

## A genetic algorithm for vehicle routing in logistic networks with practical constraints

Grzegorz Koloch,<sup>a</sup> Michał Lewandowski,<sup>b</sup> Marcin Zientara,<sup>c</sup> Grzegorz Grodecki,<sup>d</sup> Piotr Matuszak,<sup>e</sup> Igor Kantorski,<sup>f</sup> Adam Nowacki<sup>g</sup>

**Abstract.** We optimise a postal delivery problem with time and capacity constraints imposed on vehicles and nodes of the logistic network. Time constraints relate to the duration of routes, whereas capacity constraints concern technical characteristics of vehicles and postal operation outlets. We consider a method which can be applied to a brownfield scenario, in which capacities of outlets can be relaxed and prospective hubs identified. As a solution, we apply a genetic algorithm and test its properties both in small case studies and in a simulated problem instance of a larger (i.e. comparable with real-world instances) size. We show that the genetic operators we employ are capable of switching between solutions based on direct origin-to-destination routes and solutions based on transfer connections, depending on what is more beneficial in a given problem instance. Moreover, the algorithm correctly identifies cases in which volumes should be shipped directly, and those in which it is optimal to use transfer connections within a single problem instance, if an instance in question requires such a selection for optimality. The algorithm is thus suitable for determining hubs and satellite locations. All considerations presented in this paper are motivated by real-life problem instances experienced by the Polish Post, the largest postal service provider in Poland, in its daily plans of delivering postal packages, letters and pallets.

**Keywords:** rich vehicle routing problem, brownfield, hubs and satellites, genetic algorithm

**JEL:** C60, C61, L87

---

<sup>a</sup> SGH Warsaw School of Economics, Institute of Econometrics, Decision Analysis and Support Unit, ul. Madalińskiego 6/8, 02-513 Warszawa, e-mail: gkoloch@sgh.waw.pl, ORCID: <https://orcid.org/0000-0001-5506-2864>.

<sup>b</sup> SGH Warsaw School of Economics, Institute of Econometrics, Decision Analysis and Support Unit, ul. Madalińskiego 6/8, 02-513 Warszawa, e-mail: mlewan1@sgh.waw.pl, ORCID: <https://orcid.org/0000-0002-6003-1859>.

<sup>c</sup> SGH Warsaw School of Economics, Institute of Econometrics, Decision Analysis and Support Unit, ul. Madalińskiego 6/8, 02-513 Warszawa, e-mail: mziert@sgh.waw.pl, ORCID: <https://orcid.org/0000-0002-3984-2665>.

<sup>d</sup> SGH Warsaw School of Economics, Institute of Econometrics, Decision Analysis and Support Unit, ul. Madalińskiego 6/8, 02-513 Warszawa, e-mail: ggrode@sgh.waw.pl, ORCID: <https://orcid.org/0000-0001-6558-3217>.

<sup>e</sup> SGH Warsaw School of Economics, Institute of Econometrics, Decision Analysis and Support Unit, ul. Madalińskiego 6/8, 02-513 Warszawa, e-mail: pmatusz@sgh.waw.pl, ORCID: <https://orcid.org/0000-0002-5312-9698>.

<sup>f</sup> SGH Warsaw School of Economics, Institute of Econometrics, Decision Analysis and Support Unit, ul. Madalińskiego 6/8, 02-513 Warszawa, e-mail: ikanto@sgh.waw.pl, ORCID: <https://orcid.org/0000-0001-9870-0803>.

<sup>g</sup> SGH Warsaw School of Economics, Institute of Econometrics, Decision Analysis and Support Unit, ul. Madalińskiego 6/8, 02-513 Warszawa, e-mail: anowack@sgh.waw.pl, ORCID: <https://orcid.org/0000-0003-4635-6347>.

## 1. Introduction

We consider a realistic variant of a postal delivery problem with time and capacity constraints, which is a special case of a general vehicle routing problem (VRP) (see Toth & Vigo, 2002). Duration of routes is constrained since volumes need to be shipped on time (Spliet & Gabor, 2015). The capacity of vehicles is constrained, as Borčinová (2017) or Ralphs et al. (2003) notice, so only a limited volume of postal cargo can be shipped within each vehicle. Keskin et al. (2019) also assert that the capacity of postal operations outlets – nodes of the network – is constrained both in terms of vehicles that can be served by an outlet, and in terms of volumes that can be processed there.

Within any logistic plan, streams of volumes (demands) can be delivered directly or they can be shipped using some outlets as transfer locations, in which volumes are processed, merged and shipped to their final destinations. We are interested in a technology capable of identifying such outlets within the network, so that they can be recommended as prospective hubs when the network expands. It should be remembered that within a brownfield type of analysis, such prospective hubs do not have to overlap with outlets in which currently most of the volume is processed.

We use a genetic algorithm and argue that with a natural representation of solutions and with relatively simple specifications of genetic operators, subject to the representation in question, we are capable of obtaining such technology as that tested in small case studies and in a real-life problem instance. In particular, the employed genetic operators are capable of switching between direct and transfer connections, depending on what is optimal in a given problem instance, and also of selecting which volumes should be shipped directly and which should be shipped using transfer connections within a single problem instance – if the instance in question requires such a selection for optimality. All considerations presented in this paper are motivated by real-life problem instances which we analysed for the Polish Post (Poczta Polska), the largest, state-owned provider of postal services in Poland.

## 2. Problem Statement

The problem discussed in this paper can be considered as a version of a VRP which contains elements of various VRP specifications, e.g. VRPs with split pickups and split deliveries (see Casazza et al., 2018; Wassan & Nagy, 2014), VRPs with time windows (see Benjamin & Beasley, 2013; Bräysy & Gendreau, 2005), VRPs with capacity constraints (see Baldacci et al., 2012) and VRPs with multiple depot locations (see Crevier et al., 2007; Nagy & Salhi, 2005).

This problem differs from the typical ones, most importantly due to technical capacity constraints imposed on nodes of the network, and because of the fact that transported volumes can be split or merged in the nodes of the network in an arbitrary way. Additionally, we assume that an overall constraint is imposed on the length of the route (expressed here by the number of locations that can serve as transfer nodes within each route), whereas it is typical for VRPs that the length of the routes, i.e. sequences of locations visited by vehicles, is not constrained. More specifically, we assume that the route between the origin and the destination locations of a postal shipment can take the form of either a direct connection or a transfer connection, with one facility at most in which the vehicle stops along the route. Using direct connections can be beneficial, since they tend to avoid heavy traffic, and as a consequence smaller costs are incurred by vehicles and drivers. Moreover, if a substantial volume is processed on such a connection, i.e. a large vehicle (or vehicles) can be fully loaded, the number of utilised vehicles decreases, and so does the unit cost of the shipment.

On the other hand, transfer connections make it possible to split the stream into a given number of sub-streams and distributing them over the network, possibly merging with streams originating from different locations. It is also beneficial, since it reduces the number of vehicles needed to serve the deliveries and may help preventing vehicles from getting caught in heavy traffic at some locations. However, when a volume is shipped using a transfer location, it needs to be unloaded and processed by the staff and machinery of a transfer outlet, then re-loaded into either the same or another vehicle, and finally delivered to its destination location. Processing streams within postal operations outlets requires time and incurs costs.

The incorporation of these technical features of operation outlets into the definition of the analysed problem is crucial when we are interested in a brownfield analysis and the identification of prospective hubs. The solver must balance the above-mentioned elements of the problem.

The version of the postal delivery problem which we consider in this article is related to a number of other problems and approaches in the field of logistics and optimal planning, both in the context of the identification of transfer connections and in the context of the divisibility of the delivery. As regards the former, a useful, although a slightly outdated summary of formulations and solution heuristics used in point-to-point delivery systems (including postal systems) is proposed by Leung et al. (1990). Also de Camargo et al. (2013), Çetiner et al. (2010) and Karimi and Setak (2018) investigate how transfer locations are determined on the basis of demands and the topology of the postal delivery network. In the context of the

divisibility of the product stream, Kerivin et al. (2008) present an up-to-date formulation of a splittable pickup and delivery problem with reloads, Archetti and Speranza (2012) also consider a framework with split deliveries, whereas Baumung et al. (2015), for example, focus on designing optimal plans for a postal delivery network with parcels and letter mail.

The set of orders (streams, demands) that need to be processed will be denoted by  $S$ , and a single generic stream by  $s \in S$ . The execution of a stream consists in delivering a given positive volume of divisible cargo from the origin location to the destination location, using a logistic network with its operating characteristics like distances and travel times between nodes, capacity of vehicles, capacities of the nodes, time windows in which the nodes operate, costs of the utilisation of vehicles, and costs of volume processing within the nodes. The objective is to deliver all streams on time and at the smallest possible cost. In the further parts of the paper we will provide a more formal description of the analysed problem.

