Advanced Logistic Systems – Theory and Practice, 13(2), 5-20, https://doi.org/10.32971/als.2020.001

INVESTIGATION OF CONVERGENCE PROPERTIES OF ANT COLONY BASED ALGORITHMS

ANITA AGÁRDI[1]-LÁSZLÓ KOVÁCS[2]-TAMÁS BÁNYAI[3]

Abstract: The paper presents the Ant Colony Optimization (ACO) algorithms, and investigates their convergence properties. Our investigation is limited to the following ACO algorithms: Ant System, MAX-MIN Ant System, Elitist Strategy of Ant System and Rank Based Version of Ant System. In the literature can be seen, that a novel of optimization problems are solved with ACO algorithms. In our investigation, we limited the Multi-Depot Vehicle Routing Problem with Time Windows (MDVRPTW), which is an NP-hard discrete optimization problem. The paper presents a literature review in connection with ACO algorithms, and Vehicle Routing Problem (VRP). After that, the mathematical model is written. The paper also presents the implemented algorithms, and the test results based on benchmark data.
Keywords: Ant Colony Optimization, Multi-Depot Vehicle Routing Problem with Time Windows, optimization

1. INTRODUCTION

The Ant Colony Optimization (ACO) algorithms are heuristic optimization algorithms. They belong to swarm intelligence methods. The first ACO algorithm is initially proposed by Marco Doringo in 1992 his PhD thesis. His first algorithm is implemented for searching an optimal path in a graph. Since then several versions of ACO algorithms were implemented. In this article the following ACO algorithms are presented: Ant Colony System, Ant System, Elitist Strategy of Ant System, MAX-MIN Ant System, Rank Based Version of Ant System. We tested the efficiency of the algorithms with the Multi-Depot Vehicle Routing Problem with Time Windows (MDVRPTW). The Vehicle Routing Problem is a logistical problem. One of the most important tasks of logistics is the cost-efficient transportation of the right goods to the right place. The VRP can simulate the in-plant and out-plant material handling processes or can simulate the complex logistics system. In the case of the complex system the in-plant, the out-plant material handling is taken into account. In the case of the MDVRPTW, the position of depots is known in advance. Also, the number of vehicles per depot, the capacity of the vehicles are known in advance. We also know the position, the demand and the time windows of the customer. The demand of all customers must be satisfied in the time frame (time window). The objective function is the minimization of the length of the route and follows the constraints. In this article, the efficiency of the ACO algorithms is compared with Cordeau's benchmark data. Based on the test results the Rank Based Version of Ant System gave the best performance.

[1]PhD student, Department of Information Technology, University of Miskolc
agardianita@iit.uni-miskolc.hu
[2]associate professor, Department of Information Technology, University of Miskolc
kovacs@iit.uni-miskolc.hu
[3]associate professor, Institute of Logistics, University of Miskolc
alttamas@uni-miskolc.hu

2. LITERATURE REVIEW

In this chapter, we present a literature review. First, the Ant Colony Optimization is presented, then the Vehicle Routing Problem.

2.1. A literature review of Ant Colony Optimization

The Ant Colony Optimization algorithms are common heuristic algorithms. We can find 180 articles in the literature (in the Web of Science database) related to the keywords: TITLE: ("ant colony optimization") AND TOPIC: ("convergence")). Our search was conducted in February 2019; therefore, new articles may have been published since then. On subject area (Figure 1) shows that the majority of the publication related to computer science artificial intelligence. Also, some publications have been published in the following fields: Engineering electrical electronic, operations research management science, computer science interdisciplinary applications.

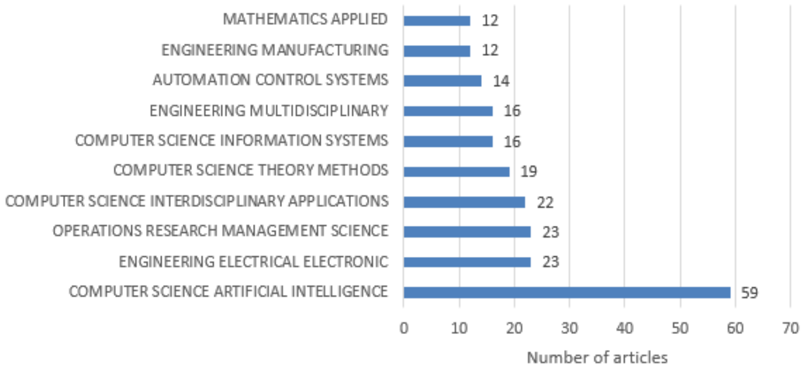

Figure 1. Classification of articles considering subject areas based on a search in WoS database using TITLE: "ant colony optimization" AND TOPIC: "convergence" keywords

As Figure 2 demonstrates, the ant colony optimization has been researched over the past ten years.

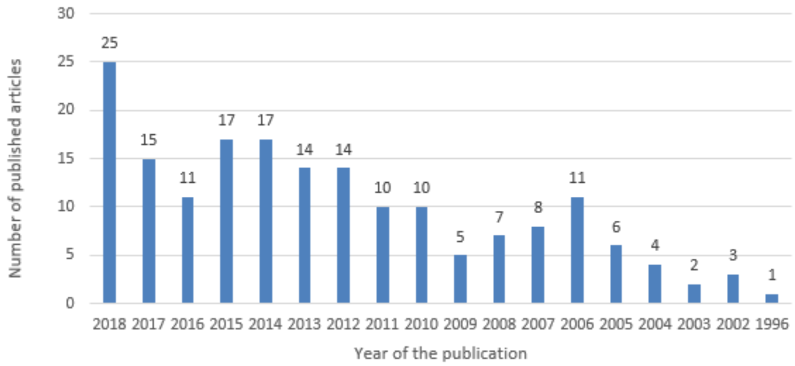

Figure 2. Classification of articles by year of the publication based on a search in WoS database using TITLE: "ant colony optimization" AND TOPIC: "convergence" keywords

