ISSN: 1803-3814 (Printed), 2571-3701 (Online)
https://doi.org/10.13164/mendel.2019.1.015

HEURISTICS FOR WASTE COLLECTION ARC ROUTING PROBLEM

Vlastimír Nevrlý✉**, Radovan Šomplák, Pavel Popela**

Institute of Process Engineering, Faculty of Mechanical Engineering, Brno University of Technology, Czech Republic
Vlastimir.Nevrly@vutbr.cz✉

Abstract

Waste management is still an expanding field which needs to be constantly enhanced so that waste transportation and treatment is as effective as possible. An important part of this process is a waste collection at the municipal level. Decision-making about daily routing for all vehicles from a heterogenous fleet substantially influences the expenses of technical services. The need of route scheduling comes also from the newly separated fractions. Transportation features include the capacity of vehicles, number and type of containers on the route, traffic light delays and many others. The mathematical model that properly describes the real practice of servicing containers has not been published yet. Moreover, routing problems are generally not solvable by exact methods, so the appropriate heuristic algorithm has been developed. A case study with obtained results is discussed. This solution serves not only to improve the current operational situation, but also to create new route schedules for increasing number of collected commodities.

Keywords: *waste management, combinatorial optimization, heuristics, routing problem, waste collection, arc routing.*

Received: 30 April 2019
Accepted: 31 May 2019
Published: 24 June 2019

1 Introduction

Effective waste management contributes to economic and environmental sustainability. As a result of increasing operational requirements and changing systems, the development of new methods in various areas is necessary. For some waste producers (municipalities), the collection plan is simple because of the small scale of the problem. Using elementary knowledge and calculations, a weekly collection plan can be created for the vehicle and adapted to the current season and needs throughout the year. However, this statement can only be made for small areas where dozens of containers of a particular type of waste occur. For larger territories, it is advisable to move to more sophisticated methods, most often based on mathematical programming and computational algorithms. An extensive up-to-date research in the field of operational research was addressed in [1], where opportunities and challenges for further development were identified. The focus of the article, in addition to the number of publications in each sector, was more general in nature, where decision-making and planning were divided into strategic, tactical and operational, also with regards to data access (deterministic, stochastic or robust).

The complex point of view for the waste management was presented in [2]. In [3] was dealt with mobile waste collection, while places to stop the vehicle were analysed. Collecting mobile sites have been selected here based on the demand of citizens for specific time windows. Four different heuristic algorithms - Tabu Search, Greedy Algorithm, Simulated Annealing (SA), Bee Colony Optimization - were tested to solve the problem. According to testing on 3 different real instances with installation and collection infrastructure (Tokyo, Philadelphia, Warsaw), the SA algorithm proved to be the most suitable for the given problem. The proposed solution to the task was to reduce the costs and emissions generated by the waste transfer. It may be useful sites for identifying warehouse, depots or waste treatment plant for the area. The facility location problem represents a challenging task, especially, when randomness is involved, the genetic algorithm showed its strength in combination with the Benders decomposition [4].

Existing systems also have drawbacks due to poor cooperation between different areas. In the case of the collection and subsequent logistic chain of packaging waste, it may be the identification of suitable locations for the warehouses, depots or waste treatment plants for the area. The paper [5] analysed the impact of the target site for waste disposal at collection costs. Several smaller areas have been integrated in the calculation to identify the potential for change. Individual collection containers were aggregated into larger areas to simplify the collection section to allow for a more macroscopic view, while the capacities of both transfer stations and terminal equipments were also taken into account. This was followed by an analysis of collection planning for recyclable waste from an economic and environmental point of view [6].

Different systems of curbside collection were analysed in [7]. Multi compartment vehicles might be desirable for some areas, however, transport requirements depends on other factors such as vehicle size, compression factors, and packing. Plastic waste collection was modelled in [8] through vehicle routing problem in terms of sustainability. Scenarios of different system were compared from cost point of view for the real world case.