## 2.1. Logistic network

A logistic network is specified by an ordered pair  $(N, E)$ , where  $N = \{1, 2, \dots, n\}$  denotes an  $n$ -element set of nodes (locations of postal operations outlets), and  $E$  denotes a set of directed edges. In the case of the company whose problem is analysed here, there is a direct connection between each pair of nodes  $(i, j) \in E$ , as long as  $i \neq j$ , hence  $E = N^2 \setminus \{(i, i), i \in N\}$ . The term direct connection between a pair of locations, say  $i \in N$  and  $j \in N$ , does not necessarily mean that the physical route between location  $i$  and location  $j$  does not go through any  $p \in N$  location (or locations). It means, however, that when delivering the stream of volume from  $i$  to  $j$ , we do not stop in postal operations outlets located in such nodes.

## 2.2. Postal operations outlets

Each postal operations outlet has the following technical characteristics:

- a) a time window in which the outlet operates:  $tw_i = [open_i, close_i]$ , where  $open_i, close_i$  are timestamps (points in time in a day), defined with the precision of up to a minute;
- b) the capacity of the outlet, in terms of the number of vehicles it can process,  $capVeh_i > 0$ ;
- c) the capacity of the outlet, in terms of the volume it can process,  $capVol_i > 0$ ;
- d) the unit cost of processing volume  $cVol > 0$ ;
- e) time needed to process a unit of volume  $tVol > 0$ .

In practice, the values of  $open_i$  and  $close_i$  are similar for the respective  $i \in N$  outlets, but as they do not have to be identical, they are indexed by  $i$ . They

represent, respectively, the earliest and the latest time at which outlet  $i$  is operational throughout the day, i.e. the earliest when any volume can enter it (the unloading of volumes from vehicles can start), and the latest any volume has to leave it (the loading of all volumes into vehicles has to finish). When a vehicle enters a postal operations outlet, what happens first is unloading the volume from it. The volume contained in a vehicle is the sum of the volumes of streams which were delivered by this vehicle<sup>1</sup> to the outlet. Each stream needs to be processed in an outlet, which takes time and incurs costs. Streams whose destination is a given outlet are processed in a simplified way, i.e. without a vehicle assignment, since they are not delivered to any further nodes of the network. But still, the time needed to process the stream without a vehicle assignment in its destination node is accounted for, along with the cost of processing.

The cargo of a stream finishing in an outlet is delivered to final destinations (post offices) outside of the model within the last mile.<sup>2</sup> If the time needed to implement the last mile around node  $i$  is denoted by  $t_i^{lm}$  (it involves uploading the volumes onto vehicles which perform the last mile, delivering the volumes to their final destinations, i.e. post offices, and unloading them there) and the deadline for the delivery of stream  $s$  is denoted by  $\bar{t}_s$  (the time by which the stream must reach the post office within the last mile), then the last possible departure time of this stream from location  $i$  is not  $close_i$ , but  $\min(close_i, \bar{t}_s - t_i^{lm})$ . Indeed, streams whose destination is location  $i$  must leave this location at the  $\bar{t}_s - t_i^{lm}$  time at the latest, and streams for which location  $i$  is a transfer node can be processed until the  $close_i$  time, and leave it by  $\bar{t}_s - t_i^{lm} < close_i$ .

Analogically, streams which originate in a given postal operations outlet are delivered to this outlet from outside the network within the first mile. If the time needed to implement the first mile around node  $i$  is denoted by  $t_i^{fm}$ , and the starting time of the implementation of the first mile for stream  $s$  is denoted by  $\underline{t}_s$ , then the effective earliest time at which this stream can enter outlet  $i$  for processing is not  $open_i$ , but  $\max(open_i, t_i^{fm} + \underline{t}_s)$ . Indeed, streams which originate in location  $i$  can leave this location at  $t_i^{fm} + \underline{t}_s$ , and streams for which location  $i$  is a transfer node can be processed starting from the  $open_i$  time, and arrive there from  $t_i^{fm} + \underline{t}_s > open_i$ . Note that the first mile correction applies only to streams which originate in node  $i$ , and the last mile correction applies only to streams whose destination is location  $i$  (i.e. it does not apply to the streams which use location  $i$  as a transfer node).

<sup>1</sup> A vehicle can ship the entire volume of a given order or a fraction of it.

<sup>2</sup> In our study the last mile is neglected. It is assumed that it is optimised locally outside of the main model.

### 2.3. Vehicles

We assume that vehicles are available on demand in each location. This is consistent with the common practice of many postal operators who often find it profitable to outsource the maintenance of their fleet of vehicles. This implies that the objective function abstracts from the possible cost which arises if a fleet of vehicles needs to be expanded (i.e. vehicles need to be purchased), but involves solely the costs of the operations of the fleet, so the costs incurred by the daily operations of the postal operator. Each vehicle has its technical characteristics, namely:

- a) the capacity in terms of the number of postal pallets it can carry  $vehVolPall_i > 0$ .  
Although the number of pallets must be an integer, we consider fractional numbers as well, which represent a pallet which is not fully loaded;
- b) the capacity in terms of the mass of the volume it can carry  $vehVolKG_i > 0$ ;
- c) the cost of vehicle utilisation per kilometre  $vehCostKM > 0$ ;
- d) the cost of vehicle utilisation per hour  $vehCostH > 0$ ;
- e) the cost of a driver per hour  $vehCostDrvH > 0$ .

### 2.4. Execution of streams

Let  $k_s \in N$  denote the origin node, and  $l_s \in N$  the destination node of stream  $s \in S$ . The volume associated with stream  $s$ , which flows over the edge  $(i, j)$ , will be denoted by  $v_{sij}$ . Naturally,  $v_{sij} \geq 0$  and  $v_{sij} \leq D_s$ , where  $D_s > 0$  denotes the volume (demand) of stream  $s$ . Since  $k_s$  and  $l_s$  are the origin node and the destination node of stream  $s$ , we know that  $v_{sk_s j} > 0$  for at least one  $j \in N \setminus \{k_s\}$ , and  $v_{sil_s} > 0$  for at least one  $i \in N \setminus \{l_s\}$ . If the flow of volume from the origin node  $k_s$  to the destination node  $l_s$  is a direct one, then  $v_{sk_s l_s} > 0$ . If a transfer connection is used (with one transfer node), we have  $0 < v_{sk_s w_{sp}} = v_{sw_{sp} l_s} \leq v_s$  for some indices  $p \in \{1, 2, \dots, m_s\}$ , where  $1 \leq m_s \leq n - 2$  denotes the number of transfer connections which are used for stream  $s$ , and, naturally,  $w_{sp} \in N \setminus \{k_s, l_s\}$ . It can also be the case that none such  $w_{sp}$  node exists in the solution (i.e.  $m_s = 0$ ), which means that stream  $s$  is processed using a (single) direct connection. It is also possible that a given stream is processed using only transfer connections, in which case  $v_{sk_s l_s} = 0$ .

### 2.5. The search space

An order can be processed in the form of a direct connection or in the form of a transfer connection. There is one way of processing an order in the form of a direct connection, whereas in relation to a transfer connection (or connections), an order

can be processed in  $\sum_{k=1}^{n-2} \binom{n-2}{k} = 2^{n-2} - 1$  ways, therefore altogether there are  $2^{n-2}$  possibilities. Since deliveries from each node of the network are carried out to all other nodes, there are  $n^2 - n$  orders placed. Hence, there are  $2^{n(n-1)(n-2)}$  possible solutions, without taking into account time and capacity constraints. We have  $n \approx 40$ , which yields  $2^{59280}$  ways in which orders can be processed, and this is without vehicle assignment. This precludes any approach based on direct enumeration of solutions.

### 3. The Algorithm

To optimise the problem outlined in the previous section, we employ a stochastic heuristic procedure which belongs to the family of genetic algorithms (see Boussaïd et al., 2013; Dréo et al., 2006 or Katoch et al., 2021), which is one of many possible approaches that are used in case of logistic problems (see Arnold & Sörensen, 2019; Baker & Ayechev, 2003; Baldacci et al., 2010; Cordeau et al., 2002; Laporte et al., 2000, or Prins, 2002). To employ such a method, a solution must be encoded in an appropriate way, as e.g. Kadri and Boctor (2018) demonstrate.

