

Ant Colony Algorithms for the Vehicle Routing Problem with Time Window, Period and Multiple Depots

Anita Agárdi (<https://orcid.org/0000-0001-9148-1214>)¹, László Kovács (<https://orcid.org/0000-0003-2703-7228>)¹, Tamás Bányai (<https://orcid.org/0000-0002-0229-4781>)²

¹Institute of Informatics, University of Miskolc, Egyetemváros H-3515 Miskolc, Hungary. E-mail: agardiana@iit.uni-miskolc.hu, kovacs@iit.uni-miskolc.hu

²Institute of Logistics, University of Miskolc, Egyetemváros H-3515 Miskolc, Hungary. E-mail: alttamas@uni-miskolc.hu

Vehicle Routing Problem is a common problem in logistics, which can simulate in-plant and out-plant material handling. In the article, we demonstrate a Vehicle Routing Problem, which contains period, time window and multiple depots. In this case, customers must be served from several depots. The position of the nodes (depots and customers), the demand and time window of the customers are known in advance. The number and capacity constraint of vehicles are predefined. The vehicles leave from one depot, visit some customers and then return to the depot. The above-described vehicle routing is solved with construction algorithms and Ant Colony algorithms. The Ant Colony algorithms are used to improve random solutions and solutions generated with construction algorithms. According to the test results the Elitist Strategy Ant System and the Rank-Based Version of Ant System algorithms gave the best solutions.

Keywords: vehicle routing problem, ant colony algorithms, improvement algorithms, construction algorithms

1 Introduction

The most important task of the logistic is delivering the right goods at the right time to the right place. The Vehicle Routing Problem (VRP) models these processes. Over the years, there have been many variations of the problem, modelling in-plant, out-plant material handling or a complete supply chain. The article presents the Vehicle Routing Problem with Time Window, Period and Multiple Depots which is suitable for in-plant and also for out-plant material handling. For the problem, we know the positions of depot and customers. Customers also have a need for goods that must be met within a time window. The vehicles start from one warehouse (depot) to serve the needs of the customers and then return to the warehouse (depot). During the task, customers do not have to be visited once, but several times within a period. The goal is to minimize the route taken by vehicles.

2 Vehicle Routing Problem

The VRP is a logistical problem, which deals with the delivery and picks up the products. In case of the basic problem the position of one depot, the position and demand of the customers, the number and capacity constraint of the transportation units (vehicles) are known in advance. The vehicles start their route at the depot. After that, visits some customers (satisfy their demand) and then return to the depot. The objective function is the minimization of the length of the route. Over the years, many VRP variations have

been developed that adapt to each logistics task. In the following, some type of Vehicle Routing Problems are described.

Capacity Constrained VRP [1] is a type when the vehicles have some capacity limit. The Single Depot VRP [2] is a VRP when the vehicles leave from a single depot. In case of Multi-Depot VRP [2], there are multiple depots. The vehicles leave the route from a depot, and after visited some customers, the vehicles return to the depot, from which they leaved. In the case of VRP with Inter-Depot Routes [3], the vehicles can return to any of the depots. Open VRP [4] is a type when the vehicles do not have to return to the depots. In the case of Environmentally Friendly VRP [5], environmentally-friendly vehicles transport the products and in the case of Electric VRP [6] electric vehicles. When vehicles do not only deliver products from the depot to the customers but also collect products from the customers to the depot, we call it Pick-up and Delivery VRP [7]. In the case of Periodic VRP [8], the customers must be visited periodically. In the case of VRP with Time Window [9] the customers have a time window. The time window can be hard [10] and soft [11]. In the case of VRP with Soft Time Window [11] the customers can be visited outside of the time window, but in this case, we get penalty points. In the case of VRP with Stochastic Demand [12], the probability of the demands is known in advance. In the case of VRP with Fuzzy Demands [13], the demands are fuzzy numbers. In the case of Cumulative VRP [14] not the minimization of the length of

the route, but the minimization of the waiting time is the objective function. In the case of VRP with Cross-Docking [15] first the products are picked up from the customers, and after all products are picked up, the products are transported to other customers. In the case of Site-Dependent VRP [16] certain customers can only be visited by certain types of vehicles. Selective VRP [17] is a VRP when not all customers must be visited. Those customers are visited who are the best in terms of profit. In the case of Two-Echelon VRP [18] the products are delivered from the depot to intermediate locations (satellites) and then from the intermediate locations to the customers. VRP with Traffic Jams [19] also takes into account the traffic factors. Rechargeable VRP [20] is a type when electric vehicles satisfy the demands of the customers. In this case, the vehicles must be recharged. VRP with Perishable Food Products Delivery [21] is a case when perishable foods are delivered.

Vehicle Routing Problems are optimization tasks belonging to the NP hard category. VRP is one of the best known optimization tasks (which also affects production), but of course other manufacturing optimization tasks can be found in the literature, such as production process planning [22], production flow [23].

3 Vehicle Routing Problem with Time Window, Period and Multiple Depots

In case of the Vehicle Routing Problem with Time Window, Period and Multiple Depots the positions of the customers and depots to be visited are known in advance. Customers are also given with special demands and a time window, which means when their demands can be served. There is also a service period. All customers have been given the number of times they should be served per period. There are also possible period-time service combinations. One of these can be selected. The number of vehicles are also given, as is the capacity limit of the vehicles. The length of vehicle routes is also maximized, and vehicles can deliver a single type of goods to customers. Figure 1 shows an example of a problem. The figure shows two depots, denoted by D1 and D2. From the D1 depot, 400 items can be transported daily and 400 items from the D2 depot. The number of customers is 9; the customers are indicated with numbers. Customers have time windows and demands. The first customer has 30 demand and the [10, 50] time window the second one has 10 demand, and the [30, 70] time window, the third has 75 demand, and [60, 120] time window, the fourth is 75 demand and [30, 150] time window, the fifth is 10 demands and [20, 100] time window, the sixth is 200 demand and [40, 100] time window the ninth has 100 demand and [40, 80] time win-

dow. The example shows a four-day period. Blue-labelled customers should be visited daily (four times per period), orange-labelled every two days (twice per period), and white candidates four days (once per period). In this example, two vehicles visit customers. The capacity limit for vehicles is 300 units. Vehicles start from the D1 or D2 depots, visit some customers (serving the customers' demands within the time window) and return to the depot. In Figure 1, on the 1st day, the first vehicle starts from the D1 depot, and then visits the first, then the 8th, then the 4th customer, and then returns to the D1 depot. The second vehicle starts from the D2 depot, then visits the customer 5, then returns to the customer 3 and then to the D2 depot. On day 2, the first vehicle starts from the D1 depot and then visits customer 1, then customer 2 and 4, and then returns to the D1 depot. On the day 2, the second vehicle starts from the D2 depot, and then visits the customer 5, then the customer 9 and 3, then return to the D2 depot. On day 3, the first vehicle starts from the D1 depot and then visits customer 1, then customer 8 and 4, and then returns to the D1 depot. The second vehicle starts from the D2 depot, then visits customer 5 and 7, and then returns to the D2 depot. On day 4, the first vehicle starts from the D1 depot, then customer 1 and 2 are visited, and then returns to the D1 depot. Then, starting from the D1 depot, visits customer 6 and 4 and returns to the D1 depot. The second vehicle starts from the D2 depot, then visits customer 5 and 9, and then returns to the D2 depot.

