

An integer programming approach to a real-world recyclable waste collection problem in Argentina

Gustavo Braier

papyro.com, Argentina
braier@papyro.com

Guillermo Durán

CONICET, Argentina
Instituto de Cálculo, FCEyN, Universidad de Buenos Aires, Argentina
Departamento de Matemática, FCEyN, Universidad de Buenos Aires, Argentina
Departamento de Ingeniería Industrial, FCFM, Universidad de Chile, Chile
gduran@dm.uba.ar

Javier Marengo

Departamento de Computación, FCEyN, Universidad de Buenos Aires, Argentina
jmarengo@dc.uba.ar

Francisco Wesner

Departamento de Computación, FCEyN, Universidad de Buenos Aires, Argentina
fwesner@dc.uba.ar

July 26, 2016

Abstract

This article reports on the use of mathematical programming techniques to optimize the routes of a recyclable waste collection system servicing Morón, a large municipality outside Buenos Aires, Argentina. The truck routing problem posed by the system is a particular case of the generalized directed open rural postman problem. An integer programming model is developed with a solving procedure built around a subtour-merging algorithm and the addition of subtour elimination constraints. The route solutions generated by the proposed methodology perform significantly better than the previously-used manually-designed routes, the main improvement being that coverage of blocks within the municipality with the model solutions is 100% by construction whereas with the manual routes as much as 16% of the blocks went unserved. The model-generated routes were

adopted by the municipality in 2014 and the national government is planning to introduce the methodology elsewhere in the country.

Keywords: rural postman problem, recyclable waste collection, vehicle routing.

1 Introduction

This study presents an integer programming approach for improving the vehicle routing of the recyclable waste collection system in the municipality of Morón, a suburb outside of the City of Buenos Aires, Argentina. Home to 320,000 inhabitants (according to the 2010 Census [13]) and covering an area of 55.6 km², the municipality is divided in five administrative districts (Castelar, Morón City, Haedo, Palomar and Villa Sarmiento). For purposes of the recyclable waste collection system every district is subdivided into as many as seven sectors, each serviced by a single recycling truck. All collections within each of the five districts are made on a single different weekday for each one, which is called the “green day” for that district. This is part of a successful initiative carried out by the municipality, which involved a strong campaign in order to get the citizens involved (see Figure 1 and reference [16]).

Prior to this study, the collection routes were improvised by the drivers themselves with no tools other than basic unannotated sector maps. The inevitable result was that in some sectors a considerable number of street segments were often left unserved. The root cause of the failure to cover certain segments was the sheer difficulty of improvising a route that would pass through all the street segments, but other factors also contributed to the problem. The drivers set out in their trucks from the municipal depot at 2pm, made their assigned rounds and dumped the collected waste at the recycling centre, finishing up 5 hours later at about 7pm. Since the same vehicles were then used for the regular evening garbage collection, this schedule had to be strictly complied with. This meant that a driver realizing some segments were skipped would often not have time to return to them. Furthermore, since the trucks do not have automatic tracking devices, just determining which segments had been skipped was not an easy task. It was the growing complaints from local residents about this uncollected waste that prompted the municipal authorities to launch the initiative leading to the implementation of the methodology proposed here.

In this vehicle routing problem for recyclable waste collection, every street must be visited by a truck. The task is further complicated by two constraints needed to incorporate conditions peculiar to this case: (a) some segments of two-way streets are narrow enough for both sides to be serviced on a single trip in either direction, meaning no second trip is necessary; and (b) certain vehicle traffic restrictions from which trucks are not exempt. More specifically, these restrictions are the Argentinean road traffic regulations prohibiting left turns at signal-controlled intersections as well as U-turns, and will be the main concern here in modelling a solvable problem for the waste collection system.

The resulting combinatorial optimization problem is a particular case of the *generalized directed open rural postman problem*, which is NP-hard. Given a directed graph $G = (V, A)$, a set of mandatory arcs $A_M \subseteq A$, and a set of groups of nodes/arcs, the generalized directed

CADA LOCALIDAD TIENE SU DÍA VERDE

TU DÍA VERDE

Lunes Villa Sarmiento
Martes Haedo
Miércoles El Palomar
Jueves Morón
Viernes Castelar

QUÉ DEBEMOS SACAR EL DÍA VERDE DE 7 A 12 HS.

