Transport and Telecommunication, 2019, volume 20, no. 4, 305–317
Transport and Telecommunication Institute, Lomonosova 1, Riga, LV-1019, Latvia
DOI 10.2478/ttj-2019-0025

TWO - ECHELON VEHICLE ROUTING PROBLEM WITH RECHARGE STATIONS

Anita Agárdi[1], László Kovács[2], Tamás Bányai[3]

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

The efficient operation of logistic processes requires a wide range of design tasks to ensure efficient, flexible and reliable operation of connected production and service processes. Autonomous electric vehicles support the flexible in-plant supply of cyber-physical manufacturing systems. Within the frame of this article, the extension of the Two-Echelon Vehicle Routing Problem with recharge stations is analyzed. The objective function of the optimization problem is the minimization of operation costs. The extension of 2E-VRP means that the second level vehicles (electric vehicles, must be recharged) come from one recharge station, then pick up the products from the satellite, visit the customers and return to the recharge station from where it started. We solved the route planning problem with the application of construction heuristics and improvement heuristics. The test results indicate that the combination of this approach provides a superior efficiency.

Keywords: vehicle routing problem; heuristics; recharge station; construction heuristics; improvement heuristics; metaheuristics

1. Introduction

In the age of the fourth industrial revolution, versatile solutions for manufacturing and logistics systems are required to increase the utilization, reliability, flexibility and cost efficiency. The diversified customers' demands present manufacturing systems with new challenges (Tamás, 2017). Matrix production may become a suitable solution through configurable production cells, the transfer of parts and tools using automated guided vehicles (AGVs) and the separation of logistics from production. The in-plant supply of manufacturing and assembly cells in matrix production is based on autonomous electric vehicles. The design and operation of autonomous electric vehicles represent a special type of vehicle routing and scheduling problems because vehicles must be recharged and therefore the solutions of traditional routing problems cannot be used to optimize the operation of AGVs. This fact was the motivation for writing this paper. After this introduction, the remaining parts of the paper are divided into five sections. Section 2 presents a literature review to summarize the research background. Section 3 presents the model framework and mathematical model of two-echelon vehicle routing problem with recharge stations. Section 4 presents the used heuristic solutions, while section 5 presents the results of the numerical analysis. Conclusions and future research directions are discussed in the last section.

2. Literature review

Within the frame of this chapter, we present a systematic literature review. Firstly, the relevant terms were defined. The optimization of the vehicle routing is a very intensively investigated research area. We can find 1513 articles in the literature (in the Web of Science database) related to the keywords: TOPIC: ("vehicle routing problem") AND TOPIC: ("optimization")) OR (TOPIC: (milkrun) AND TOPIC: (autonomous vehicle)). From this 1472 are journal articles. Our search was conducted in January 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 sciences, management and transportation. This huge number of publication shows the importance of optimization related to vehicle routing in transportation and manufacturing.

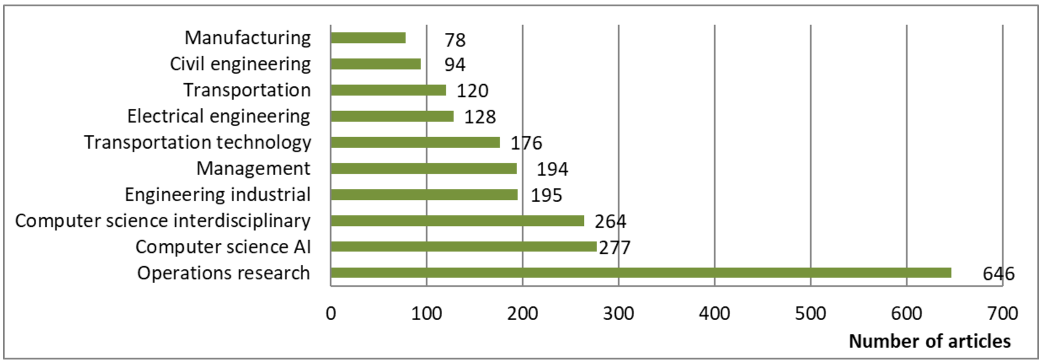

Figure 1. Classification of articles considering subject areas based on search in WoS database using TOPIC: "vehicle routing problem" AND TOPIC: "optimization" OR TOPIC: milkrun AND TOPIC autonomous vehicle keywords

As Figure 2 demonstrates, the vehicle routing problems in transportation has been researched over the past 20 years. The first article in this field was published in 1959 in the field of truck dispatching problem (Dantzig and Ramser, 1959).

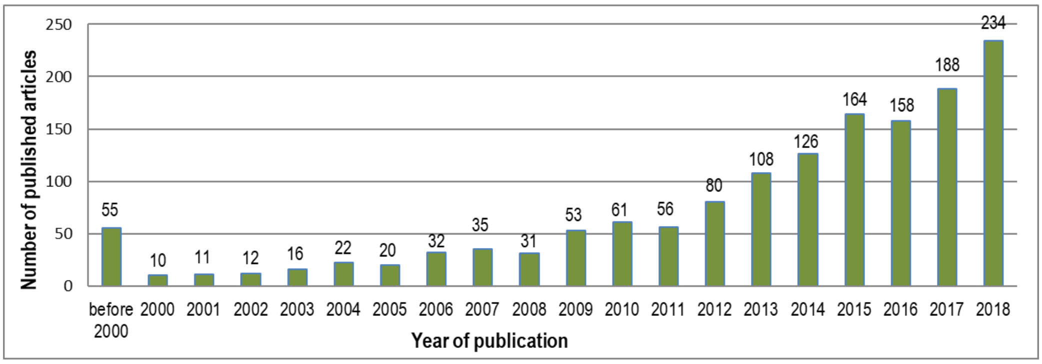

Figure 2. Classification of articles by year of publication

The distribution of the most frequently used keywords is presented in Figure 3. As the keywords show, the design of energy efficiency of supply chain solutions has a central role and different heuristic optimization methods are applied to find optimal solutions.

