On the Generalized Directed Rural Postman Problem

Technical Report LM-2012-03

Michael Drexl
Chair of Logistics Management, Gutenberg School of Management and Economics,
Johannes Gutenberg University, Mainz
and
Fraunhofer Centre for Applied Research on Supply Chain Services SCS, Nuremberg
E-mail: drexl@uni-mainz.de

25th September 2012

Abstract

The generalized directed rural postman problem (GDRPP) is a generic type of arc routing problem. In the present paper, it is described how many types of practically relevant single-vehicle routing problems can be modelled as GDRPPs. This demonstrates the versatility of the GDRPP and its importance as a unified model for postman problems. In addition, a heuristic solution approach, based on a transformation into the generalized asymmetric travelling salesman problem (GATSP), is presented. Computational experiments using a state-of-the-art heuristic for the GATSP (developed by Gutin and Karapetyan [27]) are performed, using a large set of realistic instances. The results show excellent solution quality in short computation time and thus demonstrate the practical usefulness of the transformations.
Keywords: arc routing; rural postman problems; graph transformations; heuristics; generalized asymmetric travelling salesman problem

1 Introduction

This paper studies the generalized directed rural postman problem (GDRPP), which was first considered in Drexl [18]. As its name implies, the problem is a generalization of the directed rural postman problem (DRPP). The DRPP consists in finding an optimal postman tour in a digraph, that is, a least-cost cycle traversing each arc of a specified subset of the digraph's arcs at least once. The GDRPP is, to the DRPP, what the generalized travelling salesman problem (GTSP) is to the travelling salesman problem (TSP): In the GDRPP, there are several subsets (groups, classes, clusters) of arcs and the requirement is to find a least-cost cycle traversing at least one arc from each subset at least once. The subsets need neither be a partition of the arc set, nor need they be disjoint. The GDRPP is an interesting problem in its own right, but it is also important because many single-vehicle routing problems, especially arc routing problems (ARPs), can be modelled as GDRPPs. Moreover, there are several practically relevant constraints that can also be considered when modelling a problem as a GDRPP. Applications of ARPs are numerous. Apart from the eponymous mail delivery context, there are case studies in street sweeping (Bodin and Kursh [6]), meter reading (Stern and Dror [41]), winter gritting (Eglese and Li [23]), and robot motion control (Benavent et al. [3]), among others.
The idea of transforming problems into other problems is not new. The contribution of the present paper is that it extends and unifies existing transformation procedures under a common transformation target, the GDRPP. Moreover, to verify the applicability of the proposed transformations, it presents and analyzes the results of extensive computational experiments on a large set of test instances.

In the following, it is assumed that the reader is familiar with basic graph theory concepts as treated, for example, in West [43]. The standard reference textbook on arc routing is Dror [20]; a recent survey is Corberán and Prins [17]. The present paper extends the results described in Drexl [18]. Relevant specialized literature and further related work is discussed in the respective sections, where the problems to be transformed are introduced.
The rest of the paper is structured as follows. The next section specifies the formal notation required to describe the subsequent models and transformations. In Section 3, an integer programming formulation for the GDRPP is presented. Section 4 describes the proposed problem transformations in detail. In Section 5, the results of the computational experiments are presented. The paper ends with a short conclusion in Section 6.

2 Notation

The following notation is used. $G = (V, L)$ is a graph with vertex set V and link set L. Without loss of generality, it is assumed throughout that all considered graphs are simple. This allows to unequivocally identify a link by its two adjacent vertices. The index notation f_i for the value $f(i)$ of a function f is used throughout. $c : L \to \mathbb{R}_+$ is a function denoting the *length* or the *costs of traversal of a link*. M is an arbitrarily large positive constant. Links that may be traversed in either direction are called *edges*; links that may be traversed only in one direction are called *arcs*. An arc referenced by (i, j) can only be traversed from its *tail* i to its *head* j. Two arcs (i, j) and (i', j') are called *antiparallel* if $i = j'$ and $j = i'$. Graphs containing only arcs are called *digraphs* and are denoted by $D = (V, A)$. Graphs containing edges as well as arcs are called *mixed graphs*. A graph $G = (V, L)$ is called *windy* if there are two functions $\vec{c} : L \to \mathbb{Z}_+ \uplus \{\infty\}$ and $\overleftarrow{c} : L \to \mathbb{Z}_+ \uplus \{\infty\}$ denoting the costs of traversal of a link from tail to head and from head to tail respectively. As the costs of traversal in one direction may be infinite, mixed graphs are special cases of windy graphs. For a graph $G = (V, L)$, $L := E \uplus A$, where E (A) is the edge (arc) set of G, that is, the set of links that can be traversed in both directions (possibly at different costs), and the set of links that can only be traversed from tail to head respectively. Furthermore, for a digraph $D = (V, A)$ and $A' \subseteq A$, let $HE_{A'} \subseteq V$ be the set of the heads of the arcs in A', and let $TA_{A'}$ be defined accordingly for tails. It is assumed throughout that all considered (di)graphs are (strongly) connected.
For *generalized* problems on a graph $G = (V, L)$, there are p_L groups $L_q \subsetneq L$, $q = 1, \ldots, p_L$, of links, and p_V groups $V_q \subsetneq V$, $q = 1, \ldots, p_V$, of vertices, whereof at least one element must be traversed/visited in any feasible solution. Such groups are referred to hereafter as *r-groups*. (This term is used to point out that these groups may comprise only a proper subset of links or arcs and may overlap; terms such as 'classes' or 'clusters' carry the connotation of forming a partition of a set, that is, of being exhaustive and disjoint.) The links (vertices) belonging to or *covering* an r-group are called *r-group links* (*r-group vertices*).
A link/vertex that will be traversed/visited in any optimal solution is called *required*. For simplicity of notation, let $HE_q := HE_{L_q}$, and let $TA_q := TA_{L_q}$. Without loss of generality, it is assumed that there is at least one *head-disjoint* r-group, that is, an r-group $\tilde{q}$ with $HE_{\tilde{q}} \cap \bigcup_{q=1,\ldots,p_L, q\neq\tilde{q}} HE_q = \emptyset$.
Finally, traversing a non-required or a non-r-group arc, or traversing a required or an r-group arc without servicing it, is referred to as *deadheading*.