Ant Colony Optimization consists of the following algorithms: Ant Colony System, Ant System, MAX-MIN Ant System, Rank Based Version of Ant System, Ant-Q. The algorithms simulate the food search of the ants. There are lots of problems, which have been solved by the ant colony optimization. These include the following problems:

- shop scheduling problem [1]: The authors investigate the open shop scheduling problem. They solve the problems with tabu search and ACO algorithm.
- image edge detection [2]: The authors establish a pheromone matrix, which represents the edge information, pixel position of the image. The movements of the ants correspond to the image’s intensity values.
- flowshop scheduling problem [3]: The 2-machine flowshop scheduling problem is considered in the article. Simulated Annealing search and local search algorithms are also applied along ACO algorithm.
- object segmentation [4]: In the article, the segmentation of infrared object with fuzzy entropy approach is investigated. The ACO is used to obtain the optimal parameters. Tests are compared with the Genetic algorithm.
- design of water distribution systems [5]: The optimal design of water distribution systems is investigated in the article. The problem is solved with ACO algorithms and Genetic Algorithm. The algorithms are compared, the ACO algorithms gave better solutions.
- project scheduling [6]: The paper presents the resource-constrained project scheduling problem. The authors present the combination of two pheromone evaluation methods in ACO algorithm. Presented also benchmark problems from the Project Scheduling Library.
- traveling salesman problem [7]: The ACO algorithm is presented first in the paper. The paper describes, that both for symmetric and asymmetric instances of the TSP gave the algorithm good solutions.
- knapsack problem [8]: The authors establish a new ACO algorithm, the binary ant system (BAS).
- vehicle routing problem [9]: A new approach of the ACO algorithm is presented, which simulates the decision-making processes of ant colonies as they forage for food.
- vehicle routing problem with time windows [10]: The Multiple Ant Colony System for the Vehicle Routing Problem with Time Windows is established in the paper.
- multi-depot vehicle routing problem [11]: The paper presents the Vehicle Routing Problem with virtual central depot (V-MDVRP). An improved ant colony optimization with coarse-grain parallel strategy, ant-weight strategy and mutation operation is presented in the paper.
- text feature selection [12]: The authors describe the text feature selection. This process is the most important step in text categorization. To improve the performance of text categorization the authors use ACO algorithm.
- data mining [13]: The authors present Ant-Miner (ant-colony-based data miner) algorithm. The algorithm extracts classification rules from data.

2.2. A literature review of Vehicle Routing Problem

Vehicle routing is a very intensively investigated research area. We can find 6424 articles in the literature (in the Web of Science database) related to the keywords: TITLE: ("vehicle routing problem") AND TITLE: ("optimization")). On subject area (Figure 3) shows that the majority of the publication related to operations research management science. In addition, some publications have been published in the following fields: Transportation science technology, management and computer science interdisciplinary applications.

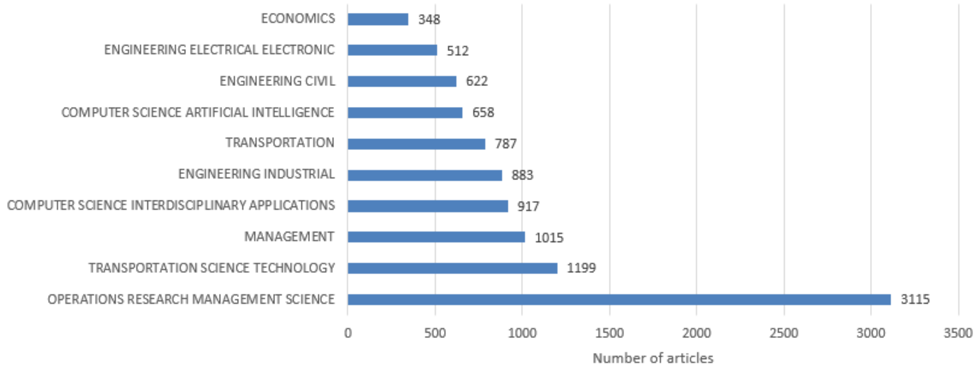

Figure 3. Classification of articles considering subject areas based on a search in WoS database using TITLE: "vehicle routing problem" AND TITLE: "optimization" keywords

As Figure 4 demonstrates, the vehicle routing problem has been researched over the past ten years.

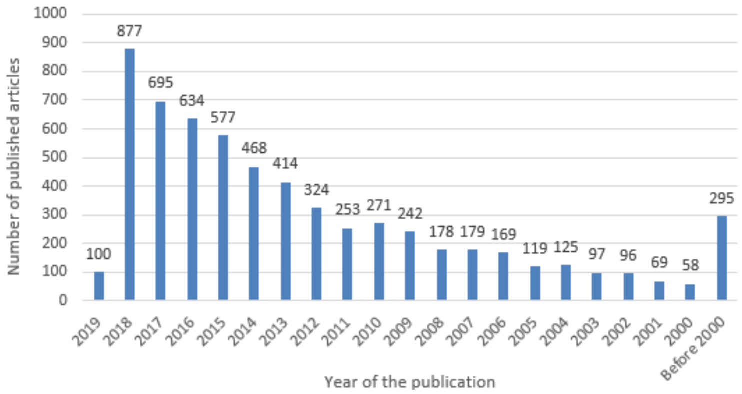

Figure 4. Classification of articles by year of the publication based on a search in WoS database using TITLE: "vehicle routing problem" AND TITLE: "optimization" keywords

Figure 5 shows the number of articles in the literature (in the Web of Science database) related to the keywords: TITLE: ("multi-depot vehicle routing problem with time windows") AND TITLE: ("optimization")). On subject area (Figure 5) shows that the majority of the publication related to operations research management science. In addition, a number of publications have been published in the following fields: computer science interdisciplinary applications, management and industrial engineering.