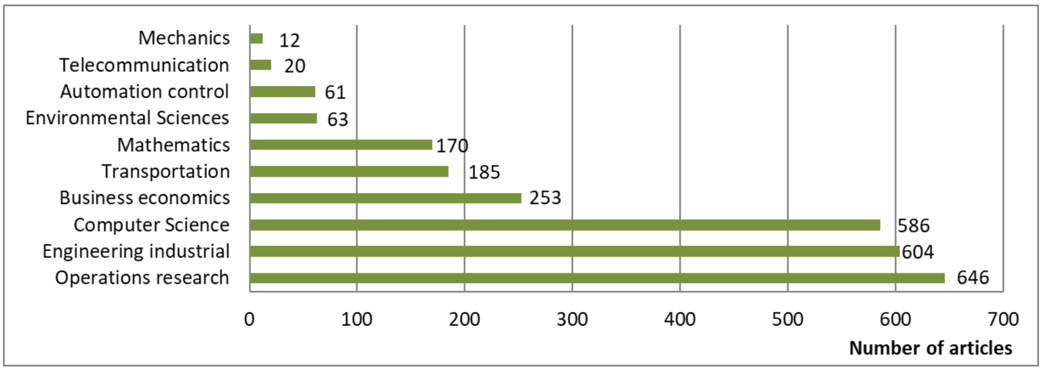

Figure 3. Classification of articles by year of publication

The Vehicle Routing Problem (VRP) is an NP-hard optimization problem in logistics. It aims at the cost-efficient delivery of products to customers. In the case of classical Vehicle Routing Problem known the position, the demand of the customers in advance. Known also the position of the central depot, the number and capacity of the vehicles in advance. The vehicles deliver the products from the depot to the customers. The goal of the problem is the minimization of the cost of transportation. The important parameters of the optimal problems are:

- Single Depot Vehicle Routing Problem (Nagy *et al.*, 2005): A single depot is available.
- Multi Depot Vehicle Routing Problem (Renaud *et al.*, 1996): There are several depots. The vehicle starts from one of the depots and after visited the customers must return to the same depot.
- Capacitated Vehicle Routing Problem (Ralphs, 2003): The customer has product demands which must be satisfied.
- Vehicle Routing Problem with Pick-up and Delivery (VRPPD) (Min, 1989): The vehicles must transport products from the depot to the customers and, at the same time collect products from customers and transport to the depot.
- Vehicle Routing Problem with Time Windows (VRPTW) (Desrochers *et al.*, 1992): Each customer has a time window. The demand of the customers must be satisfied in this interval.
- Distance Constrained Vehicle Routing Problem (DCVRP) (Li *et al.*, 1992): The vehicles can be travel from the depot to a maximal distance, then must return to the depot.
- Open Vehicle Routing Problem (OVRP) (Li *et al.*, 2007): The vehicles do not return to the depot.
- Two-Echelon Vehicle Routing Problem (2E-VRP) (Crainic *et al.*, 2010): In this type of VRP there are one or more depots and also several satellites. The products are delivered from the depot to intermediate locations, called satellites. From the satellites the products are delivered to the customers.
- Emissions Vehicle Routing Problem (EVRP) (Figliozzi, 2010): The goal of this type of VRP is not the minimization of the route, but the minimization of the fuel consumption.
- Periodic Vehicle Routing Problem (PVRP) (Cordeau *et al.*, 1997): The customers must be visited not once, they must be visited periodically. For example each customers must be visited daily, each must be visited weekly etc.
- Stochastic Vehicle Routing Problem (SVRP) (Gendreau *et al.*, 1996): In case of this problem certain factors are not known in advance. In most of the cases the demands of the customers are not known in advance.

More than 50% of the articles were published in the last 4 years. The articles that addressed the vehicle routing problems are focusing on conventional transportation vehicles, and only few of have aimed to research the routing problems of autonomous electric in-plant vehicles. Therefore, the routing of autonomous electric vehicles with charging stations still needs more attention and research.

The main contribution of this article includes: (1) the model of milkrun-based in-plant supply with autonomous electric vehicles that include routing and assignment problems; (2) description of optimization methods to solve the model; (3) validation of methods through computational results with different datasets.

3. Two - Echelon Vehicle Routing Problem with Recharge Stations

Within the frame of this chapter we are describing the model of in-plant vehicle routing problems of autonomous electric vehicles as a Two-Echelon Vehicle Routing problem (Figure 4).

The system contains only one depot, its position is fixed. The system contains also the position and the demand of the customers. We involve independent satellite and recharge stations in our model. The position and the capacity of the satellites are also known in advance. There are two types of vehicle. The first type of vehicles called first level vehicles, deliver the products from the depot to the satellites have more capacity. The second type of vehicles called second level vehicles, have less capacity, and transfer product from satellite to customers. The vehicles are electric vehicles, they must be regularly recharged at the recharge stations. The second level vehicles start from one recharge station, then visit one satellite, transfer from the satellite the products to the customers and then return to the recharge station.

In figure 4 the D indicates the depot. S1, S2, S3 and S4 indicate satellites. The numbers from 1 to 18 denote the customers and the R1, …, R11 denote the recharge stations. The recharge stations can be allocated dynamically to the vehicles, based on distance and operational status. In the picture, the R2, R3, R4, R6 and R7 recharge stations are applied. In figure 4 there are two first level vehicles. The first vehicle starts from D, and then visits S3, S1 and then returns to D. The second vehicle starts from D, then visits S4, S2, then returns to D. There are six second-level vehicles. The goal of the problem is the minimization of the route length.

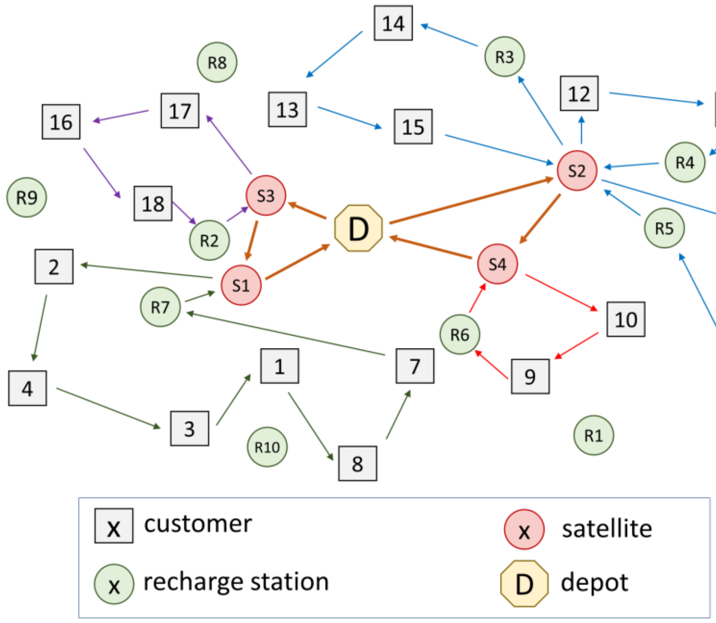

Figure 4. Two Echelon Vehicle Routing Problem with Recharge Stations

The mathematical model

We described the optimization model based on (Perboli *et al.*, 2011). In this article, the authors described the classical 2E-VRP. We used in most of the cases the notations of the authors. We extended this model with recharge stations, so some new notations and conditions are introduced, and the objective function is altered.

