A Solution for the General Routing Problem

Hassan Sherafat

Abstract—*In routing problems, the aim is to determine a least cost route for vehicles covering a specified set of locations, subject to some constraints. How the cost is calculated and minimized in these problems may include fuel consumption and other environmental criteria, such as pollution level. An example is determination of optimal routes for vehicles in solid waste collection in order to minimize the environmental impact caused by the vehicle itself. We address the problem in a broad level, known in literature as the General Routing Problem.*

Using a mathematical and computational model, in this paper we solve the problem of determining a least cost circuit which covers given subsets of arcs, edges and nodes of a mixed graph, subject to turn restrictions on nodes (restrictions that avoid bad turns of vehicles in real-life street networks). The well known problems, such as the Mixed Chinese Postman Problem and the Traveling Salesman Problem are particular cases of this general problem. Our solution is based on an efficient graph transformation that makes it possible to solve the resulting problem as a standard TSP. Computational results confirm the efficiency of the method in solving relatively large problems with good quality.

Keywords—Environment, Chinese postman problem, Turn penalties, General routing

Scope

Arc routing problems (ARP) consist of finding least cost circuits that cover links (or a subset of links) of a graph, satisfying some conditions. One of the most important contributions in solving ARP was made by Edmond and Johnson [6] who solved in polynomial time the classical Chinese Postman Problem. Thereafter, many particular instances of ARP have been formulated, most of them shown to be NP-complete, but with little progress in their solution. However, in recent years some new approaches that deal with general cases of ARP have appeared. A few of them are mentioned below.

Benavent et al. [1] have studied the Windy Rural Postman Problem – WRPP, which generalizes a great number of ARP formulations. They suggested an exact procedure, based on cutting planes, and some heuristics. An extensive computational experiment with graphs of up to 196 nodes and 316 edges showed the effectiveness of the exact procedure. It was able to solve to optimality 185 out of 288 instances. All instances were solved when a branch-and-bound option was invoked.

Hassan Sherafat

Department of Mathematics / Federal University of Sergipe
Brazil

WRPP was also studied by Benavent et al. [2], in which some heuristics inspired in heuristics of Benavent et al. [1] and a new scatter search algorithm are presented. The last algorithm was shown to have high potentialities. It was capable of solving the WRPP for relatively large problems, with deviation from the lower bound under 2%.

Another outstanding recent paper is due to Corberán et al [4]. In this article another general formulation of ARP is studied: the Mixed Rural Postman Problem, considering the existence of inconvenient or forbidden turns on nodes. Two heuristics are presented: the first one transforms ARP to an Asymmetric Travelling Salesman Problem. The second one is a constructive method based on Tabu Search metaheuristic. The second approach was shown to be more efficient, achieving average deviation of 1% from optimality for problems of up to 200 nodes and 480 links. However, the transformation method was adequate only for instances with few edges. As authors have noticed, the results obtained in instances with more than 30 edges were very poor. This limitation has been shown also by Laporte [10] in a similar transformation approach.

In this paper we present a new formulation for the ARP, which is more general than the problems listed above, and suggest an intuitive transformation method that makes it possible to solve a variety of arc and node routing problems subject to turn restrictions. Our approach is based on the same strategy line used by Laporte [10] and Corberán et al. [4]. However, we could successfully solve the problem for any graph configuration, including cases with a large number of edges. In fact, we could solve nearly to optimality mixed and undirected examples of up to 1000 edges.

I. Introduction

Let $G = (N, A, E)$ be a mixed graph in which N, A and E are the sets of nodes, arcs and edges, respectively, and let $N' \subseteq N$, $A' \subseteq A$ e $E' \subseteq E$ be the sets of required nodes, arcs and edges, i.e., the sets of links and nodes that need to be served. The problem is to construct a minimum cost circuit that contains all above required sets, subject to turn restrictions on nodes. We call this problem as *General Routing Problem* (*GRP*).

The GRP has its own applications: consider any vehicle routing problem in which many points to be served are distributed along the streets (one-way and two-way streets), while other points stand in isolated locations, and yet there are traffic regulations on crossings that should be respected. GRP is a practical and useful generalization for many routing problems. However, many other formulations may be derived from this general problem.

- If $N' \neq \varnothing$, $A' = E' = \varnothing$, the problem is reduced to the conventional TSP.

- $A = \varnothing$, $E' = E$, Undirected Chinese Postman Problem (UCPP);

- $A' = A$, $E = \varnothing$, Directed Chinese Postman Problem (DCPP);

- $A' = A$, $E' = E$, Mixed Chinese Postman Problem (MCPP);

- $A = \varnothing$, $E' \subset E$, Undirected Rural Postman Problem (URPP);

- $A' \subset A$, $E = \varnothing$, Directed Rural Postman Problem (DRPP);

- $A' \subset A$, $E' \subset E$, Mixed Rural Postman Problem (MRPP);

- $A' = A$, $E \neq \varnothing$, $E' = \varnothing$, Stacker Crane Problem (SCP).