In the following parts of the paper, we present a simplified narrative, i.e. our point is made without a detailed discussion of the possible formulations, in the form of a mixed-integer program (MIP). The implementation of the problem we deal with in the form of an MIP, which is then solved with techniques such as branch and bound, cutting plane, etc., would require the introduction of many additional variables, like index binaries showing if any volume is shipped through a given edge for each stream or not, time-related variables, etc. (see Granada-Echeverri et al., 2019; Rieck & Zimmermann, 2010 or Theurich et al. (2021). Also, as shown by Boland et al. (2017), in such a case the reformulation of the problem in the form of a time-extended graph is possible.

#### 3.1. Representation of solutions

Problem formulation presented in the previous section can be translated into a mathematical programming form with variables  $v_{sij} \geq 0$ , which denote volumes of stream  $s \in S$  assigned to edges  $(i, j)$  for  $i, j \in N$ ,  $i \neq j$ , and  $veh_{ij} \in \mathbb{Z}_+$ , which denote a number of vehicles operating on edge  $(i, j)$ . This would yield solutions of the form  $(v_{sij}, veh_{ij}, s \in S, i, j \in N, i \neq j)$  belonging to search space  $(S \times \mathbb{R}_+^{n(n-1)}, \mathbb{Z}_+^{n(n-1)})$ . Although a genetic algorithm could, in principle, be applied to such a solution representation, there are more natural, concise and efficient ways in which solutions can be represented and genetic operations performed on them.

By ‘natural’ we mean that the representation of solutions resembles a constructive nature of a genetic algorithm, i.e. the fact that it constructs new solutions from the olds ones (in the process of a crossover). A mathematical programming formulation is generic and it does not reflect the inner workings of any optimisation technique, including that of a genetic algorithm. We will employ an approach in which the representation of solutions resembles the physical process of the construction of routes within a logistic plan.

By concise we mean that only relevant information is stored in the structure (object) which represents the solution. For example, we cannot rule out that in most situations (problem instances), at each stage of the search space exploration, most variables  $v_{sij}$  will be equal to zero, since most routes do not make sense from the economic point of view. Moreover, a set of such variables can vary at different stages of the optimisation for the algorithm runs which coordinate the search towards different local minima, therefore most of such variables cannot be set to zero *a priori*. As a consequence, potentially large objects need to be stored in memory, with significant access time to their elements, capable of representing many and diverse solutions, although only a fraction of the solutions is effectively processed, i.e. visited in the process of the search space exploration.<sup>3</sup> We will employ an approach in which the representation of solutions is minimal, i.e. only the relevant information is stored.

Finally, the criterion of efficiency is related to the fact that crossover and mutation operators either produce feasible solutions, in which case it has to be a tailor-made operator, compatible with constraints imposed in the problem in question, or a repair operator must be applied to the result of the work of the operator, so that the produced solutions, possibly infeasible at first, are projected<sup>4</sup> on the feasible region of the search space. It is also possible that the algorithm allows search space explorations through infeasible regions, but, in either case, final solutions produced by the optimiser must be feasible, for example due to increasing penalty applied to the objective function for infeasibility. For the problem in question, due to a multitude of the imposed feasibility constraints and due to the way in which crossovers and mutations are typically specified for problems formulated as mixed integer mathematical programmes using variables like  $v_{sij}$  and  $veh_{ij}$ , no matter which approach is adopted, the production of feasible solutions can be computationally expensive,<sup>5</sup> especially if based on stochastic trials until a feasible solution is obtained, or can substantially restrict the way in which new solutions are

<sup>3</sup> Moreover, sparse matrices do not prove an appropriate approach due to efficiency reasons. See below.

<sup>4</sup> The term projection is used here in a casual way, without a strict mathematical meaning of a projection operator.

<sup>5</sup> Crossovers are inherently stochastic procedures, which means that producing a feasible solution by chance can take substantial computational time.



constructed from the old ones, when tailored-made feasibility-preserving operators are employed. The caveat is that such feasibility-preserving operators tend to lead to a shrinkage of the subset of the feasible region that can be effectively explored by the solver.

In both the above-described cases, the efficiency of the search space exploration deteriorates – either due to a computational burden or a low diversity of the exploration which, at the end of the day, also translates into an increased computational burden: with less diversity in the search space exploration, more iterations are needed to explore the search space effectively. In the further parts of the paper, we will take advantage of the natural way in which solutions are represented (as discussed above), so that efficient feasibility-preserving operators of reproduction and mutation can be specified. Our representation of the solution is, informatively, equivalent to a specific representation in the form of a mathematical programme, i.e. a bijective correspondence between these two representations can easily be obtained.

Let  $x_s$  denote the representation of the way in which stream  $s \in S$  is executed. If the delivery of stream  $s$  involves a direct connection, then

$$d(x_s) = ((i, j), v_{sd}), \quad (1)$$

where  $(i, j) = (k_s, l_s)$  and  $0 < v_{sd} \leq v_s$  represents the volume of  $s$  which is delivered using a direct connection. Note that  $(i, j) = (k_s, l_s)$  does not have to be stored in  $x_s$  explicitly, since the structures  $(k_s, s \in S)$  and  $(l_s, s \in S)$  must be stored anyway. If the delivery of  $s$  does not involve a direct connection, then<sup>6</sup>  $d(x_s) = \emptyset$  and  $v_{sd} = 0$ . If the delivery of stream  $s$  involves at least one transfer connection, then

$$t(x_s) = \left( \left( (i, w_{sp}, j), p \in \{1, 2, \dots, m_s\} \right), (v_{sp}, p \in \{1, 2, \dots, m_s\}) \right), \quad (2)$$

where, again,  $(i, j) = (k_s, l_s)$ ,  $w_{sp} \in N \setminus \{k_s, l_s\}$  with  $w_{sp_a} \neq w_{sp_b}$  for any  $p_a, p_b \in \{1, 2, \dots, m_s\}$ ,  $p_a \neq p_b$ , and  $0 < v_{sp} \leq v_s$  represents the volume of  $s$  which is delivered using a transfer connection that passes through node  $w_{sp}$ , i.e. the vehicle travelling from  $k_s$  to  $w_{sp}$  stops in the postal operations outlet located in  $w_{sp}$ , and the stream  $v_{sp}$  is processed in  $w_{sp}$ . Otherwise  $t(x_s) = \emptyset$ , and each  $v_{sp} = 0$ .

A solution is represented by  $x = (x_s, s \in S)$ . Such a representation is natural, since routes themselves are stored, along with volumes flowing through the edges. It is also concise, since only used connections are physically represented. As we will see, it also allows the implementation of efficient reproduction and selection operators.

<sup>6</sup> We use symbol  $\emptyset$  to denote any null, empty or nonexistent data structure.

To give an example, let us consider a small problem with three nodes  $V = \{1,2,3\}$ , and three streams  $s_1$ ,  $s_2$  and  $s_3$ , with origins and destinations given by  $(k_{s_1}, l_{s_1}) = (1,2)$ ,  $(k_{s_2}, l_{s_2}) = (1,3)$ , and  $(k_{s_3}, l_{s_3}) = (3,1)$ , and associated volumes equal to  $v_{s_1} = 10$ ,  $v_{s_2} = 20$  and  $v_{s_3} = 5$ . Let us assume that stream  $s_1$  is executed using a direct connection only, therefore we have  $d(x_{s_1}) = ((1,2), 10)$  and  $t(x_{s_1}) = \emptyset$ . Similarly, let us assume stream  $s_2$  is executed using a direct connection (volume of 12) and a transfer connection (volume of 8), therefore we have  $d(x_{s_2}) = ((1,3), 12)$  and  $t(x_{s_2}) = ((1,2,3), 8)$ . Finally, let us assume that stream  $s_3$  is executed using a transfer connection only, therefore we have  $d(x_{s_3}) = \emptyset$  and  $t(x_{s_3}) = ((3,2,1), 5)$ .

### 3.2. Initial population

For genetic optimisation, a population of solutions is needed, which we denote by  $P_k = \{x_i, i = 1, 2, \dots, |P|\}$ , where  $k \geq 0$  is an iteration index of the algorithm (with  $P_0$  being the initial population), and  $|P|$  denotes the size of the population, which is kept constant through the iterations. The initial population consists of  $|P|$  solutions, each of which is constructed according to the procedure outlined below, which is sequentially applied to each of the  $m$  streams.