Waste operations may also depend on their type and properties, e.g., the collection of kitchen waste was planned using a heuristic algorithm in [9]. A tailor-made procedure for this type of waste has been presented because of its properties such as high mass versus volume or large variation in demand between collection points (restaurants, bars, supermarkets, etc.). The test calculations of this algorithm are shown in the historic center of Genoa and compared with the currently used plan with partial savings.

Savings can also be ensured by a robust waste collection view. This was introduced in [10], where Periodic Capacitated Arc Routing Problem (PCARP) was formulated. The solution was found using the Simulated Annealing algorithm, with the demand for waste collection being considered randomly. The vehicle routing problem was solved using multi-chaotic variant of differential evolution in [11] and for capacited version in [12]. The multi-period fleet size and mix vehicle routing problem with stochastic demands was presented to make strategic fleet sizing decisions [13].

The models deal with a specific problems such as periodicity, different fleet, different capacities. However, there is no combination of Multi-Processor Multi-Trip PCARP with Heterogeneous Fleet in the literature that describes the real problem for solving tasks in the context of collection companies in the Czech Republic. Another novelty is a way of describing a task and simplifying it in the context of quality preprocessing, which greatly simplifies the model, different fractions, type of edges, etc.

2 Problem Formulation

Assuming a heterogeneous fleet (each vehicle has a different capacity) with vehicles that stand at depot. A time horizon defines days in which a vehicle can be operated (working days) and also the time of working shift is limited. Vehicles can operate within the streets during the working hours, while repeated return to processing facilities is allowed. The routing is performed on the oriented graph that describes the urban infrastructure transport network in which we have both bidirectional edges and unidirectional edges. Edges can be divided into four groups. Edges representing the exit from the depot, level crossing edges, edges with demand for collection and processing edges, i.e. edges where waste is disposed of (disposal at the landfill, incineration at the Waste-to-Energy plant or other). All considered edges do not have to be served.

A positive demand for waste collection is assigned to demand edges. If the edge is defined for both directions (the arc and its inverse arc), only one has to be serviced. Each demand edge has a specified frequency with which the waste must be collected. Based on this parameter, possible scenarios are constructed with regard to a reasonable time gap between the collection days. The goal of the mathematical model is to find scenarios for all edges and to assign vehicles in order to minimize total costs with respect to time and capacity constraints.

2.1 Notation

To formulate a mathematical model, sets, parameters, and variables must be defined. The notation and symbol description is stated in the following.

Sets	
$\mathcal{I}$	set of working days for collection, i.e., time horizon,
$\mathcal{J}$	vehicle fleet, set of available vehicles,
$\mathcal{A}$	set of all arcs,
$\mathcal{N}$	set of demand arcs and depot,
$\mathcal{N}^0$	set of demand arcs,
$\mathcal{M}$	set of processing arcs,
$\mathcal{M}_0$	set of processing arcs and depot,
$\mathcal{S}$	set of possible servicing scenarios,
$\mathcal{S}(k)$	subset of scenarios $\mathcal{S}$ induced by demand arc k,

Parameters	
$c^D_{k,l}$	transport cost from arc k to arc l $(k, l \in \mathcal{A})$,
c^S_k	servicing cost of demand arc k $(k \in \mathcal{N}^0)$,
d_l	demand of arc l $(l \in \mathcal{N})$,
Q_j	vehicle capacity j $(j \in \mathcal{J})$,
$t^D_{k,l}$	transport time from arc k to arc l $(k, l \in \mathcal{A})$,

t_k^S servicing time of arc k ($k \in \mathcal{N}^0$),

t_k^P servicing time of processing arc k ($k \in \mathcal{M}$),

$m_{k,l}^I$ indicator matrix element, i.e., existence of inverse arc l with arc k ($k, l \in \mathcal{N}^0$),

$$g_{i,s} \quad \begin{cases} 1 & \text{colection is performed on day } i \text{ in scenario } s\ (i \in \mathcal{I},\ s \in \mathcal{S}) \\ 0 & \text{otherwise,} \end{cases}$$

M big constant, i.e., number used as limit on constraints (activation and deactivation),

