

# Tactical Models based on a Multi-Depot Vehicle Routing Problem Using the Ant Colony Optimization Algorithm

P. Stodola, and J. Mazal

**Abstract**—This article deals with the metaheuristic solution to the Multi-Depot Vehicle Routing Problem (MDVRP) using the Ant Colony Optimization (ACO) algorithm. The first part of this article presents the original algorithm of the authors. It introduces pivotal principles of the algorithm, along with conducted experiments and acquired results on benchmark instances in comparison with rival state of the art methods. The primary part of the article deals with two tactical models based on our problem solution: (a) optimal supply distribution on the battlefield and (b) optimal UAVs reconnaissance. Both models have become a part of our tactical information system which serves as a tool for commanders to support them in their decision making process. The models are introduced in terms of problem formulation, implementation, and application in practical situations in the domain of the military.

**Keywords**—Ant colony optimization, multi-depot vehicle routing problem, tactical modeling

## I. INTRODUCTION

THE Multi-Depot Vehicle Routing Problem (MDVRP) is a well-known problem with many real applications in the areas of transportation, distribution and logistics [1]. In many businesses (e.g. parcel delivery, appliance repair), it is vital to find the optimal solution to this problem as it saves resources for a company, reduces its expenses, shortens time needed to distribute services, and thus makes the company more competitive.

The MDVRP problem consists in computing optimal routes for a fleet of vehicles to drop off goods or services at multiple destinations (customers); each customer should be served only once. The vehicles might start from multiple depots, each located in a different place. The important characteristic is the limited capacity of each vehicle which cannot be exceeded. After visiting the selected customers, each vehicle returns to its depot and might start a new journey to other (so far unvisited) customers with a new load.

MDVRP is an NP-hard problem as it is a generalization of the travelling salesman problem [2], therefore polynomial-time algorithms are unlikely to exist [3]. In this article, we present our original solution approach based on the Ant Colony

Optimization (ACO) theory as a new approach to this topical issue. In fact, there have already been some attempts to use this theory for this problem, but nevertheless, the results of these solutions are not of the quality as when using other contemporary methods (see Table 2). We managed to develop and design the fundamental details and parameters of this approach so that the results are comparable to other state of the art algorithms.

The primary part of this article comprises two tactical models based on our problem solution which have a practical application in the specific domain of the military. It is a model of optimal supply distribution on the battlefield and a model of optimal reconnaissance by Unmanned Aerial Vehicles (UAV).

This tactical model has been implemented into an actual tactical information system designed to support commanders in their decision making process [4]. A key goal of the model is to provide a tool to support commanders in their decision making as this system include both fundamental and advanced models of military tactics.

## II. LITERATURE REVIEW

The solution methods for VRPs can be categorized as exact, heuristic, and metaheuristic. A broad overview of various methods is offered e.g. in [5]. For examples of exact methods, see e.g. [6], or [7], to name a few. Similar to the exact methods, many of heuristics have been developed, see e.g. [8], or [9].

Very popular metaheuristic methods have emerged in the last few years. These can be classified as state space search or evolutionary algorithms. For instance, simulated annealing [10] or genetic algorithms [11], [12] belong to the main evolutionary principles.

The remainder of this section focuses on the ACO methods. The potential of the ACO algorithm has been discovered very soon since it was published [13]. It was successfully applied for various problems [14], [15], [16].

Recently, there have been publications using the ACO theory for MDVRP problems [17], [18], [19]. The solution published in [17] is compared with our algorithm as it uses the standard Cordeau's test instances for evaluation.

## III. ANT COLONY OPTIMIZATION ALGORITHM

ACO algorithm is a probabilistic technique for developing good solutions of computational problems. The principle is

P. Stodola is with University of Defence, Brno, Czech Republic (phone: +420 973 442 474; e-mail: petr.stodola@unob.cz).

J. Mazal is with University of Defence, Brno, Czech Republic (e-mail: jan.mazal@unob.cz).

adopted from the natural world where ants explore their environment to find food; the idea is based on the behavior of ants seeking a path between their colony and a source of food.

#### A. Principle of the Algorithm

Fig. 1 presents the ACO algorithm we proposed for MDVRP. The solution found by the algorithm is improved in successive generations (iterations). In point 1, the termination condition is tested, points 2 to 15 cover an individual generation. Each depot employs a colony with the specific number of ants.

```

1. while not terminated
2.   for each ant in a colony
3.     set all nodes as unvisited
4.     while number of unvisited nodes > 0 do
5.       select a depot
6.       compute ant's probability of going to
           unvisited nodes
7.       select a node according to the
           probability
8.       if ant.load + node.load > ant.capacity
           then
9.         return to the depot
10.      else
11.        visit the selected node
12.      return to the depot
13.   save the best solution if found
14.   evaporate pheromone trails
15.   update pheromone trails
16. return the best solution

```

Fig. 1 ACO algorithm in pseudo code

In each generation, all ants in all colonies move between individual customers (referred to as nodes in Fig. 1). At first, the state of all nodes is set as unvisited in point 3. The algorithm continues until all nodes are visited (just once). In point 5, the depot (colony) is selected according to the given method; points 6 to 11 apply only to the ant from the selected colony. The ant's probability is computed in point 6; it determines the chance of the ant to go to every remaining unvisited nodes. In point 7, a node to be visited is chosen according to this probability.

Point 8 checks whether the ant can visit the selected node (i.e. whether its current load allows taking the load in the node and thus not exceeding ant's maximum capacity). If not, the ant returns to its colony (emptying its load) and the algorithm continues in point 5. If yes, the ant visits the selected node (delivering node's load and marking it as visited).

In point 12, after visiting all nodes, each ant returns to its colony. Then, if the best solution found in the generation is better than the best solution found in previous ones, it is saved (see point 13). Point 14 ensures evaporating the pheromone trails and in point 15, pheromone trails are updated according to the given method. Then, the next generation begins until the

termination condition is met. The best solution found is returned at the end of the algorithm in point 16.