SI

Papeles / plásticos / vidrios / metales (latas, botellas de plástico y vidrio, diarios, revistas, cartón telas, zapatillas, llaves, tetra, juguetes, goma, bolsas de plástico o cartón, bandejas de fiambre o carnes limpias, etc).

NO

Restos de comida animal o vegetal, pañales descartables, yerba o cualquier residuo húmedo.

Prepará en tu casa una bolsa, caja o canasto para separar los residuos y sacarlos en TU DÍA VERDE.

Más información

Dirección de Políticas Ambientales

Mitre 957 1º piso. Tel.: 4483-4174/2178

politicas.ambientales@moron.gov.ar

Figure 1: One of the “green day” initiative advertisements in the printed press.

rural postman problem consists in finding a minimum-cost cycle traversing all the mandatory arcs and at least one element from each group. This NP-hard problem is a generalization of the well-known Chinese postman problem, which asks to identify a minimum-cost cycle traversing each arc at least once. The generalized directed *open* rural postman problem is to find a path with the same properties, i.e., one that traverses at least one element from each pre-specified group of nodes/arcs.

When there are no node/arc groups the problem is called the *directed rural postman problem*, which has received considerable attention in the combinatorial optimization community as regards both directed graphs [5, 11, 17, 18] and mixed graphs [6–9]. Recent works on the generalized directed rural postman problem include both integer programming and heuristic procedures [2, 10]. In this study we propose a simpler approach that is particularly amenable to the layouts of Latin American cities, and is quite straightforward to implement.

Our proposed approach uses an algorithm based on an integer programming formulation of the problem to find good-quality routes for the recycling trucks. The main theoretical contribution of this work is the introduction of a subtour-merging procedure that reduces computation times within an integer programming-based algorithm. Although this approach cannot be applied to general instances of the generalized rural postman problem since we assume zero-cost auxiliary arcs and groups formed by pairs of arcs only, it proved useful for the particular case considered here.

A valuable survey of vehicle routing problems in waste collection may be found in [14].

Applications of mathematical programming techniques to real-world waste collection systems appear in the literature for various cities around the world. Some examples are Chicago, USA [12], Kaoshiung, Taiwan [4], Lisbon, Portugal [15], Hamilton, Canada [19], Santiago, Chile [1], and Buenos Aires, Argentina [3]. Is it worth noting that the optimization problem in [3] has a different structure from that of the municipality of Morón, since the former has containers located in the collection areas.

The remainder of this article is organized in four sections. Section 2 describes the recycling truck routing problem to be solved and Section 3 formally models it as a generalized directed open rural postman problem in a suitable graph. Section 4 develops our proposed integer programming approach to solve the problem. Section 5 sets out and analyzes the results of the experiments performed using the proposed approach. Finally, Section 6 presents our conclusions.

2 Description of the problem

The Morón recyclable waste collection system divides the municipality into various sectors, each one serviced by a single truck that follows a single route and is assumed to have sufficient capacity to collect all waste in one trip. Each sector is represented by a mixed graph H whose nodes represent the sector's street corners. For each node, whether or not it has a traffic signal and the geographic coordinates of its corresponding corner are known. Two nodes bound the opposite ends of a *street segment*, defined for present purposes as a length of street running between two consecutive intersections and bordered on each side by adjacent city blocks. One-way traffic segments are represented in the graph by arcs while relatively narrow two-way traffic segments serviceable on both sides by a single truck trip are represented by the edges of the graph. In the case of broad two-way streets whose sides must be serviced separately, the segments are represented by two arcs, one in each direction. Each arc and edge is associated with a distance calculated as a straight line between two corners.

To incorporate the two traffic regulation constraints we construct a directed graph G from H . In the case of the U-turn prohibition, the graph is expanded by dividing each node into various new nodes representing all of the possibilities for arriving and departing at the corresponding corner. Auxiliary arcs are then added that connect up these nodes in such a way as to represent the legally permitted transitions from one corner to another.