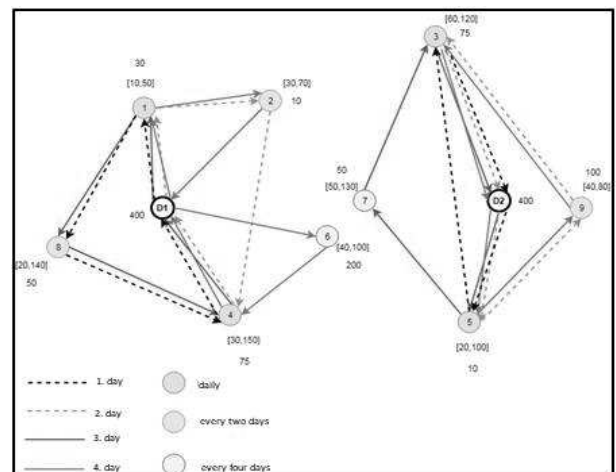


Fig. 1 Vehicle Routing Problem with Time Window, Period and Multiple Depots

3.1 The mathematical model of the Vehicle Routing Problem with Time Window, Period and Multiple Depots

In this subsection, we present the mathematical model of the Vehicle Routing Problem with Time Window, Period and Multiple Depots [24].

The following indices were used to describe our model:

i: customer index
j: customer index
r: depot index
l: period day index
k: vehicle index

\mathbb{W} : index set of the depots, where $\mathbb{W} = \{1, 2, \dots, m\}$, $|\mathbb{W}| = m$, so the number of depots is indicated with m

\mathbb{V} : index set of the customers, where $\mathbb{V} = \{m + 1, m + 2, \dots, m + n\}$, $|\mathbb{V}| = n$, so the number of customers is indicated with n

\mathbb{N} : index set of positions. The positions consists all depots and all customers, so the index set of the positions is the union of the index set of the depots and the index set of the customers:

$$\mathbb{N} = \mathbb{W} \cup \mathbb{V} \quad (1)$$

\mathbb{F} : index set of the vehicles, $\mathbb{F} = \{1, 2, \dots, p\}$, $|\mathbb{F}| = p$, so the number of vehicles is indicated with p . The index set of the vehicles is the union of the index sets of each outgoing-incoming vehicles of the depots:

$$\mathbb{F} = \bigcup_{r=1}^m \mathbb{F}_r \quad (2)$$

\mathbb{L} : index set of the possible visits within the service period, where $\mathbb{L} = \{1, 2, \dots, t\}$, $|\mathbb{L}| = t$, so the number of possible visits within the service period is indicated with t .

\mathbb{C} : set of capacity constraints of the vehicles, where $\mathbb{C} = \{c_1, c_2, \dots, c_p\}$. Each vehicle has a capacity limit for the products to be transported. The capacity limit

$$x_{ij}^{kl} = \begin{cases} 1, & \text{if vehicle } k \text{ in day } l \text{ travels from node } i \text{ to } j \\ 0, & \text{else} \end{cases} \quad (3)$$

The starting of the service time of the customer i in the day l with the vehicle k .

$$y_i^{kl} \in \mathbb{R}^+ \quad y_i^{kl} \in [a_i, b_i - s_i] \quad (4)$$

where i means the index of the node, k means the index of the vehicle and l means the period day index.

The objective function, which must be minimized, can be written in the following way:

$$Z = \sum_{i=1}^{m+n} \sum_{j=1}^{m+n} \sum_{k=1}^p \sum_{l=1}^t d_{ij} x_{ij}^{kl} \quad (5)$$

in the equation d_{ij} means the distance between node i and j , x_{ij}^{kl} is the decision variable of the traveling of vehicle k in day l from node i to j directly. In this equation all nodes must be considered (not only the customers, but the depots also be included). All vehicles and all periods must be taken into account in the amount.

Our system contains 9 constraints, which are presented below.

Constraint 1: One vehicle can visit each customer

of the vehicles must be respected.

\mathbb{DL} means the set of the maximum length of the trip, where $\mathbb{DL} = \{dl_1, dl_2, \dots, dl_p\}$. Each vehicle has a limit for the length of their trip. This limit must be also respected.

q_i : the demand of customer i . Each customer can have one product demand, because there is one type of product in our system.

s_i : the service time of customer i . Each customer can have a service time. Service time can mean the unloading time of products from the vehicles, the administration time of the products etc.

v_i : the frequency of visits customer i in a period. The customers must be visited periodically. The frequency of visits is an important component; it determines the total supply of products to each customer.

$[a_i, b_i]$: the time window of the customer i . The demand of the customers must be satisfied within a time window, where a_i means the earliest service time, and b_i is the latest service time. The time window varies from customer to customer.

d_{ij} : the distance between node i and j

M : high constant for the sub tour elimination

Z : the value of the objective function. Our objective function means the route minimization.

A decision variable must also be defined to define constraints and objective function. The decision variable can be written with the following way:

one day and after that visits the following customer.

$$\sum_{i=1}^{m+n} \sum_{k=1}^p x_{ij}^{kl} \leq 1 \quad \forall j \in \mathbb{V}, \forall l \in \mathbb{L} \quad (6)$$

$$\sum_{j=1}^{m+n} \sum_{k=1}^p x_{ij}^{kl} \leq 1 \quad \forall i \in \mathbb{V}, \forall l \in \mathbb{L} \quad (7)$$

The first equation of the constraint shows that the sum of the number of input edges is 1 or 0 per period. This means, that a customer can be visited by only one vehicle in a period day. The second equation of the constraint shows that the sum of the number of outgoing edges is 1 or 0 per period.

Constraint 2: Vehicles leave a depot and then it must arrive at the same depot after visiting customers.

$$\sum_{j=m+1}^{m+n} x_{ij}^{kl} = 1 \quad \forall k \in \mathbb{F}, \forall i \in \mathbb{W}, \forall l \in \mathbb{L} \quad (8)$$

$$\sum_{i=m+1}^{m+n} x_{ij}^{kl} = 1 \quad \forall k \in \mathbb{F}, \forall j \in \mathbb{W}, \forall l \in \mathbb{L} \quad (9)$$

The first equation demonstrates the incoming edges of the depots, and the second equation means the outgoing edges.

Constraint 3: Route continuity constraint is an important constraint. If a vehicle visits a node, after the visitation it must leave that.

$$\sum_{i=1}^{m+n} x_{ir}^{kl} = \sum_{j=1}^{m+n} x_{rj}^{kl} \quad \forall k \in \mathbb{F}, \forall r \in \mathbb{V}, \forall l \in \mathbb{L} \quad (10)$$

The equation means, that the incoming edges of each nodes must be equal with the outgoing edges of each nodes. This constraint concerns any nodes, vehicles and periods.