Firstly, the number of transfer connections is drawn from set  $\{0, 1, 2, \dots, \min(\lceil v_s \rceil, n - 2)\}$ , according to probabilities proportional to  $2^{-k}$ , respectively. If the sampled number of transfer connections is bigger than 0, we sample, uniformly and without replacement, respective routes for transfer connections  $(k_s, w_{sp_i}, l_s)$  with  $w_{sp_i} \in N \setminus \{k_s, l_s\}$  for  $i = 1, 2, \dots, |t(x_s)|$ . Secondly, if  $|t(x_s)| < n - 2$ , a direct connection  $(k_s, l_s)$  is added to  $x_s$  with the probability of 1 or  $p_d$ , respectively. Finally, the volume  $v_s$  gets distributed over the connections of  $x_s$ . Since there are, by construction,  $|t(x_s)| + |d(x_s)| \geq 1$  connections within  $x_s$ , volume  $v_s$  is partitioned into volumes  $v_{si}$ ,  $i = 1, 2, \dots, |t(x_s)| + |d(x_s)|$ , such that  $v_{si} > 0$  and  $\sum_{i=1}^{|t(x_s)| + |d(x_s)|} v_{si} = v_s$ . This makes the explicit construction of  $t(x_s)$  and  $d(x_s)$  possible, and therefore the construction of  $x_s$  as well. The procedure is applied for each  $s \in S$ , which yields generic solution  $x_i$ . The procedure is repeated for  $i = 1, 2, \dots, |P|$ .

### 3.3. Genetic operators

A genetic algorithm usually consists in the sequential application of the following three genetic operators: selection, reproduction and mutation. Selection is responsible for the stochastic promotion of high quality solutions to the reproduction phase (see Vajda et al., 2008), which in turn combines structural features of solutions

producing new solutions, possibly of higher quality than the old ones (see Kora & Yadlapalli, 2017 or Umbarkar & Sheth, 2015).<sup>7</sup> Mutation, by introducing random disturbances into offspring produced by reproduction, is mainly responsible for increasing the diversity of the search space exploration and for the prevention of premature convergence, as Lim et al. (2017) and Squillero and Tonda (2016) demonstrate. Our implementation of the genetic algorithm follows the idea, hence we have  $P_{k+1} = M(R(S(P_k)))$ , where  $S$  stands for selection,  $R$  for reproduction, and  $M$  for mutation operators.

### 3.3.1. Selection

A tournament selection is employed as a selection operator (see Bäck et al., 2018). It boils down to selecting  $|P|$  times the two solutions  $x, y \sim U(P_k)$  from  $P_k$ , i.e. two uniform draws from  $P_k$ , with the replacement and the promotion of  $x$  to the reproduction phase if  $f(x) > f(y)$ , and the promotion of  $y$  otherwise. We augment, however, the typical formulation of a tournament selection by the rule of elitism, which assumes that the  $\alpha \in (0,1)$  share of the best solutions from  $P_k$  is automatically promoted to the recombination phase, and they constitute share  $(0,1) \ni \beta \geq \alpha$  of parental pool  $S(P_k)$ . If  $\beta > \alpha$ , then the  $(\beta - \lambda)|P|$  solutions, which are integer by construction, randomly drawn from the  $\alpha$  share of  $P_k$ , are duplicated. The above shows how  $S(P_k)$  is constructed.

### 3.3.2. Reproduction

After the choice of the selection operator, we obtain population  $S(P_k) = (p_1, p_2, \dots, p_k, p_{k+1}, \dots, p_{|P|-1}, p_{|P|})$ , which contains  $\frac{|P|}{2}$  ordered tuples of solutions in the form of  $(p_k, p_{k+1})$  for  $k = 1, 3, \dots, |P| - 1$ . Each such pair of solutions – parents  $(p_k, p_{k+1})$  produces a new pair of solutions – offspring  $(o_k, o_{k+1})$ , according to a *1-stream-interchange procedure*, which we define below.

First, index  $q$  is drawn uniformly from set  $\{1, 2, \dots, m\}$ , the elements of which correspond to streams  $s \in S$ . The letter  $m$  denotes the number of streams in a given problem instance, i.e.  $|S| = m$ . So far it has not been important to emphasize, but the solution  $x = (x_s, s \in S)$  comes in the form of an ordered set, a sequence, therefore streams  $s \in S$  are indexed and we can equivalently represent the solution as  $x = (x_k, k = 1, 2, \dots, m)$ , where  $x_k$  represents the way in which the  $k$ -th sequence is executed in solution  $x$ .

<sup>7</sup> Obviously, the recombination can also produce offspring worse than ‘parents’, in particular with the objective function below the average in the population, but such offspring, in terms of the expected value, will not be promoted by the selection operator in the next iteration.

Secondly, the execution of the  $q$ -th stream in solution  $p_k$ , denoted by  $x_q(p_k)$  and represented by  $d(x_q(p_k))$  and  $t(x_q(p_k))$ , is interchanged with the execution of the  $q$ -th stream in solution  $p_{k+1}$ , denoted by  $x_q(p_{k+1})$  and represented by  $d(x_q(p_{k+1}))$  and  $t(x_q(p_{k+1}))$ . Such an interchange operation can be represented as:

$$(d(x_q(p_k)), t(x_q(p_k))) \leftrightarrow (d(x_q(p_{k+1})), t(x_q(p_{k+1}))),$$

or, in short, as  $x_q(p_k) \leftrightarrow x_q(p_{k+1})$ . The above demonstrates how  $R(S(P_k))$  is constructed.

### 3.3.3. Mutation

Mutation introduces changes in each stream  $x_s$  of each solution  $x \in R(S(P_k))$  independently, with an exogenous probability of  $p_m \in (0,1)$ . We assume three possible changes, employed with equal probabilities. The first change consists in removing a random connection from stream  $x_s$ . The connection to be removed is chosen uniformly from  $\{1,2, \dots, |t(x_s)| + |d(x_s)|\}$ . Indeed, if  $|d(x_s)| = 1$ , then index 1 represents a direct connection, and indices  $2,3, \dots, |t(x_s)| + |d(x_s)|$  represent consecutive transfer connections. If  $|d(x_s)| = 0$ , indices  $1,2, \dots, |t(x_s)| + |d(x_s)|$  represent transfer connections.

The removal of a connection implies that some volume, say  $v$ , needs to be distributed over the remaining  $|t(x_s)| + |d(x_s)| - 1$  connections. It can happen that  $|t(x_s)| + |d(x_s)| - 1 = 0$ , so in other words, that  $x_s$  contained exactly one route, which was removed. In such a case, a random connection is added to  $x_s$ , either a direct or a transfer one, with probabilities  $p_d$  and  $1 - p_d$ , respectively, and the volume from the removed connection is assigned to it. If  $|t(x_s)| + |d(x_s)| - 1 \geq 1$ , then there is at least one route which is left after the removal of the sampled route.

The second possible change involves adding a random route to stream  $x_s$ . The added route can either be a direct connection, if  $x_s$  does not have one, or a transfer connection, if  $|d(x_s)| < n - 2$ . If  $x_s$  does not already have a direct connection and it is possible to add a transfer connection, then the choice is made randomly with the probability of  $p_d'$  and  $p_t = 1 - p_d'$ , respectively. If a transfer connection is added, then the node for the connection is drawn uniformly from  $N \left\{ \{k_s, l_s\} \cup \{w_{sp}, p = 1,2, \dots, m_s\} \right\}$ . If  $x_s$  already has a direct connection, then a transfer connection is added. When a direct or transfer connection is added, a random proportion of the volume of some other connection is shifted to it. The connection

from which the volume is shifted is chosen uniformly. If the shifted volume equals the entire volume of the connection in question, it is then removed from  $x_s$ . If a volume is perfectly divisible, this does not happen, but when it has to be modelled as an integer value, such a situation can happen.

The third possible change reallocates the volume within the existing connections. More specifically, two connections are drawn at random, and their volumes are interchanged. In the way described above genetic operators are implemented.

## 4. Numerical Experiments

In this section, we present and discuss a selection of case studies. Our aim here is to verify the desired features of the algorithm, and to provide some insight into the ability of the algorithm to guide brownfield analyses for realistic-size problem instances.

Case studies are divided into the following two groups:

- a) stylised problem instances – to verify if the algorithm is capable of selecting direct and transfer connections appropriately;
- b) practical problem instances – to observe convergence and to compare the generated solution with a benchmark. The way to verify the solution is to check if it is satisfactory from the practical point of view. A possible practical benchmark is a solution in which all volumes are shipped directly from the origins to the depots.

### 4.1. Stylised case studies

To give an example of stylised case studies, we will demonstrate how it can be experimentally confirmed that the algorithm is capable of producing good-quality solutions in three cases:

- a) when streams should be executed directly;
- b) when merging of streams is optimal;
- c) when there are distinct areas in which shipments should be executed separately.