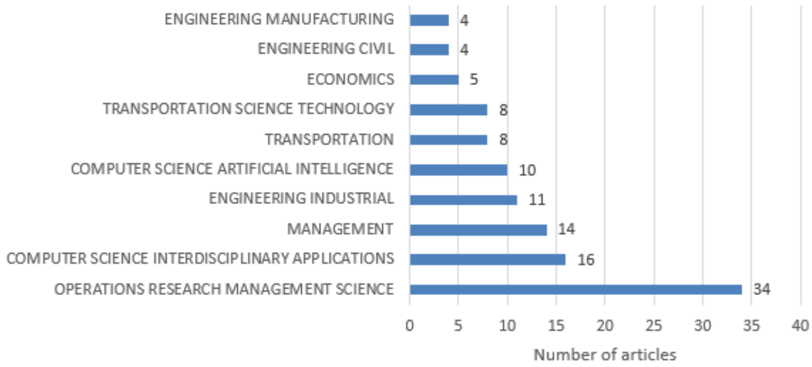

Figure 5. Classification of articles considering subject areas based on a search in WoS database using TITLE: "multi-depot vehicle routing problem with time windows" AND TITLE: "optimization" keywords

As Figure 6 demonstrates, the multi-depot vehicle routing problem with time windows has been intensively researched over the past six years.

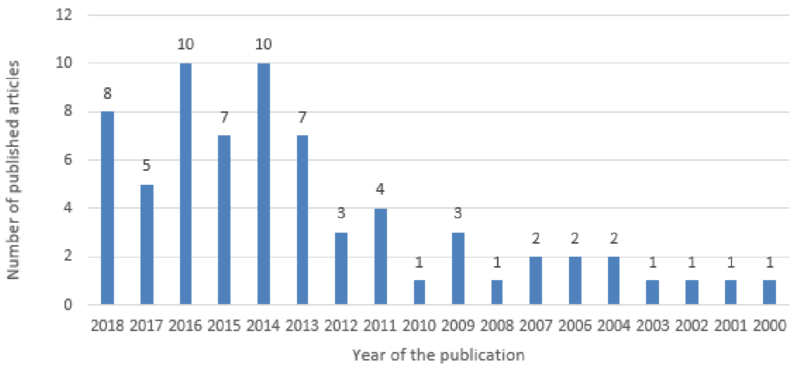

Figure 6. Classification of articles by year of the publication based on a search in WoS database using TITLE: "multi-depot vehicle routing problem with time windows" AND TITLE: "optimization" keywords

The Vehicle Routing Problem (VRP) is highly investigated. In the case of the base problem, the position and the demand of the customers are given. Also the position of the depot, the number and capacity constraints of the vehicles are given. The goal is the minimization of the length of the route. Over the years, many variations of the problem have been developed. In the following, we present the most common variants, the constraint of the VRP:

1. Multi-Depot Vehicle Routing Problem (MDVRP) [14]: In the case of MDVRP some (not only one) depot is given. The vehicles start from one depot, visit the customers then return to the depot from which have been started.
2. Open Vehicle Routing Problem (OVRP) [15]: In the case of OVRP, the vehicles do not return to the depot after visited the customers.

3. Periodic Vehicle Routing Problem (PVRP) [16]: The customer must be visited periodically (for example each customer must be visited daily, each must be visited weekly etc.)
4. Pick-up and Delivery Vehicle Routing Problem (PDVRP) [17]: Each customer has pickup demand and each customer has delivery demand. This means that each product must be delivered from the depot to the customers while each product must be collected from the customers to the depot.
5. Vehicle Routing Problem with Time Windows (VRPTW) [18]: Each customer has a time window. The customers must be visited within this interval.
6. Two-Echelon Vehicle Routing Problem (2E-VRP) [19]: The product will be delivered from the depot to the customers not directly. From the depot, the product is delivered to intermediate locations called satellites. From the satellites, the product is transferred to the customers.

3. Multi-Depot Vehicle Routing Problem with time windows

In the case of the Multi-Depot Vehicle Routing Problem (MDVRPTW) given the position of the depots and customers in advance. The customers have product demands, time windows and service times. Time window indicates an interval, in which the demand of the customer must be satisfied. The depots have a certain number of vehicles. The vehicles have capacity constraint and maximum duration time. The goal is the minimization of the length of the route. In Figure 7, the depots are indicated by D1 and D2, and the customers are indicated by integer numbers (from 1 to 10).

