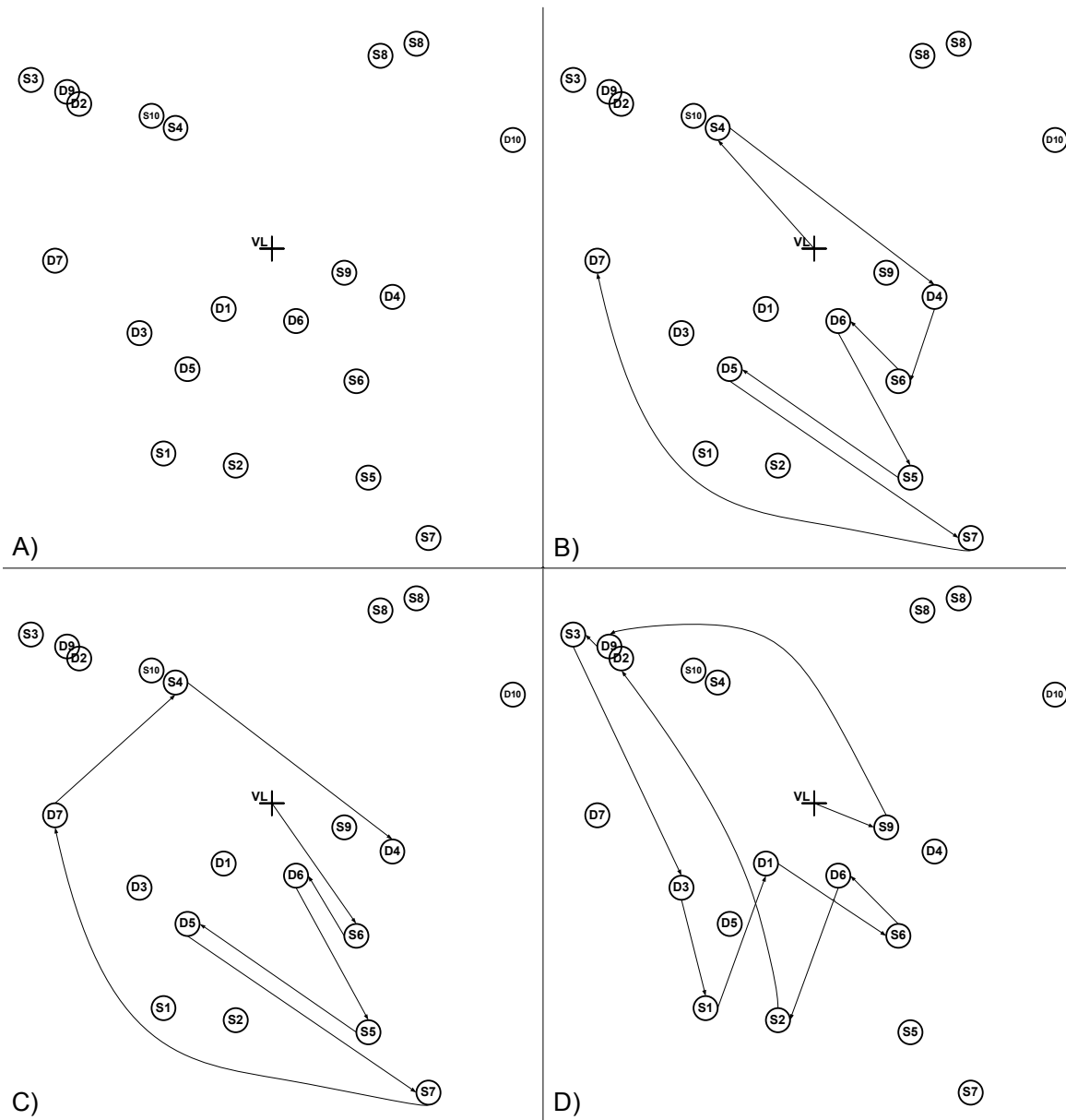


Fig. 2. The layout (A) and the three solutions (B: FB_ACO, C: profit maximising, D: distance minimising)

vector, but pheromone matrix shall be established, and by this pheromone is added not to the given storage tasks, but to the relations of storage tasks. Therefore it may be defined which collection of storage tasks shall be included into the program. In this case, of course, resistance between certain tasks (it is the distance in FB_ACO) shall be set uniformly (for example one, nil) in the relation of each storage task, therefore only information resulting from the search will play a role during selection (this may be even supplemented with quantity added to the storage tasks).

