The Tactical Model based on a Multi-Depot Vehicle Routing Problem

P. Stodola, and J. Mazal

Abstract—The Multi-Depot Vehicle Routing Problem is a famous problem formulated more than 50 years ago. Since that time, a lot of exact, heuristic and metaheuristic methods have been proposed in order to find a feasible solution for this NP-hard problem. The first part of this paper presents the original algorithm of the authors based on the Ant Colony Optimization theory. This part 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 the tactical model based on our problem solution: optimal supply distribution. The model has become a part of our tactical information system which serves as a tool for commanders to support them in their decision making process. The model is 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 a tactical model based on our problem solution which has a practical application in the specific domain of the military. It is a model of optimal supply distribution on the battlefield.

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 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.

![ACO algorithm in pseudo code](image)

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 (see below). The list of all parameters is in Fig. 3.

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).

\[
\varepsilon^k = \sum_{j \in S_u} \tau^k_{ij} \text{ for all } j \in S_w, \tag{1}
\]

where \(\varepsilon^k\) is a potential for the colony (depot) \(k\), \(\tau^k_{ij}\) 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 an index for the set of so far visited nodes, \(S_u\) is a set of so far unvisited nodes (\(S_u \subset S\)).

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).

\[
p^k_{ij} = \frac{\tau^k_{ij} \cdot \eta_{ij}^\beta \cdot \mu^k_{ij} \cdot \kappa^k_{ij} \cdot \kappa^k_{ij}}{\sum_{i \in S_u} \tau^k_{il} \cdot \eta_{il}^\beta \cdot \mu^k_{il} \cdot \kappa^k_{il}} \text{ for all } j \in S_u, \tag{2}
\]

where \(p^k_{ij}\) is a probability for an ant from the colony \(k\) in a node \(i\) to visit a node \(j\), \(\tau^k_{ij}\) 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 \(j\).
\[ \mu_k^i j \] is a so-called savings measure [8], 
\[ k^i j \] is a measure for including the influence of ant’s current load [20],
\[ \alpha, \beta, \gamma, \delta \] are coefficients controlling the influence of 
\[ r_k^i j, \eta_i j, \mu_k^i j, \delta^k j \] – see formula (2),
\[ S \] is a set of all nodes,
\[ S_u \] is a set of so far unvisited nodes (\( S_u \subset S \)).

**Number of ants in colonies** \( n_u \) (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).

\[
\tau_k^i j = (1 - \rho) \cdot \tau_k^i j \quad \text{for all} \quad i, j \in V, \tag{3}
\]

where \( \tau_k^i j \) is strength of a pheromone trail from the colony \( k \) between nodes \( i \) and \( j \),
\( \rho \) is the pheromone evaporation coefficient.

**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**

<table>
<thead>
<tr>
<th>Inst.</th>
<th>NoC</th>
<th>NoD</th>
<th>BKS</th>
<th>OBS</th>
<th>Mean</th>
<th>Stdev</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>p01</td>
<td>50</td>
<td>4</td>
<td>576.87</td>
<td>576.87</td>
<td>583.15</td>
<td>6.50</td>
<td>0.00%</td>
</tr>
<tr>
<td>p02</td>
<td>50</td>
<td>4</td>
<td>473.53</td>
<td>475.86</td>
<td>482.86</td>
<td>3.44</td>
<td>0.49%</td>
</tr>
<tr>
<td>p03</td>
<td>75</td>
<td>5</td>
<td>641.19</td>
<td>644.46</td>
<td>650.04</td>
<td>4.12</td>
<td>0.51%</td>
</tr>
<tr>
<td>p04</td>
<td>100</td>
<td>2</td>
<td>1001.59</td>
<td>1018.49</td>
<td>1035.39</td>
<td>5.69</td>
<td>1.69%</td>
</tr>
<tr>
<td>p05</td>
<td>100</td>
<td>3</td>
<td>750.03</td>
<td>755.71</td>
<td>763.09</td>
<td>3.68</td>
<td>0.76%</td>
</tr>
<tr>
<td>p06</td>
<td>100</td>
<td>3</td>
<td>876.50</td>
<td>885.84</td>
<td>899.51</td>
<td>4.89</td>
<td>1.07%</td>
</tr>
<tr>
<td>p07</td>
<td>100</td>
<td>4</td>
<td>885.80</td>
<td>895.53</td>
<td>912.48</td>
<td>5.62</td>
<td>1.10%</td>
</tr>
<tr>
<td>p08</td>
<td>249</td>
<td>2</td>
<td>4420.95</td>
<td>4445.51</td>
<td>4572.23</td>
<td>66.75</td>
<td>0.56%</td>
</tr>
<tr>
<td>p09</td>
<td>249</td>
<td>3</td>
<td>3900.22</td>
<td>3990.19</td>
<td>4145.33</td>
<td>96.89</td>
<td>2.31%</td>
</tr>
<tr>
<td>p10</td>
<td>249</td>
<td>4</td>
<td>3663.02</td>
<td>3751.50</td>
<td>3864.92</td>
<td>50.21</td>
<td>2.42%</td>
</tr>
</tbody>
</table>