T daily working hours (available time),

Variables

$L_{i,j,k}$ maximal load of vehicle j on day i after servicing of arc k, ($i \in \mathcal{I},\ j \in \mathcal{J},\ k \in \mathcal{A}$),

$$x_{i,j,k,l} \quad \begin{cases} 1 & \text{if arc } l \text{ is served after arc } k \text{ by vehicle } j \text{ on day } i \\ & (i \in \mathcal{I},\ j \in \mathcal{J},\ k, l \in \mathcal{A}) \\ 0 & \text{otherwise,} \end{cases}$$

$$\delta_{i,j,k} \quad \begin{cases} 1 & \text{if demand arc } k \text{ is served by vehicle } j \text{ on day } i \\ & (i \in \mathcal{I},\ j \in \mathcal{J},\ k \in \mathcal{N}^0) \\ 0 & \text{otherwise,} \end{cases}$$

$$y_{k,s} \quad \begin{cases} 1 & \text{if demand arc } k \text{ uses scenario } s \\ & (k \in \mathcal{N}^0,\ s \in \mathcal{S}(k)) \\ 0 & \text{otherwise.} \end{cases}$$

2.2 Model Formulation

Using the notation from 2.1, a mathematical model that matches the given problem can be formulated with objective function considering cost as in equation (1). This model will find a scenario for each demand edge and edge sequence to be served for each vehicle for all days considered so that the total cost is minimal.

$$\min \sum_{i\in\mathcal{I}} \sum_{j\in\mathcal{J}} \sum_{k\in\mathcal{A}} \sum_{l\in\mathcal{A}} c_{k,l}^D x_{i,j,k,l} + \sum_{i\in\mathcal{I}} \sum_{j\in\mathcal{J}} \sum_{k\in\mathcal{N}^0} c_k^S \delta_{i,j,k} \tag{1}$$

s.t.

$$\sum_{k\in\mathcal{A}} x_{i,j,k,h} - \sum_{l\in\mathcal{A}} x_{i,j,h,l} = 0, \qquad \forall i \in \mathcal{I},\ \forall j \in \mathcal{J},\ \forall h \in \mathcal{A}, \tag{2}$$

$$L_{i,j,k} + d_l - L_{i,j,l} \le (1 - x_{i,j,k,l})M, \qquad \forall i \in \mathcal{I},\ \forall j \in \mathcal{J},\ \forall k \in \mathcal{A},\ \forall l \in \mathcal{N}, \tag{3}$$

$$L_{i,j,k} \le Q_j, \qquad \forall i \in \mathcal{I}, \forall j \in \mathcal{J}, \forall k \in \mathcal{N}^0, \tag{4}$$

$$L_{i,j,k} = 0, \qquad \forall i \in \mathcal{I}, \forall j \in \mathcal{J}, \forall k \in \mathcal{M}_0, \tag{5}$$

$$\sum_{l\in\mathcal{A}} x_{i,j,0,l} \le 1, \qquad \forall i \in \mathcal{I}, \forall j \in \mathcal{J}, \tag{6}$$

$$\sum_{k\in\mathcal{A}} \sum_{l\in\mathcal{A}} x_{ijkl} \le M \sum_{l\in\mathcal{A}} x_{i,j,0,l}, \qquad \forall i \in \mathcal{I},\ \forall j \in \mathcal{J}, \tag{7}$$

$$\sum_{k\in\mathcal{A}} \sum_{l\in\mathcal{A}} t_{k,l}^D x_{i,j,k,l} + \sum_{k\in\mathcal{N}^0} t_k^S \delta_{i,j,k} + \sum_{k\in\mathcal{A}} \sum_{l\in\mathcal{M}} t_l^P x_{i,j,k,l} \le T, \qquad \forall i \in \mathcal{I},\ \forall j \in \mathcal{J}, \tag{8}$$

$$\delta_{i,j,l} \le \sum_{k\in\mathcal{A}} x_{i,j,k,l}, \qquad \forall i \in \mathcal{I},\ \forall j \in \mathcal{J},\ \forall l \in \mathcal{N}^0, \tag{9}$$