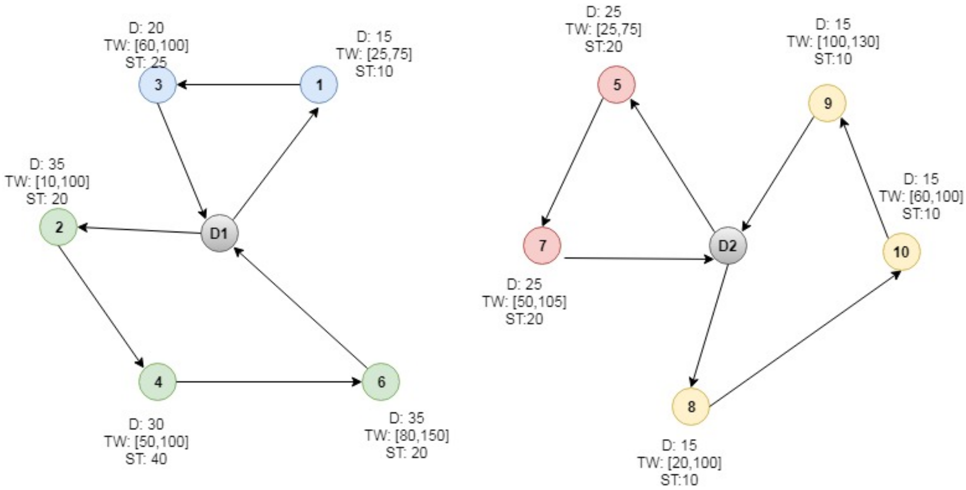

Figure 7. The Multi-Depot Vehicle Routing Problem with Time Windows

In Figure 7 the D means the demand of each customer, the TW indicates the time window (earliest and latest service time), the ST indicates the service time of the customers. The D1 and D2 depots have two vehicles. The first vehicle starts from D1, then visits customer 1, customer 3 then returns to the D1 depot. The second vehicle starts from D1, then visits customer 2, customer 4, customer 6 then returns to the D1 depot. The third vehicle starts from D2, then visits customer 5, customer 7 then returns to depot D2. The fourth vehicle starts the tour from D2 depot, then visits customer 8, customer 10, customer 9 then returns to D2 depot.

In our model we define the following input parameters:

- Customers R= $[r_i]$
- Depots F= $[f_p]$
- Vehicles $V = [v_p]$
- The demand of the customers D= $[d_i]$
- The earliest service time of the customers $E = [e_i]$
- The latest service time of the customers $T = [t_i]$
- The service time of the customers $S = [s_i]$
- The capacity constraint of the vehicles $Q = [q_{p,h}]$
- The maximum duration of a route per vehicle $RD = [rd_{p,h}]$
- Location of customers $LC = [lc_i]$
- Location of depots $LD = [ld_p]$
- Distance $DI = DI(LC, LD)$ $DI = [di_{w,g}]$ $w, g = 1 \dots n + m$
- where
- $DI^{CC} = [di_{w,g}^{CC}]$ $w, g = 1$ to n distance among customers
- $DI^{CD} = [di_{w,g}^{CD}]$ $w = 1$ to n and $g = n + 1$to $n + m$ distance in customer-depot relation
- $DI^{DC} = [di_{w,g}^{DC}]$ $w = n + 1$ to $m + n$ and $g = 1$ to n distance in depot-customer relation
- $DI^{DD} = [di_{w,g}^{DD}]$ $w = n + 1$ to $m + n$ and $g = n + 1$ to $n + m$ distance in depot-depot relation

In our model we define the following indices:

Customer	$i, j = 1\ to\ n$
Depot	$p = 1\ to\ m$
Vehicle	$h = 1$ to $max(vp)$

The following decision variable is defined:

$X = [x_{p,h,b}]$ vehicle h of depot p is assigned to customer $x_{p,h,b}$ as supply task b

$$b = 1 \dots b_{p,h}^{max} \tag{1}$$

$b_{p,h}^{max}$ is the number of assigned customers to vehicle h of depot p.

In our problem the objective function can be defined in the following way:

$$min \sum_{p=1}^{m} \sum_{h=1}^{\max_p v_p} (di_{p,x_{p,h,1}}^{DC} + di_{x_{p,h,b_{p,h}^{max}},p}^{CD} + \sum_{b=1}^{b_{p,h}^{max}-1} di_{x_{p,h,b},x_{p,h,b+1}}^{CC}) \tag{2}$$

The following constraint must be taken into account:

All customers are visited exactly once:

$$\sum_{p=1}^{m} \sum_{h=1}^{\max_p v_p} x_{i,p,h}^{*} = 1 \quad \forall i \tag{3}$$

where $X^* = [x^*_{p,h,b}]$ is a binary matrix based on the permutation matrix X and $x^*_{i,p,h} = 1$ if $x_{p,h,b} = i$ otherwise 0.

Vehicles cannot exceed their capacity limits:

$$\sum_{i=1}^{m} d_{x_{p,h,b}} \leq q_{p,h} \quad \forall p,h \tag{4}$$

Vehicles cannot exceed the maximum duration of a route:

$$\sum_{b=1}^{b_{p,h}^{max}} s_{x_{p,h,b}} + \sum_{b=1}^{b_{p,h}^{max}-1} di^{CC}_{x_{p,h,b},x_{p,h,b+1}} + di^{DC}_{p,x_{p,h,1}} + di^{CD}_{x_{p,h,b_{p,h}^{max}},p} \leq rd_{p,h} \quad \forall p,h \tag{5}$$

It is not allowed to exceed the defined time window, which can be calculated in two different ways depending on the position of the customer in the route.

- if the customer is the first one being supplied from the depot:

$$e_i \leq di^{DC}_{p,x_{p,h,1}} + s_{x_{p,h,1}} \leq t_i \qquad \forall \ \ i = x_{p,h,1} \tag{6}$$

- if the customer is supplied not as the first one in the route from the depot:

$$e_i \leq \sum_{b=1}^{i} s_{x_{p,h,b}} + \sum_{b=1}^{i} di^{CC}_{x_{p,h,b},x_{p,h,b+1}} + di^{DC}_{p,x_{p,h,1}} \leq t_i \qquad \forall i \tag{7}$$

4. REPRESENTATION OF THE PROBLEM

In this chapter, we present the representation of the Multi-Depot Vehicle Routing Problem with Time Windows (MDVRPTW). This representation is used by the ACO algorithms.