#### B. Parameters of the Algorithm

The ACO algorithm requires setting a number of parameters influencing the problem solution. Some parameters are adopted from related problems; others are new. The list of all parameters is in Fig. 4.

A crucial parameter (proposed by authors) influences *how depots are selected* (see point 5 in Fig. 1). We propose five possibilities as follows:

- *Random selection*: depot (i.e. its vehicle) is selected randomly.
- *Selection of an idle depot*: depot with the shortest distance travelled so far is selected (i.e. vehicles take turns according to their distance they travelled at the moment of selection).
- *Selection of an idle depot (probability model)*: selection probabilities for all depots are computed based on the distance travelled so far (i.e. depots with shorter routes are more likely to be selected).
- *Selection of a depot with the greatest potential*: depot with the greatest potential is selected. The potential is computed as the sum of all pheromone trails which lead to unvisited customers (at the time of selection) – see formula (1).
- *Selection of a depot with the greatest potential (probability model)*: selection probabilities for all depots are computed based on the sum of pheromone trails to unvisited customers (i.e. depots with the bigger sum are more likely to be selected).

$$\varepsilon^k = \sum_{j \in S_u} \tau_{ij}^k \text{ for all } j \in S_u, \quad (1)$$

where  $\varepsilon^k$  is a potential for the colony (depot)  $k$ ,

$\tau_{ij}^k$  is strength of a pheromone trail from the colony  $k$  between nodes  $i$  and  $j$ ,

$i$  is an index for the node with the current position of the ant from colony  $k$ ,

$S$  is a set of all nodes,

$S_u$  is a set of so far unvisited nodes ( $S_u \subset S$ ).

When testing individual options, we found out that the last two surpass the others. This can be seen in Fig. 2 showing the results of parameter tuning which was conducted on a set of testing experiments.

These testing experiments we proposed were composed of 20 different problems of various complexity (from tens to several hundred of nodes) and at least 100 executions were processed to confirm the results statistically. All results are expressed as a solution quality in % (compared to the best result which was achieved).

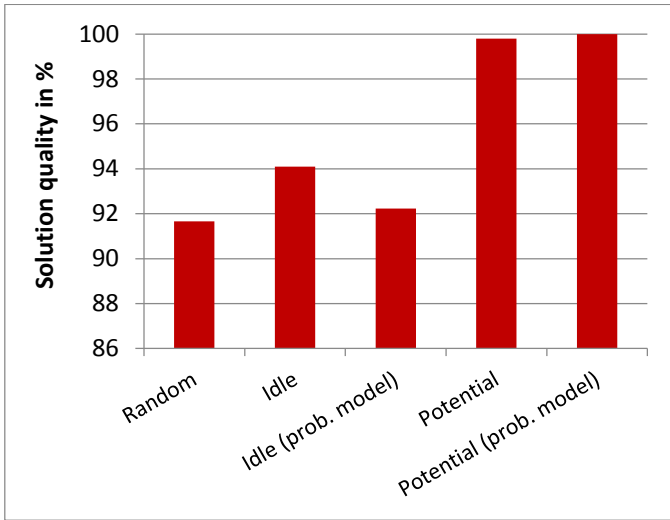


Fig. 2 Parameter tuning for depot selection

In point 6 in Fig. 1, after a depot (colony) is chosen according to the methods mentioned above, the probabilities of choosing ant's path to the one of so far unvisited nodes are computed according to formula (2).

$$p_{ij}^k = \frac{\tau_{ij}^{k\alpha} \cdot \eta_{ij}^{\beta} \cdot \mu_{ij}^{k\gamma} \cdot \kappa_{ij}^{k\delta}}{\sum_{l \in S_u} \tau_{il}^{k\alpha} \cdot \eta_{il}^{\beta} \cdot \mu_{il}^{k\gamma} \cdot \kappa_{il}^{k\delta}} \text{ for all } j \in S_u, \quad (2)$$

where  $p_{ij}^k$  is a probability for an ant from the colony  $k$  in a node  $i$  to visit a node  $j$ ,  
 $\tau_{ij}^k$  is strength of a pheromone trail from the colony  $k$  between nodes  $i$  and  $j$ ,  
 $\eta_{ij}$  is a multiplicative inverse of the distance between nodes  $i$  and  $j$ ,  
 $\mu_{ij}^k$  is a so-called savings measure [8],  
 $\kappa_{ij}^k$  is a measure for including the influence of ant's current load [20],  
 $\alpha, \beta, \gamma, \delta$  are coefficients controlling the influence of  $\tau_{ij}^k, \eta_{ij}, \mu_{ij}^k, \kappa_{ij}^k$  – see formula (2),  
 $S$  is a set of all nodes,  
 $S_u$  is a set of so far unvisited nodes ( $S_u \subset S$ ).

*Method of updating pheromone trails* (see point 15 in Fig. 1) is another key parameter. There are three possibilities as follows:

- *Update according to the best solution found in a current generation:* only the best solution found in a current generation influences the pheromone trails update process. This is an option proposed by authors.
- *Update according to the best solution found so far:* only the best solution found so far influences the pheromone trails update process.
- *Update according to all solutions found in a current generation:* all solutions found in a current generation influences the pheromone trails update process.

*Number of ants in colonies*  $n_a$  (see point 2 in Figure 1) is a parameter determining the number of different solutions to be created and evaluated within a generation. Pheromone trails are then updated according to these solutions (based on the given method mentioned above).