3 An integer programming formulation

The formulation presented in this section is based on a digraph $D = (V, A)$ with p_A r-groups $1, \ldots, p_A$ of arcs. It assumes that r-group 1 is a head-disjoint r-group of minimal cardinality, that is, $|HE_1| \leq |HE_q|$ for all head-disjoint r-groups $q \in \{1, \ldots, p_A\}$. The formulation uses two types of variables: General integer variables x_{ij}, for all $(i, j) \in A$, indicating the number of times

arc (i, j) is traversed in a solution, and binary variables y_i, for all $i \in V$, indicating whether or not vertex i is visited. The formulation is as follows:

(GDRPP):

$$\text{minimize} \sum_{(i,j)\in A} c_{ij}x_{ij} \tag{1}$$

subject to

$$\sum_{(i,j)\in A_q} x_{ij} \geq 1 \quad \forall\, q \in \{1,\ldots,p_A\} \tag{2}$$

$$\sum_{(h,i)\in A} x_{hi} - \sum_{(i,j)\in A} x_{ij} = 0 \quad \forall\, i \in V \tag{3}$$

$$x_{ij} - (|A|+1)y_i \leq 0 \quad \forall\, i \in V, (i,j) \in A \tag{4}$$

$$\sum_{(j,k)\in A: j\in S\not\ni k} x_{jk} - y_i - y_{i'} \geq -1 \quad \forall\, i \in HE_1,\, TA_q \ni i' \in S \subseteq V \setminus \{i\}, q \in \{2,\ldots,p_A\} \tag{5}$$

$$x_{ij} \in \mathbb{Z}^0_+ \quad \forall\, (i,j) \in A \tag{6}$$

$$y_i \in \{0,1\} \quad \forall\, i \in V \tag{7}$$

Constraints (2) ensure that each r-group is covered, constraints (3) guarantee flow conservation, and constraints (4) establish the connection between the x and the y variables.
A difficulty with generalized ARPs is that not every vertex need be incident to a required link, because there need not be required links at all: If all r-groups have cardinality greater than one, it is impossible to tell in advance whether a certain link will be traversed in any optimal solution or not. If there is no r-group q with $\bigcap_{(i,j)\in A_q}\{i,j\} \neq \emptyset$, there is not even a required vertex. This complicates the specification of subtour elimination constraints. Formulation (GDRPP) uses constraints (5) to eliminate subtours. These constraints require that, for each vertex i in the head-disjoint r-group HE_1 and each vertex i' that belongs to a subset S of V not containing i and is the tail of an r-group arc from an r-group other than 1, if both vertices are visited, then an arc leaving S must be used.
Violated constraints (5) can be separated by computing the maximum flow between all pairs of vertices i, i' with $HE_1 \ni i \neq i' \in TA_q, q \in \{2,\ldots,p_A\}$ in the support graph. For each pair i and i' where the maximum flow is less than one, the respective constraint (5) can then be checked and added if it is violated. (This procedure does not work on graphs without head-disjoint r-groups. If a vertex i is the head of two arcs belonging to different r-groups, the maximum flow from i to i must be computed, which is impossible with standard max-flow algorithms without modifying the graph. This is why head-disjoint r-groups were introduced.)

4 Transformations

In this section, various types of uncapacitated routing problems are considered. They can be classified along several orthogonal dimensions:

- Type of routing application
  - Arc routing
  - Vertex routing
  - Combined arc and vertex routing
- Graph type
  - Undirected
  - Directed
  - Mixed
  - Windy

- Generalized, clustered, and hierarchical models
- Additional real-world constraints
  - Turn penalties
  - Street segment sides
  - Zigzag service
  - Different costs for servicing and deadheading in different directions
  - Use of public transport
  - Open tours

In the following, formulations such as 'problem A is transformed into (is modelled as) problem B' are used as shortcuts for 'a graph representing an instance of problem B is created from a graph representing an instance of problem A'. All of the presented transformations have to be interpreted as a proof of concept. They just describe one of several possibilities of how to (polynomially) transform one problem into another. No claim is made as to whether the presented transformations are the simplest or the most elegant ones.

4.1 Standard arc routing problems

The archetypal arc routing problem is the Chinese postman problem (CPP, Guan [26]), which seeks a least-cost cycle in a graph, traversing each link at least once. The problem of finding a least-cost cycle traversing only a specified subset of the links of a graph at least once is called rural postman problem (RPP, Orloff [37]). Depending on the type of graph, these problems are referred to as undirected, directed, mixed, or windy CPP/RPP.

For routing problems on graphs containing links that may be traversed in both directions, one part of the solution consists in determining a direction of traversal for each such link that is to be used in the solution. This simply means the selection of one out of two alternatives. Hence, it is common in the literature to represent an edge or a windy link by two variables whose values indicate the number of traversals of the link in the respective direction, cf. Win [44]. Therefore, to model the above standard arc routing problems as GDRPPs, all links that can be traversed in both directions are replaced by two antiparallel arcs with the corresponding traversal costs. Each such pair of antiparallel arcs corresponding to a required link forms one r-group in the GDRPP, and each required original arc forms one r-group, too.

4.2 The clustered generalized directed rural postman problem

This problem is encountered in the context of postal delivery: The streets (or street segments, or street segment sides) of a town are partitioned into (not necessarily connected) 'delivery sections' or 'service units' that must be serviced consecutively in an arbitrary order. The use of streets from different delivery sections for deadheading is allowed. Not all streets necessarily need be serviced. The problem can be modelled as a GDRPP as follows. First, all undirected, windy or zigzag links are replaced by appropriate arcs and the appropriate r-groups are created. Then, all service units are made strongly connected by adding deadheading arcs corresponding to shortest paths between the respective end vertices in the original graph, so that each vertex of a service unit can be reached from any other vertex of the service unit. After that, for each service unit in the original graph, a 'super-cluster' containing all r-groups corresponding to links of the respective service unit is created. Finally, M is added to the costs of each arc connecting two super-clusters. See Figure 1.