We use the following notations in the description of the optimization:

- $D = \{d_0\}$ depot
- $S = \{s_1, s_2, \dots, s_{n_s}\}$ satellites.
- $C = \{c_1, c_2, \dots, c_{n_c}\}$ customers.
- $\bar{D} = \{\bar{d}_1, \bar{d}_2, \dots, \bar{d}_{n_c}\}$ demands of the customers.
- $R = \{r_1, r_2, \dots, r_{n_r}\}$ recharge stations.
- n_s number of satellites.
- n_c number of customers.
- n_r number of recharge stations.
- m_1 the number of vehicles in depot-satellite level.
- m_2 the number of vehicles in satellite-customer level.
- m_{s_k} capacity of the satellite (indicates the maximum demand of the satellite).
- K^1 capacity of the vehicles in the first level (depot-satellite level).
- K^2 capacity of the vehicles in the second level (satellite-customer level).
- L^2 the maximal distance of the vehicles in the satellite-customer level.
- c_{ij} the cost of the (i, j) edge.
- Q^1_{ij} the amount of transported products from the depot-satellite (i, j) edge.
- Q^2_{ijk} the amount of transported products from the satellite-customer (i, j) edge, which starts from the k. satellite.
- x_{ij} the number of vehicles, which passes through depot-satellite level (i, j) edges.
- y^k_{ij} value can be 0 or 1. 1, if the satellite-customer level (i, j) edge belongs to the k. satellite.
- z_{kj} 1, if customer c_i is served from k. satellite.
- λ^k_{ij} 1, if after the recharge station the i. customer is visited by the k.vehicle and after the j. customer the vehicle returns to the recharge station.

The products must be delivered from the d_0 depot to the customers $(C = \{c_1, c_2, \dots, c_{n_c}\})$. $\bar{d}_i$ indicates the demand of c_i customer. In the depot-satellite level m_1 indicates the number of vehicles. These vehicles have the same K^1 capacity. The m_2 indicates the maximum number of vehicles in satellite-customer level. Each vehicle has maximum m_{s_k} capacity. In the satellite-customer level the vehicles have same K^2 capacity. The satellite-customer vehicles are electric vehicles. These vehicles must be recharged after taking certain distance, so these vehicles must visit the recharge station. These vehicles start from the recharge station and after taking their route come back to their recharge station.

In order to simplify the mathematical model, we introduce an auxiliary variable:

$$\bar{D}_k = \sum_{j\in C} \bar{d}_j z_{kj}, \forall k \in S. \tag{1}$$

The formula indicates the sum of the route, which belongs to the k. satellite.

We use the following objective function, which minimalize the route and the related costs:

$$\min \sum_{i,j\in D\cup S, i\neq j} c_{ij}x_{ij} + \sum_{k\in S}\sum_{i,j\in S\cup C, i\neq j} c_{ij}y_{ij}^k + \sum_{k\in S}\sum_{i,j\in S\cup C, i\neq j} c_{ij}\lambda_{ij}^k. \tag{2}$$

The optimal route-plan must satisfy the following conditions:

The number of routes does not exceed the number of vehicles in the depot-satellite level:

$$\sum_{i\in S} x_{0i} \leq m_1. \tag{3}$$

If $k = d_0$, the whole depot-satellite level route starts and ends in the depot. If k is satellite, the vehicles take a route, which also consists of the given satellite:

$$\sum_{j\in S\cup D, j\neq k} x_{jk} = \sum_{i\in S\cup D, i\neq k} x_{ki} \ \ \forall k \in S \cup D. \tag{4}$$

The number of routes does not exceed the number of available vehicles in the depot-satellite level:

$$\sum_{k\in S}\sum_{j\in S} y_{kj}^k \leq m_{2.} \tag{5}$$

The capacity limit of each satellite, which constraints the maximal number of satellite-customer routes, which belongs to each satellites:

$$\sum_{j\in C} y_{kj}^k \leq m_{s_k} \ \forall k \in S. \tag{6}$$

The satellite-customer level vehicles can take a maximal distance, after taking, they must be recharged:

$$\sum_{k\in S}\sum_{i,j\in S\cup C, i\neq j} c_{ij}y_{ij}^k + \sum_{k\in S}\sum_{i,j\in S\cup C, i\neq j} c_{ij}\lambda_{ij}^k \leq L^2. \tag{7}$$

The satellite-customer routes starts from a recharge station, and ends at the same recharge station:

$$\sum_{j\in C} \lambda_{ij}^k = \sum_{j\in C} \lambda_{ji}^k \ \ \forall j \in R. \tag{8}$$

The product traffic is the same of the demand of the customer, except the depot. In case of the depot, the outcome product traffic is the same of the demand of all customers. This condition eliminates sub-tours, which does not consist the depot.

$$\sum_{i\in S\cup D, i\neq j} Q_{ij}^1 - \sum_{i\in S\cup D, i\neq j} Q_{ji}^1 = \begin{cases} \bar{D}_j \ if\ j\ is\ not\ depot \\ \sum_{i\in V_c} -d_i \ \ otherwise \end{cases} \forall j \in S \cup D. \tag{9}$$

The capacity limit in the depot-satellite level:

$$Q_{ij}^1 \leq K^1 x_{ij} \ \forall i,j \in S \cup D, i \neq j. \tag{10}$$

The product traffic equals to the demand of the given customer, except the satellite. In case of the satellite, the outcoming product traffic equals the demands of the customers which belong to the given satellite. This condition eliminates subtours, which does not contain satellites.