Other formulations are possible; among them, we can mention the Windy Postman Problem (WPP), which has the same structure as UCPP, except that for each edge $e_{ij} \in E$ the traverse cost c_{ij} may be different from the traverse cost in the opposite direction, c_{ji}. For a detailed discussion about the ARP and its variations, see Eiselt et al. [7] and Eiselt et al. [8].

Among all cases listed above, only the UCPP and DCPP are well solved and have exact polynomial solutions [6]. All other cases are shown to be NP-complete problems, of which the TSP has received more attention from researchers in recent years and has gained some good, newly developed, approximate solutions, which make it possible to solve medium, even large-scale problems [18], [14].

In the following sections we introduce an intuitive transformation method that makes it possible to solve GRP, including any particular form of the CPP or RPP, as a standard TSP. Furthermore, we show how the node restrictions (prohibited left-turns and U-turns), can be handled by this method. Although such prohibitions are considered as strong constraints in the real-life problems and could be associated with any of the above formulations, they are usually neglected in most theoretical approaches for solving routing problems.

II. **Transformation**

Some authors have suggested transforming the Arc Routing Problem into a Node Routing Problem (NRP), which is then solved as a TSP. Pearn [13] devises a method to transform the undirected case of an ARP with m edges to an equivalent NRP with $3m+1$ nodes. He doesn't report any computational experience. Laporte [10] suggests a different and interesting approach for performing such a transformation, which transforms an ARP to an equivalent Generalized Traveling Salesman Problem, in the first step, and in the second, this problem to a standard TSP. As describedby the author, the method behaves well when the graph is directed or

has only a few undirected edges. It fails when there are too many edges. Corberán et al. [4] suggest a similar approach for MRPP in which the ARP is directly transformed to an Asymmetric Traveling Salesman Problem (ATSP). Computational results reported by the authors also show deficiencies in solving problems with relatively more edges. Dror et al. [5] employ the Generalized TSP approach to solve a special case of DRPP, named as the Directed Clustered Rural Postman Problem. They have succeeded in solving problems with up to 81 nodes.

In this section, we present a detailed description of the proposed graph transformation. Let $G = (N, A, E)$ be a strongly connected mixed graph, as defined in the previous section. Without loss of generality, we describe the construction of a transformed graph for the case of the MCPP. So, all arcs in A and all edges in E are required. Thus, in order to avoid complex notations, we will employ A and E as required sets. The set N is not required explicitly; however, any feasible solution for the MCPP will contain all nodes in N. The generalization of this procedure for GRP is discussed in a later section.

Let $n = |N|$, $r = |A|$ and $m = |E|$ be the cardinality of the above sets, and let D be a cost matrix associated to G. The procedure described below transforms G into a graph with $r + 2m$ nodes.

Step 1

Graph $G_1 = (N_1, A_1 \cup E_1)$ is constructed from G , as follows;

Initially let $N_1 = \varnothing$, $A_1 = \varnothing$, $E_1 = \varnothing$.

For any link $\ell = (i, j) \in A \cup E$ of graph G , create the following components in G_1:

- Two nodes $n_{i\ell}$, $n_{j\ell}$ in N_1;

- One arc $(n_{i\ell}, n_{j\ell})$ in A_1, if ℓ is an arc type link; or

- Two arcs $(n_{i\ell}, n_{j\ell})$, $(n_{j\ell}, n_{i\ell})$ in E_1 , if ℓ is an edge type link in the original graph G .

In any case, the cost associated to any arc $(n_{i\ell}, n_{j\ell}) \in A_1 \cup E_1$ will be same as the cost in the original link ℓ .

Now, G_1 is a directed graph with $r + m$ disconnected components, in which any node $i \in N$ has k copies in N_1, where k is the number of links incident to i in G . Let D_1 bethe cost matrix associated to G_1.

Step 2

Now, we proceed to construct $G_2 = (N_1, A_1 \cup E_1 \cup B_1)$ from G_1. Let $P_i = \{i_1, i_2, ..., i_k\}$ be the set of copies of node $i \in N$ in N_1, and initially let $B_1 = \varnothing$. For each node $i \in N$ scan the set P_i as follows: for every pair of nodes $i_s, i_t \in P_i$, create the arc (i_s, i_t) in B_1, if, and only if, i_s is a terminal node of any arc in $A_1 \cup E_1$ and i_t is an initial node of any arc in $A_1 \cup E_1$. Now G_2 is a strongly connected directed graph. We call the arcs in B_1 as *connecting arcs*, which can be interpreted as passages on the nodes of the original graph, from one link to another. Therefore, the penalties due to the bad turns may be applied at this moment, as appropriate traverse costs applied to the arcs in B_1. Implementation of node restrictions is discussed in a following section. If there are no penalties, such costs will be zero. Let D_2 be the actual cost matrix, associated with G_2.

Step 3

We form the complete graph $G_3 = (N_1, A_1 \cup E_1 \cup S_1)$, where S_1 represents the set of shortest paths between all pairs of nodes $i, j \in N_1$, in G_2, so that the arc $(i, j) \notin A_1 \cup E_1$. Note that all connecting arcs in B_1 are now included in S_1. However, their costs may be reduced by means of the shortest distances. Consider D_3 as such a full cost matrix.