$$\sum_{k\in\mathcal{A}} x_{i,j,k,l} \le \delta_{i,j,l} M, \qquad \forall i \in \mathcal{I}, \forall j \in \mathcal{J}, \forall l \in \mathcal{N}^0, \tag{10}$$

$$m_{k,l}^I (y_{k,s} - y_{l,s}) = 0, \qquad \forall k, l \in \mathcal{N}^0 : l > k, \forall s \in \mathcal{S}(k), \tag{11}$$

$$\sum_{s\in\mathcal{S}(k)} y_{k,s} = 1, \qquad \forall k \in \mathcal{N}^0, \tag{12}$$

$$\sum_{s\in\mathcal{S}(k)} g_{i,s} y_{k,s} - \sum_{j\in\mathcal{J}} (\delta_{i,j,k} + \sum_{l\in\mathcal{N}^0} m_{k,l}^I \delta_{i,j,l}) = 0, \qquad \forall k \in \mathcal{N}^0,\ \forall i \in \mathcal{I}, \tag{13}$$

$$L_{i,j,k} \geq 0, \qquad \forall i \in \mathcal{I}, \forall j \in \mathcal{J},\ \forall k \in \mathcal{A}, \tag{14}$$

$$x_{i,j,k,l} \in \{0,1\}, \qquad \forall i \in \mathcal{I},\ \forall j \in \mathcal{J},\ \forall k,l \in \mathcal{A}, \tag{15}$$

$$\delta_{i,j,k} \in \{0,1\}, \qquad \forall i \in \mathcal{I},\ \forall j \in \mathcal{J},\ \forall k \in \mathcal{N}^0, \tag{16}$$

$$y_{k,s} \in \{0,1\}, \qquad \forall k \in \mathcal{N}^0,\ \forall s \in \mathcal{S}(k). \tag{17}$$

The objective function (1) minimizes the cost for transportation through the streets (both servicing and transferring), which are generally assumed different for mutually inverse arcs. This means that in terms of costs, it is distinguished between the direction in which the street is passed. This can be caused, for example, by a steep rise in the street. The equations (2) ensure continuity between successive arcs i.e. if the vehicle enters the arc, it must also leave it. Constraints (3) prevent the creation of forbidden subtours and, at the same time, recalculate the current vehicle load after visiting the given arc. In the next equations (4), the vehicle load is limited by its capacity. Constraints (5) indicate that the vehicle load in processing plants and in depot is zero. The equations (6) ensure that every vehicle can start the trip from depot at most once per day of the time horizon. However, the number of collection routes is not limited - the waste can transported to processing facilities multiple times (when all other constraints are met). If the vehicle services any arc, then it has to leave the depot (7). Operating time of the vehicle is restricted in equations (8), while counting transportation time, servicing time and a waste unloading time at the processing plant. Constraints (9) and (10) describe the relationship between $\delta_{i,j,k}$ and $x_{i,j,k,l}$. One binary variable indicates the other and vice versa, while for $x_{i,j,k,l}$ the value is aggregated. Arcs that are inverse to each other must met equations (11), where the same collection planning scenario is chosen for both. The choice of just one scenario for each demand arc is provided by the equation (12). The equations (13) define the collection plan for all arcs. The selected scenario is applied for demand arcs or their inversions, if they exist. The equations (14) indicate the non-negativity of the load. Finally, (15), (16) and (17) introduce variables as binary.

3 Problem Solution

Mathematical models such as the one in the previous section are very difficult to solve [14]. Since there are enormous number of binary variables for the real instance of the model, it is preferable to use heuristic approach. The input data such as the matrix of distances and times has to be defined prior the calculation, i.e. the distance of the end node of one arc to the initial node of the second arc. The shortest path among all arcs has to be found. The Dijkstra algorithm is used in this step [15]. This procedure eliminates the crossing arcs. To get the final route the predecessor matrix is used [14]. It defines the sequence of arcs across the shortest path. The algorithm of route computation is described in the next subsection.