$$\sum_{i\in V_c\cup k, i\neq j} Q_{ijk}^2 - \sum_{i\in V_c\cup k, i\neq j} Q_{jik}^2 = \begin{cases} z_{kj}\bar{d}_j \ if\ j\ is\ not\ satellite \\ \quad -\bar{D}_j \ otherwise \end{cases} \forall j \in C \cup S, \forall k \in S. \tag{11}$$

Capacity limit in the satellite-customer level:

$$Q_{ijk}^2 \leq K^2 y_{ij}^k \ \forall i,j \ \in S \cup C, i \neq j, \forall k \in S. \tag{12}$$

All of the products must be transported to the satellites, does not transfer products back to the depot:

$$\sum_{i\in S} Q_{id_0}^1 = 0. \tag{13}$$

All of the products must be transported to the customers, does not transfer products back to the satellites:

$\sum_{j\in C} Q^2_{jkk} = 0\ \forall k \in S.$ (14)

The j. customer is served by the k. satellite, when the customer belongs to the k. satellite:

$y^k_{ij} \le z_{kj}\ \forall i \in S \cup C, \forall j \in C, \forall k \in S.$ (15)

$y^k_{ij} \le z_{kj}\ \forall i \in S, \forall j \in C, \forall k \in S.$ (16)

Just one satellite-depot level route can consists one customer:

$\sum_{i\in V_s \cup V_c} y^k_{ij} = z_{k_j}\ \ \forall k \in S, \forall j \in C.$ (17)

$\sum_{i\in V_s} y^k_{ij} = z_{k_j}\ \ \forall k \in S, \forall j \in C.$ (18)

Each customers can belong to only one satellite:

$\sum_{i\in V_s} z_{ij} = 1\ \forall j \in C.$ (19)

The satellite-customer routes, which belong to the k. satellite, can be taken after the service of the depot-satellite level.

$y^k_{kj} \le \sum_{l\in S\cup D} x_{kl}\ \ \forall k \in S, \forall j \in C.$ (20)

$y^k_{ij} \in \{0,1\}, z_{kj} \in \{0,1\},\ \lambda^k_{ij} \in \{0,1\},\ \ \forall k \in S \cup D, \forall i,j \in C.$ (21)

$x_{kj} \in \mathbb{Z}^+, \forall k,j \in S \cup D.$ (22)

$Q^1_{ij} \ge 0, \forall i,j \in S \cup D, Q^2_{ijk} \ge 0, \forall i,j \in S \cup D, \forall k \in S.$ (23)

4. Applied optimization

4.1. Construction algorithms

We use construction algorithms based on the Traveling Salesman Problem (TSP) algorithms. The construction algorithms always take locally the best steps. The applied construction algorithms start from a randomly selected customer. The Nearest Neighbour algorithm takes then the unvisited customer, which is the nearest to the last selected customer. The Arbitrary Insertion algorithm takes then the unvisited customer random, and insert into two selected customers, which minimize the cost of the route. The applied construction algorithms end, when all customers are selected (Rosenkrantz *et al.*, 1977).

4.2. Improvement algorithms

These algorithms improve one or more initial solutions iteratively. The Genetic algorithm (GA) simulates natural processes, inspired by the evolution. The Genetic algorithm operates on the population of the solutions. The population consists of individuals. The Genetic algorithm most often represents individuals with bits. Common is also the permutation representation. The individuals have fitness values. The fitness values indicate the goodness of the individuals. The general steps of the genetic algorithms (Maulik *et al.*, 2000):

- Step 1: The initialization of the population
- Step 2: Calculation of the fitness values of the individuals
- Step 3: Some individuals will belong unaltered to the next generation
- Step 4: The chosen parent will be recombined
- Step 5: The chosen new individuals will be mutated
- Step 6: The new individuals will be evaluated
- Step 7: The next generation of the population will be chosen according to their fitness values
- Step 8: The 3.-7. steps will be iterated after the stop condition is not met (for example, iteration count or running time is reached, the fitness values are altered slightly etc.)

The first step of GA is the initialization of the population with (in most cases random) individuals. Then the fitness of each individual will be calculated. The calculation of the fitness values is problem dependent. The fitness function must indicate the goodness of the solutions. The better the individual is, the higher is the fitness function. In the next step, some individuals will be transferred unaltered to the next generation. This is called elitism strategy. Each selected individuals will be recombined. After recombination, new individuals appear. These new individuals are called children. Selected children will be mutated. After these steps, the new individuals will be evaluated. The last step is the uploading of the new population.

There are several crossing and mutation algorithms. We applied the Order Crossover (OX) the Cycle Crossover (CX) and the Partially Matched Crossover (PMX). The 2-opt operator was used for mutation.

The Order Crossover (OX) (Braun, 1990) is the following: we choose two crossing points randomly, which gives a fitting section (Figure 5). First we delete the elements of the first individual in the fitting section in the other individual, and holes are formed (denote Letter H). In the following the fitting section must be empty. The parent will have the first non-hole (non-H) gene the first element of the child. We can place the other genes in place by always taking them the next non-empty (non-H) gene in the parent and placed in the next free space in the offspring so that the fit section is empty. In the child, we replace the empty section (letter H in the figure) with genes in the middle of the other parent's chromosome.

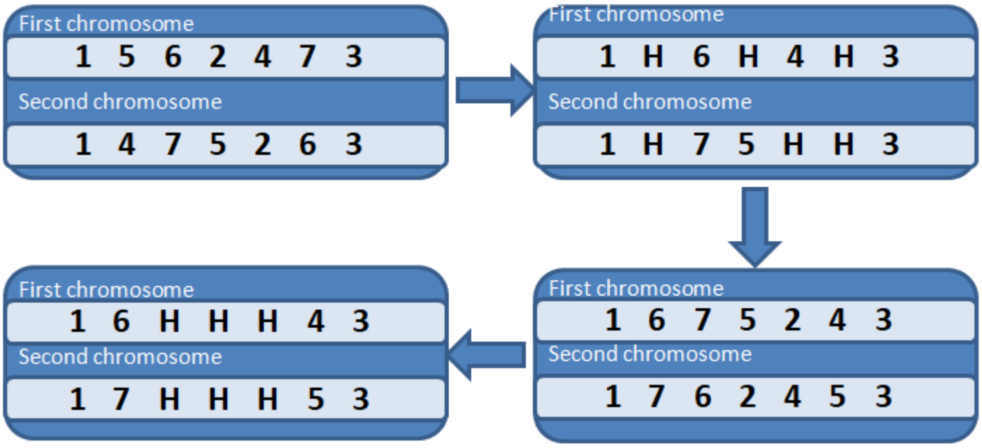

Figure 5. Order Crossover (OX)