*Pheromone evaporation coefficient*  $\rho$  determines the speed of evaporating pheromone trails at the end of each generation (point 14 in Figure 1) – see Formula (3).

$$\tau_{ij}^k = (1 - \rho) \cdot \tau_{ij}^k \text{ for all } i, j \in V, \quad (3)$$

where  $\tau_{ij}^k$  is strength of a pheromone trail from the colony  $k$  between nodes  $i$  and  $j$ ,  
 $\rho$  is the pheromone evaporation coefficient.

*Pheromone updating coefficient*  $\sigma$  controls the influence of individual solutions when updating pheromone trails (see point 15 in Fig. 1). The right ratio of the pheromone evaporation and updating coefficients is very important since bad values can cause the fast evaporation of pheromone trails on the one hand or the solution to get stuck in a false optimum on the other.

Formulas (4) to (6) present the update of pheromone trails for different methods chosen (according to the best solution in a generation (4), or according to the best solution found so far (5), or according to all solutions in a generation (6) respectively).

$$\tau_{ij}^k = \tau_{ij}^k + x_{ij}^k \cdot \sigma \cdot \frac{d_{min}}{d_1} \text{ for all } i, j \in V, \quad (4)$$

$$\tau_{ij}^k = \tau_{ij}^k + x_{ij}^k \cdot \sigma \text{ for all } i, j \in V, \quad (5)$$

$$\tau_{ij}^k = \tau_{ij}^k + \sigma \cdot \sum_a x_{ij}^k \cdot \frac{d_a}{d_{tot}} \text{ for all } i, j \in V, \quad (6)$$

where  $\tau_{ij}^k$  is strength of a pheromone trail from the colony  $k$  between nodes  $i$  and  $j$ ,  
 $x_{ij}^k = \begin{cases} 1 & \text{if } i \text{ precedes } j \text{ on route } k, \\ 0 & \text{otherwise.} \end{cases}$   
 $\sigma$  is the pheromone updating coefficient,  
 $d_{min}$  is the distance of the best solution found so far,  
 $d_1$  is the distance of the best solution found in a generation,  
 $d_a$  is the distance of the  $a$ -th solution,  $a = 1, \dots, n_a$ ,  
 $d_{tot}$  is the sum of distances of all the solutions in a generation ( $d_{tot} = \sum_a d_a$ ).

### C. Experiments and Results

As benchmark problems, we chose Cordeau's MDVRP instances taken from [21], namely p01, p02, p03, p04, p05, p06, p07, p08, p09, p10, p11, p12, p15, p18, and p21 (instances p13, p14, p16, p17, p19, and p20 were not included in the experiments as they incorporate the constraint on the maximum length of a single route, which the algorithm does not support).

Table 1 presents the results. We conducted 100 tests on each instance and registered the best solution found, the mean and

standard deviation. The last column shows the difference between our results and the best solutions known so far which were received from [21]. The best known solutions were achieved by various algorithms during the history of benchmark instances.

Table 1 Results for MDVRP benchmark problems

| Inst. | NoC | NoD | BKS     | OBS     | Mean    | Stdev | Error |
|-------|-----|-----|---------|---------|---------|-------|-------|
| p01   | 50  | 4   | 576.87  | 576.87  | 583.15  | 6.50  | 0.00% |
| p02   | 50  | 4   | 473.53  | 475.86  | 482.86  | 3.44  | 0.49% |
| p03   | 75  | 5   | 641.19  | 644.46  | 650.04  | 4.12  | 0.51% |
| p04   | 100 | 2   | 1001.59 | 1018.49 | 1035.39 | 5.69  | 1.69% |
| p05   | 100 | 2   | 750.03  | 755.71  | 763.09  | 3.68  | 0.76% |
| p06   | 100 | 3   | 876.50  | 885.84  | 899.51  | 4.89  | 1.07% |
| p07   | 100 | 4   | 885.80  | 895.53  | 912.48  | 5.62  | 1.10% |
| p08   | 249 | 2   | 4420.95 | 4445.51 | 4572.23 | 66.75 | 0.56% |
| p09   | 249 | 3   | 3900.22 | 3990.19 | 4145.33 | 96.89 | 2.31% |
| p10   | 249 | 4   | 3663.02 | 3751.50 | 3864.92 | 50.21 | 2.42% |
| p11   | 249 | 5   | 3554.18 | 3657.16 | 3760.60 | 38.94 | 2.90% |
| p12   | 80  | 2   | 1318.95 | 1318.95 | 1320.48 | 1.90  | 0.00% |
| p15   | 160 | 4   | 2505.42 | 2510.11 | 2576.27 | 18.46 | 0.19% |
| p18   | 240 | 6   | 3702.85 | 3741.80 | 3812.25 | 37.22 | 1.05% |
| p21   | 360 | 9   | 5474.84 | 5631.12 | 5788.19 | 46.64 | 2.85% |

NoC – number of customers, NoD – number of depots  
BKS – best known solution, OBS – our best solution

Table 2 compares results obtained via our algorithm with other results published. Algorithms called GA1 [12], GA2 [22], and GA3 [11] are based on genetic algorithm principles. GJ stands for Gillett and Johnson's algorithm [23]; CGW stands for Chao, Golden and Wasil's algorithm [24]. FIND (Fast improvement, INtensification, and Diversification) is a tabu search based algorithm [9], and finally ACO is another version of an algorithm based on the ACO theory [17]. Best solution values in Table 2 are indicated by bold numbers.