Constraint 4: Time window constraint is an important constraint; it determines the order of services of vehicles. The product demand of the customer cannot be satisfied before and after the time window. Each customer can have own time window.

$$y_i^{kl} + s_i - y_j^{kl} \leq M(1 - x_{ij}^{kl}) \quad \forall k \in \mathbb{F}, \forall i \in \mathbb{N}, \forall j \in \mathbb{V}, \forall l \in \mathbb{L} \quad (13)$$

In the equation M means a big positive integer

Constraint 6: The capacity constraint of the vehicle can be also taken into account.

$$\sum_{i=m+1}^{m+n} q_i \sum_{j=1}^{m+n} x_{ij}^{kl} \leq C^l \quad \forall k \in \mathbb{F}, \forall l \in \mathbb{L} \quad (14)$$

In the equation q_i means the product demand of customer i , x_{ij}^{kl} is the decision variable, and C^l is the capacity limit of the vehicle. Based on the equation each vehicle must comply with its capacity limit in each period.

Constraint 7: The maximum length of the route of the vehicle may not exceed:

$$\sum_{i=1}^{m+n} \sum_{j=1}^{m+n} d_{ij} x_{ij}^{kl} \leq dl_k \quad \forall k \in \mathbb{F}, \forall l \in \mathbb{L} \quad (15)$$

This equation means that the route of all vehicles can be maximized in each period.

Constraint 8: The system does not contain any inter-depot routes.:

$$x_{ij}^{kl} = 0 \quad \forall k \in \mathbb{F}, \forall i, j \in \mathbb{W}, \forall l \in \mathbb{L} \quad (16)$$

This equation means that there is no depot-depot route. Considering the former constraint, it also follows that there is only a depot-customer, customer-customer, and customer-depot path in the system.

Constraint 9: The number of visits of each customer per the whole period can be also taken into account.

$$\sum_{i=1}^{m+n} \sum_{k=1}^p \sum_{l=1}^t x_{ij}^{kl} = v_i \quad \forall j \in \mathbb{V} \quad (17)$$

$$\sum_{j=1}^{m+n} \sum_{k=1}^p \sum_{l=1}^t x_{ij}^{kl} = v_j \quad \forall i \in \mathbb{V} \quad (18)$$

In the equation, v_i means the frequency of visits of customer i . The sum of the incoming edges of each customer must be equal with the frequency of visits of each customer.

Customer's visit days should be out of the customer set assigned to customers.

3.2 The application of the Vehicle Routing Problem with Time Window, Period and Multiple Depots in transportation systems

We have chosen the Vehicle Routing Problem with Time Window, Period and Multiple Depots because

$$a_i \leq y_i^{kl} \quad \forall k \in \mathbb{F}, \forall i \in \mathbb{N}, \forall l \in \mathbb{L} \quad (11)$$

$$y_i^{kl} + s_i \leq b_i \quad \forall k \in \mathbb{F}, \forall i \in \mathbb{N}, \forall l \in \mathbb{L} \quad (12)$$

In the equation a_i means the earliest time of the service, and b_i means the latest time of the service. y_i^{kl} means the starting of the service time of the customer i in the day l with the vehicle k . s_i indicates the service time of customer i . The service can be performed within the time window.

Constraint 5: Sub-tour elimination provides a feasible tour.

many logistics tasks can be modelled with it. In the following, the below transportation problems are detailed:

- **Food delivery:** delivery of food to shops. It consists of the delivery of daily food (for example bakery), perishable foods (for example dairy products, meats), non-perishable foods and drinks (conserves, frozen foods, soft drinks, alcohol drinks). Food delivery is delivery from the producer to the store.
- **Delivery of raw materials:** The raw material is delivered from the producer to the factory.
- **Waste collection:** waste transportation means collecting household garbage and industrial waste.
- **Postal services:** this includes the delivery of letters and newspapers from the post office to the homes.
- **In-plant material handling:** means transport from the warehouse to the production line and then transport from the production line to the warehouse.
- **Delivery of money:** it means that the used money is collected from shops and banks and then delivered to the banknote manufacturer. Or vice versa, the money is transported from the banknote manufacturer to the shops and banks.
- **Patient transport:** means that patients are transported to the hospital.
- **Maintenance:** maintenance companies perform maintenance activities at each node.

Vehicles have a capacity limit for the products to be transported. During many transportation tasks, vehicles can leave from multiple depots and then return

there after visited the customers (for example in the case of transportation of food or durable product). But even in the case of in-plant material handling, we can choose which warehouses to depart from during the transport between the production units and the warehouses. The period is a very important factor for transportation tasks. In many cases, it may be necessary to serve the customer's demand for goods periodically. In almost all transportation activities, for example, food transportation, maintenance activity, trash transportation, certain postal deliveries (newspaper, flyer), and money transportation. Periodicity is also important during in-plant transportation because there must be a continuous flow of material between production and the warehouse. In some cases, there is no periodicity, such as emergency patient transport or travel agency ad hoc trips. The time window is an important factor in out-plant material handling. The time window means that customer demands can only be served within a certain time interval. This constraint can be occurred in case of food and durable products transportation, in case of maintenance tasks, postal deliveries, trash transportation, and money transportation. During in-plant delivery, the time window can also be important, if the raw material is brought to the production line, the raw material must arrive at the production line within a certain time interval. It is also important when transporting the product from the production line that the finished product is delivered within a certain time interval. With these in mind, the Vehicle Routing Problem with Time Window, Period and Multiple Depots is a general task that is often required during different types of transportation systems.

4 Construction Algorithms

In this section, we describe the implemented construction algorithms. The construction algorithms construct one solution. They give locally the best steps, but with the exclusive usage of these algorithms usually, we can not get the global optimum. The running time of these algorithms is low. During these algorithms we construct a tour (TSP is solved), only the evaluation of the permutation is different from the TSP problem (this is called route-first cluster second technique [25]).

Nearest Neighbour algorithm

In the algorithm, we always visit the closest city to the last selected city. So, the algorithm takes the best steps locally, but it can easily skip points that can only be crawled at a high cost. [26]

Insertion Heuristics

The Nearest Insertion, Cheapest Insertion, Farthest Insertion and Arbitrary Insertion algorithm

belongs to the Insertion Heuristics group. The Nearest Insertion algorithm always chooses the city that is closest to the tour. The Cheapest Insertion algorithm always selects the city where the insertion cost is minimal. The Arbitrary Insertion algorithm selects the city arbitrary. The Farthest Insertion always chooses the city that is farthest to the tour. [27]

Greedy

The algorithm continuously builds the path from the individual edges, always selecting the shortest edge that has not yet been selected, which does not develop an n -peak tour (where n is the number of cities) and where the peak level is not bigger than two. [28]

5 Ant Colony Algorithms

Ant Colony Optimization is one of the best known metaheuristics after Genetic Algorithm [29], Simulated Annealing [30] and Particle Swarm Optimization [30].

Ant Colony Optimization is based on ants' behaviour. Ants release a hormone (pheromone) that is used to mark the path to food. This is also perceived by other ants. The more ants traversing the route, the higher the hormone content of the road, and the lesser use of pheromone on less-used roads. Thus, ants will choose roads with higher pheromone content. For shorter routes, ants can place several "markers" per unit section, so the ants will probably choose the shorter path. [31]

ACO is a generic term and includes many specific algorithms. Examples of such algorithms include the Ant System (AS), Ant Colony System (ACS), MAX-MIN Ant System (MMAS), and Rank Based Version of Ant System (RBVAS).