3 Results

The main results of the developed ant colony algorithms (parameters: 20 initial databases, 90 runs, 26 000 versions):

- sufficient number of examined versions (iterations, ants) : 3...5 percent of the theoretically possible versions;
- FB_ACO always gives better solution (on the average the dif-

ference is 43 percent, the standard deviation is 41 percent), than the methods (distance minimising – Dacey method [2] –, profit maximising) which are based on only one parameter.

For example a concrete problem may be seen in Fig. 2 (Part A) (freight tasks: 1...10, start point of tasks: S1...S10, destination point of tasks: D1...D10, VL: vehicle location, restrictive condition: distance/150). Table 1 includes the remuneration for the completion (FD). Table 2 presents the main results of the three different (FB_ACO, profit maximising, distance minimising) algorithms: FB_ACO gives the highest profit and the shortest route (the price/value rate is the best). Moreover, compared to the two mentioned methods, FB_ACO is very reliable.

Tab. 1. The remuneration for the completion

Freight tasks	1	2	3	4	5	6	7	8	9	10
FD (unit)	20	40	33	44	56	58	78	28	35	40

Tab. 2. The results of the examined algorithms

Results	Methods		
	FB_ACO	Profit maximising	Distance minimising
Freight tasks and sequence	4 → 6 → 5 → 7	6 → 5 → 7 → 4	9 → 3 → 1 → 6 → 2
Distance (unit)	145	153	148
Profit (unit)	90,9	83,4	37,7
Profit/distance	0,63	0,55	0,25

4 Conclusions

In the electronic freight and warehouse exchanges multifold optimum search is available, from which the one presented in this essay in details, the one on the methods on the optimization of organizing routes and storage use is one of the most promising tools of the development of such online fairs, on the one hand, and of becoming a determinative agent of the freight and storage market, on the other hand, and are essential for the support of users' procedures. With the help of the presented methods, by the filtering of local optimums, shortly a solution is found, which to a freight capacity starting at a certain location adds, almost optimally, freight tasks related to each other, and to a certain storage capacity selects storage tasks ensuring optimal use. The algorithms were coded and tested in MS Visual Basic language, as well as formulas and their characteristics. Compared to other linear programming methods it may be definitely stated that the established ant colony algorithm may be used effectively and flexibly in the solution of several problems of optimum search mentioned in the essay and arising from the same.

References

- 1 **Bonabeau E, Dorigo M, Theraulaz G**, *Swarm Intelligence: From Natural to Artificial Systems*, Oxford University Press, 1999.
- 2 **Dacey M F**, *Selection of an Initial Solution for the Travelling Salesman Problem*, *Operations Research* **8** (1960), 133-134.
- 3 **Dorigo M**, *Optimization, Learning and Natural Algorithms*, Politecnico di Milano, Italy, 1992. PhD thesis.
- 4 **Dorigo M, Gambardella L M**, *Ant Colony System: A Cooperative Learning Approach to the Travelling Salesman Problem*, *IEEE Transactions on Evolutionary Computation* **1** (1997), 53-66.
- 5 **Dorigo M, Stützle T**, *Ant Colony Optimization*, MIT Press, 2004.
- 6 **Kovács G**, *The structure, modules, services, and operational process of modern electronic freight and warehouse exchanges*, *Periodica Polytechnica ser. Transport Engineering* **37** (2009), no. 1-2, 33-38, DOI 10.3311/pp.tr.2009-1-2.06.