Case a). Assume that the objective function is given by the total length of all routes of all streams counted separately, i.e. without merging volumes:

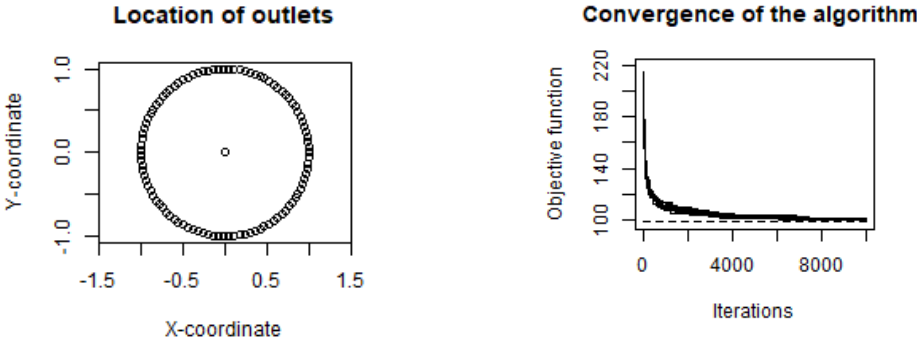
$$f(x) = \sum_{s \in S} \left( \chi_{sd} d_{(k_s, l_s)} + \chi_{st} \left( \sum_{p=1}^{m_s} d_{(k_s, w_{sp})} + d_{(w_{sp}, l_s)} \right) \right), \quad (3)$$

where:

$$\chi_{sd} = \begin{cases} 1, & d(x_s) \neq \emptyset \\ 0, & d(x_s) = \emptyset \end{cases} \text{ and } \chi_{st} = \begin{cases} 1, & t(x_s) \neq \emptyset \\ 0, & t(x_s) = \emptyset \end{cases}, \quad (4)$$

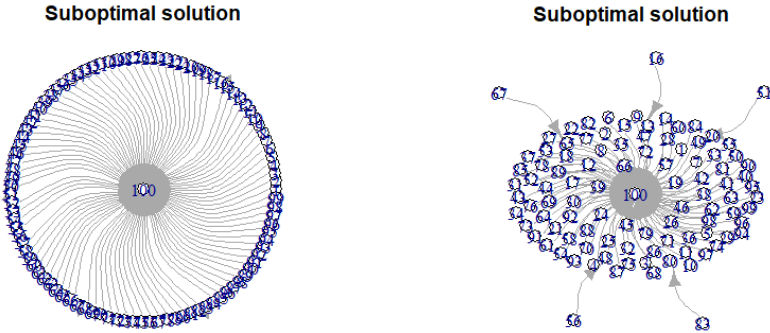
i.e.  $\chi_{sd}$  and  $\chi_{st}$  are indicator variables which show if  $x_s$  involves a direct connection ( $\chi_{sd}$ ), or at least one transfer connection ( $\chi_{st}$ ), and  $d_{(i,j)}$  denotes the distance between locations  $i \in N$  and  $j \in N$ . In such a case, the optimal solution boils down to using only direct connections, no matter how the network is organised. We also assume that  $v_s = v$  for all  $s \in S$  and that  $vehCap = v$  for all vehicles, but due to the specification of  $f$ , having  $vehCap > v$  does not change the fact that the optimal solution involves only direct connections. For the graphical presentation of the results, we assumed that depots of respective streams are located in points  $(x_i, y_i)$ ,  $i = 1, 2, \dots, n$ , on a circle with the center in  $(0,0)$ , i.e. for  $x_i = r \cos(\phi_i)$  and  $y_i = r \sin(\phi_i)$ , where  $\phi_i = i \frac{2\pi}{n+1}$ . The centre of the circle serves as a single destination point for all streams, as shown in Figure 1.

**Figure 1.** Location of origin nodes – the circle, and of the destination node – the center, convergence of optimisation



Source: authors' calculations based on stylised data described in Case a).

**Figure 2.** Best suboptimal solution generated by the algorithm



Source: authors' calculations based on stylised data described in Case a).

For  $n = 100$ , which is large enough from the practical perspective, we know that the optimal value of the objective function is  $f = 99$ , hence it is easy to observe how the algorithm converges. Figure 1 plots the value of the objective function over 10,000 iterations for 10 independent runs. The dashed horizontal line presents the level of the optimal solution. In all runs, the algorithm converges closely to the optimal value. Figure 1 shows 10,000 iterations with  $f$  converging to, respectively: 101.52, 100.13, 100.64, 100.32, 102.06, 100.20, 100.68, 100.82, 101.08 and 100.39.

Figure 2 plots the best sub-optimal solution generated by the algorithm, with  $f(x) = 100.13$ , which is within 1.1% from the optimal solution. The left panel presents the solution with outlets aligned as in the input data of the problem. We know that there are transfer connections in this solution, but the alignment of the locations makes them hard to see. Therefore, on the right panel, we also present the same sub-optimal solution, but with a different alignment of nodes, so that we can observe that there are five transfer connections still left in the solution.<sup>8</sup>

The conclusion is that the employed genetic operators allow the selection of solutions based on direct connections if the problem instance in question requires such coordination.

Case b1). Assume that the locations of all the outlets are the same as in Case a), and also that the volume of each stream stays the same, i.e.  $v_s = v$  for all  $s \in S$ . Assume also that the objective function is modified, so that it is beneficial to merge volumes:

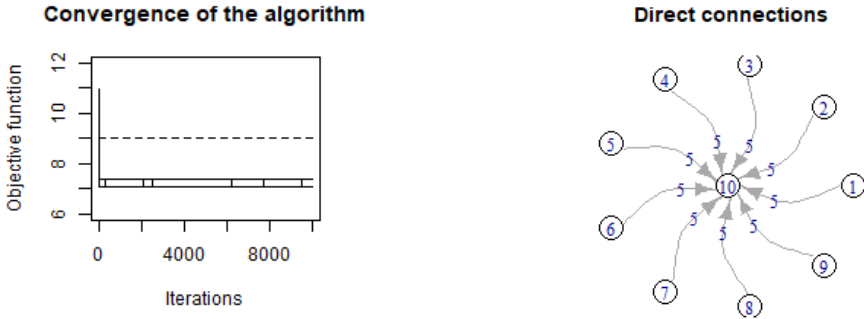
$$f(x) = \sum_{\substack{(i,j) \in N^2 \\ i \neq j}} veh_{ij} d_{(i,j)}, \quad (5)$$

where  $veh_{ij}$  represents the number of vehicles assigned to the  $(i, j)$  edge, i.e. the minimal number of vehicles capable of loading volume  $v_{ij} = \sum_{s \in S} v_{sij}$  which flows through this edge. To make the merging of volumes possible, the capacities of vehicles are increased to a value larger than  $v$ . We assume that  $vehCap = 3v$  for each vehicle. Now it seems reasonable to set up local hubs and coordinate the direct flows to the destination point from the hubs.

---

<sup>8</sup> Within 25,000 iterations, all the runs converged to the optimal solution.

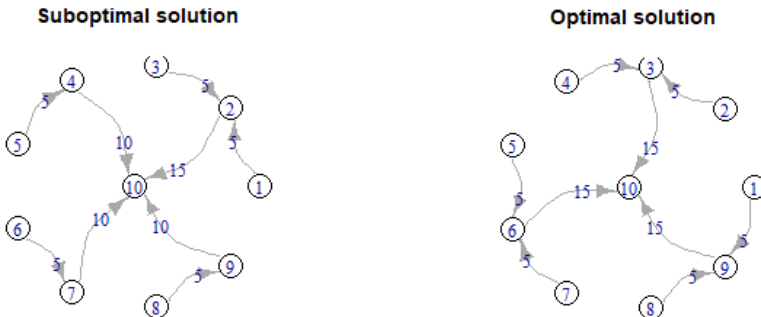
**Figure 3.** Convergence of the objective function and solution based on direct connections only



Source: authors' calculations based on stylised data described in Case b1).

For clarity, in Figures 3 and 4 we present the results for  $n = 10$ , but an analogical behaviour of the algorithm is observed for greater values of  $n$ .<sup>9</sup> Figure 3 shows the trajectory of the objective function over 10,000 iterations for 10 independent runs of the algorithm, where the dashed horizontal line represents the value of the objective function when only direct connections are used, with  $f = 9$  (the left panel), along with a solution which uses only direct connections (the right panel). Figure 4 presents two types of solutions generated by the algorithm, one of which is optimal with  $f(x) = 7.10$  (the right panel), while the other is sub-optimal with  $f(x) = 7.42$  (the left panel). As arrows on the figure suggest, we assumed that  $v = 5$ .