Ant Optimization was initially applied on graphs. In the algorithm, ants construct paths from node to node. Finding solutions for ants is assisted by pheromone paths and heuristic information. You can associate on the $\tau_{ij}(t)$ pheromone quantity with the edge of the graph (i, j) , which may change continuously during the algorithm. t denotes the iteration variable. [31]

At first, ants randomly select their nodes. The choice of the next node of the ant depends on the amount of pheromone in the edge between i and j and the length of the edge. The ants will then always choose the node that is close to them, and the connecting edge has high pheromone content. Because of the regularity of a solution, ants have to know which node they have touched to construct a path. After every ant has constructed its way, the pheromones are updated. The algorithm does this by reducing the pheromone content of the edges with a constant value, and then placing the ants on the pheromone at each edge they went through. When edges are refreshed, edges on which many ants travelled and/or form a shorter path will

receive a greater amount of pheromone. [31]

5.1 Ant System

Constructing the tour [31]

Ants select their first node at random and apply a probability-based decision rule in each path preparation step. The probability of ant k , that is in the city i , and goes to city j , in iteration t :

$$p_{ij}^k = \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} [\tau_{il}(t)]^\alpha [\eta_{il}]^\beta} \quad \text{if } j \in N_i^k \quad (19)$$

In the formula, $\eta_{ij} = 1$ i.e. the reciprocal of the distance between the two nodes, and $\tau_{ij}(t)$ the pheromone content of (i, j) in iteration t . α and β determine the effect of the amount of pheromone and

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{1}{L^k(t)} & \text{if ant } k \text{ goes through the } (i, j) \text{ edge} \\ 0 & \text{else} \end{cases} \quad (21)$$

In the formula, $L^k(t)$ is the length of the path of the k . ant, and $\tau_{ij}^k(t)$ the pheromone content of (i, j) edge in the t . iteration.

5.2 Ant Colony System

The Ant Colony System algorithm was developed to develop the Ant System algorithm. ACS differs from AS in three ways: [31]

1. ACS uses a much more powerful operational selection rule.
2. The algorithm adds pheromone only to the edges that belong to the best global solutions.
3. Whenever the ants use the (i, j) edge, so when goes from city i . to j . lose a pheromone from the edge.

Construction the path:

Creating the path is the same as the Ant System algorithm's path constructing strategy.

Updating Global Pheromone [31]

In the ACS algorithm, only the globally best ant can take pheromone after every iteration. Pheromone update formula:

$$\tau_{ij}(t+1) = (1 - \rho) * \tau_{ij}(t) + \rho * \Delta\tau_{ij}^{gb}(t) \quad (22)$$

In the formula $\Delta\tau_{ij}^{gb}(t) = \frac{1}{L^{gb}} \cdot \tau_{ij}(t)$ is the pheromone content of the (i, j) edge in iteration t .

Local Pheromone Update [31]

In addition to the global pheromone update, the Ant Colony System algorithm also performs a local

$$\tau_{ij}(t+1) = (1 - \rho) * \tau_{ij}(t) + \sum_{k=1}^m \Delta\tau_{ij}^k(t) + w * \Delta\tau_{ij}^{gb}(t) \quad (24)$$

In the formula $\Delta\tau_{ij}^{gb}(t) = \frac{1}{L^{gb}} \cdot \tau_{ij}(t)$ is the pheromone content of the (i, j) edge in iteration t , $w > 0$ weight controls how globally the best solution influences the pheromone update on edges that are the best.

distance between nodes. N_i^k denotes the nodes that the ant has not yet been affected.

Updating Pheromones [31]

When the ants constructed their way, we update the pheromones. In the first step, the amount of pheromone must be reduced by a constant value, and the ants will "mark" the edges they visit. Pheromone update formula:

$$\tau_{ij}(t+1) = (1 - \rho) * \tau_{ij}(t) + \sum_{k=1}^m \Delta\tau_{ij}^k(t) \quad (20)$$

in the formula $0 < \rho \leq 1$ is the rate of evaporation of the pheromone, and τ_{ij} is the pheromone content of the (i, j) edge in the t . iteration.

pheromone update. Once they have passed through the route, they will do so immediately. The formula of the local pheromone update:

$$\tau_{ij} = (1 - \xi) * \tau_{ij} + \xi * \tau_{ij}^0 \quad (23)$$

In the formula $0 < \xi < 1$. τ_{ij}^0 is the initial pheromone content. During the local update, the edges that have already been selected are not as attractive to the following ants.

5.3 MAX-MIN Ant System

The MMAS is a further development of the Ant System algorithm. In the MMAS, the paths are similar to those of AS. The differences are: [31]

1. Only one ant can put pheromone after each iteration.
2. Limiting the strength of pheromone paths to an $[\tau_{min}, \tau_{max}]$ interval where $\forall \tau_{ij} (\tau_{min} \leq \tau_{ij} \leq \tau_{max})$ avoids stagnation.
3. The initial value of the pheromone paths will be the highest value of the interval. This at the beginning of the algorithm provides faster detection of the search space.

5.4 Elitist Strategy of Ant System

Elitist Strategy of Ant System has been developed to improve the Ant System algorithm. The idea is to place the emphasis on the best route so far when updating the pheromone. The pheromone update is done using the following formula: [32]

5.5 Rank Based Version of Ant System

The Rank Based Version of Ant System algorithm is another modification of the Ant System algorithm. This algorithm is based on the Elitist Strategy of Ant System because it updates pheromones based on the

globally best route. Also, a certain number of ants that have travelled the shortest route in a given iteration can put a pheromone. In the algorithm, ants are ranked by their path, i.e. $(L_1(t) \leq L_2(t) \leq \dots \leq L_m(t))$. The pheromone level depends on the rank of the ant (which is denoted by w). Only the best ($w -$

$$\tau_{ij}(t+1) = (1 - \rho) * \tau_{ij}(t) + \sum_{r=1}^{w-1} (w - r) * \Delta\tau_{ij}^r(t) + w * \Delta\tau_{ij}^{gb}(t) \quad (25)$$

In the formula $\Delta\tau_{ij}^r(t) = \frac{1}{L^r(t)}$ and $\Delta\tau_{ij}^{gb}(t) = \frac{1}{L^{gb}}$, $\tau_{ij}(t)$ if the pheromone content of the (i, j) edge in iteration t .

6 Implementation and evaluation

In this section, we present our representation technique and evaluation for the Vehicle Routing Problem with Time Window, Period and Multiple Depots.

During the problem, we introduced a two-part representation technique. This two-part technique can be seen in Figure 2.

Suppose we have 9 customers. In addition, a four-day period was defined. The first part of the figure (permutation part) represents the order of the customers, i. e. 1, 2, 6, 4, 8, 3, 9, 5, 7. Period part (day 1) indicates customers to visit on the first day, period part (day 2) indicates customers to visit on the second day, and so on. Based on these, in the first day the customers should be visited with the following order: 3, 2, 4, 1, 5, 9, while in the second day should be visited in the following order: 5, 7, 4, 3, 8, 1.