3.1 Algorithm

The proposed algorithm is based on a combination of a genetic algorithm with local search (LS). Programming language C++ was used for its implementation. Specifically, the idea of maintaining two separate populations was derived. Individuals were kept in a set of feasible and infeasible solutions. In addition, the so-called biased fitness and diversification were used to select individuals who survive and go to the next generation. Solution representation is explicit here to avoid complex and time-consuming decoding processes while running the algorithm. The solution is then in the form of demand arc sequences for each route of each vehicle for all days of the time horizon. Due to the crossover process, the processing facilities and the depot are not included in this sequence, but given implicitly. The key component of this algorithm is the fitness function, which is considered for solution Sol as follows:

$$fit(S) = E(S) + w^Q v^Q(S) + w^T v^T(S), \tag{18}$$

where $v^Q(S)$ and $v^T(S)$ are exceeded values of capacity and time, w^Q and w^T are the respective penalization and $E(S)$ defines the total cost.

The initialization of populations is performed through two algorithms that create reasonable routes, i.e. Route-Merging and Path-Scanning, see [16]. Algorithm 1 describes the main scheme in the pseudocode.

The stopping criterion depends on the decision, it can be the total number of iterations, the running time of the algorithm, the number of diversification operations performed, or a combination of the previous ones. The crossover process is a phase where two P^1 and P^2 ancestors are selected to create a new individual C. This selection is for both ancestors realized by binary tournament, i.e. two individuals are randomly selected from the entire population (merge of feasible and infeasible populations) and choose the one with the better (less) fitness.

Algorithm 1 General scheme of the algorithm

```
Initialize population;
while stopping criteria is not met do
    Select parents $P^1$ and $P^2$
    Perform crossover of $P^1$ and $P^2$ to yield offspring $C$
    Perform local search or repair of $C$ with probabilities $P_{LS}$, resp. $P_{Rep}$
    if $C$ is feasible then Insert $C$ into feasible subpopulation
    end if
    if $C$ is infeasible then Insert $C$ into infeasible subpopulation
    end if
    if One of the subpopulation reached its maximum size then
        Perform survivor selection with this subpopulation
    end if
    if best solution not improved for $It_{DIV}$ iterations then
        Diversify population
    end if
end while
return best feasible solution
```

These two individuals then enter their own process of crossover, which begins by selecting the day from the time horizon i. On this day, the vehicle is randomly picked from both and their routes are combined to get a new individual C. At this stage, the assignment of the scenario to demand arcs must be performed in order to meet all requirements of individual C.

After crossover, there is a phase of local search or repair of the solution. These adjustments try to make solutions feasible and to improve their fitness. There are considered three possible movements from the initial solution.

- *Single Insertion* (SI): moves the demand arc from the sequence to another position.
- *Double Insertion* (DI): moves two consecutive demand arcs from a sequence to another positions.
- *Swap*: the positions of two demand arcs are swapped.

All of these movements take place on the same day to keep the assigned scenario. Both possible directions (inversion to arc) are tested. These movements are performed in random order while the solution is being improved. The repair phase uses local search and, in addition, it adds penalty parameters w^Q and w^T to the fitness function during iterations to emphasize the importance of getting a feasible solution.

Another part of the algorithm is the survivor selection, which is performed when a specified number of individuals in one population is reached. The aim is to reduce the size of the population to a certain level, and then resume with crossover. The main priority is to remove the clones, which are very similar individuals. Having two similar individuals in the population has no benefit in terms of the genetic algorithm. Hamming distance is used to find similarity, see [17]. Furthermore, the biased fitness is introduced to select individuals who will be removed from the population. This selection is based on two characteristics of how good a given solution is and how much it contributes to the diversity of the population. The last procedure to manage the population is diversification. This process is performed when there is no improvement during a certain number of iterations. From both populations, some individuals are removed by selecting survivors and replaced by several new individuals using the initialization phase. This process will dramatically change the diversity of the population and revive further searches.