**Figure 4.** The suboptimal solution and the optimal solution with coordination of volumes in selected nodes



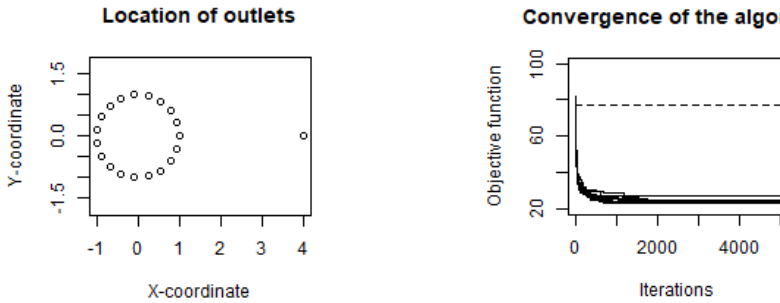
Source: authors' calculations based on simulated data described in Case b1).

<sup>9</sup> Since emergent properties of solutions, i.e. the selection of direct connections, when needed, and the selection of transfer connections with the coordinated merging of volumes, when needed, come as a consequence of the way in which genetic operators were implemented, they do not depend on the size of the problem. Efficiency is bigger for smaller test cases, therefore in the current section we also include the presentation of a real-life size of the problem.



Case b2). Now let us assume that the locations of nodes are given as presented in the left panel of Figure 5. Location of origin nodes is the same as in Case a) and b1), with the exception of the location of the single destination node, which was shifted from (0,0) to (0,4). Let us assume that  $vehCap = 100v$ , which in practical terms means that the capacities of vehicles are unlimited. The objective is defined as in Case b1). The right panel of Figure 5 presents the trajectory of the objective function over 5,000 iterations for 10 independent runs of the algorithm. The dashed horizontal line represents the value of the objective function when only direct connections are used.

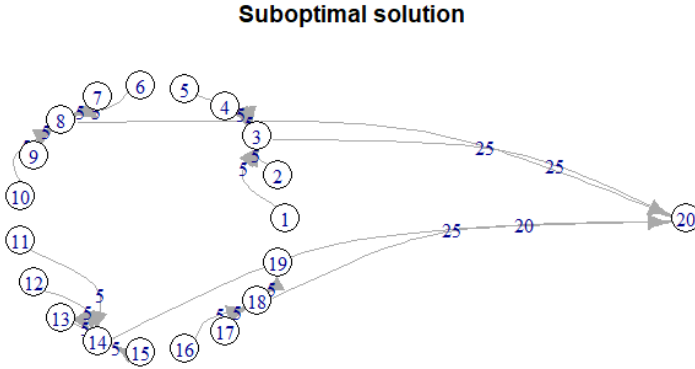
**Figure 5.** Location of origins and destinations and the convergence of the objective function



Source: authors' calculations based on stylised data described in Case b2).

Figure 6 presents the optimal solution generated by the algorithm with  $f(x) = 23.25$ . For clarity, we present an example of a solution for  $n = 20$ , but an analogical behaviour of the algorithm is observed for larger values of  $n$ . The figure shows that the algorithm is capable of coordinating the execution of streams by setting up four hubs – for regions NE, NW, SE and SW, to which volumes are shipped from nearby locations and from which merged volumes are shipped by single vehicles directly to the destination location. Please note that streams originating in locations selected as hubs are shipped directly to their destination points. The conclusion from Cases b1) and b2) is that the employed genetic operators are capable, whenever necessary, of selecting hubs in which volumes are merged and from which they are transferred directly to their destinations.

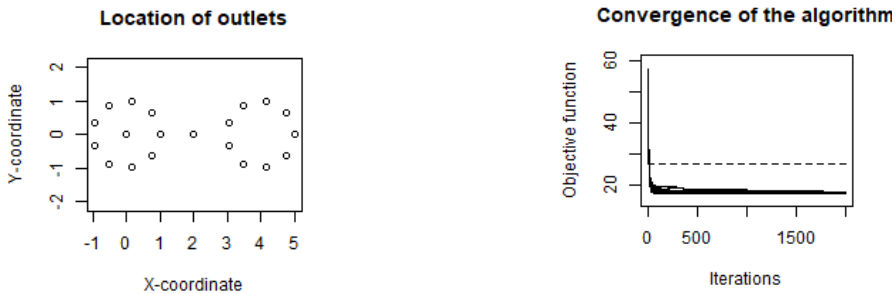
**Figure 6.** A solution in which four locations are selected as hubs where volume is aggregated and transported to the destination location by direct connections



Source: authors' calculations based on stylised data described in Case b2).

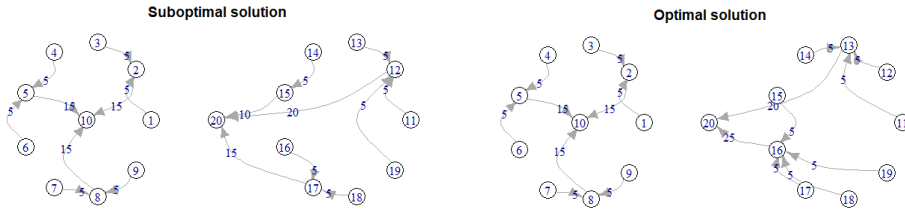
Case b3). This case is analogical to b2), but we introduce two regions – one consists of locations on the circle centred at (0,0), as in the a) and b1) Cases, and the other is analogical to the b2 Case, but with the circle centred in (4,0) and the single destination in (2,0). In Figure 7, locations are presented on the left panel and the convergence of the objective on the right one (the dashed line represents the benchmark solution with  $f = 28.14$ ). Figure 8 presents both the optimal and the suboptimal solutions, the latter being very close to the optimal solution as far as the value of  $f$  is concerned ( $f = 17.47$  for the optimal solution, and  $f = 17.49$  for the suboptimal one). Note that although the difference in the objective is small in the two presented instances, the implied logistic organisation of the East region is substantially different, especially as far as the status of node 16 is concerned (which was given the status of a hub node in the right case, but no such status in the left one). Also the hub status in the NE region is interchanged between nodes 12 and 13.

**Figure 7.** Location of depots and destinations, convergence of the objective function



Source: authors' calculations based on stylised data described in Case b3).

**Figure 8.** Two solutions with a very similar value of the objective function, yet with a substantially different organisation of the East region



Source: authors' calculations based on stylised data described in Case b3).

**4.2. Practical case study**

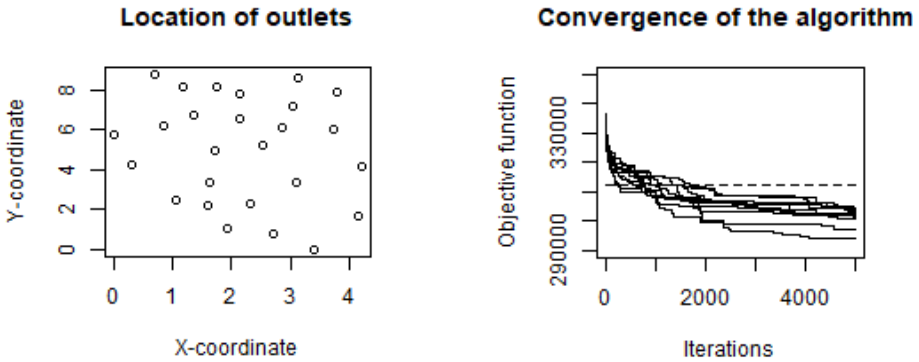
To show some sample results of a brownfield analysis supported by our algorithm, let us consider a problem in which locations correspond to 26 chosen facilities from the logistic network of an existing postal service provider (left panel of Figure 9), with distances and travel times calculated on the basis of real measurements, with realistic assumptions regarding demands and technical characteristics of vehicles and outlets, and with the objective defined as the cost of the execution of the plan.<sup>10</sup> The cost is calculated by a simulation of the execution of the solution and the calculation of the following variables:  $V$  – total volume processed, i.e.  $V = \sum_{i \in N} v(i)$ , where  $v(i)$  denotes the total volume processed in outlet  $i \in N$ ,  $D$  – total distance travelled by vehicles,  $T$  – total time in which the vehicles travelled,  $T_D$  – total time of the work of drivers (which is longer than  $T$  since it includes situations when drivers have to wait for the volume to be processed in transfer nodes). These variables are then crossed with cost characteristics  $cVol$ ,  $vehCostKM$ ,  $vehCostH$  and  $vehCostDrvH$ . The right panel of Figure 9 presents the trajectory of the objective function for 1,000 iterations for 10 independent runs of the algorithm. The dotted line at  $f = 312,300$  represents the cost of the benchmark solution. We present a sample solution which is at least 5% better than the benchmark. This suboptimal solution obtains  $f = 294,000$ . Figure 10 presents the outflows and inflows of the volume for the respective nodes of the network, for the benchmark solution (in blue) and for the presented solution (in red). Please note that for each node, both the outflow and the inflow of volume is larger for the suboptimal solution than for the benchmark one, in which outflows and inflows correspond to volumes originating and finishing in the respective nodes. The residual outflow and inflow (the net flow) equals the sum of the total outflow from a node decreased by the outflow which

<sup>10</sup> Due to property rights, all data were transformed, so that units do not represent any meaningful quantities.

originates in the node and the total inflow of the node decreased by the inflow which finishes in that node. The net flow grows due to the utilisation of transfer connections and the merging of volumes.