The permutation part								
1	2	6	4	8	3	9	5	7
The period part (1. day)								
3	2	4	1	5	9			
The period part (2. day)								
5	7	4	3	8	1			
The period part (3. day)								
4	8	3	5	1				
The period part (4. day)								
4	5	9	3	6	2	1		

Fig. 2 Representation of the problem

Permutation and period parts are evaluated by taking the order of the customers in the permutation section and seeing the day of the period part. According to this, we list the customers for a given day. Results after the evaluation are presented in Figure 3. In the first day the following customers must be visited, in that order: 1, 2, 4, 3, 9, 7. In the second day the

1) ant can put pheromone in the given iteration. The globally (the previous) best ant can put pheromone with the highest weight. The best ants of the iterations can always put pheromone with weight $\max\{0, w - r\}$. The update formula for this modified pheromone is: [31]

following customers must be visited: 1, 8, 4, 3, 7, 5, in the third day the 1, 8, 4, 3, 5, in the fourth day 1, 2, 6, 4, 3, 9, 5.

The period part (1. day)					
1	2	4	3	9	5
The period part (2. day)					
1	8	4	3	7	5
The period part (3. day)					
1	8	4	3	5	
The period part (4. day)					
1	2	6	4	3	9

Fig. 3 Results of the evaluation

The next step in the evaluation is to take each day one after the other - for the cities that always belong to that day - and do the following:

1. Starting from the first element of the permutation section, we continually collect the elements (cities) until the capacity limit or the time window barrier.

2. The order of the received cities is assigned to the depot closest to the first and last cities.

7 Test results

The test runs were examined for four data sets. The data sets contained 48 customers. We tested the effectiveness of all the Ant Colony Optimization techniques we mentioned. The following abbreviations are used: the AS+R means that the Ant System algorithm improves only random solutions, the AS+C,R means that the Ant System algorithm improves the solutions of the construction algorithms and randomly generated solutions, the ACS+R means that the Ant Colony System algorithm improves only randomly generated solutions, the ACS+C,R means that the Ant Colony System algorithm improves the solution of the construction algorithms and also randomly generated solutions. The ESAS+R means that the Elitist Strategy of Ant System algorithm improves only randomly generated solutions, the ESAS+C,R means that the Elitist Strategy of Ant System algorithm improves the solution of construction algorithms and also randomly

generated solutions. MMAS+R means that the MAX-MIN Ant System algorithm improves only randomly generated solutions. The MMAS+C,R means that the MAX-MIN Ant System algorithm improves the solution of constructed algorithms and randomly generated solutions. The RBVAS+R means that the Rank-Based Version of Ant System improves randomly generated solutions, the RBVAS+C,R means that the Rank Based Version of Ant System algorithm improves the solutions of construction algorithms and also randomly generated solutions.

During the test runs, the parameters of the Ant System algorithm (AS+R) had the following parameters: number of iterations $20 * \text{number of customers}$, number of ants: 70, $\rho = 0.8$, $\alpha = 1$, $\beta = 2$ and 70 random solutions. AS + C,R differs from AS + R in that instead of the initial 70 random solutions, the initial 10 Nearest Neighbor solutions, 10 Nearest Insertion solutions, 10 Cheapest Insertion solutions, 10 Arbitrary Insertion solutions, 10 Farthest Insertion solutions, 10 Greedy solutions and it also contains 10 randomly generated solutions.

The Ant Colony System (ACS + R) had the following parameters: number of iterations $20 * \text{number of customers}$, number of ants: 70, $\rho = 0.8$, $\alpha = 1$, $\beta = 2$, $\xi = 0.8$ and 70 random solutions. ACS + C,R differs from ACS + R in that instead of the initial 70 random solutions, the initial 10 Nearest Neighbor solutions, 10 Nearest Insertion solutions, 10 Cheapest Insertion solutions, 10 Arbitrary Insertion solutions, 10 Farthest Insertion solutions, 10 Greedy solutions and it also contains 10 randomly generated solutions.

The MAX-MIN Ant System (MMAS + R) had the following parameters: number of iterations $20 * \text{number of customers}$, number of ants: 70, $\rho = 0.8$, $\alpha = 1$, $\beta = 2$, $\tau_{min} = \frac{\tau_{max}}{2 * \text{Number of locations}}$, $\tau_{max} = \frac{1}{\rho * T^{gb}}$ and 70 random solutions. MMAS + C,R differs from MMAS + R in that instead of the initial 70 random solutions, the initial 10 Nearest Neighbor solutions, 10 Nearest Insertion solutions, 10 Cheapest Insertion solutions, 10 Arbitrary Insertion solutions, 10 Farthest Insertion solutions, 10 Greedy solutions and it also contains 10 randomly generated solutions.

The Elitist Strategy of Ant System algorithm (ESAS+R) had the following parameters: number of iterations $20 * \text{number of customers}$, number of ants: 70, $\rho = 0.8$, $\alpha = 1$, $\beta = 2$, $w = 6$ and 70 random solutions. ESAS + C,R differs from ESAS + R in that instead of the initial 70 random solutions, the initial 10 Nearest Neighbor solutions, 10 Nearest Insertion solutions, 10 Cheapest Insertion solutions, 10 Arbitrary

Insertion solutions, 10 Farthest Insertion solutions, 10 Greedy solutions and it also contains 10 randomly generated solutions.

The Rank Based Version of Ant System algorithm (RBVAS+R) had the following parameters: number of iterations $20 * \text{number of customers}$, number of ants: 70, $\rho = 0.8$, $\alpha = 1$, $\beta = 2$, $w = 6$ and 70 random solutions. RBVAS + C,R differs from RBVAS + R in that instead of the initial 70 random solutions, the initial 10 Nearest Neighbor solutions, 10 Nearest Insertion solutions, 10 Cheapest Insertion solutions, 10 Arbitrary Insertion solutions, 10 Farthest Insertion solutions, 10 Greedy solutions and it also contains 10 randomly generated solutions.

Tab. 1 Parameter settings of the ACO algorithms

Number of iterations	20*Number of locations	
Number of Ants	70	
ρ	0.8	
α	1	
β	2	
ξ	0.8	
w	6	
τ_{min}	$\frac{\tau_{max}}{2 * \text{Number of locations}}$	
τ_{max}	$\frac{1}{\rho * T^{gb}}$ (T^{gb} =length of the global best route)	
Initial solution		
Nearest Neighbour	0	10
Nearest Insertion	0	10
Cheapest Insertion	0	10
Arbitrary Insertion	0	10
Farthest Insertion	0	10
Greedy	0	10
Random	70	10

Table 1. indicates the parameter settings of the ACO algorithms. The number of locations was 48 in our test case. Figure 4. illustrates the running result of the first dataset, Figure 5. shows the results of the second, Figure 6. the third and Figure 7. the fourth test results. In the case of each dataset, we give the best the average and the worst results of 10 runnings.