Table 2 Best solutions values obtained by various algorithms

| Inst. | Our          | GA1    | GA2    | GA3    | GJ     | CGW          | FIND          | ACO    |
|-------|--------------|--------|--------|--------|--------|--------------|---------------|--------|
| p01   | <b>576.9</b> | 591.7  | 622.2  | 598.5  | 593.2  | <b>576.9</b> | <b>576.9</b>  | 620.5  |
| p02   | 475.9        | 483.1  | 480.0  | 478.7  | 486.2  | 474.6        | <b>473.5</b>  | -      |
| p03   | 644.5        | 694.5  | 706.9  | 699.2  | 652.4  | <b>641.2</b> | <b>641.2</b>  | -      |
| p04   | 1018.5       | 1062.4 | 1024.8 | 1011.4 | 1066.7 | 1012.0       | <b>1003.9</b> | 1585.9 |
| p05   | 755.7        | 754.8  | 785.2  | -      | 778.9  | 756.5        | <b>750.3</b>  | -      |
| p06   | 885.8        | 976.0  | 908.9  | 882.5  | 912.2  | 879.1        | <b>876.5</b>  | -      |
| p07   | 895.5        | 976.5  | 918.1  | -      | 939.5  | 893.8        | <b>892.6</b>  | 1257.9 |

|     |               |        |        |   |        |        |               |        |
|-----|---------------|--------|--------|---|--------|--------|---------------|--------|
| p08 | <b>4445.5</b> | 4812.5 | 4690.2 | - | 4832.0 | 4511.6 | 4485.1        | -      |
| p09 | 3990.2        | 4284.6 | 4240.1 | - | 4219.7 | 3950.9 | <b>3937.8</b> | 9633.2 |
| p10 | 3751.5        | 4291.5 | 3984.8 | - | 3822.0 | 3727.1 | <b>3669.4</b> | -      |
| p11 | 3657.2        | 4092.7 | 3880.7 | - | 3754.1 | 3670.2 | <b>3649.0</b> | -      |
| p12 | <b>1319.0</b> | 1421.9 | 1319.0 | - | -      | 1327.3 | <b>1319.0</b> | -      |
| p15 | <b>2510.1</b> | 3059.2 | 2579.3 | - | -      | 2610.3 | 2551.5        | -      |
| p18 | <b>3741.8</b> | 5462.9 | 3903.9 | - | -      | 3877.4 | 3781.0        | -      |
| p21 | <b>5631.1</b> | 6872.1 | 5926.5 | - | -      | 5791.5 | 5656.5        | -      |

We can see that our algorithm managed to find better solutions in all cases when compared with the genetic principle based algorithms (GA1, GA2, GA3) and also in case of Gillett and Johnson's algorithm (GJ). The results are also better in 7 cases (and in 1 case the same) in comparison with the CGW algorithm and in 4 cases (and in 2 cases the same) in comparison with the algorithm FIND.

The last column of Table 2 shows the results for another version of the algorithm based on the ant colony optimization theory. As we can see, the results for this algorithm do not compare well with any other algorithm presented; in case of the instance p09, the error is more than 140% compared to the best known solution.

In addition to the experiments mentioned above, we launched our algorithm on another set of Cordeau's test instances (taken from [21]), namely pr01 to pr10. The results are shown in Table 3. In two cases (pr02 and pr07), we managed to improve the best known solution.

Table 3 Results for another set of MDVRP benchmark problems

| Instance | NoC | NoD | BKS     | OBS     | Error  |
|----------|-----|-----|---------|---------|--------|
| pr01     | 48  | 4   | 861.32  | 861.32  | 0.00%  |
| pr02     | 96  | 4   | 1288.37 | 1277.12 | -0.87% |
| pr03     | 144 | 4   | 1782.58 | 1802.16 | 1.10%  |
| pr04     | 192 | 4   | 2072.52 | 2114.06 | 2.00%  |
| pr05     | 240 | 4   | 2343.66 | 2380.75 | 1.58%  |
| pr06     | 288 | 4   | 2675.16 | 2709.41 | 1.28%  |
| pr07     | 72  | 6   | 1085.61 | 1077.33 | -0.76% |
| pr08     | 144 | 6   | 1666.60 | 1686.96 | 1.22%  |
| pr09     | 216 | 6   | 2153.10 | 2182.10 | 1.35%  |
| pr10     | 288 | 6   | 2811.49 | 2817.31 | 0.21%  |

NoC – number of customers, NoD – number of depots  
BKS – best known solution, OBS – our best solution

#### IV. OPTIMAL SUPPLY DISTRIBUTION MODEL

The ACO algorithm has been integrated into our tactical information system designed to support command decision-making.

This model seeks distribution patterns to provide supplies to friendly elements operating in the area of interest as efficiently as possible. Efficiency is based on the nature of the task at hand; the objective might be to minimize the sum of distances travelled by all vehicles, or minimize the time of the whole operation, or minimize the total fuel consumed by all vehicles.

The system provides a user friendly interface enabling to add, edit and delete nodes (depots and customers). Fig. 3 shows the main dialog for this model. As an example, 4 depots (labelled A to D) and 18 customers were included.

When all nodes are added (including their maximum load/capacity in kilograms), the MDVRP algorithm is executed. Values of algorithm's parameters are set and used to select the best options for the task at hand (see Fig. 4). Note the parameter called number of cores used; this parameter represents the number of cores of a multi-core processor used for the

execution since the ACO algorithm can be parallelized.