We used a two-part representation. The first part is depot-vehicle compliance (depot-vehicle part). This part contains the number of depots, as many vehicles have each depot, for example $1, 1, 2, 2$.

The second part is the sequence of the customers (customer part). This part contains all customers (number of customers) exactly once, for example $1, 3, 2, 4, 6, 5, 7, 8, 10, 9$.

The evaluation of the two-part representation is the following: First, the sequence of customers is taken into account. Taking the elements of this sequence until one constraint is broken. This sequence of customers will belong to the first vehicle. The depot of the first vehicle is the first element of the depot-vehicle part. Then taking the elements of the sequence of the customers from the last taken customer until all constraints are satisfied. This will be the customers of the second vehicle. The second element of the depot-vehicle part will indicate the depot of the second vehicle. Iterating the above-written steps until all customers are not visited. If all customers are visited, we get the solution. If not all customers are visited, but the depot-vehicle compliance has no more elements, then all vehicles are used, but not all customers are visited, so the solution does not satisfy all constraints. To avoid these solutions, we use penalty points. The fitness function is the distance travelled and the number of missed customers.

5. ANT COLONY OPTIMIZATION ALGORITHMS

The base ACO algorithms [20] are extended with depot and customer part described above. The basic steps are represented in Figure 8.

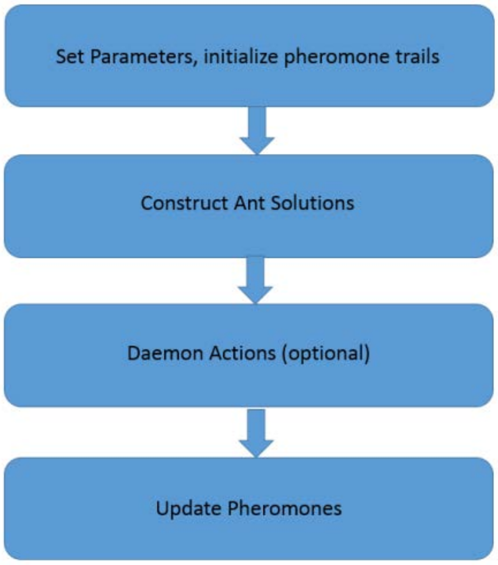

Figure 8. The basic steps of the Ant Colony Optimization algorithms

5.1. Ant System

The algorithm consists of the following steps [20]:

1. Generating the depot part and the customer part for each ant. The generation occurs randomly.

2. Initializing the pheromones of the edges.
   The following formula is used where $\tau_{ij}(t) = 1$:

$$\tau_{ij}(t+1) = (1-\rho) * \tau_{ij}(t) + \sum_{k=1}^{m} \Delta\tau_{ij}^{k}(t) \tag{8}$$

   In the formula $0 < \rho \leq 1$ is the rate of the pheromone evaporation. The amount of pheromone, which the ant puts on (i, j) edge:

$$\Delta\tau_{ij}^{k}(t) = \begin{cases} \frac{1}{L^k(t)} \; if\ the\ k.ant\ passes\ through\ the\ (i,j)\ \text{edge} \\ 0 \quad else \end{cases} \tag{9}$$

   where $L^k(t)$ is the length of the route of the $k.$ ant.

3. The depot part is generated randomly. Each ant makes their route (customer part), in the following way:
   3.1. Selecting a city randomly.

3.2. Then we calculate the following probabilities for all (i, j)cities where i is the last city visited by the ant so far, j is a city that has not been visited by the ant yet. The probability that the ant travel from $i.$ city to $j.$ city is:

$$p_{ij}^{k} = \frac{[\tau_{ij}(t)]^{\alpha} * [\eta_{ij}]^{\beta}}{\sum_{l \in N_i^k} [\tau_{il}(t)]^{\alpha} * [\eta_{il}]^{\beta}} \quad if\ j \in N_i^k \tag{10}$$

In the formula $\eta_{ij} = \frac{1}{d_{ij}}$, and $\tau_{ij}(t)$ gives the pheromone of the (i, j) node in the $t.$ iteration.

3.3. Based on the probabilities, the next city of the ant is chosen.
3.4. Repeat the 3.2.-3.3. steps, until all cities have been selected.

4. Once all the ants have completed their tour, the pheromone on the edges is updated. The (8) formula is used for the pheromone upgrade.

5. Storing the best ant (depot part and customer part) so far.

6. The 3.-5. steps are repeated until the stopping condition is not met. The algorithm returns with the best solution.

5.2. Ant Colony System

The algorithm consists of the following steps [20]:

1. Generating a path (depot and customer part) for each ant. The generation occurs randomly.

2. Initializing the pheromones of the edges.
The following formula is used, where $\tau_{ij}(t) = 1$:

$$\tau_{ij}(t+1) = (1-\rho) * \tau_{ij}(t) + \rho * \Delta\tau_{ij}^{gb}(t) \tag{11}$$

In the formula $0 < \rho \leq 1$ is the pheromone evaporation and $\Delta\tau_{ij}^{gb}(t) = \frac{1}{L^{gb}}$.

3. The depot part is generated randomly. Each ant makes its way (customer part), as follows:
3.1. We select a city randomly.
3.2. The following probabilities are calculated for all (i, j) cities, where i is the last city visited by the ant so far and j is a city that has not been visited by the ant yet. The equation (10) is used for the calculation for the probability.
3.3. Based on the probabilities, we choose the next city of the ant.
3.4. Perform a local pheromone update using the formula below:

$$\tau_{ij} = (1-\xi) * \tau_{ij} + \xi * {\tau_{ij}}^{0} \tag{12}$$

In the formula $0 < \xi < 1$ and ${\tau_{ij}}^{0}$ is the initial pheromone value, τ_{ij} is the pheromone content of the (i, j) edge. During the local pheromone update, we do not make the already selected edges promising to other ants.