Step 4

We now deduce the final transformed graph $G_4 = (N_2, E_1 \cup S_2)$, as described below: eliminate all arcs in G_3 that belong to A_1, unifying their initial and final nodes. This should be done as follows: for each arc $(i, j) \in A_1$, exclude the node j, exclude all arcs $(v, j) \in S_1$, exclude all arcs $(i, v) \in S_1$, and exclude the arc (i, j) itself. All arcs $(j, v) \in S_1$ come to be (i, v), with no alteration in their costs. These modifications reduce the set of nodes N_1 to N_2, and the set of shortest paths S_1 to S_2. Note that each original arc in G is now represented by a single node in G_4. Furthermore, the cost of any arc in E_1 is changed to $-M$, where M is a big positive number.

Figure 1 shows an illustrative example, in which the transformation is done on a graph G, for solving a MCPP. Graph G_2 represents the transformed graph after step 2, and G_4 is the complete final transformed graph.

Mixed Graph G Transformed Graph G_2
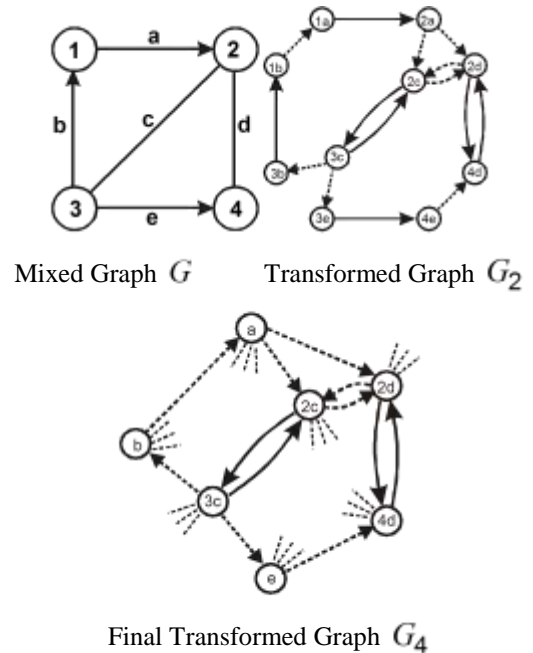
Final Transformed Graph G_4

Figure 1.   An example for the graph transformation

III.   The TSP Solution

Whenever the transformation is done, the next step is to solve a Traveling Salesman Problem on the transformed graph. First we show the correspondence between a postman tour in G and a Hamiltonian Circuit in G_4.

As we saw, the graph $G = (N, A, E)$ is transformed to the $G_4 = (N_2, E_1 \cup S_2)$. Every edge in G is represented by a pair of opposite arcs in G_4. E_1 is the set of such pairs. Moreover, every arc in G is represented by a single node in G_4. Let $N_a \subseteq N_2$, be the set of such nodes, and $N_e \subseteq N_2$ be the set of terminal nodes of all arcs in E_1. Obviously, $N_a \cap N_e = \varnothing$ and $N_a \cup N_e = N_2$.

Consider now the circuit C_C as a feasible postman tour in G; which covers all its arcs and edges (and possibly containing some extra copies of them). If we trace the tour on the transformed graph G_4, we will obtain a circuit C_H, which contains at least one arc of every pair of arcs in E_1, and all nodes in N_a. As C_H contains one arc of each pair of arcs in E_1, then it contains all nodes in N_e. Thus, C_H contains all nodes in $N_a \cup N_e = N_2$; therefore it is a Hamiltonian Circuit in G_4. It is important to note that the extra copies of

links in an optimal postman tour in G form shortest paths between some pairs of nodes. In G_4, such paths are represented by traverses through arcs in S_2, without any additional passage on nodes in N_2.

Reciprocally, we show that a hamiltonian circuit in G_4 corresponds to a postman tour in G. Any optimum, or near optimum TSP solution in G_4 provides a circuit that fulfills the following characteristics: contains one arc from any pair of arcs in E_1, and all nodes in N_a. In fact, as G_4 is a complete graph, there exists a hamiltonian circuit that contains all nodes of N_2 exactly once (including N_a). And if such a circuit is an optimum, or near optimum TSP solution, it contains one arc of each pair of arcs in E_1, just because their costs are big negative numbers. It means that the corresponding circuit in G should contain all edges in E and all arcs in A.

Therefore, there exists a one-to-one relation between a TSP solution in G_4 and a postman tour on the mixed graph G.

A discussion about the solution techniques for the TSP is outside the scope of this paper. Any method to solve the asymmetric case of TSP may be employed. However, our approach is motivated by the recent improvements in this field. Johnson et al. [9] and Voudouris et al. [18] have suggested approximate solutions that have been applied successfully to large scale Traveling Salesman Problems. In our computational tests, we adopted the Guided Local Search metaheuristic, developed by the latter author. Rodrigues [14] presents computational tests for this method, using medium and large sized test graphs, up to 14.000 nodes, with known optimal solutions. Near optimum solutions have been obtained in a relatively short time.