The procedure of SI movement is described in detail, while the others are based on the similar idea. The process begins by selecting the day from the time horizon where the SI will take the part. It goes through each arc of the sequence of all vehicles to test the improvement. For tested arc TA (two consecutive arcs in the case of DI), the fitness contribution is calculated according to the equation:

$$contribution = c^D_{m,TA} + c^S_{TA} + c^D_{TA,n} - c^D_{m,n}, \tag{19}$$

where m and n are the previous and the next arcs in the sequence.

Then the neighbourhood $N(TA)$ is searched and for the position n of each arc is calculated contribution of arc TA. If the contribution at new position is lesser than the original one, then the solution is evaluated and its feasibility checked (vehicle capacity and operating time). The pseudocode of the algorithm is displayed in Algorithm 2.

Algorithm 2 Local Search algorithm

```
Choose $i \in Sol$
for all $j, k, l \in Sol[i]$ do
    $TA := Sol[i, j, k, l]$
    Calculate $oldContribution$
    for all $n \in N(TA) : n$ used in $Sol[i]$ do
        Calculate $newContribution$
        if $newContribution < oldContribution$ then
            Copy old solution and perform move
            Check feasibility
            if feasible then
                Replace by new sub-solution and return improved Solution
            end if
        end if
    end for
end for
```

The second most challenging operation is crossover. Similarly to LS, the method utilises more accurate neighbourhood and recalculation with contribution.

The inputs of the algorithm are two ancestors P^1 and P^2. The algorithm starts with random selection of day i from time horizon, vehicle $j_1 \in P^1[i]$ and vehicle $j_2 \in P^2[i]$. Then a new individual C is created by copying the P^1 solution, and its vehicle j_1 is replaced by a j_2 vehicle while the capacity of individual C remains the same. Arcs occurring multiple times at day i are deleted from the solution. The change of scenarios for all arcs of $P^1[i, j_1] \cup P^2[i, j_2]$ follows according to the ancestor P^2 and removing the arcs that have changed. Now a set R that contains all missing arcs is created.

For the next steps, the parameters $conCost$, $conTvQ$ and $conTvT$ are introduced. This means the contribution of the arc, the increase in capacity constraint violation and the increase in time constraint violation. There are also parameters with the B flag, indicating the same properties for the best n^B position to insert.

The algorithm then goes through all the non-assigned arcs $e \in R$, and for each of them selects from the neighbourhood $N(e)$ the arc position n^B such that all the previously mentioned properties are the smallest. This position is then used for insertion of arc e and the fitness function is recalculated. Algorithm 3 describes the whole approach.

Algorithm 3 Crossover algorithm

```
Input: $P^1, P^2$
Randomly choose $i \in \mathcal{I}$
Randomly choose $j_1 \in P^1[i]$ and $j_2 \in P^2[i]$
$C := P^1$
Replace $j_1$ of $C$ by $j_2$ and remove redundant tasks in period $i$ of $C$
for all $e \in P^1[i, j_1] \cup P^2[i, j_2]$ do
    replace scenarios for $e$ of $C$ with that in $P^2$
end for
Remove all tasks of $C$ whose scenario was changed
Add all missed tasks to set $R$
for all $e \in R$ do
    $BconCost, BconTvQ, BconTvT, n^B := infinity$
    for all $n \in N(e)$ : scenarios of $e$ and $n$ are equal do
        Calculate conCost, conTvQ and conTvT
        if $conCost \leq BconCost \wedge conTvQ \leq BconTvQ \wedge conTvT \leq BconTvT$ then
            $BconCost := conCost$
            $BconTvQ := conTvQ$
            $BconTvT := conTvT$
            $n^B := n$
        end if
    end for
    Insert $e$ to position of $n^B$ and recalculate fitness of $C$
end for
```

In this way, a new individual C is created. This individual is further sent to other procedures - LS or Repair, depending on the feasibility of the given solution, see algorithm 1.