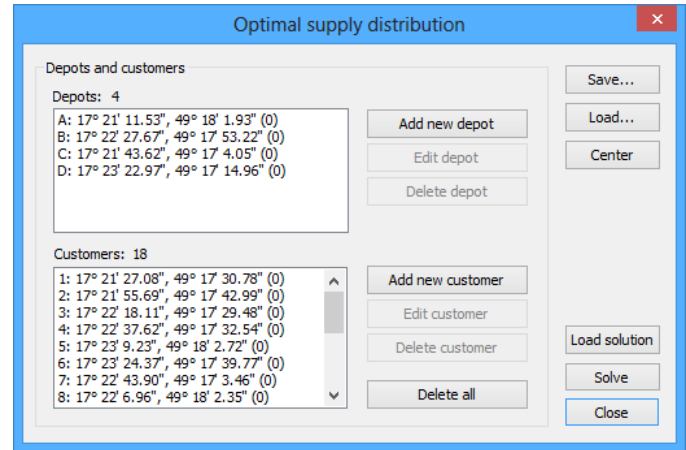


Fig. 3 Dialog for the optimal supply distribution model

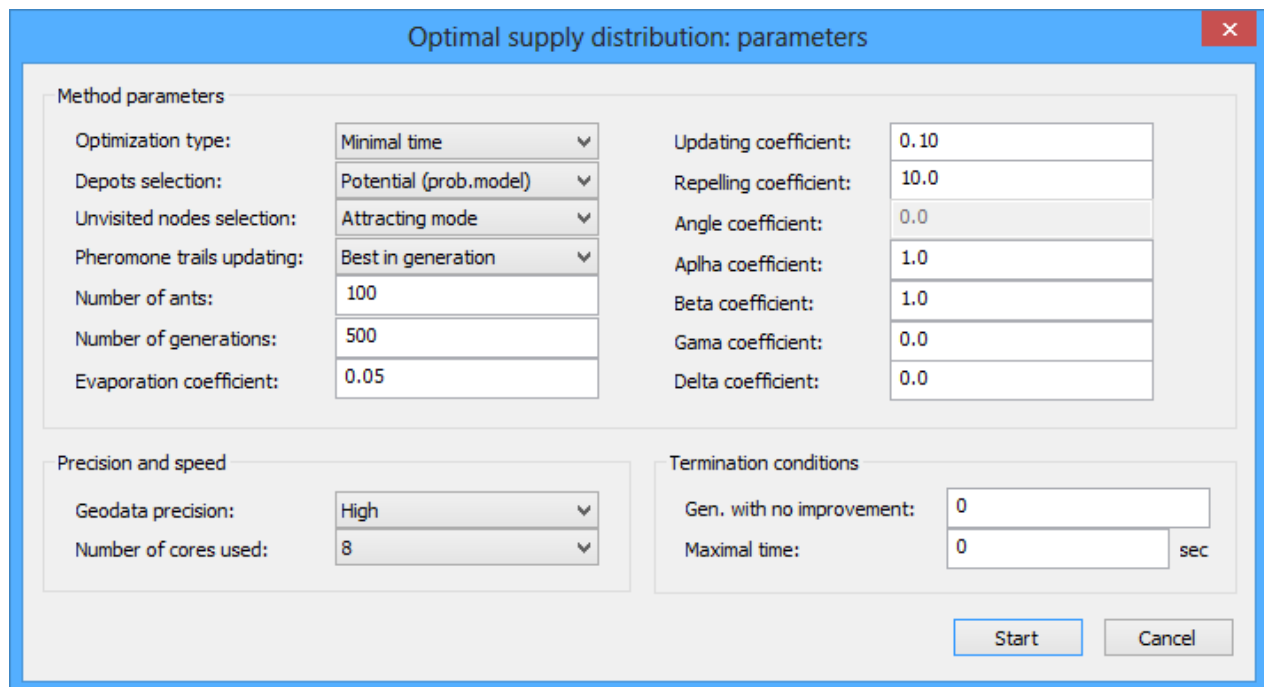


Fig. 4 Parameters for the ACO algorithm

Final routes for all vehicles are displayed both textually and graphically – see Fig. 5. Depots are shown as blue hexagons, customers as blue circles, and the red lines present the optimal

routes for individual vehicles. Although the example is rather simple, the same system can be used for tasks with many depots and hundreds of customers.