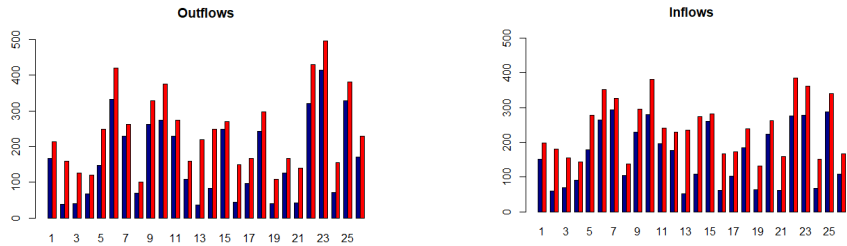
Net flows are shown in Figure 11 as nominal values (the upper panel), as percentages of outflows and inflows of the suboptimal solution (the middle panel), and as percentages of outflows and inflows of the benchmark solution (the lower panel). When the net flow is big in absolute terms, as well as relatively to the outflow which originates in the node and relatively to the inflow which finishes in the node, the node is identified as a hub. Note that, for example, nodes 13 and 14, which, relatively to some other nodes in the network, are not the origin or destination of any great volume of shipments, are considered as prospective hubs, whereas nodes 22 and 23, where a substantial amount of volume originates and finds destination, are considered so to a lesser extent.

**Figure 9.** Location of points and trajectory of the objective function



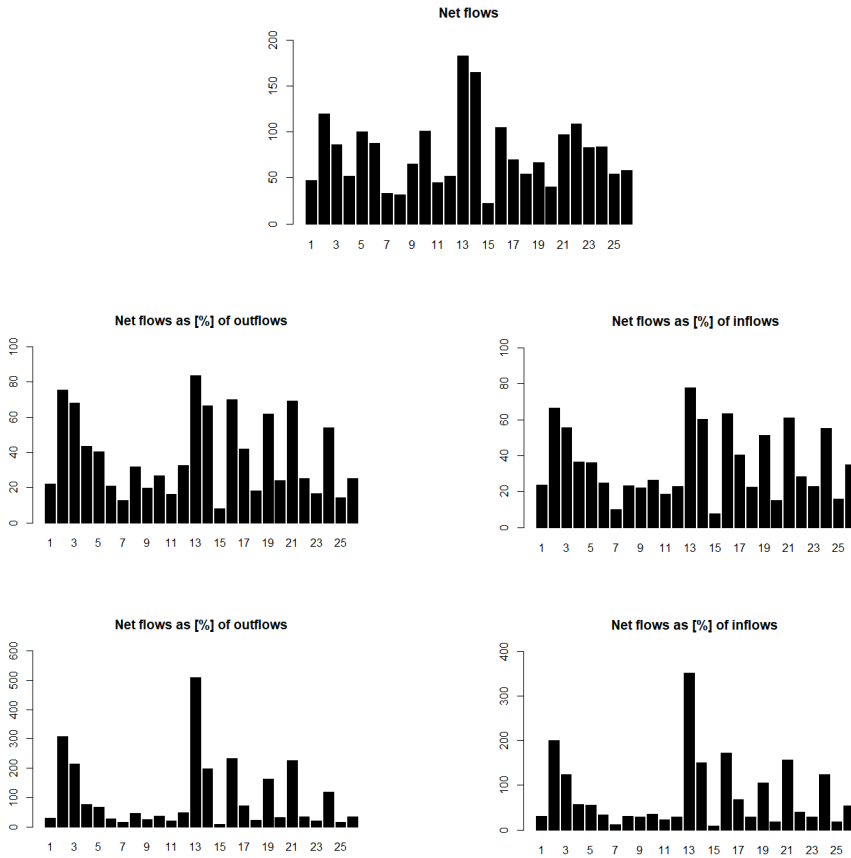
Source: authors' calculations based on a subset of scaled real data.

**Figure 10.** Inflows and outflows of volume for respective facilities for the benchmark solution (in blue) and for the suboptimal one (in red)



Source: authors' calculations based on a subset of scaled real data.

**Figure 11.** Net flows assigned to outlets in terms of nominal values (upper panel), [%] of outflows and inflows of the suboptimal solution (the middle panel), [%] of outflows and inflows of the benchmark solution (lower panel)



Source: authors' calculations based on a subset of scaled real data.

### 5. Conclusions

We studied a realistic variant of a postal delivery problem with pickups and deliveries, split pickups and deliveries, time constraints and capacity constraints, the latter imposed both on vehicles and operations outlets. Our main conclusion is that the implementation of the genetic algorithm we used (with a relatively simple specification of genetic operators) makes it possible to select streams which should be delivered directly and those which should be merged with other streams along the routes, and allows the determination of transfer nodes within the network. The implementation with such features, when used in a real-life case study with relaxed

constraints on the capacities of outlets, allows the determination which outlets will be able to serve as prospective hubs when the network expands. From the practical point of view, such a conclusion means that the algorithm in question can be helpful in the brownfield type of analyses of postal companies, especially large ones, which operate on a country-wide infrastructure. Strong trends that we nowadays observe on the market of postal shipments, which are driven e.g. by a substantial surge in the volumes of orders that consumers place via the Internet (vs. stationary shopping), which will most probably continue and strengthen in the future, pose a serious challenge for the operator. On the other hand, such a situation opens the way for the development of optimisation techniques in the process of both tactical and strategic planning. Our research focused on one aspect of this kind of planning, namely on the identification of prospective hubs within the postal network, and on the approach to this problem. We assumed that transfer connections could pass through arbitrary transfer locations, which allowed the identification of hub locations. The next step would be to introduce an optimisation model in which hub locations would be fixed and transfer connections could be executed only through them, whereas peripheral locations would never serve as transfer nodes. We plan to add such second part of the model to obtain a closed, two-step procedure for postal optimisation. Moreover, in the model presented in this study, we accounted only for the overall time feasibility constraints, i.e. the time of logistic operations and transport which did not exceed the time windows in which outlets operate. The second part of the model would need to take into consideration time constraints resulting from vehicles' queuing in the network nodes and from the necessity of waiting in a hub location for the volumes originating from peripheral locations to be merged there. The above would be a meaningful follow-up to the study presented in this paper.

## Acknowledgements

The research results presented in this paper were obtained with the support of the National Centre for Research and Development's grant no. POIR.04.01.04-00-0054/17-00.

## References

- Archetti, C., & Speranza, M. G. (2012). Vehicle routing problems with split deliveries. *International Transactions in Operational Research*, 19(1-2), 3-22. <https://doi.org/10.1111/j.1475-3995.2011.00811.x>.
- Arnold, F., & Sörensen, K. (2019). What makes a VRP solution good? The generation of problem-specific knowledge for heuristics. *Computers & Operations Research*, 106, 280-288. <https://doi.org/10.1016/j.cor.2018.02.007>.