The Cycle Crossover (CX) (Starkweather *et al.*, 1991) is based on the fact that an element is transferred from the first or second parent to the new individual. Take the first gene pair as the first element of the first parent and the first element of the second parent. Then the first parent's gene gets into the first offspring and the second parent's gene goes to the second offspring. Since the gene set must be complete (permutation) for the offspring, so should the second child have a gene such as the first gene of the first parent. The first parent gene has already been given to the first child, so the second parent must have the second parent gene. This procedure should continue until the cycle is round. The other genes are split so that the first parent genes are given to the second child, while the second parent genes are given by the first child (Figure 6).

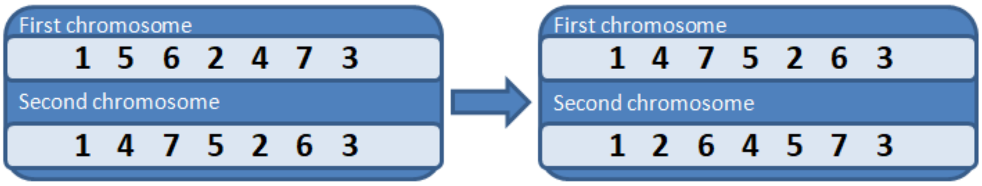

Figure 6. Cycle Crossover (CX)

The Partially Matched Crossover (PMX) (Starkweather *et al.*, 1991) also works with a fitting section. The genes in the fitting section are paired. Children are obtained by making copies of the parents and then changing the genes defined by the pairs of numbers (Figure 7).

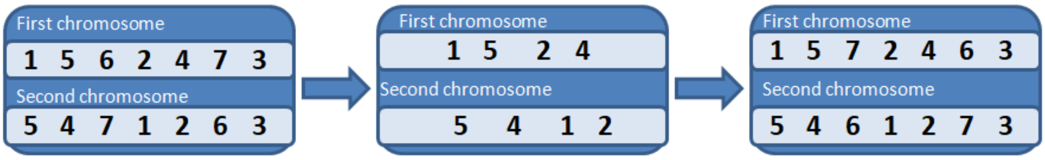

Figure 7. Partially Matched Crossover (PMX)

We used the 2-opt operator as mutation (Braun, 1990) to replace two edges (Figure 8).

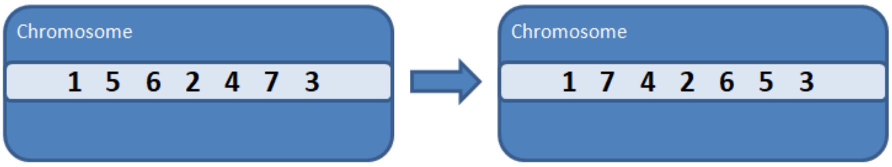

Figure 8. 2-opt operator

The Hill Climbing algorithm improves one solution iteratively. In the first step, the algorithm starts from a solution (mostly random solution). This solution is called actual solution. Then the algorithm tries to generate one or more neighbours of the actual solution. Then the best neighbour will be chosen. If the best neighbour is better than the actual solution, the best neighbour will be the actual solution. (Katayama *et al.*, 2000). The neighbour is created with 2-opt operator.

5. Implementation and evaluation

We used permutation representation. We created a three-part of representation. This means, that the whole representation consists of three permutations (Figure 9). The first permutation indicates the depot-satellite level (contains the number of each satellite), the second permutation indicates the satellite-customer level (contains the number of each customers), and the third permutation indicates the recharge stations (contains the number of recharge stations).

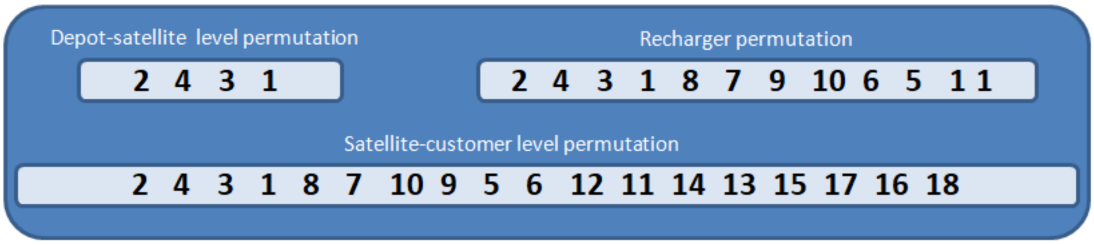

Figure 9. Permutation representation of the problem

The evaluation of the permutation consists of the following steps. The satellite-customer permutation is chosen in the first step of the evaluation. The customers are chosen after the vehicle reaches the capacity limit. The selected customers will belong to the same vehicle. Figure 10.a demonstrates this step.

The elements of the satellite-customer permutation are selected sequentially, after the capacity of the satellite-customer level vehicle is not met. The process is continued after not all customers are chosen. Figure 10.b demonstrates a part of a possible solution. The capacity constraint of the satellites determines from which depot the vehicles transport the product. The vehicles are selected in sequence after the capacity of the satellite is met. These vehicles (and the customers of these vehicles) will belong to the same satellite. Then the next vehicles will be chosen sequentially etc. All vehicles must belong to one satellite. Figure 10.c represents this method.

The next step is to determine which vehicle will serve which satellites. This mechanism is similar to the vehicles-customers binding. The difference is that not all satellites must have vehicles. It can be occurred, that a satellite has no demand, so the satellite does not serve customers. Figure 10.d demonstrates a possible solution. Then the recharge station must be bounded with satellite-customer level vehicles. The recharge stations of the permutation and the satellite-customer level vehicles will be selected in order. The first element of the recharge permutation will belong to the first satellite-customer level vehicle; the second element of the recharge permutation will belong to the second vehicle and so on. This process can be seen in Figure 10.e. This process is iterated until no more satellite-customer level vehicles are. The remaining recharge stations are not taking into account. The solution can be seen in Figure 10.f.