An adaptive parameter control has been introduced to improve the algorithm's performance. This control changes the parameters w^Q and w^T according to the ratio of frequencies of feasible and infeasible solutions. The smaller the proportion is (generating more infeasible solutions), the greater are the penalties, and vice versa.

4 Testing

The presented algorithm was tested on the real network of Jihlava. The considered network contains 1,467 vertices and 3,529 edges. Other input parameters (costs of operation and capacities of vehicles, production and demand for waste collection) were appropriately generated on the basis of demographic data and waste management expert estimates. The best performance was obtained with probabilities 0.57 and 0.61 for P_{LS}, resp. P_{Rep}. The need of randomness in evolutionary algorithms was discussed and numerically tested in [18]. The functionality and speed of the algorithm were studied on these data. The maximum population size was set to 100 for both feasible and infeasible. The average calculation time per iteration was 0.69 second. Fitness value development was highly dependent on the penalization rates (w^Q and w^T). Initially, the algorithm quickly improves the solution but with increasing iterations, the improvement interval increases. Once the solution has not been improved for 30 iterations, a diversification procedure is introduced, which adds diversity to the population and thus revives the entire crossover procedure.

The real case study was conducted for another city with 8,594 nodes. The calculation was performed for specific district and route, where 264 arcs including inverse arcs were linked with 160 containers for paper waste collection. The output was compared with the currently operated route and savings were identified. The operated route distance is 71.4 km, while the result of the algorithm suggested route of 66.7 km length. This reveals the potential of almost 7 % of savings, which corresponds with 1,400 EUR annually per one route.

5 Conclusion

An advanced tool for supporting planning in the field of waste logistics at a micro-regional level is needed to effectively implement and meet goals. Tasks from waste collection mostly fall into operations research field. However, such supply chain models are very complex and have hight time requirements for computing the solution, which is mainly due to the combinatorial nature of the problem. Thus, it is necessary to proceed with a heuristic approach. In this paper, the combination of genetic algorithm and local search was presented in pseudocodes and the whole approach was described. These tasks are also demanding from the input data point of view, therefore it is necessary to perform thorough preprocessing and statistical evaluation. Tasks related to waste collection, fleet sizing, locating containers, and assessing the overall economic and environmental impact must meet operational requirements to credibly model the reality.

The use of defined tasks is designed to support investment decision-making and project implementation, primarily by minimizing costs, impact on the population or the environment. The heuristic algorithm was used for real infrastructure and better performance was found for already operated routes. Potential savings of almost 7 % were identified for studied problem. If such lesser cost would be realizable for all routes and vehicles during the year, it would be saved up to tens of thousands of euros on costs.

New limitations and decision criteria can be implemented in the algorithm as well as in the model. In heuristic algorithms, the fitness for emissions, costs and time can be defined and compared in the future. In addition, the option for collection days might be added. One variant considers fixed collection days according to the current plan. In the second variant, the aim is to create a schedule of collection days for all containers. So far, allowed combinations of collection days have been generated based on frequency. Identical solutions often appear in the current population of acceptable solutions, so the next challenge is to prevent the creation of clones that already exist in the population. Thus, a clone test may be performed before a new solution is included in the population.

Acknowledgement: The authors gratefully acknowledge financial support provided by ERDF within the research project No. CZ.02.1.01/0.0/0.0/16_026/0008413 "Strategic Partnership for Environmental Technologies and Energy Production". The authors also sincerely thank for the financial support given by the Ministry of Education, Youth and Sports of the Czech Republic under the standard specific research project FCH/FSI-J-19-5794.

References

[1] Barbosa-Póvoa, A. P., da Silva, and C., Carvalho, A. 2018. Opportunities and challenges in sustainable supply chain: An operations research perspective. *European Journal of Operational Research* 268, 2, pp. 399–431. DOI: 10.1016/j.ejor.2017.10.036

[2] Málek, M., Šomplák, R., Popela, P., and Kůdela, J. 2018. Stochastic integer waste management problem solved by a modified progressive hedging algorithm. *Mendel* 24, 2, pp. 17–22.