Dror and Langevin [21] considered the clustered DRPP (without the 'generalized' component), transformed it into the GATSP, and solved it exactly by a Lagrangian-based method due to Noon and Bean [36]. The largest instances they solved had 581 vertices and 770 arcs.

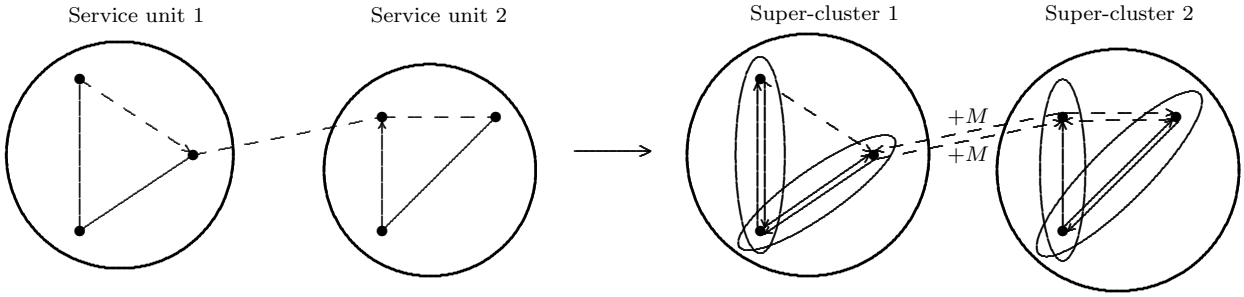

Figure 1: Clustered generalized DRPP

4.3 Hierarchical postman problems

Several authors consider so-called hierarchical postman problems (HPPs), see, e.g., Ghiani and Improta [24], Cabral et al. [11]. In HPPs, there are disjoint clusters of links as in the clustered version just described, and there is a given order (a 'hierarchy') in which the clusters have to be serviced, that is, the sequence of clusters is fixed. This can be modelled as a special case of a clustered DRPP by adding M to all intercluster arcs for clusters that must be serviced consecutively, and by adding $2M$ to all other intercluster arcs.

4.4 Turn penalties

There is a considerable amount of literature on ARPs with turn penalties, starting with the paper by Caldwell [13]. Other important contributions are Wattleworth and Shuldiner [42], Kirby [31], Kirby and Potts [32], Wyskida and Gupta [46], Bodin and Kursh [6], Bodin and Kursh [7], McBride [35], Roy and Rousseau [38], Añez et al. [1], Benavent and Soler [4], Bousonville and Kopfer [9], Clossey et al. [14], Corberán et al. [15], Winter [45], Arkin et al. [2], and Soler et al. [40]. Some authors develop solution algorithms working directly on the graphs with turn penalties, others use transformations into arc or vertex routing problems without turn penalties. A detailed review would go beyond the scope of this paper. One possibility to transform a graph G_{TP} with turn penalties into a graph G without, which is used for the computational experiments described in Section 5, works as follows: A graph G is created that contains one arc for each possible direction of traversal of each link of G_{TP}. These arcs form the set S. Then, additional non-r-group arcs from the head of each arc $(i, j) \in S$ to the tail of each other arc $(i', j') \in S$ are inserted if and only if the corresponding turn in G_{TP} is allowed. The costs of these additional arcs are set to the corresponding turn penalty. An example (adapted from Grünert and Irnich [25], p. 615) is given in Figure 2.
Other possibilities for transforming a graph with turn penalties into one without are described in Caldwell [13], Añez et al. [1], and Benavent and Soler [4].

4.5 Street segment sides

Irnich [29] considers several real-world ARP extensions occurring in postal delivery. One of them is that 'for delivery by foot or by bicycle, it may be necessary to distinguish between a traversal on the left-hand side (lhs) or right-hand side (rhs) of a street segment', which 'offers the possibility to favour tours that do not cross main roads too often in order to make the tour faster and safer for the postman' (p. 54). This can be modelled in a GDRPP by introducing deadheading arcs reflecting the different costs for traversing a street segment between two junctions/vertices in either direction and on either side of the street.

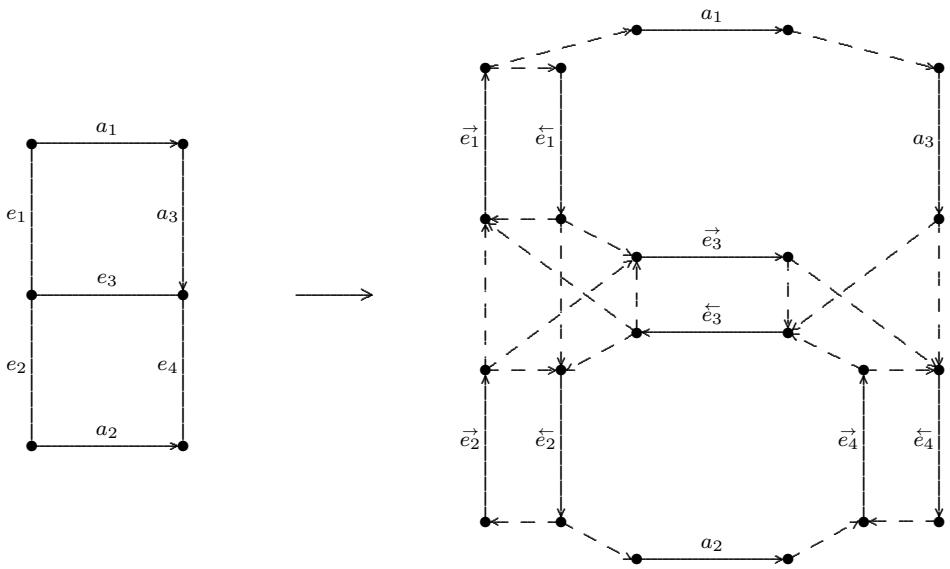

Figure 2: Transformation of a graph with turn penalties into one without

4.6 Zigzag service

The issue of zigzag service also arises in the context of postal delivery. Imagine a postman who must deliver mail in a street segment between two road junctions. There are houses on both sides of the street segment. If there is not too much traffic in the segment, the postman has several possibilities for servicing it: He may service one side at a time (and not necessarily both sides contiguously), which means that he must traverse the street segment at least twice, and he may service both sides simultaneously (in 'zigzag mode'), which means that he must traverse the street segment only once. See Figure 3.