- Baker, B. M., & Ayechew, M. A. (2003). A genetic algorithm for the vehicle routing problem. *Computers & Operations Research*, 30(5), 787–800. [https://doi.org/10.1016/S0305-0548\(02\)00051-5](https://doi.org/10.1016/S0305-0548(02)00051-5).
- Bäck, T., Fogel, D. B., & Michalewicz, Z. (Eds.). (2018). *Evolutionary Computation 1: Basic Algorithms and Operators*. CRC Press. <https://doi.org/10.1201/9781482268713>.
- Baldacci, R., Mingozzi, A., & Roberti, R. (2012). Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints. *European Journal of Operational Research*, 218(1), 1–6. <https://doi.org/10.1016/j.ejor.2011.07.037>.
- Baldacci, R., Toth, P., & Vigo, D. (2010). Exact algorithms for routing problems under vehicle capacity constraints. *Annals of Operations Research*, 175(1), 213–245. <https://doi.org/10.1007/s10479-009-0650-0>.
- Baumung, M. N., Gündüz, H. I., Müller, T., & Sebastian, H.-J. (2015). Strategic Planning of Optimal Networks for Parcel and Letter Mail. In H.-J. Sebastian, P. Kaminsky & T. Müller (Eds.), *Quantitative Approaches in Logistics and Supply Chain Management* (pp. 81–103). Springer. [https://doi.org/10.1007/978-3-319-12856-6\\_4](https://doi.org/10.1007/978-3-319-12856-6_4).
- Benjamin, A. M., & Beasley, J. E. (2013). Metaheuristics with disposal facility positioning for the waste collection VRP with time windows. *Optimization Letters*, 7(7), 1433–1449. <https://doi.org/10.1007/s11590-012-0549-6>.
- Boland, N., Hewitt, M., Marshall, L., & Savelsbergh, M. (2017). The Continuous Time Service Network Design Problem. *Operations Research*, 65(5), 1303–1321. <https://doi.org/10.1287/opre.2017.1624>.
- Borčinová, Z. (2017). Two models of the capacitated vehicle routing problem. *Croatian Operational Research Review*, 8(2), 463–469. <https://doi.org/10.17535/crorr.2017.0029>.
- Boussaïd, I., Lepagnot, J., & Siarry, P. (2013). A survey on optimization metaheuristics. *Information Sciences*, 237, 82–117. <https://doi.org/10.1016/j.ins.2013.02.041>.
- Bräysy, O., & Gendreau, M. (2005). Vehicle Routing Problem with Time Windows, Part I: Route Construction and Local Search Algorithms. *Transportation Science*, 39(1), 104–118. <https://doi.org/10.1287/trsc.1030.0056>.
- Casazza, M., Ceselli, A., & Calvo, R. W. (2018). A branch and price approach for the Split Pickup and Split Delivery VRP. *Electronic Notes in Discrete Mathematics*, 69, 189–196. <https://doi.org/10.1016/j.endm.2018.07.025>.
- Çetiner, S., Sepil, C., & Süral, H. (2010). Hubbing and routing in postal delivery systems. *Annals of Operations Research*, 181(1), 109–124. <https://doi.org/10.1007/s10479-010-0705-2>.
- 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), 756–773. <https://doi.org/10.1016/j.ejor.2005.08.015>.
- Cordeau, J.-F., Gendreau, M., Laporte, G., Potvin, J.-Y., & Semet, F. (2002). A guide to vehicle routing heuristics. *Journal of the Operational Research Society*, 53(5), 512–522. <https://doi.org/10.1057/palgrave.jors.2601319>.
- de Camargo, R. S., de Miranda, G., & Løkketangen, A. (2013). A new formulation and an exact approach for the many-to-many hub location-routing problem. *Applied Mathematical Modelling*, 37(12–13), 7465–7480. <https://doi.org/10.1016/j.apm.2013.02.035>.

- Dréo, J., Pétrowski, A., Siarry, P., & Taillard, E. (2006). *Metaheuristics for Hard Optimization: Methods and Case Studies*. Springer-Verlag. <https://doi.org/10.1007/3-540-30966-7>.
- Granada-Echeverri, M., Toro, E. M., & Santa, J. J. (2019). A mixed integer linear programming formulation for the vehicle routing problem with backhauls. *International Journal of Industrial Engineering Computations*, 10(2), 295–308. <https://doi.org/10.5267/j.ijiec.2018.6.003>.
- Kadri, R. L., & Boctor, F. F. (2018). An efficient genetic algorithm to solve the resource-constrained project scheduling problem with transfer times: The single mode case. *European Journal of Operational Research*, 265(2), 454–462. <https://doi.org/10.1016/j.ejor.2017.07.027>.
- Karimi, H., & Setak, M. (2018). A bi-objective incomplete hub location-routing problem with flow shipment scheduling. *Applied Mathematical Modelling*, 57, 406–431. <https://doi.org/10.1016/j.apm.2018.01.012>.
- Katoch, S., Chauhan, S. S., & Kumar, V. (2021). A review on genetic algorithm: past, present, and future. *Multimedia Tools and Applications*, 80(5), 8091–8126. <https://doi.org/10.1007/s11042-020-10139-6>.
- Kerivin, H. L. M., Lacroix, M., Mahjoub, A. R., & Quilliot, A. (2008). The splittable pickup and delivery problem with reloads. *European Journal of Industrial Engineering*, 2(2), 112–133. <https://doi.org/10.1504/EJIE.2008.017347>.
- Keskin, M., Laporte, G., & Çatay, B. (2019). Electric vehicle routing problem with time-dependent waiting times at recharging stations. *Computers & Operations Research*, 107, 77–94. <https://doi.org/10.1016/j.cor.2019.02.014>.
- Kora, P., & Yadlapalli, P. (2017). Crossover Operators in Genetic Algorithms: A Review. *International Journal of Computer Applications*, 162(10), 34–36. <https://doi.org/10.5120/ijca2017913370>.
- Laporte, G., Gendreau, M., Potvin, J.-Y., & Semet, F. (2000). Classical and modern heuristics for the vehicle routing problem. *International Transactions in Operational Research*, 7(4-5), 285–300. <https://doi.org/10.1111/j.1475-3995.2000.tb00200.x>.
- Leung, J. M. Y., Magnanti, T. L., & Singhal, V. (1990). Routing in Point-to-Point Delivery Systems: Formulations and Solution Heuristics. *Transportation Science*, 24(4), 245–260. <https://doi.org/10.1287/trsc.24.4.245>.
- Lim, S. M., Sultan, A. B. M., Sulaiman, M. N., Mustapha, A., & Leong, K. Y. (2017). Crossover and Mutation Operators of Genetic Algorithms. *International Journal of Machine Learning and Computing*, 7(1), 9–12. <http://doi.org/10.18178/IJMLC>.
- 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), 126–141. <https://doi.org/10.1016/j.ejor.2002.11.003>.
- Prins, C. (2002). Efficient Heuristics for the Heterogeneous Fleet Multitrip VRP with Application to a Large-Scale Real Case. *Journal of Mathematical Modelling and Algorithms*, 1(2), 135–150. <https://doi.org/10.1023/A:1016516326823>.
- Ralphs, T. K., Kopman, L., Puleyblank, W. R., & Trotter, L. E. (2003). On the Capacitated Vehicle Routing Problem. *Mathematical Programming*, 94(2–3), 343–359. <https://doi.org/10.1007/s10107-002-0323-0>.
- Rieck, J., & Zimmermann, J. (2010). A new mixed integer linear model for a rich vehicle routing problem with docking constraints. *Annals of Operations Research*, 181(1), 337–358. <https://doi.org/10.1007/s10479-010-0748-4>.



- Spliet, R., & Gabor, A. F. (2015). The Time Window Assignment Vehicle Routing Problem. *Transportation Science*, 49(4), 721–731. <https://doi.org/10.1287/trsc.2013.0510>.
- Squillero, G., & Tonda, A. (2016). Divergence of character and premature convergence: A survey of methodologies for promoting diversity in evolutionary optimization. *Information Sciences*, 329, 782–799. <https://doi.org/10.1016/j.ins.2015.09.056>.
- Theurich, F., Fischer, A., & Scheithauer, G. (2021). A branch-and-bound approach for a Vehicle Routing Problem with Customer Costs. *EURO Journal on Computational Optimization*, 9, 1–11. <https://doi.org/10.1016/j.ejco.2020.100003>.
- Toth, P., & Vigo, D. (Eds.). (2002). *The Vehicle Routing Problem*. Society for Industrial and Applied Mathematics. <https://doi.org/10.1137/1.9780898718515>.
- Umbarkar, A. J., & Sheth, P. D. (2015). Crossover operators in genetic algorithms: a review. *ICTACT Journal on Soft Computing*, 6(1), 1083–1092. <http://doi.org/10.21917/ijsc.2015.0150>.
- Vajda, P., Eiben, A. E., & Hordijk, W. (2008). Parameter Control Methods for Selection Operators in Genetic Algorithms. In G. Rudolph, T. Jansen, N. Beume, S. Lucas & C. Poloni (Eds.), *Parallel Problem Solving from Nature – PPSN X* (pp. 620–630). Springer. [https://doi.org/10.1007/978-3-540-87700-4\\_62](https://doi.org/10.1007/978-3-540-87700-4_62).
- Wassan, N. A., & Nagy, G. (2014). Vehicle Routing Problem with Deliveries and Pickups: Modelling Issues and Meta-heuristics Solution Approaches. *International Journal of Transportation*, 2(1), 95–110. <http://dx.doi.org/10.14257/ijt.2014.2.1.06>.