3.5. Repeat 3.2.-3.4. steps until all cities have been selected.

4. If every ant makes its way (customer part) then we update the pheromone content of the edges. The Ant Colony System calls this as a global pheromone update because in this algorithm only the best ant can put pheromone. This is calculated with equation (11).

5. Storing the best solution so far (customer and depot part).

6. The 3.-5. steps are repeated until the stopping condition is not met. The algorithm gives the best solution.

5.3. MAX-MIN Ant System

The algorithm consists of the following steps [20]:

1. Determination of the τ_{min} and τ_{max} values. In our algorithm the $\tau_{min} = \frac{\tau_{max}}{2*n}$, the $\tau_{max} = \frac{1}{\rho * T^{gb}}$, where n indicates the number of customers and T^{gb} indicates the global best tour.

2. Depot and customer part generation for each ant. The generation occurs randomly.
3. The initialization of the pheromones of the edges. During the initialization, the following formula is used where $\tau_{ij}(t) = 1$:

$$\tau_{ij}(t+1) = (1-\rho) * \tau_{ij}(t) + \Delta\tau_{ij}^{best} \tag{13}$$

   In the formula (13) $\Delta\tau_{ij}^{best} = \frac{1}{L^{best}}$. Only the best ant puts pheromone. If the pheromone value decreases below the value of τ_{min} then the pheromone value will be τ_{min}, if the value grows above τ_{max}, then the value will be τ_{max}.

4. The depot part is generated randomly. Each ant makes the customer part, in the following way:
   4.1. We select a city randomly.
   4.2. The following probabilities are calculated for all (i, j) cities, where i is the last visited city of the ant, and j is a city that has not been visited by the ant yet. The probability that the ant travels from i. city to j. city gives the (10) formula.
   4.3. Based on the probabilities, we choose the next city of the ant.
   4.4. Repeat step 4.2.-4.3. until all cities have been selected.

5. If every ant moved (depot and customer) then we update the pheromone content of the edges. The equation (13) is used for updating the pheromone.

6. Storing the so far best solution.

7. Updating the τ_{min} and τ_{max} values.

8. The 4.-7. steps are repeated until the stopping condition is not met. The algorithm returns with the best solution (depot and customer part).

5.4. Elitist Strategy of Ant System

The algorithm consists of the following steps [21]:

1. Generating depot and customer part for each ant. The generation occurs randomly.
2. Initialization of the pheromone of the edges. The following formula is used where $\tau_{ij}(t) = 1$:

   $$\tau_{ij}(t+1) = (1-\rho) * \tau_{ij}(t) + \sum_{k=1}^{m} \Delta\tau_{ij}^{k}(t) + w * \Delta\tau_{ij}^{gb}(t) \tag{14}$$

   In the formula $\Delta\tau_{ij}^{gb}(t) = \frac{1}{L^{gb}}$, the $0 < \rho \leq 1$ is the pheromone evaporation, while the $w > 0$ controls the weight, the globally best solution in which extent affects the pheromone update at the edges which belong to the global best solution.
3. The depot part is generated randomly. Each ant makes its way, as the following:
   - 3.1. We select a city randomly.
   - 3.2. The following probabilities are calculated for all (i, j) cities, where i is the last visited city, j is a city that has not been visited by the ant. The probability that the ant goes from the i. city to the j. city determines the (10) formula.
   - 3.3. Based on the probabilities, we choose the next city of the ant.
   - 3.4. Repeat 3.2.-3.3. steps until all cities have been selected.
4. If each ant made its way (permutation) then we update the pheromone content of the edges. The equation (14) is used for the pheromone updating.
5. Storing the best solution so far (depot and customer part).
6. The 3.-5. steps are repeated until the termination condition is not met. The algorithm returns with the best solution.

5.5. Rank Based Version of Ant System

The algorithm consists of the following steps [20]:

1. The depot and customer part generation for each ant. The generation occurs randomly.
2. The initialization of the pheromone of the edges. During the initialization the following rules are used:
   - 2.1. Based on the tour of the global best ant, and based on the tour of the $(w-1)$ best ants in the iteration, the amount of deposited pheromone also depends on the rank of the ant (indicated with r). Only in the iteration $(w-1)$ best ant can deposit pheromone. The global best ant can put pheromone with w weight (with the maximal weight). The pheromone updating formula is the following, where $\tau_{ij}(t) = 1$:

     $$\tau_{ij}(t+1) = (1-\rho) * \tau_{ij}(t) + \sum_{r=1}^{w-1} (w-r) * \Delta\tau_{ij}^{r}(t) + w * \Delta\tau_{ij}^{gb}(t) \tag{15}$$

     In the formula $\Delta\tau_{ij}^{r}(t) = \frac{1}{L^{r}(t)}$ and $\Delta\tau_{ij}^{gb}(t) = \frac{1}{L^{gb}}$.
3. The depot part is generated randomly. Each ant makes the customer part, in the following way:

3.1. Selecting a city randomly.
3.2. The following probabilities are calculated for all (i, j) cities, where i is the last visited city of the ant and j is a city that has not been visited. The formula (10) gives the probability that the ant travels form city i. to city j.
3.3. Based on the probabilities the next city of the ant is chosen.
3.4. The 3.2.-3.3. steps are repeated until all cities are chosen.