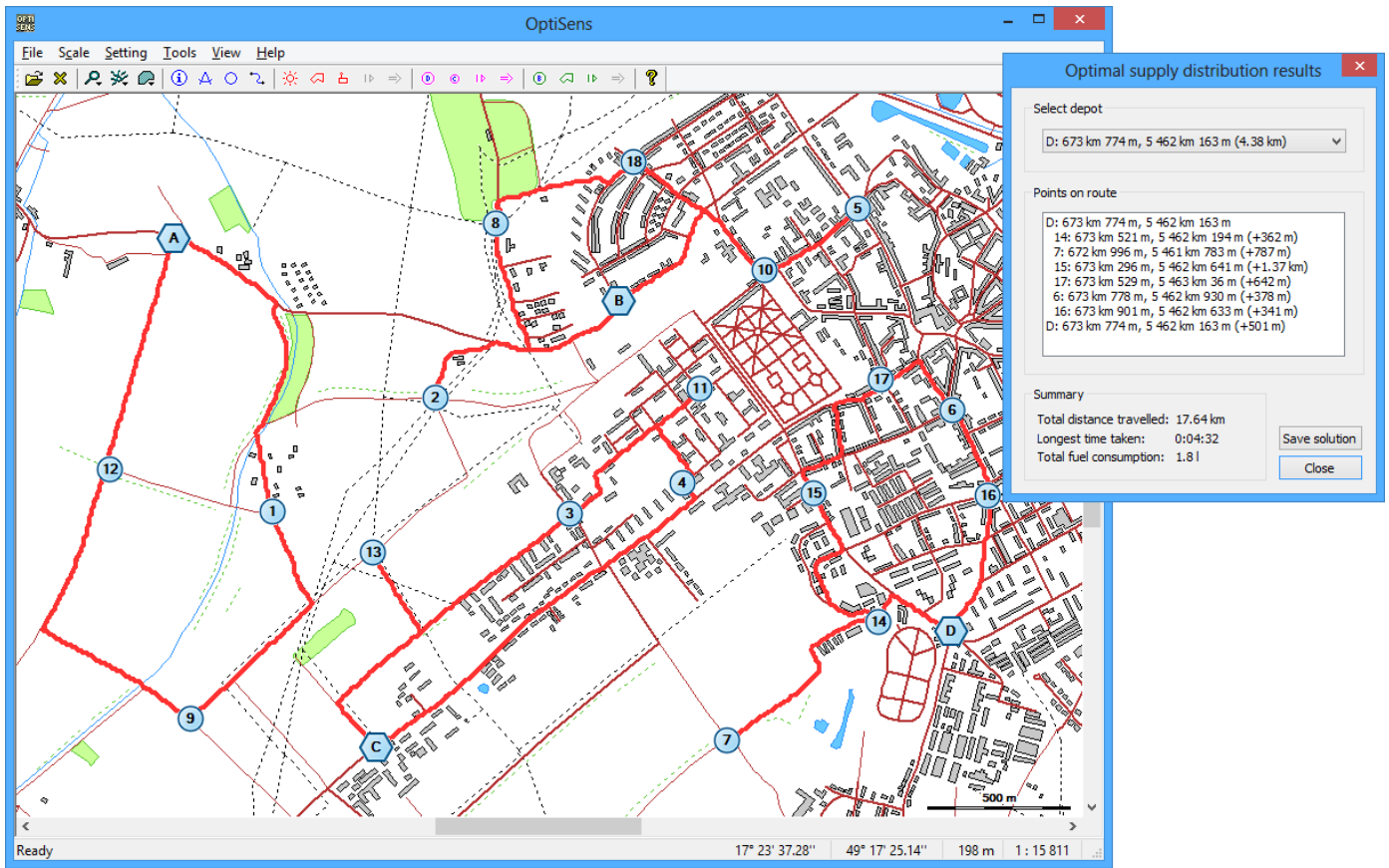


Fig. 5 Solution to the example situation

V. OPTIMAL UAVS RECONNAISSANCE MODEL

The second tactical model implemented is designed for optimal aerial reconnaissance of the area if interest. It can be used by one or more teams (squads) equipped with unmanned aerial vehicle(s). The model requires defining the area of interest. This is done via a system interface where the area is created using a polygon. The position(s) of UAV base(s) also need to be specified. The goal is to explore the defined area of interest as efficiently as possible; most often the objective is to minimize the overall time of the operation.

Fig. 6 presents the main dialog for this model depicting an example with 3 UAV bases (labelled A to C), i.e. 3 unmanned aerial vehicles are available. The average velocity for each vehicle can also be specified (see Fig. 7). Note that the area of interest in the example is defined by a polygon with total area of 989,891 m<sup>2</sup>.

The optimization starts by spreading out the number of checkpoints uniformly inside the area of interest; these checkpoints are customers for the MDVRP problem, i.e. each checkpoint is visited just once by any vehicle. The number of checkpoints depends on the distance parameter (see Fig. 6) which defines the approximate distance between neighbor checkpoints. It is assumed that by visiting all checkpoints the area of interest is completely explored.

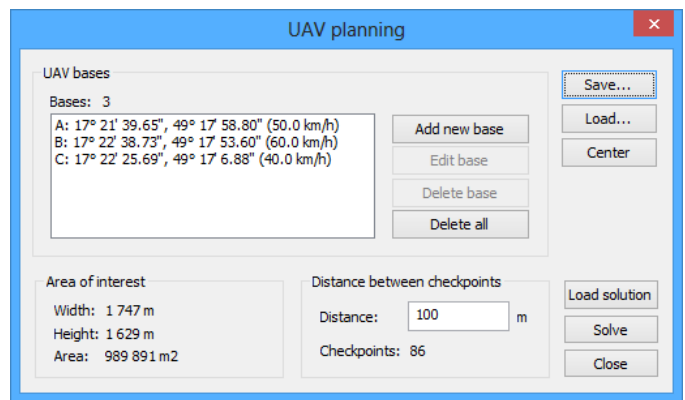


Fig. 6 Dialog for the optimal UAVs reconnaissance planning

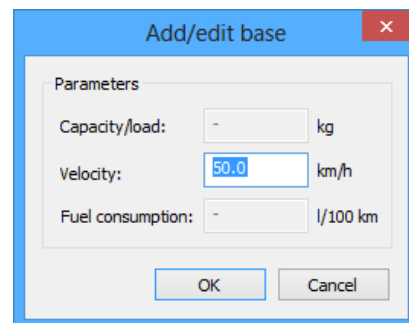


Fig. 7 Dialog for specifying the average velocity for UAVs

In Fig. 8, the solution for the example situation is shown. The solution offers a very good time (close to optimal) required for the entire reconnaissance of the given area (note that the average velocities of individual vehicles are not the same). Blue

lines show the exact routes for every vehicle; the system is designed and prepared for uploading the routes immediately into the real vehicles in the real situation on the battlefield.