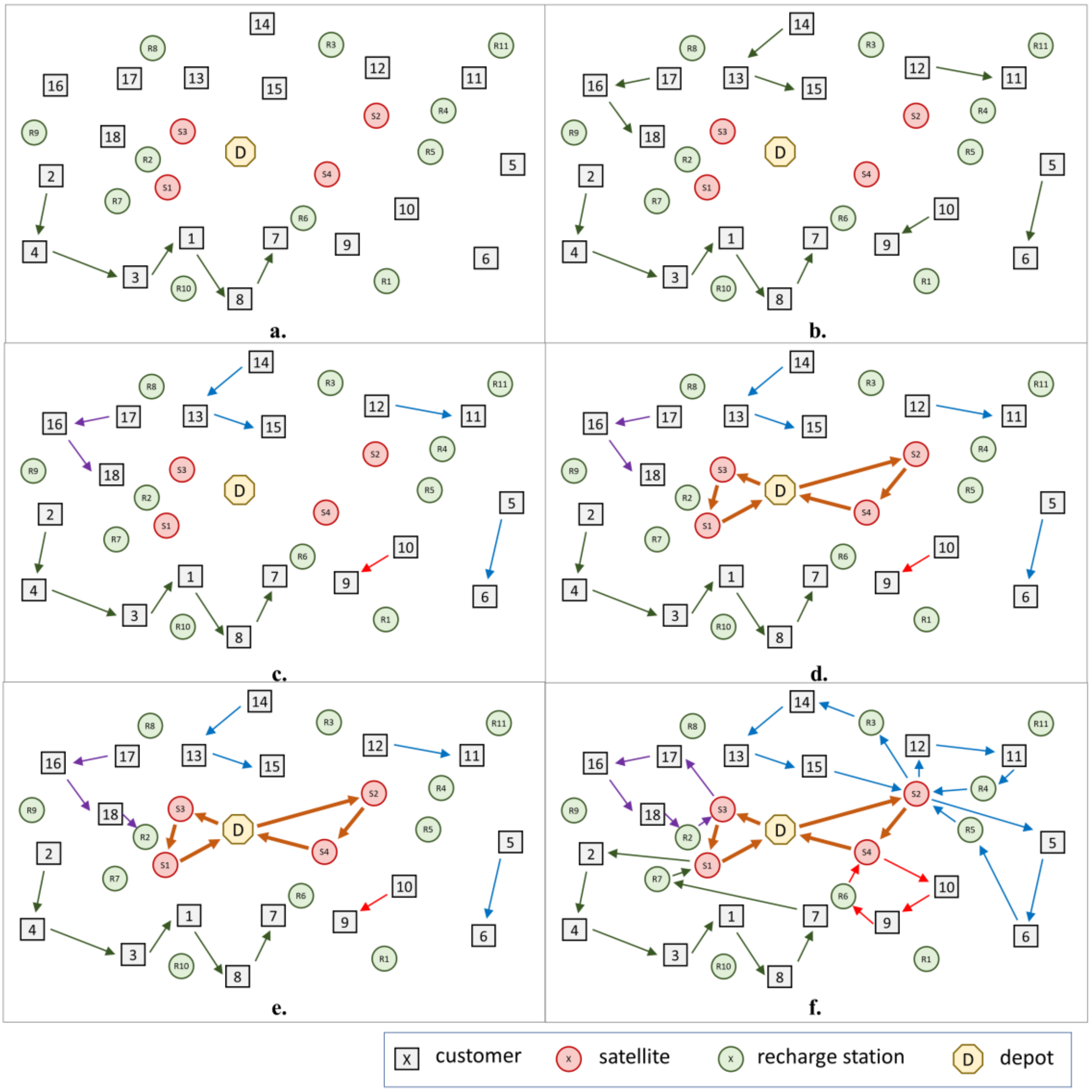

Figure 10. **a)** The coherent customers and their sequence; **b)** The sequence of the customers; **c)** Vehicles – satellites binding; **d)** Satellite-vehicles binding; **e)** Binding satellites to satellite-customer level vehicles; **f)** Binding recharge stations with satellite-customer level vehicles

6. Test results

The algorithms was implemented in Java. The test results were made in the following hardware environment: CPU: Intel i5 1,8 GHz, RAM: 8GB, Operating System: Windows 10 Pro. Based on the test results the combination of construction algorithm and genetic algorithm were the best in the average.

Table 1. Structure of the dataset

Dataset	Number of vehicles	Number of customers	Number of satellites
01	4	48	10
02	4	48	10
03	4	96	10
04	4	96	10

Table 2. Parameters of the genetic algorithm

Number of iteration	100* number of customers	
OX crossover	0.3	0.3
CX crossover	0.3	0.3
PMX crossover	0.3	0.3
Number of elitism	10	10
Initial solutions created with Nearest Neighbour Algorithm	30	0
Initial solutions created with Arbitrary Insertion Algorithm	30	0
Randomly created initial solutions	30	90

Table 3. Parameters of the hill climbing algorithm

Number of iteration	5000* number of customers		
Initial solutions created with Nearest Neighbour Algorithm	1	0	0
Initial solutions created with Arbitrary Insertion Algorithm	0	1	0
Randomly created initial solutions	0	0	1