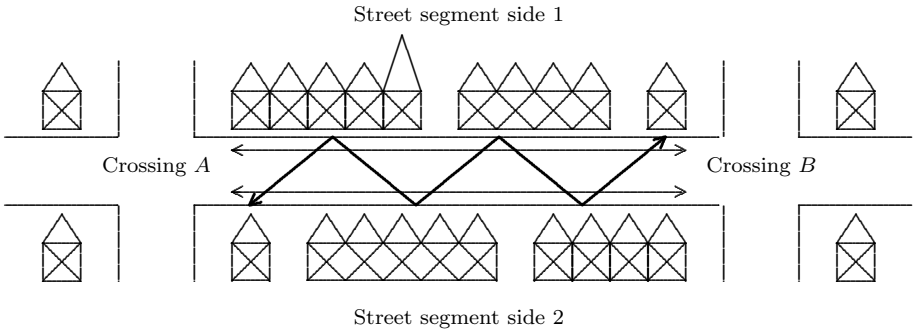

Figure 3: Street with zigzag service option

Thus, zigzag service of a required segment means that it can be serviced by traversing it once in each direction, by traversing it twice from tail to head, by traversing it twice from head to tail, or by traversing it once in any direction in 'zigzag mode'. Figure 4 shows how this can be modelled by two overlapping r-groups. For a potential zigzag service street segment, two vertices, eight arcs, and two r-groups are introduced. There are two non-r-group deadheading arcs, one for each direction, two antiparallel r-group arcs for each street segment side, and two antiparallel r-group arcs for zigzagging. To service the street segment, the postman must either use one of the zigzag arcs or two 'street segment side arcs', one for each street segment side. Grouping the arcs in two overlapping r-groups, one for each street segment side, such that the two zigzag arcs belong to the intersection of the r-groups, yields the desired result. (In Figure 4, there is only one deadheading arc per direction. If street segment sides are to be distinguished

also for deadheading as described in Section 4.5, there has to be an additional deadheading arc in either direction.)

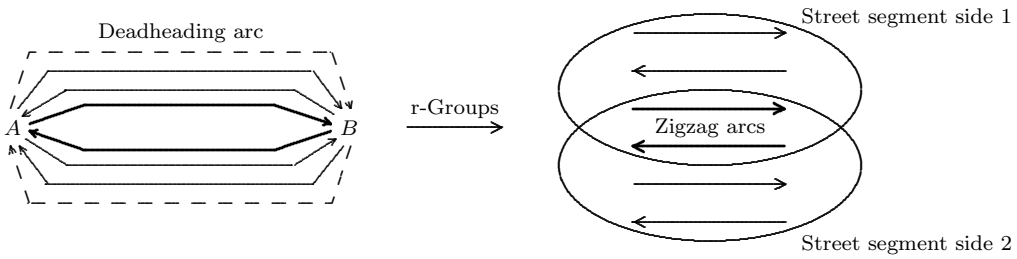

Figure 4: Modelling of zigzag service in a GDRPP

An alternative approach for considering zigzag service is proposed by Irnich [28].

4.7 Different costs for servicing and deadheading in different directions

An extension considered by Lacomme et al. [33] is the possibility of two different costs per link for traversal with and without servicing. For windy graphs, this must be generalized to four different costs, depending on the direction of traversal and on whether the link is being serviced or not. This, too, can be modelled within a GDRPP, by introducing two service and two deadheading antiparallel arcs for each such windy link, where the two service arcs form one r-group, similar to the procedure for zigzag service.

4.8 Use of public transport

An additional aspect, described in Irnich [29], is the use of public transport: A postman does not have to use his bicycle or walk to get from the post office to his delivery district, but may use public transport such as bus, tramway, or underground. This can simply be modelled by introducing shortcut arcs from/to the vertex corresponding to the post office to/from any other relevant vertex.

4.9 Open tours

Open tours, that is, tours that do not have to end where they started, have recently been discussed by several authors in the context of capacitated vehicle routing problems (e.g., Brandão [10]). Also in postal applications, it is possible that the postman ends his workday immediately after the last delivery and returns to the post office only on the next day (Irnich [29]). This can be modelled as follows. First, a fixed starting vertex s (corresponding to the post office) is specified by creating an r-group that contains all arcs emanating from s. Then, one arc with cost M is added from each vertex where the tour may end to s, and all these 'return' arcs are put into one r-group. In this way, it is ensured that the post office is visited (left) at the beginning of the tour, and that exactly one return arc is used.

4.10 The generalized travelling salesman problem

The best-known combinatorial optimization problem is the travelling salesman problem (TSP), which basically comes in two variants. On undirected graphs, it is called symmetric TSP (STSP); on directed graphs, it is called asymmetric TSP (ATSP). The generalized travelling salesman problem (GTSP) is a TSP on a graph whose vertex set is partitioned into disjoint clusters with the requirement that exactly (or, sometimes, at least) one vertex from each cluster be visited on any feasible tour.
The transformation (or, rather, interpretation) of a GATSP into (as) a GDRPP is simple: Each cluster of vertices is replaced by one r-group of arcs containing all arcs emanating from