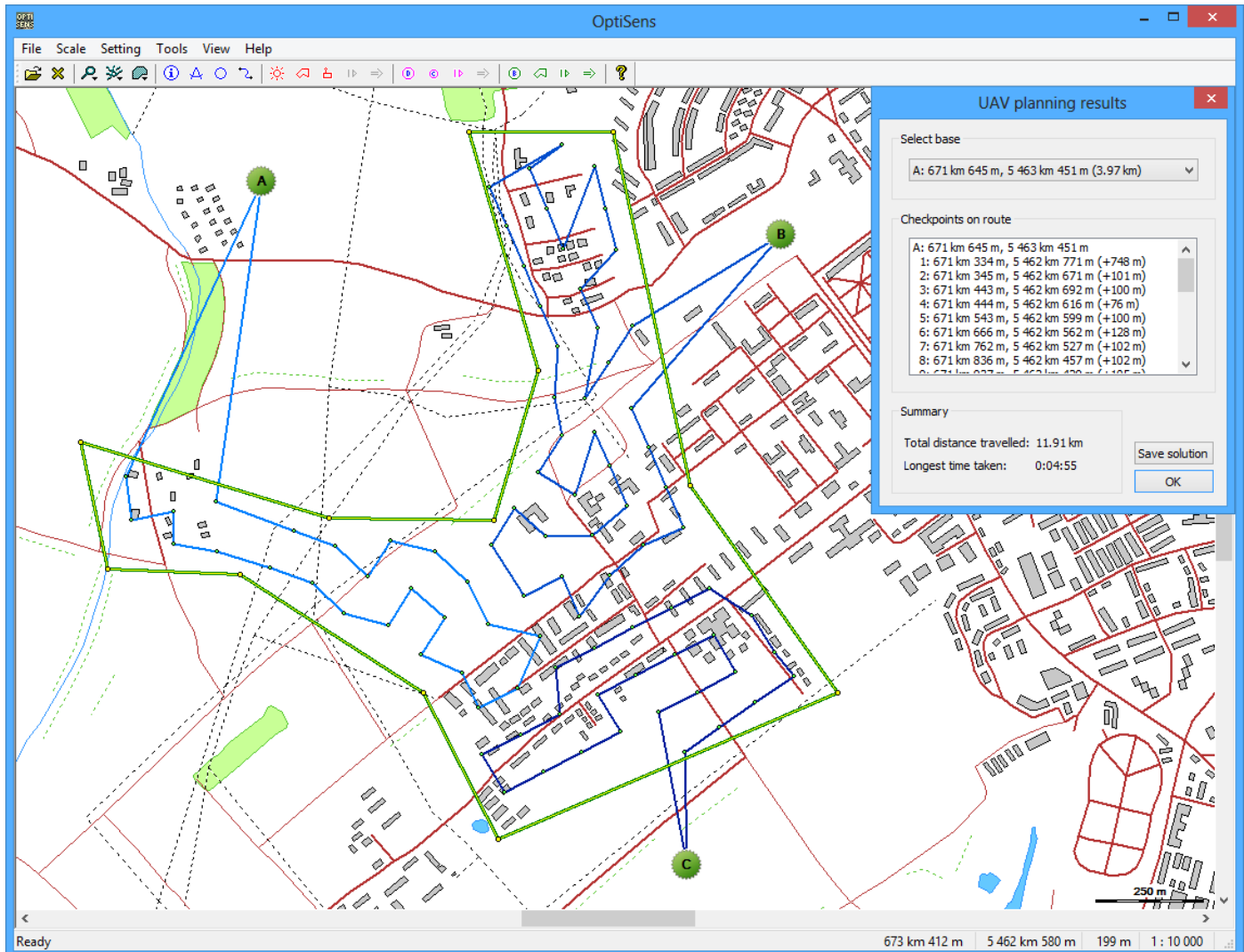


Fig. 8 Solution to the example situation

## VI. CONCLUSION

The paper presents the approach proposed by authors to the capacitated MDVRP problems based on the ant colony optimization theory. We have developed some new parameters and options not published yet (e.g. methods of selecting depots, method of updating pheromone trails according to the best solution found in a generation), thus contributing to the ACO theory. The new parameters we designed and verified participate on the very good results which the algorithm was able to achieve.

The strengths of the proposed algorithm are as follows:

- Fast convergence close to the optimal solution.
- High quality of solutions (comparable to the state of the art methods).

- Universal applicability (to metric, non-metric, and asymmetric problems).
- Possibility of distributed parallel processing.
- Application of the algorithm without any modification to solve classic VRP or capacitated VRP problems.

The proposed algorithm is of considerable significance in practical application in the domain of the military. It has been implemented into our tactical information system designed to support commanders' decision-making in order to provide the interface to solve the tactical task.

There are also a lot of ways of improving the current version of the algorithm and the system in the future.

Some future perspectives are as follows:

- Distribution of some computation to a GPU processor.
- Distribution of processing not only to the cores of a multi-core processor but also among more computers (to the GRID networks for instance).
- Development of other methods than empirical approach how to find the best parameter setting for various tasks.
- Extension of the algorithm for solving other problems (for instance MDVRP with Time Windows or with Pick-up and Delivering).