Table 4. Test results in the case of different algorithms

Genetic algorithm improving construction heuristics and random solutions				**Genetic algorithm improving random solutions**		
Dataset	**Best result**	**Worse result**	**Average result**	**Best result**	**Worse result**	**Average result**
01	1544.3562	1826.0787	1614.3596	1737.1068	1963.5961	1875.0753
02	1566.5477	1566.5477	1609.3894	1799.8128	2046.4477	1912.7631
03	2231.0849	2528.9775	2338.6010	2999.8053	3436.3459	3252.7300
04	2219.3018	2478.2863	2337.2244	3110.5496	3519.8378	3291.7636
Nearest Neighbour algorithm solutions				**Arbitrary Insertion algorithm solutions**		
Dataset	**Best result**	**Worse result**	**Average result**	**Best result**	**Worse result**	**Average result**
01	2210.2310	2438.4807	2306.9814	2040.1331	2827.8947	2275.3639
02	2134.8915	2492.9391	2272.3484	2104.7768	2558.3083	2262.9433
03	2524.9931	3008.8530	2827.8970	2792.8133	3301.8926	3044.7333
04	3110.5496	3519.8378	3291.7636	2751.6701	3104.0569	2967.3083
Hill Climbing algorithm improving Nearest Neighbour algorithm				**Hill Climbing algorithm improving Arbitrary Insertion algorithm**		
Dataset	**Best result**	**Worse result**	**Average result**	**Best result**	**Worse result**	**Average result**
01	1597.1717	1901.6917	1726.8008	1543.8055	1802.4046	1672.6482
02	1637.1992	1910.1925	1742.3684	1483.0881	1723.6742	1627.4466
03	2139.6349	2410.2512	2287.9834	2225.5516	2528.9172	2371.1426
04	2129.6757	2427.4679	2349.8574	2293.1954	2550.7962	2398.4233
Hill Climbing algorithm improving random solutions						
Dataset	**Best result**		**Worse result**		**Average result**	
01	1850.9431		2071.8965		1934.6546	
02	1800.7846		2203.3704		1915.8581	
03	2766.0555		3299.6057		3090.4349	
04	2752.9495		3282.1967		3050.4361	

The Hill Climbing algorithm with Nearest Neighbour algorithm provided good solutions. The Hill Climbing algorithm with Arbitrary Insertion algorithm is also efficient. The Arbitrary Insertion algorithm gave better solutions than the Nearest Neighbour algorithm when improvement algorithms are not applied. When improvement algorithms are applied then both algorithms are equally effective.

The improvement algorithms gave better solutions if construction algorithms are improved then random solutions are improved. Improving Arbitrary Insertion algorithm with Genetic algorithm and Hill Climbing algorithm is effective. About from 2000-2500 travel units the solutions were improved with 500 travel unit applying improving algorithms. With improving the Nearest Neighbour algorithm from the 2200-2500-3000 travel unit about with 700-1000 travel unit the solution is improved.

Figure 6 demonstrates how the total distance depends on the number of customers. The figure relates the solutions given by only construction heuristics, the improvement of construction heuristics and random solutions. The number of customers was taken between 10-45, increasing by five. The number of satellites was four, and the number of recharge stations was ten. Test results were created for two datasets. The two datasets were altered by the position of the customers. In Figure 6 the AI means the Arbitrary Insertion algorithm, the HC means random solutions improved by Hill Climbing algorithm, the HC_AI means the Arbitrary Insertion algorithm improved by Hill Climbing algorithm, the HC_NN means the Nearest Neighbour algorithm improved by Hill Climbing algorithm, the GA means random solutions improved by Genetic algorithm, the GA_AI_NN means random solutions, Arbitrary Insertion algorithm and Nearest Neighbour algorithm improved by Genetic Algorithm, and the NN means Nearest Neighbour algorithm.

Based on running results it can be concluded, that using only construction algorithms we get weaker results. Using both construction and improvement algorithms (and using only improvement algorithms) with small number of customers (10-20) the algorithms are efficient.