4. If every ant constructs the path (customer part) then we update the pheromone content of the edges based on the route of the global best ant and based on the route of some best ants of the iteration. The amount of pheromone deposited depends on the rank of the ant (indicated by r). Only the $(w - 1)$ best ants in the iteration can deposit pheromone. The global best ant puts pheromone with w weight. The pheromone updating formula is equation (15).

5. Storing the so far best solution.

6. The 3.-5. steps are repeated until the termination condition is not met. The algorithm gives the best solution (depot and customer part) in the end.

6. RUNNING RESULTS

We tested the efficiency of the algorithms with Cordeau's benchmark data [22]. In the benchmark data, the number of vehicles per depot is 2, but we gave maximum of 4 vehicles per depot. In Table II the number of vehicles means the whole number of vehicles, not the number of vehicles per depot. The benchmark data were run ten times. Table II indicates the minimum solution, the maximum solution and the average solution of the runs. Based on the results, the Rank Based Version of Ant System gave the best performance. The second best was the Elitist Strategy of Ant System algorithm. The worst performance was given by MAX-MIN Ant System algorithm. The Figure 9. shows the solution (performed by the Rank Based Version of Ant System) of the Multi-Depot Vehicle Routing Problem with Time Window.

Table I.

The characters of the benchmark data

	Ds[4]	***NoVpD***[5]	***NoC***[6]	***NoD***[7]	***MDoR***[8]	***CCoV***[9]	***BNR***[10]
1	pr01	2	48	4	500	200	1083.98

[4] Dataset
[5] Number of vehicles per depot
[6] Number of customers
[7] Number of depots
[8] Maximum duration of the route
[9] Capacity constraint of the vehicles
[10] Best known result

Table II.

Results

		ACO[11]	AS[12]	ESAS[13]	MMAS[14]	RBVAS[15]
Minimum solution	**Route length**	1910.78	1790.71	1770.12	2129.59	*1601.94*
	Running time (min)	4.69	3.56	3.98	2.63	5.55
	Number of vehicles	13	11	13	12	12
Maximum solution	**Route length**	2324.04	1994.24	1910.35	2370.44	*1861.50*
	Running time (min)	2.39	3.08	4.46	2.97	4.83
	Number of vehicles	14	15	14	15	13
Average solution	**Route length**	2141.08	1900.35	1857.74	2272.68	*1721.06*
	Running time (min)	3.69	3.84	3.95	3.40	5.40
	Number of vehicles	14	13	13.4	13.3	12.6

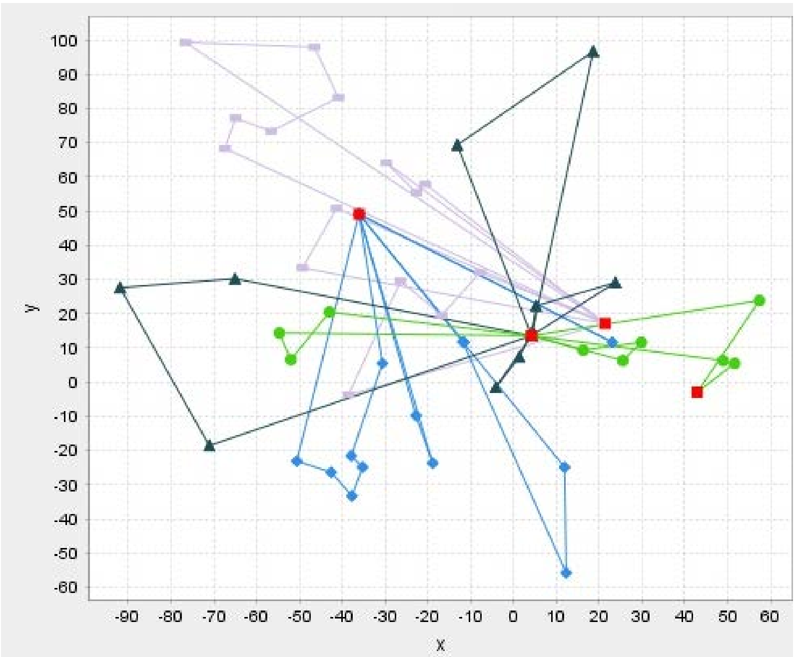

Figure 9. The solution of Multi-Depot Vehicle Routing Problem with Time Window

[11] Ant Colony System
[12] Ant System
[13] Elitist Strategy of Ant System
[14] MAX-MIN Ant System
[15] Rank Based Version of Ant System

7. SUMMARY

In this article, the Ant Colony Optimization algorithms are described. The ACO algorithms are heuristic algorithms and simulate the ants while searching for food. We presented the following ACO algorithms: Ant Colony System, Ant System, Elitist Strategy of Ant System, MAX-MIN Ant System, Rank Based Version of Ant System. The Multi-Depot Vehicle Routing Problem with Time Windows (MDVRPTW) was solved with the ACO algorithms. The MDVRPTW is a logistical problem, which can simulate the in-plant and the out-plant material handling, or a complex system. In the case of MDVRPTW, the position of the depots is known in advance. Also, the number and the capacity constraint of the vehicles are known in advance. The position, the demand and the time window of the customer are also known in advance. The vehicles transport the goods from the depot to the customer. The objective function of the problem is the minimization of the transportation cost while following the above-mentioned constraints. The performance of the algorithms was tested with Cordeau's benchmark data. Based on the running result the Rank Based Version of Ant System gave the best solutions. This optimization solution can be added to different simulation methods to improve its performance [23].

ACKNOWLEDGEMENTS

"The described article/presentation/study was carried out as part of the EFOP-3.6.1-16-00011 "Younger and Renewing University – Innovative Knowledge City – institutional development of the University of Miskolc aiming at intelligent specialisation" project implemented in the framework of the Szechenyi 2020 program. The realization of this project is supported by the European Union, co-financed by the European Social Fund."