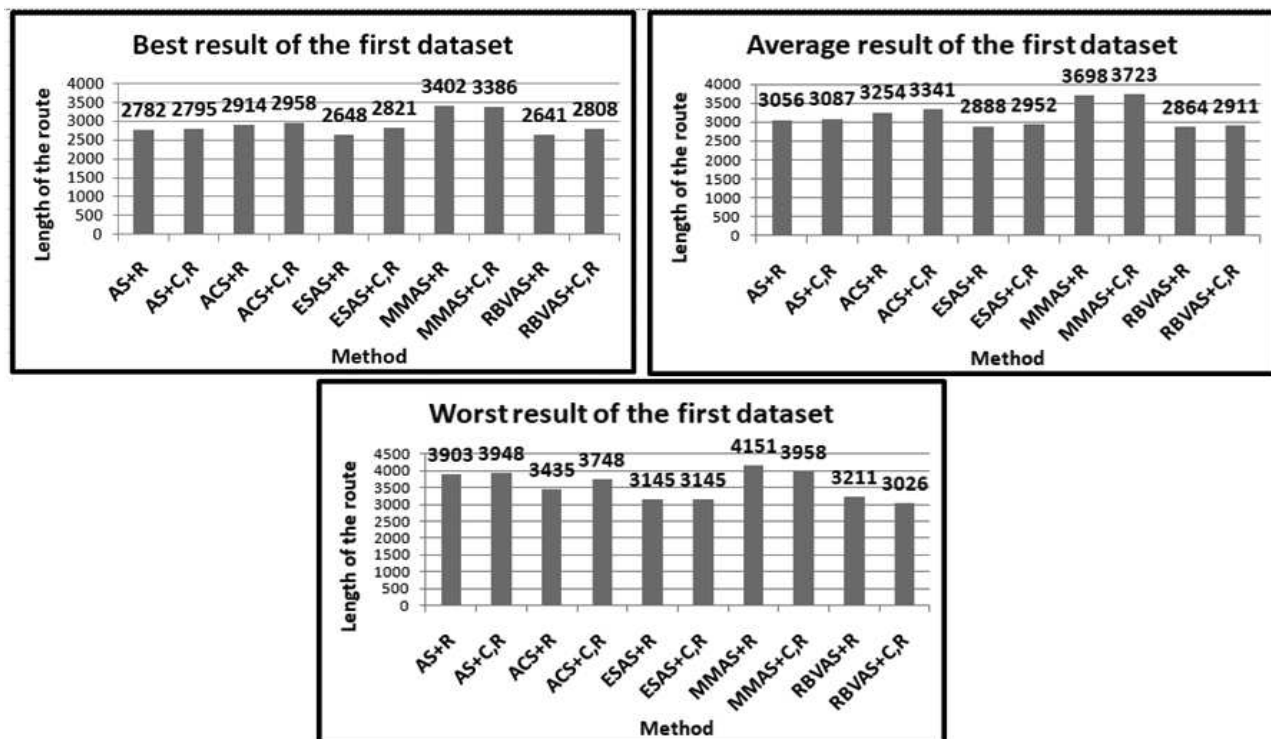


Fig. 4 The results of the first dataset

In the case of the first dataset (Figure 4.), the values of best result are between 2641 and 3402. This means 761 difference values. The best techniques were the ESAS algorithm (2648 fitness value) and the RBVAS algorithm (2641 fitness value). The improvement of construction algorithms is better, than improving only randomly generated solutions.

The average result values of the first dataset are between 2864 and 3723. The best techniques were the

ESAS algorithm (2888 fitness value) and the RBVAS algorithm (2864 fitness value). The worst technique was the MMAS algorithm (3723 fitness value). The difference between the best and worst fitness values is 859.

The worst result values of the first dataset are between 3026 and 4151 fitness values. The MMAS algorithm has the worst solution, and RBVAS has the best fitness value.

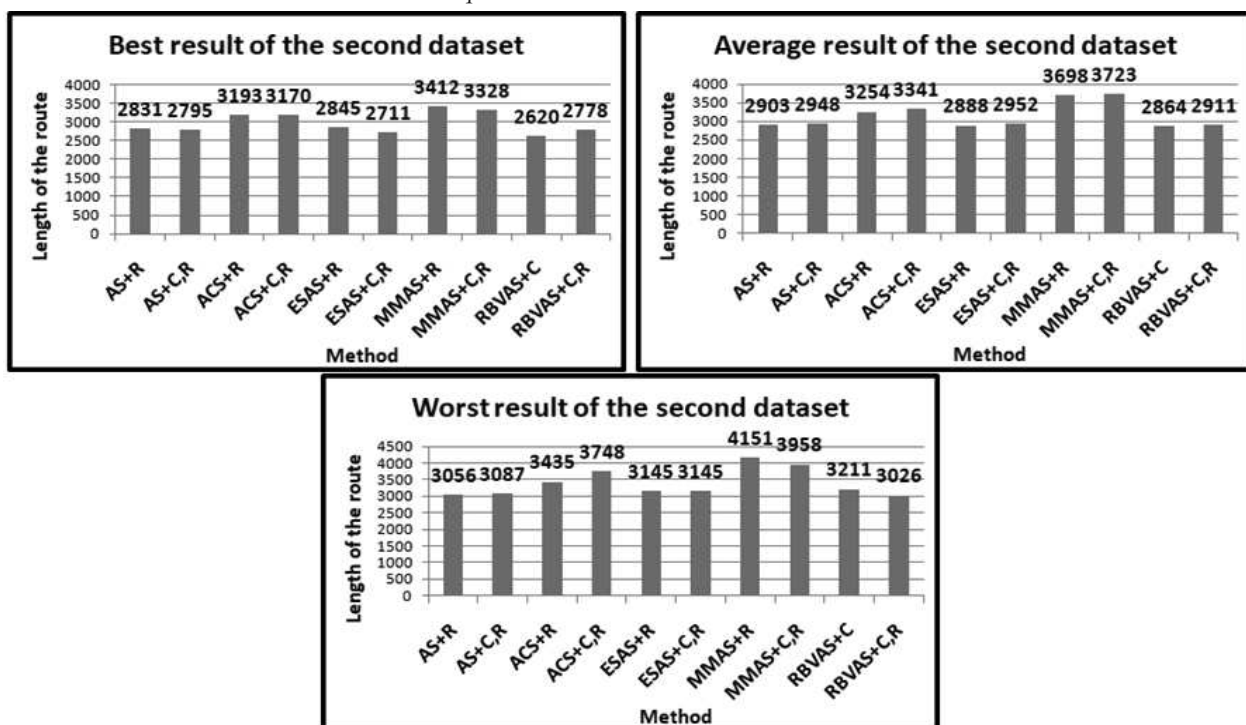


Fig. 5 The results of the second dataset

In the case of the second dataset (Figure 5.) also the ESAS and the RBVAS algorithm has the best performance. The MMAS algorithm has the worst performance.

In the best result of the second dataset figure can be seen, that the best values are between 2620 and 3412. The difference between the two values is 792. The best value belongs to the RBVAS algorithm, and the worst value belongs to MMAS algorithm.

In the average result of the second dataset figure

can be seen, that the average values are between 2864 and 3723 fitness values. The difference between these two values is 859. The MMAS gave the worst results, and the RBVAS gave the best results.

In the worst result of the second dataset figure can be seen, that the worst result values are between 3026 and 4151 fitness values. The 3026 fitness value belongs to RBVAS algorithm, and the 4151 fitness value belongs to MMAS algorithm.