(or, equivalently, leading to) one of the vertices in the cluster. An alternative procedure is to replace each cluster of vertices by an r-group of arcs with costs of zero and with a one-to-one correspondence between the original vertices and the arcs replacing them, and to set the heads (tails) of all original arcs leading to (emanating from) an original vertex equal to the tail (head) of its corresponding newly added arc.
The inverse transformation is also straightforward. Adopting ideas from Laporte [34], a GDRPP instance can be transformed into a GATSP instance as follows. Each r-group arc is replaced by a vertex. Vertices v_{ij}, $v_{i'j'}$ corresponding to arcs (i,j), (i',j') covering different r-groups are connected by two antiparallel arcs with costs equal to the costs of a shortest path from j to i' plus c_{ij}, and from j' to i plus $c_{i'j'}$ respectively. All r-groups remain logically the same, the only difference is that they now contain vertices instead of arcs.
Laporte [34] considers transformations of several types of arc routing problems (though not the GDRPP) into ATSPs by first transforming the ARPs into GATSPs as described in the previous paragraph and then transforming the GATSPs into ATSPs by a procedure due to Noon and Bean [36]. Laporte [34] solves to optimality (transformations of) mixed CPPs with up to 440 arcs and 10 edges and mixed RPPs with up to 660 arcs and 5 edges.
When solving the GTSP, it is interesting that existing algorithms (cf. Noon and Bean [36], Dror and Langevin [21]) assume disjoint clusters. However, already Noon and Bean [36] state that this means no loss of generality, as a problem with overlapping clusters can be transformed into a problem with disjoint clusters. This can be done as follows. For each vertex i covering $q > 1$ r-groups, $q-1$ duplicates are added to the underlying digraph, so that, in the modified digraph, there are q vertices corresponding to i. To each of i's clusters, one of these q vertices is assigned, and each of these vertices has the same entering and outgoing arcs as i, with the same costs. In addition, arcs with zero cost are added between each pair of duplicates of i. In this way, when a duplicate of i is visited, it is always best to visit all other duplicates of i (and thus covering all clusters originally covered by i) before moving on to another vertex and another cluster.

4.11 The generalized directed general routing problem

The general routing problem (GRP), which was introduced by Orloff [37], constitutes the connection between postman and travelling salesman problems. It is usually described for undirected graphs, but the directed version of the problem can be described as follows. Given a digraph, a specified subset of its arcs and a specified subset of its vertices, the problem is to find a least-cost cycle traversing/visiting each arc/vertex of the respective subsets at least once. The extension of this problem corresponding to the GDRPP is the generalized directed general routing problem (GDGRP).
The GDGRP contains, as special cases, the symmetric and the asymmetric travelling salesman problem as well as their 'generalized' versions, and also the windy general routing problem examined in Corberán et al. [16]. The transformation of a GDGRP into a GDRPP works as described for the GATSP. If at least one element of a subset S of $V \uplus A$ is to be visited/traversed, this can be modelled by adding all arcs (i,j) with $i \in S \cap V$ to S.
Blais and Laporte [5] solve the undirected, directed, and mixed GRP (not the generalized GRP) by transformations into GATSP, ATSP, and undirected, directed, and mixed RPP. The transformations into GATSP and ATSP work as described in Section 4.1. In addition, for each required vertex in the GRP, there is a corresponding vertex in the GATSP and the ATSP. The transformation of an undirected, directed, or mixed GRP into a corresponding RPP consists simply in replacing each required vertex by a corresponding link with costs of zero, as described in Section 4.10. Blais and Laporte [5] solve to optimality directed GRPs with up to 4,000 required vertices and 3,000 required arcs, undirected GRPs with up to 90 required vertices and 10 required edges or 10 required vertices and 90 required edges, and mixed GRPs with up to 110 required vertices, 200 required arcs and 25 required edges.

4.12 Complexity results

After the detailed description of the transformations, some remarks on the difficulty of the considered problems are appropriate, since the interest of transformations from one problem type into another is, in general, limited to *NP*-hard problems. With the exception of the undirected and directed CPP, for which Edmonds and Johnson [22] give polynomial algorithms, all considered arc routing problems are *NP*-hard (Dror [19]). All considered vertex routing and rural postman problems are even *NP*-hard in the strong sense. The former is shown in Johnson and Papadimitriou [30], the latter is easily seen by reducing the STSP to the undirected RPP (URPP): A graph of an STSP instance is polynomially transformed into a graph of a URPP instance by creating a duplicate of each vertex and connecting each original vertex with its duplicate by an edge with costs of zero. The additional edges are required; all original edges are non-required. Moreover, Corberán et al. [15] prove that postman problems with turn penalties are *NP*-hard in the strong sense. Consequently, the GDRPP itself is strongly *NP*-hard as well.

5 Computational experiments

Given the simplicity of transforming GDRPPs and GTSPs into one another, and considering that no heuristic solution procedure for the GDRPP, other than the coarse heuristic described in Drexl [18], has been proposed up to now, it is reasonable to try to use existing heuristics for the GTSP (of which there are several; see, for example, Snyder and Daskin [39], Bontoux et al. [8], Cacchiani et al. [12]) to solve GDRPPs. To this end, the memetic algorithm presented by Gutin and Karapetyan [27] (henceforth abbreviated by 'GK') is used in this paper. Contrary to most other approaches, it can be directly applied to the asymmetric GTSP, the computational behaviour compares favourably with other procedures and can be considered state of the art, and its authors have kindly made it available over the Internet (`http://www.sfu.ca/~dkarapet`). The algorithm consists of the following four steps. First, an initial generation of solutions is created by means of a semirandom construction heuristic. Second, each of the first generation of solutions is brought to a local optimum by local search, and duplicate solutions are eliminated. Third, a next generation is produced, using reproduction, crossover, and mutation operators. Fourth, the next generation is again locally optimized by local search. Steps two to four are iterated until no improved solution is found in several subsequent generations. Six different local search procedures are used.

Drexl [18] has created a set of 420 test instances. Three types of instances were created. Each type consists of several classes differing with respect to the number of vertices, arcs, and r-groups. The first type consists of random GDRPP instances. The characteristics of the created instance classes are shown in Table 1.

No. of created instances	No. of vertices	No. of arcs	No. of r-groups	Max. % r-group arcs covering one r-group	Max. % r-groups covered by one arc
10	50	500	10	2	20
10	50	500	15	2	20
10	50	500	20	2	10
30	50	500	25	2	10
30	50	500	50	2	5
30	100	1,000	100	1	4

Table 1: Characteristics of created random GDRPP instances

For all classes, n instances with disjoint and n instances with not necessarily disjoint r-groups were created, where n is the value in the first column of Table 1. The entries in the last two columns refer only to the instances with not necessarily disjoint r-groups. The instances with 10, 15, and 20 r-groups were created by deleting 15, 10 and 5 r-groups from the instances with 25 r-groups. All instances had 50 % r-group arcs. The arc costs were selected randomly from $[1; 100]$.