IV. Turn Restrictions

In contrast to the present approach, and that employed by Corberán et al. [4], most other techniques solve the arc routing problems in two distinct stages (see [7]): the first is to determine a minimum cost augmentation of the graph; and the second, to construct a tour which covers the required components of the graph. With turn restrictions, the second stage becomes a relevant problem. The augmented graph obtained in the first stage may require new augmentation in order to comply with the turn prohibitions, as we can see in the following case.

Suppose that in the mixed graph shown in figure 2(a), the U-turns in nodes 1 and 2 are prohibited. Figure 2(b) represents an optimal augmentation for the MCPP, obtained by any method. However, there is no feasible postman tour in this graph, initiating for example at node 1; any attempt results in a U-turn, for example, the sequence (c, f, g, e, b, d, d', a). In fact, any feasible solution requires additional augmentation, for example, that shown in figure 2(c). In this latest, the tour may be: (c, f, g, e, b, d, g', e', b', a).
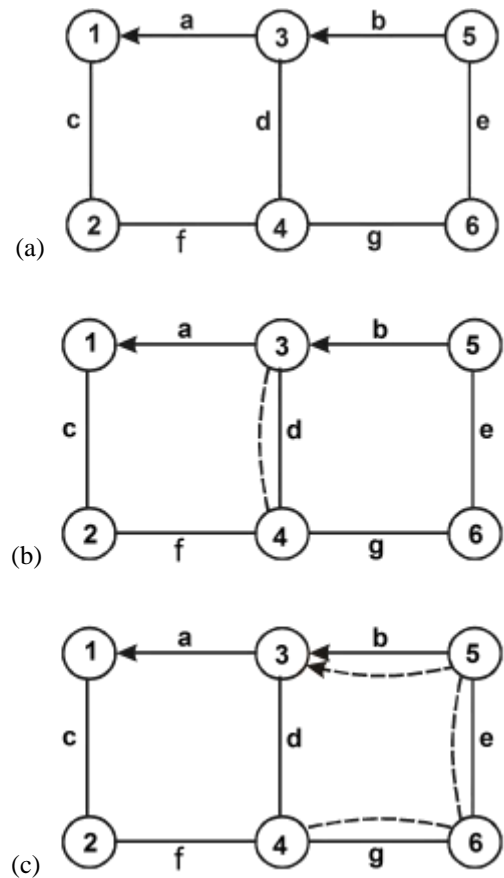

Figure 2. Ilustrating example for implication of node restrictions

In our approach, the cost matrix D_4 may be structured, not only by link costs, but also by node traverse penalties. Therefore, the TSP solution provides a complete feasible solution to the arc routing problem, which minimizes the whole cost: link distances and bad-turn penalties. This is one of the main advantages of the method presented here.

Now, we consider the implementation of turn restrictions, using a typical situation as shown in the figure 3(a). The example may be viewed as a part of a bigger graph G. Suppose that in this graph the U-turns and left-turns described by the sequences $(c,1,b)$, $(a,1,a)$ and $(c,1,c)$ should be avoided. Figure 3(b) represents the transformed graph G_2, obtained at the end of second step of transformation procedure. Considering that the connecting arcs (such as $(1c,1b)$) represent the passage on nodes from one link to another, a natural way for managing bad turns is to apply appropriate costs to such connecting arcs. In G_2, however, there are noconnecting arcs referring to U-turns, nor the left turn may be prohibited, suppressing (or penalizing) only the connecting arc $(1c,1b)$. It is important to note that such an action does not inhibit the possibility of a "round connection", such as $(c,1c,1a,1b,b)$, which may be encountered in the third step of the algorithm, when the shortest paths are

calculated. Such a roundconnection is an infeasible solution that may substitute the left turn.

To solve this problem, we apply a new expansion on G_2 as follows: for all pairs of arcs in E_1, we split each terminal node into two nodes, so that every arc of the pair has its distinct terminal nodes. Figure 3(c) shows the expanded graph G_2'. Now, it is possible to put appropriate penalties for all connecting arcs, including $(1a,1a')$, $(1c,1c')$ and $(1c,1b)$ which represent the prohibited U-turns and left turn. In the third step of the above algorithm, the shortest paths procedure should be applied to the expanded graph G_2'. Now, the shortest paths calculated from nodes $1a$, $1c$ and $1d$ to all other nodes will find feasible turns for each case, which may be formed by a longer path than a simple connecting arc. In the forth step, the expanded graph will be reduced to a graph with identical dimensions as G_4 in the previous algorithm. These operations are detailed in the following section for the case of General Routing Problem.
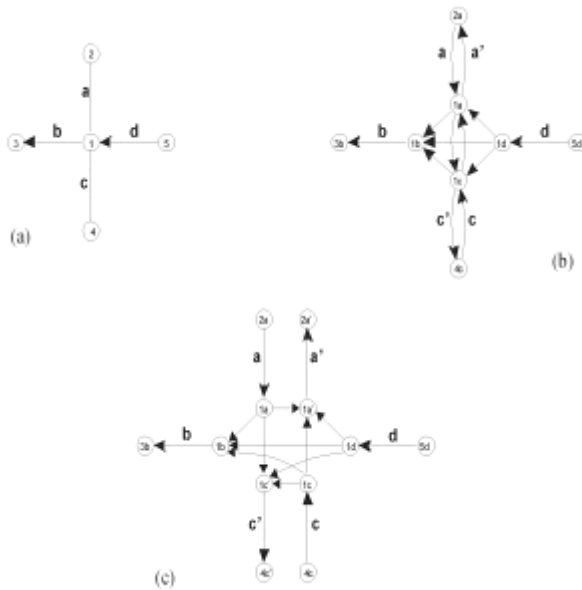