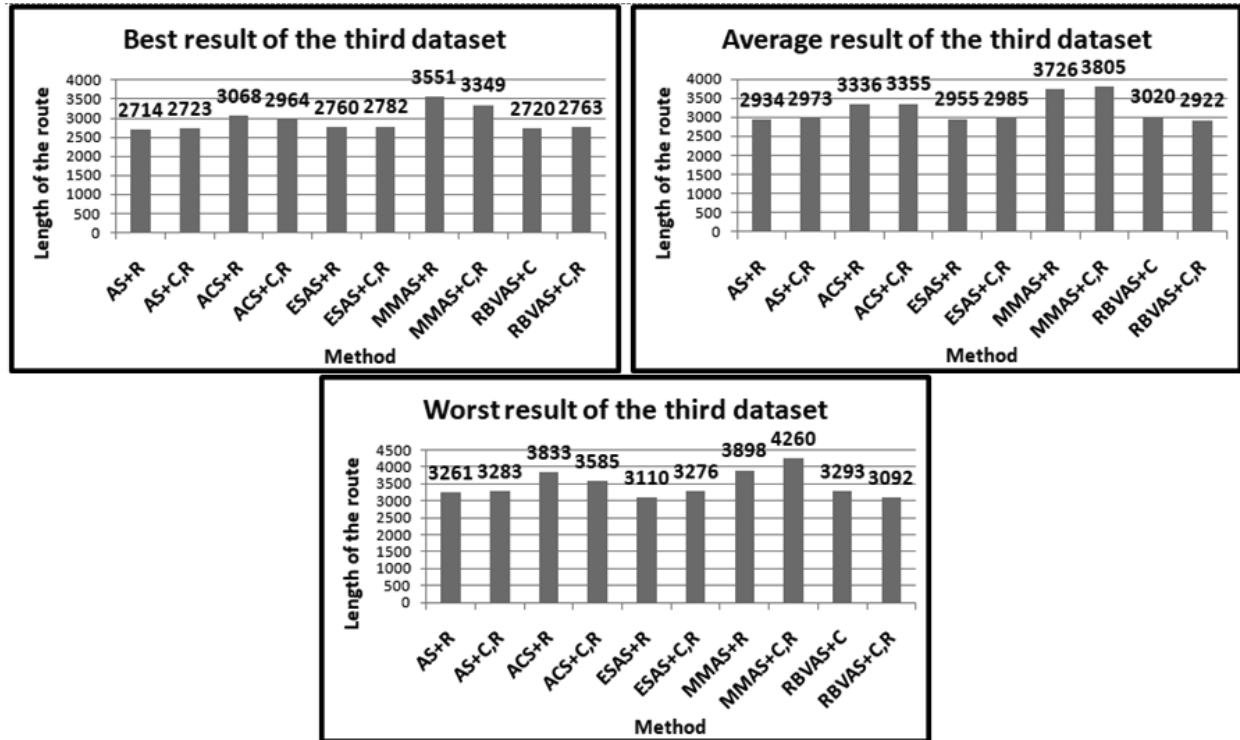


Fig. 6 The results of the third dataset

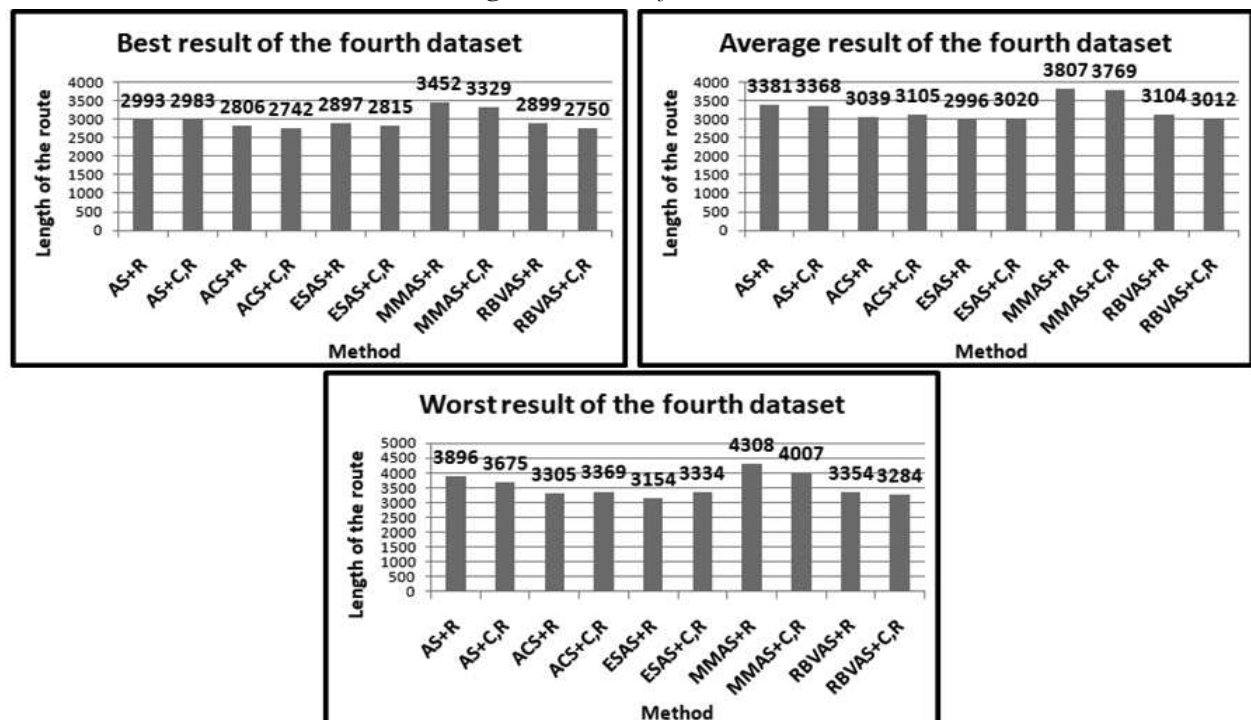


Fig. 7 The results of the fourth dataset

The third dataset (Figure 6.) shows the same as the previous datasets. The MMAS algorithm has the worst, and the ESAS and RBVAS has the best performance.

In the best result of the third dataset figure can be seen, that the best results are between 2714 and 3551. The difference of these two values is 837. The 2714 fitness value belongs to AS algorithm, and the 3551 belongs to MMAS algorithm.

The average result of the third dataset figure present, that the average results are between 2922 and 3805. The 2922 value belongs to RBVAS, and the 3805 value belongs to MMAS.

The worst result of the third dataset figure presents, that the worst results are between 3092 and 4260. The 3092 value belongs to RBVAS and the 4260 value belongs to MMAS algorithm.

The fourth dataset (Figure 7.) shows the same as the other, above presented three datasets. The best algorithms were the ESAS and RBVAS algorithm and the worst was the MMAS algorithm.

The best result of the fourth dataset figure presents, that the best results values are between 2742 and 3452. The 2742 value belongs to ACS algorithm, and the 3452 value belongs to MMAS algorithm.

The average result of the fourth dataset figure demonstrates that the average results values are between 3012 and 3807. The 3012 fitness value belongs to RBVAS, and the 3807 value belongs to MMAS algorithm.

Then considering the worst result of the fourth dataset figure, it can be established, that the worst results are between 3154 and 4308 values. The 3154 value belongs to ESAS algorithm, 4308 value belongs to MMAS algorithm.