A simple example of a node expansion is illustrated in Figure 2. In panel (a) of the figure the node is shown before expansion in the original graph H and represents a crossing between a one-way street (the horizontal arcs) and a two-way street (the vertical edges). In panel (b) the node has been expanded into six new nodes representing each possible arrival and departure alternative at the intersection, and the edges have been replaced by two parallel and opposing arcs. The new nodes are connected by auxiliary arcs (the dashed lines) indicating the permitted transitions between them.

According to the municipal authorities, the recycling trucks have no difficulties negotiating turns at the intersections. This implies that the auxiliary arcs do not add appreciably to the

cost and are therefore considered to have null distance. Note that the particular construction applied in this setting does not generate cycles composed of auxiliary arcs only, hence no zero-cost cycles appear in the resulting graph.

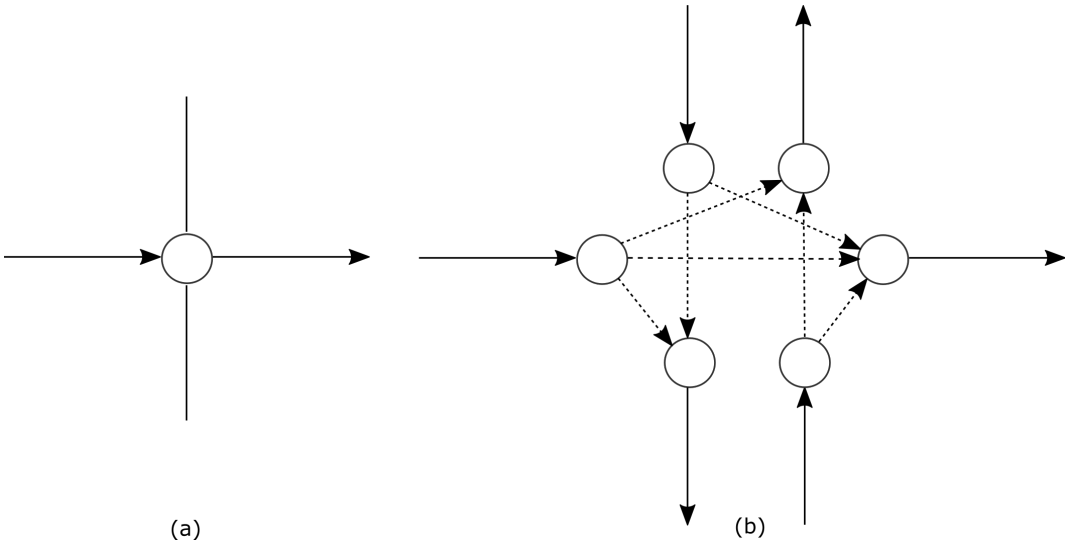


Figure 2: Expansion of a graph node.

The traffic regulations make a necessary exception to the U-turn prohibition for cul-de-sacs. To incorporate this exemption, our graph as developed so far is modified slightly by adding auxiliary arcs to dead-end nodes.

In addition to U-turns, the traffic regulations prohibit left turns from a two-way street at signal-controlled intersections save for the rare occasions where they are explicitly permitted. The graph expansion discussed above readily accommodates this constraint by simply eliminating the auxiliary arcs representing the illegal turns. An example of how this is done for an intersection of two two-way streets with traffic lights is shown in Figure 3.

The starting point for the route of a given sector must be located on the sector border closest to the recycling truck depot. Any node satisfying this condition may be chosen as the starting node. The choice of ending node does not have a major impact on the overall performance given that the trucks go directly back to the recycling depot once they have completed their routes. Since sectors are not very large, this additional cost can be ignored. An alternative approach would have been to add a vertex representing the depot and link it to the starting nodes, thus incorporating the cost from the depot to the sector in the objective function, but since doing so would not have significantly modified either the computations or the generated solutions, this consideration was not adopted.