Figure 3.   Additional transformation to handle turn restrictions

V.   **The General Routing Problem**

In this section, the transformation procedure, previously described for MCPP, is generalized for GRP.

Given a strongly connected mixed graph $G = (N, A, E)$, with N', A' and E' as required sets, as defined earlier. Let $N_T \subseteq N$ be the set of nodes that appear as terminal node of any link in $A' \cup E'$. This implies that all nodes in N_T are implicitly required. Let $N_X \subseteq N'$ be the set of all required nodes, which are not covered by any link in $A' \cup E'$.

Therefore we have a larger set of required nodes $N_R = N_X \cup N_T$.

The proposed procedure initially transforms the graph G with n nodes to another with $2\,r + 4m$ nodes, where $n = |N|$, $r = |A|$ and $m = |E|$. However, the final transformed graph will be reduced to one with $n_x + r' + 2m'$ nodes, where $n_x = |N_X|$, $r' = |A'|$ e $m' = |E'|$. The general transformation procedure is summarized as follows:

Step 1

First, construct the graph $G_1 = (N_1, A_1 \cup E_1)$ from G. Let initially $N_1 = \varnothing$, $A_1 = \varnothing$, and $E_1 = \varnothing$.

For every arc $a = (i, j) \in A$, create the following components in G_1:

- two nodes n_{ia}, n_{ja} in N_1;

- one arc (n_{ia}, n_{ja}) in A_1.

For every edge $e = (i, j) \in E$, create the following components in G_1:

- four nodes n_{ieu}, n_{iev}, n_{jeu}, e n_{jev} in N_1;

- two arcs (n_{ieu}, n_{jeu}), (n_{jev}, n_{iev}) in E_1.

The cost associates with any created arc in $A_1 \cup E_1$ will be the same original link cost in $A \cup E$. Let D_1 be the cost matrix associated with G_1. Now, G_1 is a disconnected and directed graph with $r + 2m$ components.

Note that G_1 is constructed not only by required links, but also by non-required links. The reason for transforming the entire graph is to permit application of turn penalties to all nodes, including those which are not required, since they may appear in the final route.

Let $P_i = \{i_1, i_2, ..., i_k\}$ be the set of copies of node $i \in N$ in N_1, as created in step 1. We call $P_i \subset N_1$ an *image* of node $i \in N$ on N_1. Then, for every set P_i, $i \in N_X$, choose a unique node for representing the respective required node i in the transformed graph G_1. Let $N_{1X} \subset N_1$ be the set of nodes chosen in this manner. All nodes in N_{1X} will remain in the final transformed graph for appearing in the TSP tour. The question that which node in every set P_i should be chosen, in real life logistic problems may be responded taking in account the exact location of the point to be served. For example, in figure 3 (c) node 1 is represented by six copies. However, each

copy has a distinct position in the original network. For instance, visiting node 1 at $1c$ position is different than visiting it at $1c'$ position.

More precisely, the set of nodes N_1 may be classified in four subsets:

$N_{1A} \subseteq N_1$ is the set of nodes associated with required arcs A';

$N_{1E} \subseteq N_1$ is the set of nodes associated with required edges E';

$N_{1X} \subseteq N_1$ is the set of nodes associated with required nodes $N_X \subseteq N'$, and

$N_{1D} \subseteq N_1$ is the set of nodes which are not in any of sets N_{1A}, N_{1E}, e N_{1X}.

Denote $N_{1R} = N_{1A} \cup N_{1E} \cup N_{1X}$ as the image of implicit or explicitly required nodes in G_1, and let A_{1R} and E_{1R} be the sets of arcs in G_1, associated respectively to the required sets A' and E'.

Step 2

Construct the graph $G_2 = (N_1, A_1 \cup E_1 \cup B_1)$ from G_1, as described below:

Let B_1 be the set of connecting arcs, initially $B_1 = \varnothing$. For each node $i \in N$ scan the set $P_i = \{i_1, i_2, ..., i_k\}$ (image of node i in N_1) as follows: For each pair of nodes $i_s, i_t \in P_i$, create an arc (i_s, i_t) in B_1, if, and only if, i_s is a final node of any arc in $A_1 \cup E_1$ and i_t is an initial node of any arc in $A_1 \cup E_1$.