REFERENCES

[1] Blum, C., & Sampels, M. (2004). An ant colony optimization algorithm for shop scheduling problems. *Journal of Mathematical Modelling and Algorithms*, *3*(3), 285-308. https://doi.org/10.1023/B:JMMA.0000038614.39977.6f

[2] Tian, J., Yu, W., & Xie, S. (2008). An ant colony optimization algorithm for image edge detection. In *Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence)*. 751-756. https://doi.org/10.1109/CEC.2008.4630880

[3] T'kindt, V., Monmarché, N., Tercinet, F., & Laügt, D. (2002). An ant colony optimization algorithm to solve a 2-machine bicriteria flowshop scheduling problem. *European Journal of Operational Research*, *142*(2), 250-257. https://doi.org/10.1016/S0377-2217(02)00265-5

[4] Tao, W., Jin, H., & Liu, L. (2007). Object segmentation using ant colony optimization algorithm and fuzzy entropy. *Pattern Recognition Letters*, *28*(7), 788-796. https://doi.org/10.1016/j.patrec.2006.11.007

[5] Maier, H. R., Simpson, A. R., Zecchin, A. C., Foong, W. K., Phang, K. Y., Seah, H. Y., & Tan, C. L. (2003). Ant colony optimization for design of water distribution systems. *Journal of water resources planning and management*, *129*(3), 200-209. https://doi.org/10.1061/(ASCE)0733-9496(2003)129:3(200)

[6] Merkle, D., Middendorf, M., & Schmeck, H. (2002). Ant colony optimization for resource-constrained project scheduling. *IEEE transactions on evolutionary computation*, *6*(4), 333-346. https://doi.org/10.1109/TEVC.2002.802450

[7] Dorigo, M., & Gambardella, L. M. (1997). Ant colonies for the travelling salesman problem. *Biosystems*, *43*(2), 73-81. https://doi.org/10.1016/S0303-2647(97)01708-5

[8] Kong, M., Tian, P., & Kao, Y. (2008). A new ant colony optimization algorithm for the multidimensional knapsack problem. *Computers & Operations Research, 35*(8), 2672-2683. https://doi.org/10.1016/j.cor.2006.12.029

[9] Bell, J. E., & McMullen, P. R. (2004). Ant colony optimization techniques for the vehicle routing problem. *Advanced engineering informatics, 18*(1), 41-48. https://doi.org/10.1016/j.aei.2004.07.001

[10] Barán, B., & Schaerer, M. (2003, February). A Multiobjective Ant Colony System for Vehicle Routing Problem with Time Windows. *Applied informatics* 97-102.

[11] Yu, B., Yang, Z. Z., & Xie, J. X. (2011). A parallel improved ant colony optimization for multi-depot vehicle routing problem. *Journal of the Operational Research Society, 62*(1), 183-188. https://doi.org/10.1057/jors.2009.161

[12] Aghdam, M. H., Ghasem-Aghaee, N., & Basiri, M. E. (2009). Text feature selection using ant colony optimization. *Expert systems with applications, 36*(3), 6843-6853. https://doi.org/10.1016/j.eswa.2008.08.022

[13] Parpinelli, R. S., Lopes, H. S., & Freitas, A. A. (2002). Data mining with an ant colony optimization algorithm. *IEEE transactions on evolutionary computation, 6*(4), 321-332. https://doi.org/10.1109/TEVC.2002.802452

[14] Renaud, J., Laporte, G., & Boctor, F. F. (1996). A tabu search heuristic for the multi-depot vehicle routing problem. *Computers & Operations Research, 23*(3), 229-235. https://doi.org/10.1016/0305-0548(95)O0026-P

[15] Brandão, J. (2004). A tabu search algorithm for the open vehicle routing problem. *European Journal of Operational Research, 157*(3), 552-564. https://doi.org/10.1016/S0377-2217(03)00238-8

[16] Gaudioso, M., & Paletta, G. (1992). A heuristic for the periodic vehicle routing problem. *Transportation Science, 26*(2), 86-92. https://doi.org/10.1287/trsc.26.2.86

[17] Dethloff, J. (2001). Vehicle routing and reverse logistics: the vehicle routing problem with simultaneous delivery and pick-up. *OR-Spektrum, 23*(1), 79-96. https://doi.org/10.1007/PL00013346

[18] Desrochers, M., Desrosiers, J., & Solomon, M. (1992). A new optimization algorithm for the vehicle routing problem with time windows. *Operations research, 40*(2), 342-354. https://doi.org/10.1287/opre.40.2.342

[19] Crainic, T. G., Mancini, S., Perboli, G., & Tadei, R. (2011). Multi-start heuristics for the two-echelon vehicle routing problem. In *European Conference on Evolutionary Computation in Combinatorial Optimization*. 179-190. https://doi.org/10.1007/978-3-642-20364-0_16

[20] Stützle, T., & Dorigo, M. (1999). ACO algorithms for the traveling salesman problem. *Evolutionary algorithms in engineering and computer science*, 163-183.

[21] Bullnheimer, B., Hartl, R. F., & Strauss, C. (1997). *A new rank based version of the Ant System.* A computational study.

[22] Cordeau's benchmark data. http://neo.lcc.uma.es/vrp/vrp-instances/multiple-depot-vrp-with-time-windows-instances/

[23] Kaczmar, I. (2019) *Komputerowe modelowanie i symulacje procesów logistycznych w środowisku FlexSim.* Wydawnictwo Naukowe PWN. ISBN 978-83-01-20544-7