In certain cases, particularly in sectors with many one-way streets, the recycling trucks may have to make wide detours to access certain corners. To reduce the size and therefore the distance costs of such detours, the drivers will sometimes cross briefly into a neighbouring sector. To incorporate such cases into the model, the street segments immediately beyond

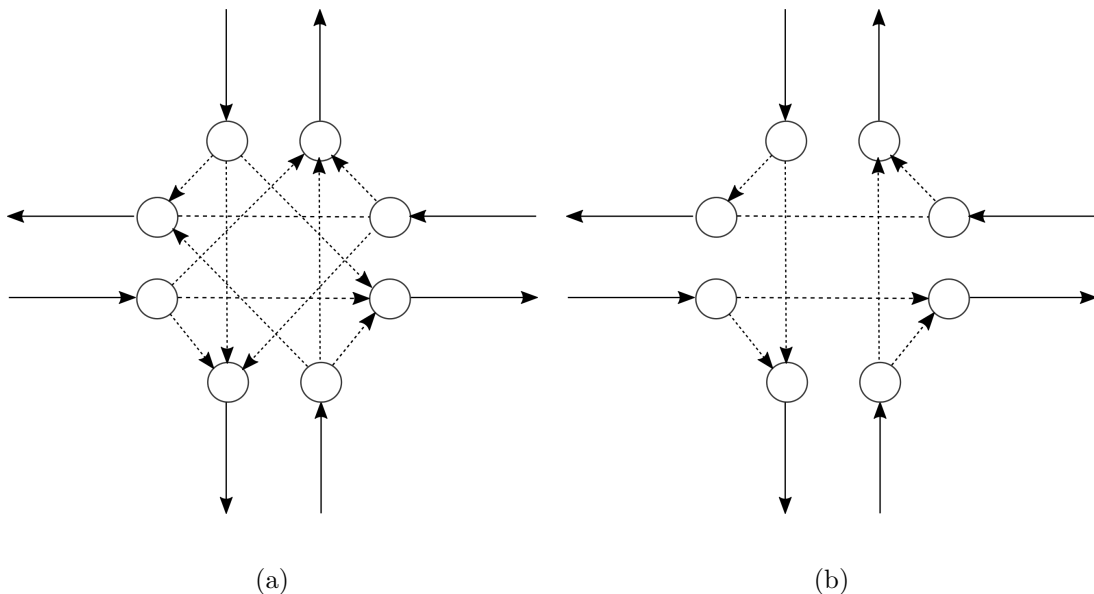


Figure 3: Permitted turns at an intersection (a) without traffic lights and (b) with traffic lights.

the borders of a sector are tagged, thus forming a perimeter band of segments in the adjacent sectors the drivers do not service but can use for detour reduction. Including this option requires no changes to the model since the segments so tagged are treated in the same fashion as auxiliary arcs.

An example of such a perimeter detour band is given in Figure 4 for a typical sector in the municipality. The street segments making up the band are shown in dark lines. Note in this case that there are no tagged segments along the north (top) or south (bottom) of the sector. This is because the north side is closed off by a highway while the south side borders on a different municipality for which up-to-date information on street directionality was not available.

3 Integer programming model

Let $G = (V, A)$ be the directed graph in which the nodes V correspond to all possible alternatives for arriving at the intersections' corners and A is composed of arcs that can be traversed only in one specified direction. Also, let $E \subseteq \{\{i, j\} : i \in V, j \in V, i \neq j\}$ represent two-way street segments that may be travelled in either direction. The input data include a set $A_M \subseteq A$ of mandatory arcs representing one-way segments that must be travelled in the specified direction. A weight function $w : A \rightarrow \mathbb{R}$ associates a weight with each arc, and we assume that the auxiliary arcs between nodes representing street corners all have zero weight. Finally, the input data also include a set $I \subseteq V$ of initial nodes specifying the allowed starting

$$\min \sum_{ij \in A} w_{ij} x_{ij} \quad (1)$$

$$x_{ij} \geq 1 \quad \forall ij \in A_M \quad (2)$$

$$x_{ij} + x_{ji} \geq 1 \quad \forall ij \in E \quad (3)$$

$$s_i + \sum_{j:ji \in A} x_{ji} = \sum_{j:ij \in A} x_{ij} + t_i \quad \forall i \in I \quad (4)$$

$$\sum_{j:ji \in A} x_{ji} = \sum_{j:ij \in A} x_{ij} + t_i \quad \forall i \in V \setminus I \quad (5)$$

$$\sum_{i \in I} s_i = 1 \quad (6)$$

$$\sum_{i \in V} t_i = 1 \quad (7)$$

$$\sum_{ij \in \delta^+(S)} x_{ij} \geq 1 - \sum_{ij \in L_e \cap A(V \setminus S)} x_{ij} \quad \forall S \subseteq V, \forall e \in E \quad (8)$$