Now, G_2 is a directed and strongly connected graph. The set B_1 is formed by connecting arcs which represent passages on nodes, from one link to another. These arcs may be penalized with appropriate costs, according to inconvenience of each turn. Among them, there are connecting arcs which represent U-turns. Let $U_{1R} \subseteq B_1$ be the subset of connecting arcs which refer to U-turns on required nodes N_{1R}. D_2 is the actual cost matrix, associated to G_2.

Step 3

Form the complete graph $G_3 = (N_{1R}, A_{1R} \cup E_{1R} \cup S_1)$, where N_{1R} represents the image of required nodes in G_2, and $S_1 \supset B_1$ represents set of shortest paths between all pairs of nodes $i, j \in N_{1R}$, calculated in G_2, so that $(i, j) \notin A_{1R} \cup E_{1R}$.

The connecting arcs in B_1 are now included in S_1. However, their costs may have been reduced by means of shortest distances. Consider D_3 as distance matrix associated with the complete graph G_3. It is structured, not only with arc costs, but also with all bad turn penalties attributed to nodes.

Step 4

Construct the final transformed graph $G_4 = (N_{2R}, E_{2R} \cup S_2)$, by removing all arcs in A_{1R} and all connecting arcs in U_{1R} from G_3. This is done by unifying the initial and final nodes, as described below:

For each arc $(i, j) \in A_{1R}$, identify all arcs $(i, v) \in S_1$ attributing to them the cost of the correspondent arc $(j, v) \in S_1$. Then eliminate node j and all arcs incident to it.

For each arc $(i, j) \in U_{1R}$, identify all arcs $(v, i) \in S_1$ attributing them the cost of the correspondent arc $(v, j) \in S_1$. Then eliminate node j and all arcs incident to it. The pairs of arcs $(i, j), (k, l) \in E_{1R}$, which represent required edges in graph G, will continue to exist, but their initial nodes should have been changed, becoming $(l, j), (j, l) \in E_{2R}$.

These modifications reduce the node set N_{1R} to N_{2R}, pairs of arcs E_{1R} to E_{2R} and the shortest path set S_1 to S_2. Note that any required arc in A' and any required node in N_X are now represented by a single node, and every required edge in E' by two nodes mutually connected by a pair of opposite arcs in G_4.

In order to complete the transformation, change the cost of all arcs in E_{2R} to $-M$, where M is a big positive number.

The algorithm described above, transforms the General Routing Problem, defined on the mixed graph G, to a pure node routing problem, defined on the directed graph G_4. The correspondence between these two problems may be verified in a way similar to thatshown for the MCPP version of the algorithm, considering that each required node in the original graph is represented by a single node in the transformed graph, each required arc also by a single node, and each required edge by two nodes interconnected by a pair of arcs with highly negative costs.

The computational complexity of the algorithm is bounded by $O\left(\left(2|A| + 4|E|\right)^3\right)$, due to the shortest paths calculations in its 3rd Step. Therefore, the transformation phase of solution method is an efficient procedure. However, the global

52

efficiency depends on method to be used for the second phase (TSP solution).

VI. Computational Results

The procedure described above was implemented on a personal computer equipped with Pentium IV 2.0 GHz Processor. In order to compare computational results with other recent expressive works, we oriented the tests towards solving the Mixed Chinese Postman Problem and the Mixed Rural Postman Problem with Turn Penalties - MRPPTP.

For MCPP the tests were done on randomly generated pseudo-manhattan graphs. These were constructed on a grid of $n =$ p$\times$ q nodes, in which any node may be connected to no more than eight other nodes, except the frontier and the corner nodes, which may be connected at most to five and three nodes, respectively. A predefined number of arcs and edges are randomly distributed among the nodes, assuring the connectivity condition for the graph. The graph is of pseudo-manhattan type, therefore crossing diagonal links are allowed. For each value of n (node number), five graphs with increasing percentage of directed links were generated: with 0%, 25%, 50%, 75% and 100% of directed links. So, not only the mixed case was handled, but also the two extreme cases, totally directed and totally undirected cases were included in the tests. Such graphs were passed through the transformation steps and then solved by a TSP code based on a Guided Local Search approach [14].

Table 1 shows detailed information about these computational tests. In each case a lower bound on the MCPP circuit distance is provided, which is calculated by solving the CPP over the undirected network, derived from the original one by ignoring all arc directions. Also, for each tested graph a good known solution is reported, obtained from a special routine included in the random graph generator, which permits constructing the network together with a good feasible solution for the MCPP. This solution stands very close to the optimal solution.

As we can see, the solutions obtained have a maximum deviation of 7.9%, in relation to the corresponding lower bounds, considering all mixed cases. It is worth noting that for the totally undirected problems, the lower bound is also the optimal solution; however, for totally directed cases, the lower bound may be much lower than the optimal MCPP solution.

Our solutions outperform or are equal to the good known solutions, in almost all cases. In only three cases did we observe deviations, all of which were under 0.9%.

Since the Guided Local Search used for the second phase of the proposed method is an iterative procedure it is possible to obtain less approximate solutions in shorter times. For example, solving the graph Mix500A25 (with 500 nodes, 250 arcs and 750 edges), at the execution time $t_2 = 60$ seconds, we could register a solution that stands 7.2% above the Lower Bound.