[3] Nowakowski, P., Szwarc, K., and Boryczka, U. 2018. Vehicle route planning in e-waste mobile collection on demand supported by artificial intelligence algorithms. *Transportation Research Part D: Transport and Environment* 63, pp. 1–22. DOI: 10.1016/j.trd.2018.04.007

[4] Kůdela, J. and Popela, P. 2018. Two-stage stochastic facility location problem: GA with benders decomposition. *Mendel* 21, pp. 53–58.

[5] Ramos, T. R. P., Gomes, M. I., and Barbosa-Póvoa, A. P. 2014. Assessing and improving management practices when planning packaging waste collection systems. *Resources, Conservation and Recycling* 85, pp. 116–129. DOI: 10.1016/j.resconrec.2013.12.013

[6] Ramos, T. R. P., Gomes, M. I., and Barbosa-Póvoa, A. P. 2014. Economic and environmental concerns in planning recyclable waste collection systems. *Transportation Research Part E: Logistics and Transportation Review* 62, pp. 34–54. DOI: 10.1016/j.tre.2013.12.002

[7] Zbib, H. and Wøhlk, S. 2019. A comparison of the transport requirements of different curbside waste collection systems in Denmark. *Waste Management* 87, pp. 21–32. DOI: 10.1016/j.wasman.2019.01.037

[8] Bing, X., de Keizer, M., Bloemhof-Ruwaard, J. M., and van der Vorst, J. G. A. J. 2014. Vehicle routing for the eco-efficient collection of household plastic waste. *Waste Management* 34, 4, pp. 719–729. DOI: 10.1016/j.wasman.2014.01.018

[9] Laureri, F., Minciardi, R., and Robba, M. 2016. An algorithm for the optimal collection of wet waste. *Waste Management* 48, pp. 56–63. DOI: 10.1016/j.wasman.2015.09.020

[10] Tirkolaee, E. B., Mahdavi, I., Esfahani, M. M. S. 2018. A robust periodic capacitated arc routing problem for urban waste collection considering drivers and crew's working time. *Waste Management* 76, pp. 138–146. DOI: 10.1016/j.wasman.2018.03.015

[11] Viktorin, A., Hrabec, D., and Pluháček, M. 2016. Multi-chaotic differential evolution for vehicle routing problem with profits. In *30th European Conference on Modelling and Simulation*. European Council for Modelling and Simulation (ECMS), Regensburg, pp. 245–251.

[12] Davendra, D., Zelinka, I., Bialic-Davendra, M., Šenkeřík, R., and Jašek, R. 2011. Discrete self organising migrating algorithm for the task of capacitated vehicle routing problem. *Mendel* 17, pp. 259–265.

[13] Pasha, U., Hoff, A., and Hvattum, L. M. 2018. The multi-period fleet size and mix vehicle routing problem with stochastic demands. *Computational Methods in Applied Sciences* 45, pp. 121–146. DOI: 10.1007/978-3-319-54490-8_9

[14] Wolsey, L. A. 1998. *Integer programming*. Wiley, New York.

[15] Dijkstra, E. W. 1959. A note on two problems in connexion with graphs. *Numerische Mathematik* 1, 1, pp. 269–271. DOI: 10.1007/BF01386390

[16] Golden, B. L., Dearmon, J. S., and Baker, E. K. 1983. Computational experiments with algorithms for a class of routing problems. *Computers and Operations Research* 10, 1, pp. 47–59. DOI: 10.1016/0305-0548(83)90026-6

[17] Vidal, T., Crainic, T. G., and Gendreau, M. 2012. A Hybrid Genetic Algorithm for Multidepot and Periodic Vehicle Routing Problems. *Operations Research* 60, 3, pp. 611–624.

[18] Zelinka, I., Šenkeřík, R., and Pluháček, M. 2013. Do evolutionary algorithms indeed require randomness?. In *2013 IEEE Congress on Evolutionary Computation*. IEEE, Cancun, Mexico, pp. 2283–2289. DOI: 10.1109/CEC.2013.6557841