$$x_{ij} \in \mathbb{Z}_+ \quad \forall ij \in A \quad (9)$$

$$s_i \in \{0, 1\} \quad \forall i \in I \quad (10)$$

$$t_i \in \{0, 1\} \quad \forall i \in V \quad (11)$$

The objective function attempts to minimize the total cost of the waste collection route, that is, the sum of the individual costs of the segments travelled by the vehicles. Cost is in principle a function of travel time, which is therefore the real factor we wish to minimize, but since travel time data per segment are not available, we use distance as a proxy. Constraints (2) impose that each mandatory arc be travelled at least once while constraints (3) require that each edge be travelled at least once in any direction. Constraints (4)-(5) ensure that the solution is truly a path by adding the standard flow conservation condition at each node. Thus, the number of times the circuit enters a node must equal the number of times the circuit exits it, except for the initial and final nodes whose respective entries and exits number one less. Constraints (6) and (7) guarantee the initial and final nodes are unique.

The addition of the auxiliary arcs to reflect traffic regulation restrictions brings with it certain complications. Since not all the arcs in the graph need to be travelled, requiring indegree to be equal to outdegree will not by itself guarantee that the model's feasible solutions are connected routes. Indeed, constraints (2)-(7) allow subtours to form, an undesirable occurrence that must be avoided. To address this problem, constraints (8) are added to the model, thus avoiding this situation [2]. Finally, constraints (9)-(11) specify the possible values for the model variables. Recall that vehicle capacity is assumed to be sufficient, so capacity constraints are not required.

-
1. Create the relaxed model $\mathcal{M} := (1)-(7)$ and $(9)-(11)$.
 2. Solve \mathcal{M} .
 3. If no solution can be found for \mathcal{M} , return “infeasible” and stop.
 4. If the best solution found for \mathcal{M} has no subtours, return this solution and stop.
 5. If the subtours can be merged with the path from the starting node to the ending node, then merge the subtours with this path, return the obtained solution and stop.
 6. Otherwise, add to \mathcal{M} a *standard* subtour-eliminating constraint for each subtour in the solution and go to Step 2.
-

Table 1: Algorithm pseudocode.

4 The solution procedure

We now propose a procedure for solving the integer programming model presented in the preceding section. The algorithm is summarized in Table 1. The relaxed model constituted by the constraints (2)-(7) and (9)-(11) is solved in Steps 1 and 2. If the resulting solution contains no subtours (Step 4), it is optimal and the algorithm ends; otherwise, a subtour exists and must be eliminated.

In many cases, subtours can be eliminated by the following straightforward *subtour-merging procedure*. Given a subtour S and the *main path* P (i.e., the path starting (resp. ending) at the unique vertex $i \in V$ with $s_i = 1$ (resp. $t_i = 1$)), the procedure attempts an interchange of auxiliary arcs in order to merge S and P . The procedure is summarized in Table 2. Given the main path and a subtour, the procedure scans all of their arcs for any of the following configurations and, if found, executes the indicated action:

- **Configuration A:** If the main path and the subtour meet at some intermediate node, as in Figure 5a, then they are joined as in Figure 5b.
- **Configuration B:** If the subtour meets the last node in the main path, as in Figure 5c, then they are joined as in Figure 5d.
- **Configuration C:** If the subtour meets the first node in the main path, as in Figure 5e, then they are joined as in Figure 5f.

This procedure is applied for every subtour in the solution until they have all merged into the main path or, if they fail to do so, are handled using the constraint technique. Since the cost of all auxiliary arcs joining street-corner nodes is the same, the modified route remains optimal, hence the algorithm stops and returns the obtained solution in this case.