The time consumed by transformation procedure is significantly smaller than the time used by the TSP procedure. For example, for graphs with 500 nodes and 1000 links, the average transformation time is 21.5 seconds, while the average time used by TSP routine is 161.8 seconds.

TABLE I.   COMPUTATIONAL TESTS FOR MCPP

Graph Name	Node Number	Link Number	% Arcs	Best Known Solution	Obtained Solution
Mix100A00	100	200	0	21410	21410
Mix100A25	100	200	25	21537	21537
Mix100A50	100	200	50	22138	22103
Mix100A75	100	200	75	22203	22174
Mix100A100	100	200	100	24205	24205
Mix200A00	200	400	0	42976	42976
Mix200A25	200	399	25	44144	43945
Mix200A50	200	400	50	44405	44261
Mix200A75	200	400	75	46191	46097
Mix200A100	200	400	100	47193	47193
Mix300A00	300	600	0	64612	64612
Mix300A25	300	599	25	65006	64734
Mix300A50	300	600	50	66517	66502
Mix300A75	300	600	75	68190	68080
Mix300A100	300	600	100	72673	72673
Mix400A00	400	800	0	86208	86208
Mix400A25	400	798	25	86214	86394
Mix400A50	400	800	50	89079	89014
Mix400A75	400	800	75	91592	91494
Mix400A100	400	799	100	96517	96326
Mix500A00	500	999	0	107470	108435
Mix500A25	500	1000	25	108370	108649
Mix500A50	500	1000	50	110640	110598
Mix500A75	500	1000	75	114268	114268
Mix500A100	500	1000	100	122080	122080

The largest MCPP that we attempted to solve was one with 1000 nodes and 2000 links, of which 50% were directed. We obtained a 9.7% above lower bound solution, in a total time of $t_3 = 473$ seconds.

For solving the Mixed Rural Postman Problem with Turn Penalties the tests were done on the same set of test graphs used by Corberán et al. [4]. In this paper, the authors present two heuristic procedures for MRPPTP, named as Heuristic 1 and Heuristic 2. They provided computational tests with randomly generated graphs, and compared them with exact solutions, whenever available.

The above test problems consist of 216 mixed, randomly generated, general graphs, with $|N| \in [40, 200]$, $|A| \in [90, 440]$, and $|E| \in [10, 40]$. In each instance, all edges are considered as required, but $|A_R|$ arcs are randomly

selected to be required. These test problems were classified in 24 sets, according to the number of required links. Finally, were assigned the following turn penalties: 0 for going straight ahead, 1 for turning right, 3 for turning left, and all U-turn were considered forbidden.

TABLE II. COMPUTATIONAL TESTS FOR MIXED RURAL POSTMAN PROBLEM WITH TURN PENALTIES

Set (1)	$\|A_R\|$ (2)	$\|E_R\|$ (3)	$\|N^T\|$ (4)	C/C_1 (9)	C/C_2 (10)
D11	[36, 98)	10	[56, 118)	0,9517	0,9844
D12		20	[76, 138)	0,9252	0,9830
D13		30	[96, 158)	0,8945	0,9861
D14		40	[116, 178)	0,8847	0,9843
D21	[98, 140)	10	[118, 160)	0,9582	0,9840
D22		20	[138, 180)	0,9250	0,9826
D23		30	[158, 200)	0,9021	0,9840
D24		40	[178, 220)	0,8918	0,9887
D31	[140, 182)	10	[160, 202)	0,9670	0,9831
D32		20	[180, 222)	0,9403	0,9771
D33		30	[200, 242)	0,9187	0,9829
D34		40	[220, 262)	0,9062	0,9845
D41	[182, 260)	10	[202, 280)	0,9724	0,9966
D42		20	[222, 300)	0,9491	0,9953
D43		30	[242, 320)	0,9377	0,9939
D44		40	[262, 340)	0,9110	0,9942
D51	[260, 330)	10	[280, 350)	0,9752	0,9962
D52		20	[300, 370)	0,9546	0,9961
D53		30	[320, 390)	0,9390	0,9979
D54		40	[340, 410)	0,9306	0,9954
D61	[330, 440]	10	[350, 460]	0,9778	0,9994
D62		20	[370, 480]	0,9490	0,9987
D63		30	[390, 500]	0,9497	0,9965
D64		40	[410, 520]	0,9355	0,9980
Average				**0,9336**	**0,9894**

Ser: set of randomly generated test graphs

$\|A_R\|$: range of required arcs of the set;

$\|E_R\|$ number of required edges of the set;

$\|N^T\|$: range of node number of transformed graph (dimension of final distance matrix) of the set;

C/C_1: average ratio between the calculated circuit costs and solutions obtained by Heuristic 1 of Corberán et al. [4];

C/C_2: average ratio between the calculated circuit costs and solutions obtained by Heuristic 2 of Corberán et al. [4];

Table 2 shows statistical results of computational tests using the proposed method, compared with two Heuristics of Corberán et al. [4], and the exact solution, whenever existing. Average deviation between obtained solutions and optimal solution is 0,15%. Among all 216 instances, the worst deviation from a known optimal solution stands on 0,99%. If compared with heuristics of Corberán et al. [4], the proposed method in average outperforms Heuristic 1 and Heuristic 2 by 6,64% and 1,06%, respectively. However, the computational time for our method is significantly greater than time consumed by either heuristics.