#### REFERENCES

- [1] G. B. Dantzig, and J. H. Ramser, "The Truck Dispatching Problem," in *Management Science*, vol. 6, no 1, pp. 80-91, 1959.
- [2] C. Contardo, and R. Martinelli, "A new exact algorithm for the multi-depot vehicle routing problem under capacity and route length constraints," in *Discrete Optimization*, vol. 12, pp. 129-146, 2014.
- [3] M. R. Garey, and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York: W. H. Freeman & Co., 1990.
- [4] P. Stodola, and J. Mazal, "Planning Algorithm and its Modifications for Tactical Decision Support Systems," in *International Journal of Mathematics and Computers in Simulation*, vol. 6, no 1, pp. 99-106, 2012.
- [5] G. Laporte, "The Vehicle Routing Problem: An Overview of Exact and Approximate Algorithms," in *European Journal of Operational Research*, vol. 59, no 3, pp. 345-358, 1992.
- [6] G. Laporte, H. Mercure, and Y. Nobert, "An Exact Algorithm for the Asymmetrical Capacitated Vehicle Routing Problem," in *Networks*, vol. 16, no 1, pp. 33-46, 1986.
- [7] J. Gromicho, J. J. van Hoorn, A. L. Kok, and J. M. J. Schutten, "Restricted Dynamic Programming: A flexible framework for solving realistic VRPs," in *Computers and Operations Research*, vol. 39, no 5, pp. 902-909, 2012.
- [8] G. Clarke, and J. W. Wright, "Scheduling of Vehicles from a Central Depot to a Number of Delivery Points," in *Operations Research*, vol. 12, no 4, pp. 568-581, 1964.
- [9] J. Renaud, G. Laporte, and F. F. Boctor, "A Tabu Search Heuristic for the Multi-Depot Vehicle Routing Problem," in *Computers & Operations Research*, vol. 23, no 3, pp. 229-235, 1996.
- [10] Z. J. Czech, and P. Czarnas, "Parallel Simulated Annealing for the Vehicle Routing Problem with Time Windows," in *10th Euromicro Workshop on Parallel, Distributed and Network-based Processing*, Canary Islands, pp. 376-383, 2002.
- [11] P. Surekha, and S. Sumathi, "Solution To Multi-Depot Vehicle Routing Problem Using Genetic Algorithms," in *World Applied Programming*, vol. 1, no 3, pp. 118-131, 2011.
- [12] R. Thangiaha, and S. Salhi, "Genetic Clustering: An Adaptive Heuristic for the Multidepot Vehicle Routing Problem," in *Applied Artificial Intelligence*, vol. 15, no 4, pp. 361-383, 2001.
- [13] M. Dorigo, "Optimization, Learning and Natural Algorithms," Ph.D. dissertation, Milan: Politecnico di Milano, 1992.
- [14] M. Dorigo, and L. C. Gambardella, "Ant Colonies for the Traveling Salesman Problem," in *Biosystems*, vol. 43, no 2, pp. 73-81, 1997.
- [15] A. R. M. de Silva, and G. L. Ramalho, "Ant System for the Set Covering Problem," in *IEEE International Conference on Systems, Man, and Cybernetics*, Tucson, vol. 5, pp. 3129-3133, 2001.
- [16] J. E. Bell, and P. R. McMullen, "Ant Colony Optimization Techniques for the Vehicle Routing Problem," in *Advanced Engineering Informatics*, vol. 18, no 1, pp. 41-48, 2004.
- [17] T. C. M. Caldeira, *Optimization of the Multi-Depot Vehicle Routing Problem: an Application to Logistics and Transport of Biomass for Electricity Production*. Lisbon: Technical University of Lisbon, 2009.
- [18] Jianhua Ma, and Jie Yuan, "Ant Colony Algorithm for Multiple-Depot Vehicle Routing Problem with Shortest Finish Time," in *Communications in Computer and Information Science*, vol. 113, pp. 114-123, 2010.
- [19] K. S. V. Narasimha, E. Kivelevitch, and M. Kumar, "Ant Colony Optimization Technique to Solve Min-Max MultiDepot Vehicle Routing Problem," in *American Control Conference (ACC)*, Montreal, pp. 3980-3985, 2012.
- [20] B. Bullnheimer, R. F. Hartl, and C. Strauss, "Applying the Ant System to the Vehicle Routing Problem," in *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization*, Springer US, pp. 285-296, 1999.
- [21] NEO Web, *Networking and Emerging Optimization*. Malaga: University of Malaga, 2015, <http://neo.lcc.uma.es/vrp/vrp-instances/multiple-depot-vrp-instances/> [Accessed on 30 January 2015].
- [22] B. Ombuki, F. and Hanshar, "An Effective Genetic Algorithm for the Multi-Depot Vehicle Routing Problem," Technical Report No. CS-04-10, St. Catharines, Canada: Brock University, 2004.
- [23] B. E. Gillett, and J. G. Johnson, "Multi-Terminal Vehicle-Dispatch Algorithm," in *Omega*, vol. 4, no 6, pp. 711-718, 1976.
- [24] M. I. Chao, B. L. Golden, and E. Wasil, "A New Heuristic for the Multi-Depot Vehicle Routing Problem that Improves upon Best-known Solutions," in *American Journal of Mathematical and Management Sciences*, vol. 13, no 3-4, pp. 371-406, 1993.
- [25] D. Valis, L. Zak, and O. Pokora, "Contribution to System Failure Occurrence Prediction and to System Remaining Useful Life Estimation based on Oil Field Data," in *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, vol 229, no 1, pp. 36-45, 2015.
- [26] D. Valis, and L. Zak, "Contribution to Oil Data Assessment for System Condition Determination," in *European Safety and Reliability Conference, ESREL 2014*, Wroclaw, pp. 287-292, 2014.
- [27] K. Silinger, and M. Blaha, "Possibilities of Effectiveness Increasing of Some Fire Control Preparation Measurements in the 25. Field Artillery of the Czech Republic Army," in *Recent Researches in Computational Intelligence and Information Security*, Jakarta, Island of Java, Indonesia: WSEAS Press, pp. 97-101, 2011.
- [28] M. Blaha, and K. Silinger, "Task Formation Commander's Operation Order for Fire Support," in *Applied Mathematics, Computational Science and Engineering*, Varna: Eurompt, pp. 380-388, 2014.
- [29] Junhai Ma, and Jing Zhang, "Measure of Bullwhip Effect Considering Stochastic Disturbance Based on Price Fluctuations in a Supply Chain with Two Retailers Based on Price Fluctuations in a Supply Chain with Two Retailers," in *WSEAS Transactions on Mathematics*, vol 14, pp. 116-126, 2015.
- [30] L. Chaabane, and M. Abdelouahab, "A Hybrid Method Applied to Multiple Sequence Alignment Problem," in *WSEAS Transactions on Computers*, vol 14, pp. 465-473, 2015.
- [31] J. Farlik, A. Stefek, and J. Casar, "Multi-Agent System as Operational Center Support," in *Proceedings of the 16th International Conference on Mechatronics*, Mechatronika 2014, Brno, pp. 458-462, 2015.
- [32] J. Farlik, "New Approaches to Operational Centres Modelling," in *Proceedings of the 15th International Conference on Mechatronics*, Mechatronika 2012, Prague, 2015.
- [33] J. Gogola, "Comparison of Selected Stochastic Mortality Models," in *International Journal of Mathematical Models and Methods in Applied Sciences*, vol 9, pp. 159-165, 2015.
- [34] J. Bongiorno, and A. Mariscotti, "Evaluation of Performances of Indexes Used for Validation of Simulation Models Based on Real Cases," in *International Journal of Mathematical Models and Methods in Applied Sciences*, vol 9, pp. 133-147, 2015.