The second instance type consists of transformed windy rural postman problems with turn penalties (WRPPTP). Three classes were created: 10 vertices and 40 links, 20 vertices and 80 links, and 30 vertices and 120 links. For each class, 30 instances were created. Each instance had 50 % required links. 50 % of the links were symmetric edges, and 25 % of the links were arcs. The link costs were selected randomly from $[1; 100]$, and the turn penalties were selected randomly from $[0; 100]$. 3 % of the turns were forbidden.
The third type consists of windy rural postman problems with zigzag service (WRPPZZ) with different costs for servicing and deadheading in different directions. Three classes of WRPPZZ instances were created: 30 vertices and 100 links, 40 vertices and 150 links, and 50 vertices and 200 links. For each type, 30 instances were created. Each instance had 50 % required links. 50 % of the links were symmetric edges, 25 % of the links were arcs, and 25 % of the links were zigzag links. The deadheading costs were selected randomly from $[1; 100]$. The costs for one-sided service were computed by multiplying the deadheading costs by a random factor of between 1 and 2. The costs for zigzag service were computed by summing up the costs for the two one-sided services of the respective link in the respective direction and by multiplying the sum by a random factor of between 1 and 2.
To solve the instances with the GK heuristic, they were transformed into GATSPs with disjoint clusters as described in Section 4.10. To assess the quality of the solutions obtained with the GK heuristic, the instances were solved as integer programs (IPs) in formulation (1)–(7), which was implemented in C++ using IBM Ilog Cplex Concert technology (`http://www.cplex.com`), V 11.0, in single thread mode and with strong branching. Most instances could be solved to optimality within a few seconds. All but two instances could be solved to optimality within 90 minutes of CPU time. The remaining two instances could be solved in less than 30 minutes after specifying the best known solution of the GK heuristic as upper bound. Both GK and IP were run on a laptop computer with a 2.16 GHz processor and 2 GB of main memory under Win XP SP 2, 32-bit.
The following Table 2 presents the instance sizes in formulation (1)–(7), that is, the columns 'No. vertices', 'No. arcs', 'No. constraints', and 'No. r-Groups' refer to the respective values in the representation of the instances as GDRPPs in formulation (1)–(7). The numbers in these columns are slightly higher than what might be expected looking at formulation (1)–(7) and Table 1. This is due to the addition of a head-disjoint r-group, which was automatically added to all instances without prior checking whether or not an instance contains such an r-group already. It must be noted that the largest asymmetric GTSP benchmark instance solved in the literature (as cited in Gutin and Karapetyan [27]) contains only 89 clusters, so that the largest random GDRPP and the WRPPZZ instances considered in this paper can be considered very large in this respect.

Instance type	No. instances	No. vertices (y_i vars.)	No. arcs (x_{ij} vars.)	No. constraints (without SECs (5))	No. r-Groups (average)	Disjoint r-groups?
Random 50_500_10_n	10	58	516	585	11	n
Random 50_500_10_y	10	55	510	576	11	y
Random 50_500_15_n	10	58	516	589	16	n
Random 50_500_15_y	10	55	510	580	16	y
Random 50_500_20_n	10	58	515	594	21	n
Random 50_500_20_y	10	54	508	584	21	y
Random 50_500_25_n	30	58	515	599	26	n
Random 50_500_25_y	30	54	509	589	26	y
Random 50_500_50_n	30	52	503	606	51	n
Random 50_500_50_y	30	51	502	605	51	y
Random 100_1000_100_n	30	101	1,002	1,205	101	n
Random 100_1000_100_y	30	100	1,002	1,204	101	y
WRPPTP 10_40	30	142	586	748	20	n
WRPPTP 20_80	30	282	1,197	1,522	43	n
WRPPTP 30_120	30	420	1,776	2,256	60	n

(continued on next page)

Instance type	No. instances	No. vertices (y_i vars.)	No. arcs (x_{ij} vars.)	No. constraints (without SECs (5))	No. r-Groups (average)	Disjoint r-groups?
WRPPZZ 30_100	30	31	371	500	98	n
WRPPZZ 40_150	30	41	559	749	149	n
WRPPZZ 50_200	30	51	736	983	196	n
Overall average		107	735	910	68	

Table 2: Instance sizes in formulation (1)–(7)

The computational results are shown in the subsequent Table 3. For the GK heuristic, three runs were performed for each instance, because the algorithm contains a random component; the IP solver was run only once. For the second, third, and fourth column, each row indicates the minimum, average, and maximum values over all instances (and all runs) of the respective type. The values in the column 'GK relative gap to optimum' are computed as $(OF_{GK} - OF_{opt})/OF_{opt}$, where OF_{GK} is the minimum, average, and maximal objective function value of the GK heuristic over all three runs, and OF_{opt} is the optimal objective function value. The column 'GK solved to optimality' indicates the percentage of instances of each type for which an optimal solution was found in at least one of the three runs.