VII. Conclusions

Transportation and logistics have a considerable impact on environment, but optimization tools may minimize its effects.In this paper, we defined and presented a solution method for the General Routing Problem, which consists of determining a least cost circuit that covers given subsets of arcs, edges and nodes of a mixed graph, subject to turn restrictions on nodes. The Mixed Chinese Postman Problem, the Rural Postman Problem and their variations are particular cases that can be handled by this method. Our solution is based on an efficient graph transformation that makes it possible to solve the resulting problem as a standard TSP.

Computational results for MCPP and MRPPTP confirm the efficiency of the method for solving large number of random instances. For MCPP case, near optimum problems of up to 500 nodes and 1000 links were solved. For MRPPTP case, 216 test problems used by Corberán et al. [4] were solved, for all of which the average deviation from the optimal solution stood around 0,15%, with a worst deviation of 0,99%. The proposed GRP method outperforms in solution quality both heuristics presented by latest authors.

The proposed method solves one of the most general formulations of the routing problems, without loss of quality and performance when applied to particular cases. Moreover, it admits node restrictions (prohibited left-turns and U-turns), which are considered as strong constraints in the real-life problems and may be associated with any routing formulation, although neglected in most theoretical approaches.

References

[1] Benavent E., Corberán A., Piñana E., Plana I., e Sanchis J.M. New Heuristic Algorithms for the Windy Rural Postman Problem, Technical Report, Universidad de Valencia, España, May 2003.

[2] Corberán A. , Marti R. , Romero A., Heuristics for the Mixed Rural Postman Problem, Computers & Operations Research, Vol.27, No. 2, pag. 183-203, 2000.

[3] Corberán A., Martí R., Martinez E., e Soler D., The Rural Postman Problem on Mixed Graphs with Turn Penalties, Computers & Operations Research Vol.29, pag. 887 - 903, 2002.

[4] Dror M., Langevin A., A Generalized Traveling Salesman Problem Approach to the Directed Clustered Rural Postman Problem, Transportation Science, Vol.31, No. 2 pag. 187-192, 1997.

[5] Edmonds J., Johnson E. L., Matching, Euler Tours and the Chinese Postman Problem, Math. Program. 5, Pag. 88-124, 1973.

[6] Eiselt H. A., Gendreau M., Laporte G., Arc Routing Problems, Part I: The Chinese Postman Problem, Operations Research, Vol. 43, No. 2, pag. 231-242, 1995.

[7] Eiselt H. A., Gendreau M., Laporte G., Arc Routing Problems, Part II: The Rural Postman Problem, Operations Research, Vol. 43, No. 3, pag. 339-414, 1995

[8] Johnson D. S., McGeoch L. A., The Traveling Salesman Problem: a Case Study, Aarts, E. & Lenstra, J. K. (eds). Local Search in Combinatorial Optimization, Wiley, New York, pag: 215-310, 1997.

[9] Laporte G., Modeling and Solving Several Classes of Arc Routing Problems as Traveling Salesman Problems, Computers & Operations Research, Vol.24, No. 11, 1997.

[10] McBride R., Controlling Left and U-Turns in the Routing of Refuse Collection Vehicles, Computers & Ops Res. Vol.9, No. 2, 1982.

[11] Papadimiriou C. H., On the Complexity of the Edge Traversing, J. Assoc. Comput. Mach. 23, pag: 544-554, 1976.

[12] Pearn W. L., Transforming Arc Routing Into Node Routing Problems, Computers & Ops. Res. Vol.14, No. 4, 1987.

[13] Rodrigues M. A. P., Problema do Caixeiro Viajante, Um Algoritmo para Resolução de Problemas de Grande Porte Baseado em Busca Local Dirigida – Master Dissertation, PPGEP, Universidade Federal de Santa Catarina, Brasil, 2000.

[14] Sherafat H., Algoritmos Heurísticos de Cobertura de Arcos, Doctorate Thesis, PPGEP, Universidade Federal de Santa Catarina, Brasil, 2004.

[15] Sherafat H., Circuitos Eulerianos Sujeitos a Restrições nos Vértices, Anais de XXVIII SBPO, Vol.2, pag. 575-580, 1996.

[16] Sherafat H., Uma Solução para o Problema do Carteiro Chinês no Grafo Misto, Anais de XXI SBPO, Vol.1, pag. 157-170, 1988.

[17] Voudouris C., Tsang E., Guided Local Search and its Application to the Travelling Salesman Problem, European Journal of Operations Research 113, pag. 469-499, 1999.

About Author (s):

Transportation and logistics have a considerable impact on environment, but optimization tools may minimize its effects.In this paper, we defined and presented a solution method for the General Routing Problem, which consists of determining a least cost circuit that covers given subsets of arcs, edges and nodes of a mixed graph, subject to turn restrictions on nodes.