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

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, ITensification, 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**

<table>
<thead>
<tr>
<th>Inst.</th>
<th>Our</th>
<th>GA1</th>
<th>GA2</th>
<th>GA3</th>
<th>GJ</th>
<th>CGW</th>
<th>FIND</th>
<th>ACO</th>
</tr>
</thead>
<tbody>
<tr>
<td>p01</td>
<td>576.9</td>
<td>591.7</td>
<td>622.2</td>
<td>598.5</td>
<td>593.2</td>
<td><strong>576.9</strong></td>
<td>576.9</td>
<td>620.5</td>
</tr>
<tr>
<td>p02</td>
<td>475.9</td>
<td>483.1</td>
<td>480.0</td>
<td>478.7</td>
<td>486.2</td>
<td>474.6</td>
<td><strong>473.5</strong></td>
<td>-</td>
</tr>
<tr>
<td>p03</td>
<td>644.5</td>
<td>694.5</td>
<td>706.9</td>
<td>699.2</td>
<td>652.4</td>
<td><strong>641.2</strong></td>
<td>641.2</td>
<td>-</td>
</tr>
<tr>
<td>p04</td>
<td>1018.5</td>
<td>1062.4</td>
<td>1024.8</td>
<td>1011.4</td>
<td>1066.7</td>
<td>1012.0</td>
<td><strong>1003.9</strong></td>
<td>1585.9</td>
</tr>
<tr>
<td>p05</td>
<td>755.7</td>
<td>754.8</td>
<td>785.2</td>
<td>-</td>
<td>778.9</td>
<td>756.5</td>
<td><strong>750.3</strong></td>
<td>-</td>
</tr>
<tr>
<td>p06</td>
<td>885.8</td>
<td>976.0</td>
<td>908.9</td>
<td>882.5</td>
<td>912.2</td>
<td>879.1</td>
<td><strong>876.5</strong></td>
<td>-</td>
</tr>
<tr>
<td>p07</td>
<td>895.5</td>
<td>976.5</td>
<td>918.1</td>
<td>-</td>
<td>939.5</td>
<td>893.8</td>
<td><strong>892.6</strong></td>
<td>1257.9</td>
</tr>
<tr>
<td>p08</td>
<td><strong>4445.5</strong></td>
<td>4812.5</td>
<td>4690.2</td>
<td>-</td>
<td>4832.0</td>
<td>4511.6</td>
<td>4485.1</td>
<td>-</td>
</tr>
<tr>
<td>p09</td>
<td>3990.2</td>
<td>4284.6</td>
<td>4240.1</td>
<td>-</td>
<td>4219.7</td>
<td>3950.9</td>
<td><strong>3937.8</strong></td>
<td>9633.2</td>
</tr>
<tr>
<td>p10</td>
<td>3751.5</td>
<td>4291.5</td>
<td>3984.8</td>
<td>-</td>
<td>3822.0</td>
<td>3727.1</td>
<td><strong>3669.4</strong></td>
<td>-</td>
</tr>
<tr>
<td>p11</td>
<td>3657.2</td>
<td>4092.7</td>
<td>3880.7</td>
<td>-</td>
<td>3754.1</td>
<td>3670.2</td>
<td><strong>3649.0</strong></td>
<td>-</td>
</tr>
<tr>
<td>p12</td>
<td>1319.0</td>
<td>1421.9</td>
<td>1319.0</td>
<td>-</td>
<td>-</td>
<td>1327.3</td>
<td><strong>1319.0</strong></td>
<td>-</td>
</tr>
<tr>
<td>p15</td>
<td><strong>2510.1</strong></td>
<td>3059.2</td>
<td>2579.3</td>
<td>-</td>
<td>-</td>
<td>2610.3</td>
<td>2551.5</td>
<td>-</td>
</tr>
<tr>
<td>p18</td>
<td>3741.8</td>
<td>5462.9</td>
<td>3903.9</td>
<td>-</td>
<td>-</td>
<td>3877.4</td>
<td>3781.0</td>
<td>-</td>
</tr>
<tr>
<td>p21</td>
<td><strong>5631.1</strong></td>
<td>6872.1</td>
<td>5926.5</td>
<td>-</td>
<td>-</td>
<td>5791.5</td>
<td>5656.5</td>
<td>-</td>
</tr>
</tbody>
</table>

NoC – number of customers, NoD – number of depots
BKS – best known solution, OBS – our best solution
instance p09, the error is more than 140% compared to the best known solution.

IV. OPTIMAL SUPPLY DISTRIBUTION MODEL

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

This subsystem 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. 2 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. 3). 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.

Final routes for all vehicles are displayed both textually and graphically — see Fig. 4. 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.
V. CONCLUSION

The paper presents the approach proposed by authors to the capacitated MDVRP problems based on the ant colony optimization theory. We have developed same 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