Instance type	Running time IP [CPU seconds] (min. / avg. / max.)	Running time GK [CPU seconds] (min. / avg. / max.)	GK relative gap to optimum [%] (min. / avg. / max.)	GK solved to optimality [%]
Random 50_500_10_n	0.1 / 1.1 / 4.1	0.3 / 0.3 / 0.4	0.0 / 0.0 / 0.0	100.0
Random 50_500_10_y	0.0 / 1.3 / 6.5	0.2 / 0.2 / 0.3	0.0 / 0.0 / 0.4	100.0
Random 50_500_15_n	0.0 / 20.6 / 136.4	0.5 / 0.7 / 0.9	0.0 / 0.0 / 0.0	100.0
Random 50_500_15_y	0.1 / 0.8 / 3.3	0.3 / 0.5 / 0.6	0.0 / 0.3 / 2.9	90.0
Random 50_500_20_n	0.1 / 6.0 / 48.9	1.0 / 1.2 / 1.5	0.0 / 0.4 / 4.6	90.0
Random 50_500_20_y	0.0 / 1.1 / 4.7	0.6 / 0.8 / 1.2	0.0 / 0.6 / 2.9	80.0
Random 50_500_25_n	0.0 / 9.1 / 124.1	1.3 / 2.0 / 4.5	0.0 / 1.0 / 5.5	70.0
Random 50_500_25_y	0.0 / 2.7 / 44.4	0.8 / 1.4 / 3.5	0.0 / 0.7 / 4.9	70.0
Random 50_500_50_n	0.0 / 0.4 / 1.6	1.7 / 3.0 / 6.8	0.0 / 1.3 / 4.3	20.0
Random 50_500_50_y	0.0 / 0.4 / 1.5	1.5 / 2.8 / 7.1	0.0 / 1.2 / 3.7	23.3
Random 100_1000_100_n	0.1 / 4.9 / 29.5	16.7 / 43.8 / 129,7	0.2 / 3.5 / 8.8	0.0
Random 100_1000_100_y	1.0 / 6.9 / 30.0	13.5 / 34.4 / 123.5	0.7 / 3.0 / 6.1	0.0
WRPPTP 10_40	0.0 / 2.1 / 18.7	0.0 / 0.1 / 0.4	0.0 / 0.1 / 2.2	93.3
WRPPTP 20_80	0.1 / 48.4 / 384.6	0.5 / 1.8 / 5.1	0.0 / 0.7 / 4.1	53.3
WRPPTP 30_120	1.4 / 716.2 / 6,071.4	2.0 / 7.0 / 22.3	0.0 / 1.5 / 3.1	13.3
WRPPZZ 30_100	0.0 / 0.0 / 0.1	4.6 / 19.9 / 67.0	0.0 / 0.3 / 1.0	36.7
WRPPZZ 40_150	0.0 / 3.6 / 105.1	39.4 / 109.1 / 353.1	0.0 / 0.5 / 1.8	6.7
WRPPZZ 50_200	0.0 / 0.1 / 0.6	118.7 / 402.5 / 1,491.6	0.0 / 0.6 / 1.4	0.0
Overall	0.0 / 57.5 / 6,071.4	0.0 / 44.9 / 1,491.6	0.0 / 1.1 / 8.8	41.0

Table 3: Computational results

The following observations can be made in Table 3:

- The solution quality of the heuristic is very good for all instances; the average deviation from the optimal solution is only 1.1 %.
- The instances with overlapping r-groups are more difficult (take longer) to solve with the heuristic than those with disjoint r-groups. For the exact algorithm, the situation is similar, but not so pronounced. With respect to solution quality, the comparison between the instances with overlapping and those with disjoint r-groups is undetermined.
- The running time of the exact algorithm depends strongly on the number of arcs (which determines the number of variables and constraints), whereas the running time of the heuristic depends on the number of r-groups.
- The exact algorithm has a high variation in computation time also for instances of the same type; running times are much less predictable than for the heuristic.

Irnich [29] reports that mail delivery districts (to be served by one postman) for urban areas in Germany range in size from 19 to 100 delivery street segments with between 0 and 39 zigzag segments. This means that the largest test instances of each type reflect the average size of such a district. Hence, a viable solution approach for many different types of uncapacitated real-world single-postman problems is to model them as GDRPPs and then transform them into GATSPs.

6 Conclusion

It has been demonstrated that the generalized directed rural postman problem provides a flexible and generic modelling approach for a broad range of single-vehicle routing problems. The computational experiments show that GDRPPs of realistic size can be solved quickly and with high solution quality by a transformation into a generalized asymmetric travelling salesman problem. The memetic algorithm developed by Gutin and Karapetyan [27] for the GATSP is very robust with respect to instance structure and computes close-to-optimal solutions for a wide variety of problems.
The presented IP formulation already works quite well for instances with not too many vertices and arcs, and the time needed for its exact solution by a standard branch-and-cut solver is rather insensitive to the number of r-groups. Therefore, an interesting topic for further study is to improve this formulation by finding strong valid inequalities to tighten its LP relaxation. Furthermore, since mail delivery problems larger than those used in the computational experiments will most probably require more than one postman, future research could extend the models and algorithms presented in this paper to generalized versions of capacitated arc routing problems.

Acknowledgement This research was funded by the Deutsche Forschungsgemeinschaft (DFG) under grant no. IR 122/5-1. This support is gratefully acknowledged. The author also thanks G. Gutin and D. Karapetyan for making the code of their GATSP heuristic freely available.

References

[1] Añez J, de la Barra T, Pérez B (1996):
Dual Graph Representation of Transport Networks
Transportation Research Part B 30: 209–216

[2] Arkin E, Bender M, Demaine E, Fekete S, Mitchell J, Sethia S (2005):
Optimal Covering Tours with Turn Costs
SIAM Journal on Computing 35: 531–566

[3] Benavent E, Carrotta A, Corberán A, Sanchis J, Vigo D (2007):
European Journal of Operational Research 176: 855–869

[4] Benavent E, Soler D (1999):
The Directed Rural Postman Problem with Turn Penalties
Transportation Science 33: 408–418

[5] Blais M, Laporte G (2003):
Exact Solution of the Generalized Routing Problem Through Graph Transformations
Journal of the Operational Research Society 54: 906–910

[6] Bodin L, Kursh S (1978):
A Computer-Assisted System for the Routing and Scheduling of Street Sweepers
Operations Research 26: 525–537

[7] Bodin L, Kursh S (1979):
A Detailed Description of a Computer System for the Routing and Scheduling of Street Sweepers
Computers & Operations Research 6: 181–198

[8] Bontoux B, Artigues C, Feillet D (2010):
A Memetic Algorithm with a Large Neighborhood Crossover Operator for the Generalized Traveling Salesman Problem
Computers & Operations Research 37: 1844–1852