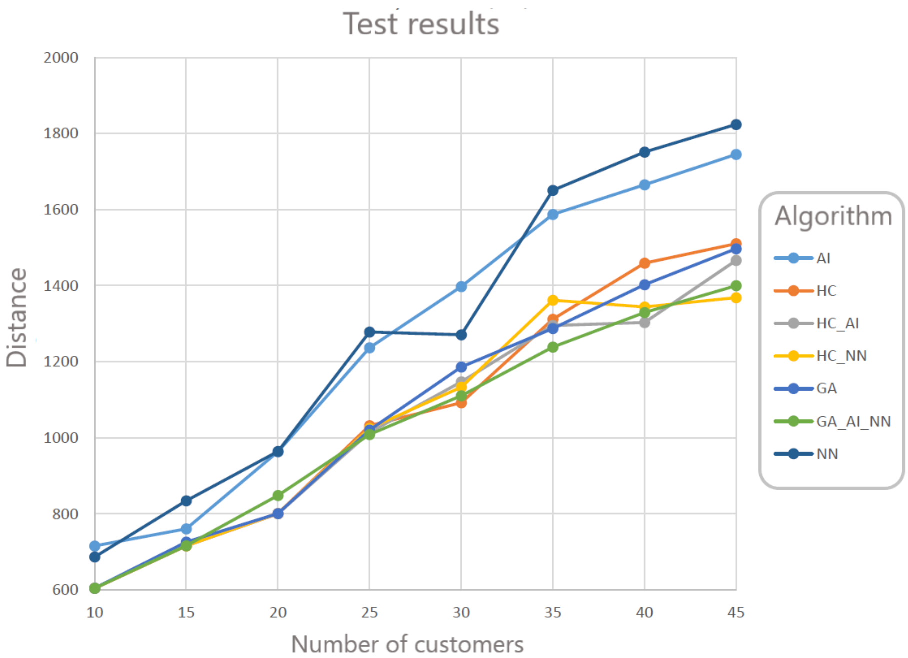

Figure 11. Running results depending on the number of customers

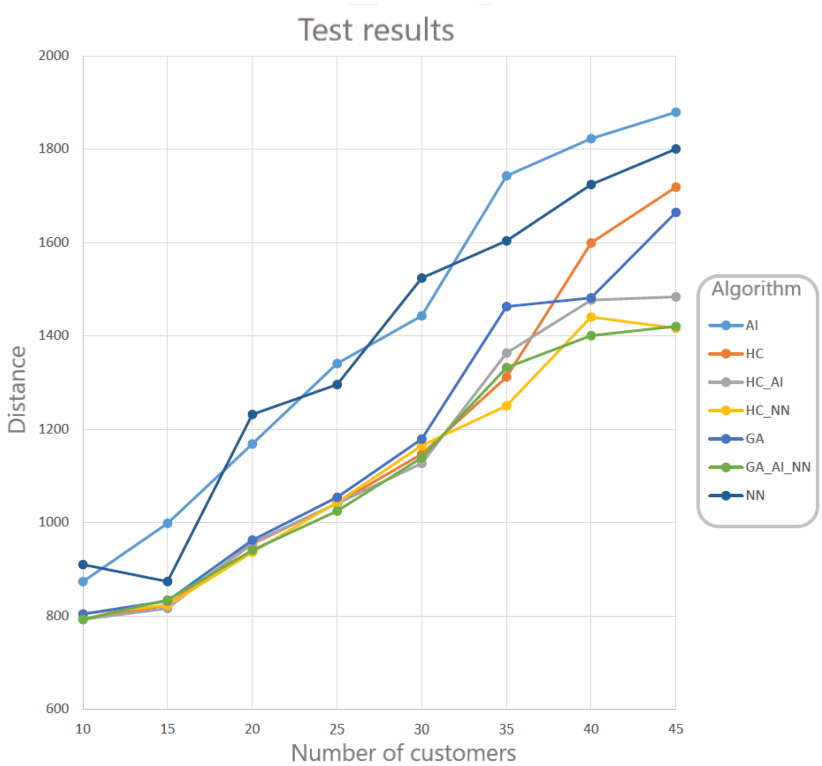

Figure 12. Running results depending on the number of customers

7. Conclusions

Our investigation is the Two-Echelon Vehicle Routing Problem (2E-VRP). In 2E-VRP, the products are delivered not immediately from to the depot to the customers. The products are delivered from the depot to intermediate location called satellites. From the satellites, the products are delivered to the customers. In 2E-VRP, there are two types of vehicles. The first type of the vehicles delivers the product from the depot to the satellites. The second type of the vehicles delivers the product from the satellites to the customers. We extended the problem with recharge stations because the second type of the vehicles is autonomous vehicles, they must be recharged. In our model, the second level vehicles start their route from one recharge station, then travel to the satellite, then visits customers, and after visited the customers returns to the recharge station.

We solved the extension of 2E-VRP with construction and improvement heuristics. The construction heuristics were the Nearest Neighbour algorithm and the Arbitrary Insertion algorithm. The improvement algorithms were the Hill Climbing algorithm and the Genetic algorithm. Test results are evaluated. Based on the test results the improvement of the construction heuristics was the best way to solve the problem.

Acknowledgements

Supported BY the ÚNKP-18-2-I New National Excellence Program of the Ministry of Human Capacities.

References

1. Braun, H. (1990) On solving travelling salesman problems by genetic algorithms. *International Conference on Parallel Problem Solving from Nature*, 129-133.
2. Cordeau, J. F., Gendreau, M., and Laporte, G. (1997) A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks: an International Journal,* 30(2), 105-119. DOI:10.1002/(SICI)1097-0037(199709)30:23.3.CO;2-N.
3. Crainic, T. G., Perboli, G., Mancini, S., and Tadei, R. (2010) Two-echelon vehicle routing problem: a satellite location analysis. *Procedia-Social and Behavioral Sciences,* 2(3), 5944-5955. DOI:10.1016/j.sbspro.2010.04.009.
4. Dantzig, G.B., and Ramser, J.H. (1959) The Truck Dispatching Problem. *Management Science,* 6(1), 80–91. DOI:10.1287/mnsc.6.1.8.
5. Desrochers, M., Desrosiers, J., and Solomon, M. (1992) A new optimization algorithm for the vehicle routing problem with time windows. *Operations Research,* 40(2), 342-354. DOI:10.1287/opre.40.2.342.
6. Figliozzi, M. (2010) Vehicle routing problem for emissions minimization. *Transportation Research Record: Journal of the Transportation Research Board,* (2197), 1-7. DOI:10.3141/2197-01.
7. Gendreau, M., Laporte, G., and Séguin, R. (1996) Stochastic vehicle routing. *European Journal of Operational Research,* 88(1), 3-12. DOI:10.1016/0377-2217(95)00050-X.
8. Katayama, K., Sakamoto, H., and Narihisa, H. (2000) The efficiency of hybrid mutation genetic algorithm for the travelling salesman problem. *Mathematical and Computer Modelling,* 31(10-12), 197-203. DOI:10.1016/S0895-7177(00)00088-1.
9. Li, C. L., Simchi-Levi, D., and Desrochers, M. (1992) On the distance constrained vehicle routing problem. *Operations Research*, 40(4), 790-799. DOI:10.1287/opre.40.4.790.
10. Li, F., Golden, B., and Wasil, E. (2007) The open vehicle routing problem: Algorithms, large-scale test problems, and computational results. *Computers & Operations Research*, 34(10), 2918-2930. DOI:10.1016/j.cor.2005.11.018.
11. Maulik, U., and Bandyopadhyay, S. (2000) Genetic algorithm-based clustering technique. *Pattern Recognition,* 33(9), 1455-1465. DOI:10.1016/S0031-3203(99)00137-5.
12. Min, H. (1989) The multiple vehicle routing problem with simultaneous delivery and pick-up points. *Transportation Research Part A: General,* 23(5), 377-386. DOI:10.1016/0191-2607(89)90085-X.
13. Nagy, G., and Salhi, S. (2005) Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries. *European Journal of Operational Research,* 162(1), 126-141. DOI:10.1016/j.ejor.2002.11.003.
14. Perboli, G., Tadei, R., and Vigo, D. (2011) The two-echelon capacitated vehicle routing problem: models and math-based heuristics. *Transportation Science,* 45(3), 364-380. DOI:10.1287/trsc.1110.0368.

15. Ralphs, T. K., Kopman, L., Pulleyblank, W. R., and Trotter, L. E. (2003) On the capacitated vehicle routing problem. *Mathematical programming,* 94(2-3), 343-359. DOI:10.1007/s10107-002-0323-0.
16. Renaud, J., Laporte, G., and Boctor, F. F. (1996) A tabu search heuristic for the multi-depot vehicle routing problem. *Computers & Operations Research,* 23(3), 229-235. DOI:10.1016/0305-0548(95)O0026-P.
17. Rosenkrantz, D. J., Stearns, R. E., and Lewis, II, P. M. (1977) An analysis of several heuristics for the traveling salesman problem. *SIAM journal on computing,* 6(3), 563-581. DOI:10.1137/0206041.
18. Starkweather, T., McDaniel, S., Mathias, K. E., Whitley, L. D., Whitley, C. (1991) A Comparison of Genetic Sequencing Operators. *ICGA* 69-76.
19. Tamás, P. (2017) Decision Support Simulation Method for Process Improvement of Intermittent Production Systems. *Sustainability*, 7(9), 950, DOI:10.3390/app7090950.