8 Conclusion

In this article, we presented the Vehicle Routing Problem with Period, Time Window and Multiple Depots. We presented the mathematical model of the problem than we solved with construction and improvement algorithms. The construction algorithms were the following: Nearest Neighbour, Nearest Insertion, Farthest Insertion, Cheapest Insertion, Arbitrary Insertion and Greedy algorithm. The improvement algorithms were the Ant Colony Optimization (ACO) algorithms, which are the followings: Ant System (AS), Ant Colony System (ACS), Elitist Strategy of Ant System (ESAS), MAX-MIN Ant System (MMAS), Rank-Based Version of Ant System (RBVAS). We used the ACO algorithms to improve only randomly generated solutions and to improve to results of the construction algorithms and also randomly generated solutions. The results indicated, that the ESAS and the RBVAS algorithms were the most effective, and the MMAS

was the worst. The improvement of construction algorithms had better solutions than improving only randomly generated solutions.

9 References

- [1] TOTH, P., & VIGO, D. (2002). The vehicle routing problem. *Society for Industrial and Applied Mathematics*
- [2] NAGY, G., & SALHI, S. (2005). Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries. *European journal of operational research*, 162(1), pp. 126-141
- [3] CREVIER, B., CORDEAU, J. F., & LAPORTE, G. (2007). The multi-depot vehicle routing problem with inter-depot routes. *European Journal of Operational Research*, 176(2), pp. 756-773
- [4] LI, F., GOLDEN, B., & WASIL, E. (2007). The open vehicle routing problem: Algorithms, large-scale test problems, and computational results. *Computers & operations research*, 34(10), pp. 2918-2930.
- [5] MARINELLI, M., COLOVIC, A., & DELL'ORCO, M. (2018). A novel dynamic programming approach for two-echelon capacitated vehicle routing problem in city logistics with environmental considerations. *Transportation research procedia*, 30, 147-156.
- [6] BRUGLIERI, M., PEZZELLA, F., PISACANE, O., & SURACI, S. (2015). A Variable Neighborhood Search Branching for the Electric Vehicle Routing Problem with Time Windows. *Electron. Notes Discret. Math.*, 47, 221-228.
- [7] MIN, H. (1989). The multiple vehicle routing problem with simultaneous delivery and pick-up points. *Transportation Research Part A: General*, 23(5), pp. 377-386.
- [8] CAMPBELL, A. M., & WILSON, J. H. (2014). Forty years of periodic vehicle routing. *Networks*, 63(1), 2-15.
- [9] LIU, F. H., & SHEN, S. Y. (1999). The fleet size and mix vehicle routing problem with time windows. *Journal of the Operational Research society*, 50(7), 721-732.
- [10] TAŞ, D., DELLAERT, N., VAN WOENSEL, T., & DE KOK, T. (2014). The time-dependent vehicle routing problem with soft time windows and stochastic travel times. *Transportation Research Part C: Emerging Technologies*, 48, 66-83.

- [11] CALVETE, H. I., GALÉ, C., OLIVEROS, M. J., & SÁNCHEZ-VALVERDE, B. (2007). A goal programming approach to vehicle routing problems with soft time windows. *European Journal of Operational Research*, 177(3), 1720-1733.
- [12] BERTSIMAS, D. J. (1992). A vehicle routing problem with stochastic demand. *Operations Research*, 40(3), 574-585.
- [13] CAO, E., & LAI, M. (2010). The open vehicle routing problem with fuzzy demands. *Expert Systems with Applications*, 37(3), 2405-2411.
- [14] NGUEVEU, S. U., PRINS, C., & CALVO, R. W. (2010). An effective memetic algorithm for the cumulative capacitated vehicle routing problem. *Computers & Operations Research*, 37(11), 1877-1885.
- [15] MOUSAVI, S. M., VAHDANI, B., TAVAKKOLI-MOGHADDAM, R., & HASHEMI, H. (2014). Location of cross-docking centers and vehicle routing scheduling under uncertainty: A fuzzy possibilistic-stochastic programming model. *Applied Mathematical Modelling*, 38(7-8), 2249-2264.
- [16] SCHWARZE, S. (2016). Pricing strategies for the site-dependent vehicle routing problem. *OR spectrum*, 38(1), 137-173.
- [17] ARAS, N., AKSEN, D., & TEKIN, M. T. (2011). Selective multi-depot vehicle routing problem with pricing. *Transportation Research Part C: Emerging Technologies*, 19(5), 866-884.
- [18] GRANGIER, P., GENDREAU, M., LEHUÉDÉ, F., & ROUSSEAU, L. M. (2016). An adaptive large neighborhood search for the two-echelon multiple-trip vehicle routing problem with satellite synchronization. *European Journal of Operational Research*, 254(1), 80-91.
- [19] MAŃDZIUK, J., & ŚWIECHOWSKI, M. (2017). UCT in capacitated vehicle routing problem with traffic jams. *Information Sciences*, 406, 42-56.
- [20] BRUGLIERI, M., PEZZELLA, F., PISACANE, O., & SURACI, S. (2015). A variable neighborhood search branching for the electric vehicle routing problem with time windows. *Electronic Notes in Discrete Mathematics*, 47, 221-228.
- [21] CHEN, H. K., HSUEH, C. F., & CHANG, M. S. (2009). Production scheduling and vehicle routing with time windows for perishable food products. *Computers & operations research*, 36(7), 2311-2319.
- [22] KELLNER, T., KYNCL, J., PITRMUC, Z., BERANEK, L., KANAK, M., & KYNCL, M. (2019). Production process planning in Additive manufacturing and conventional machining technology manufacturing system. *Manufacturing Technology*, 19(2), 232-237.
- [23] MODROVSKÝ, R. (2019). Optimization of Production Flow through the CRAFT Method. *Manufacturing Technology*, 19(1), 114-117.
- [24] YOSHIZAKI, H. T. Y. (2009). Scatter search for a real-life heterogeneous fleet vehicle routing problem with time windows and split deliveries in Brazil. *European Journal of Operational Research*, 199(3), 750-758.
- [25] PRINS, C., LACOMME, P., & PRODHON, C. (2014). Order-first split-second methods for vehicle routing problems: A review. *Transportation Research Part C: Emerging Technologies*, 40, 179-200.
- [26] KIZILATES, G., & NURIYEVA, F. (2013). On the nearest neighbor algorithms for the traveling salesman problem. In *Advances in Computational Science, Engineering and Information Technology*, pp. 111-118. Springer, Heidelberg.
- [27] HAHSLER, M., & HORNIK, K. (2007). TSP-Infrastructure for the traveling salesperson problem. *Journal of Statistical Software*, 23(2), pp.1-21.
- [28] NILSSON, C. (2003). Heuristics for the traveling salesman problem. *Linköping University*, pp.1-6.
- [29] KANTOR, M., CHALUPA, M., SOUČEK, J., BÍLKOVÁ, E., NOWAK, P. (2020). Application of Genetic Algorithm Methods for Water Turbine Blade Shape Optimization, *Manufacturing Technology*, 20(4), 453-458.
- [30] ZHANG, W., HOU, L., GAN, Y., XU, C., BU, X., LIN, H. (2019). Parallel Optimization of the Balancing and Sequencing for Mixed-model Assembly Lines, *Manufacturing Technology*, 19(3), 537-544.
- [31] STÜTZLE, T., & DORIGO, M. (1999). ACO algorithms for the traveling salesman problem. *Evolutionary algorithms in engineering and computer science*
- [32] BULLNHEIMER, B., HARTL, R. F., & STRAUSS, C. (1997). A new rank based version of the Ant System. A computational study