[9] Bousonville T, Kopfer H (2000):
A Hybrid Evolutionary Algorithm for the Mixed Rural Postman Problem with Turn Penalties
Technical Report, Bremer Institut für Betriebstechnik und angewandte Arbeitswissenschaft, Universität Bremen

[10] Brandão J (2004):
A Tabu Search Algorithm for the Open Vehicle Routing Problem
European Journal of Operational Research 157: 552–564

[11] Cabral E, Gendreau M, Ghiani G, Laporte G (2004):
Solving the Hierarchical Chinese Postman Problem as a Rural Postman Problem
European Journal of Operational Research 155: 44–50

[12] Cacchiani V, Fernandes Muritiba A, Negreiros M, Toth P (2010):
A Multistart Heuristic for the Equality Generalized Traveling Salesman Problem
Networks 57: 231–239

[13] Caldwell T (1961):
On Finding Minimum Routes in a Network with Turn Penalties
Communications of the ACM 4: 107–108

[14] Clossey J, Laporte G, Soriano P (2001):
Solving Arc Routing Problems with Turn Penalties
Journal of the Operational Research Society 52: 433–439

[15] Corberán A, Martí R, Martínez E, Soler D (2002):
The Rural Postman Problem on Mixed Graphs with Turn Penalties
Computers & Operations Research 29: 887–903

[16] Corberán A, Plana I, Sanchis J (2007):
A Branch & Cut Algorithm for the Windy General Routing Problem and Special Cases
Networks 49: 245–257

[17] Corberán A, Prins C (2010):
Recent Results on Arc Routing Problems: An Annotated Bibliography
Networks 56: 50–69

[18] Drexl M (2007):
On Some Generalized Routing Problems
PhD Thesis, Faculty of Business and Economics, RWTH Aachen University
URL `http://darwin.bth.rwth-aachen.de/opus3/volltexte/2007/2091/pdf/Drexl_Michael.pdf`

[19] Dror M (2000):
Arc Routing: Complexity and Approximability
in:
Dror M (ed):
Arc Routing: Theory, Solutions, and Applications
Kluwer, Boston
133–169

[20] Dror M (ed) (2000):
Arc Routing: Theory, Solutions, and Applications
Kluwer, Boston

[21] Dror M, Langevin A (1997):
A Generalized Traveling Salesman Problem Approach to the Directed Clustered Rural Postman Problem
Transportation Science 31: 187–192

[22] Edmonds J, Johnson E (1973):
Matching, Euler Tours and the Chinese Postman
Mathematical Programming 5: 88–124

[23] Eglese R, Li L (1992):
Efficient Routeing for Winter Gritting
Journal of the Operational Research Society 43: 1031–1034

[24] Ghiani G, Improta G (2000):
An Efficient Transformation of the Generalized Vehicle Routing Problem
European Journal of Operational Research 122: 11–17

[25] Grünert T, Irnich S (2005):
Optimierung im Transport
Band II: Wege und Touren
Shaker, Aachen

[26] Guan M (1962):
Graphic Programming Using Odd or Even Points
Chinese Mathematics 1: 273–277

[27] Gutin G, Karapetyan D (2010):
A Memetic Algorithm for the Generalized Traveling Salesman Problem
Natural Computing 9: 47–60

[28] Irnich S (2005):
A Note on Postman Problems with Zigzag Service
INFOR 43: 33–39

[29] Irnich S (2008):
Solution of Real-World Postman Problems
European Journal of Operational Research 190: 52–67

[30] Johnson D, Papadimitriou C (1985):
Computational Complexity
in:
Lawler L, Lenstra J, Rinnooy Kan A, Shmoys D (eds):
The Traveling Salesman Problem
Wiley, Chichester
37–85

[31] Kirby R (1966):
A Minimum Path Algorithm for a Road Network with Turn Penalties
ARRB Proceedings 3: 434–442

[32] Kirby R, Potts R (1969):
The Minimum Route Problem for Networks with Turn Penalties and Prohibitions
Transportation Research 3: 397–408

[33] Lacomme P, Prins C, Ramdane-Chérif W (2004):
Competitive Memetic Algorithms for Arc Routing Problems
Annals of Operations Research 131: 159–185

[34] Laporte G (1997):
Modeling and Solving Several Classes of Arc Routing Problems as Traveling Salesman Problems
Computers & Operations Research 24: 1057–1061

[35] McBride R (1982):
Controlling Left- and U-Turns in the Routing of Refuse Collection Vehicles
Computers & Operations Research 9: 145–152

[36] Noon C, Bean J (1991):
A Lagrangian Based Approach for the Asymmetric Generalized Traveling Salesman Problem
Operations Research 39: 623–632

[37] Orloff C (1974):
A Fundamental Problem in Vehicle Routing
Networks 4: 35–64

[38] Roy S, Rousseau J (1989):
The Capacitated Canadian Postman Problem
INFOR 27: 58–73

[39] Snyder L, Daskin M (2006):
A Random-Key Genetic Algorithm for the Generalized Traveling Salesman Problem
European Journal of Operational Research 174: 38–53

[40] Soler D, Martínez E, Micó J (2008):
A Transformation for the Mixed General Routing Problem with Turn Penalties
Journal of the Operational Research Society 59: 540–547

[41] Stern H, Dror M (1979):
Routing Electric Meter Readers
Computers & Operations Research 6: 209–223

[42] Wattleworth J, Shuldiner P (1963):
Analytical Methods in Transportation: Left-Turn Penalties in Traffic Assignment Models
Journal of the Engineering Mechanics Division, Proceedings of the American Society of Civil Engineers 89: 97–126

[43] West D (1996):
Introduction to Graph Theory
Prentice-Hall, Upper Saddle River

[44] Win Z (1987):
Contributions to Routing Problems
PhD Thesis, Faculty of Mathematics and Natural Sciences, University of Augsburg

[45] Winter S (2002):
Modeling Costs of Turns in Route Planning
GeoInformatica 6: 345–361

[46] Wyskida R, Gupta J (1972):
IE's Improve City's Solid Waste Collection
Industrial Engineering 46: 12–15