-
1. If there exist $(i_1, j_1) \in P$ and $(i_2, j_2) \in S$ such that $(i_1, j_2) \in A$ and $(i_2, j_1) \in A$, return $[(S \cup P) \setminus \{(i_1, j_1), (i_2, j_2)\}] \cup \{(i_1, j_2), (i_2, j_1)\}$ and finish the procedure.
 2. Let t be the last node of P . If there exists $(i, j) \in S$ such that $(t, j) \in A$, return $[(S \cup P) \setminus \{(i, j)\}] \cup \{(t, i)\}$ and finish the procedure.
 3. Let s be the first node of P . If there exists $(i, j) \in S$ such that $(i, s) \in A$ and $j \in I$, return $[(S \cup P) \setminus \{(i, j)\}] \cup \{(i, s)\}$ and finish the procedure.
 4. Otherwise, return “fail”.
-

Table 2: The subtour-merging procedure given the main path $P \subseteq A$ and a subtour $S \subseteq A$.

This construction allows us to efficiently find an optimal solution from an infeasible solution containing subtours, and proved quite efficient in practice (see Section 5).

Despite its great efficiency, however, this *subtour-merging* technique does not work with every subtour that forms. As an example, the solution generated by the model may be such as the one in Figure 6. In this case, the subtour (around the right block) cannot be merged with the main path (the path around the left block) by the above method.

In this case, the main algorithm sketched in Table 1 adds a standard subtour-eliminating constraint to the \mathcal{M} , i.e., if S is the set of vertices in the cycle to be eliminated, then the algorithm adds the following constraint:

$$\sum_{ij \in \delta^+(S)} x_{ij} \geq 1. \quad (12)$$

Note that we do not add the model constraints (8) to \mathcal{M} , but we add the simpler constraints (12) instead. This allows us to use existing, efficient, and quite standard code in our implementation. The addition of the standard subtour-eliminating constraint (12) is performed through the well-known technique of finding all the connected components of the obtained solution. If there is more than one connected component, then the components not containing the starting and ending nodes are subtours and the corresponding constraints are added to the model. Note that the addition of the stronger inequalities (12) instead of the original model constraints (8) may cut off optimal solutions and increase the objective value of the obtained solution. However, for the instances considered in this work these effects were negligible, as Section 5 shows.

The definitive strategy, then, is to combine this merging technique with the dynamic addition of subtour-eliminating constraints. If a solution has no subtours, it is returned (Step 4). If the solution has one or more subtours but in every case they can be merged with the main path, this is done and the merged solution is returned (Step 5). Finally, if the solution has subtours that cannot be merged to the main path, a subtour-eliminating constraint (12) for each such subtour is added and the procedure is repeated (Step 6).

Note that subtour-eliminating cuts are always –and only– inserted when the branch-and-bound enumeration for solving \mathcal{M} terminates. Detection of subtours is thus straightforward since the variables take integer values. An alternative approach would be to try to separate inequalities (12) dynamically during the branch-and-bound procedure (thus turning Step 2 from the procedure into a branch-and-cut approach), but the detection of violated subtour-eliminating constraints over fractional solutions is not straightforward. For this reason we opted to search for subtours only once the model has been solved and an integer solution has been obtained.

5 Computational results

In this section we present and analyze the results of the experiments conducted to evaluate the effectiveness of the algorithms described in the previous sections in the municipality of Morón. The experiments were executed on a PC with a 3Ghz AMD Phenom II x4 945 processor and 4GB of RAM running Linux Ubuntu 12.04 (32 bits). The solver used for solving the integer programs was SCIP. Although there are commercial solvers that perform better than SCIP on benchmark instances, for our purposes this choice was adequate.

To solve the integer programming models a maximum of 10 minutes was allowed at each iteration. This limit was chosen after determining that the majority of the sectors were solved to optimality within that time while for those that were not, the optimality gap was very small. Thus, if the solution procedure found a feasible solution before the time limit was reached, that solution was used in Steps 4 and 5, otherwise, the procedure stopped and a failure message was returned.

The results of our experiments on the effectiveness of the subtour-merging algorithm are presented in Table 3. As can be seen in the first column, the data all relate to Castelar district, which was chosen for this test because of the great diversity of its street patterns. The district’s eight sectors vary greatly in size, shape (some rectangular, others more irregular) and street directionality (some mainly one-way, others primarily two-way). Columns 6 and 7 report the number of required iterations and the solution times for the various sectors without the subtour-merging algorithm (i.e., with a simple branch-and-cut procedure dynamically adding the standard subtour elimination constraints (12) for every non-feasible integer solution). The final three columns show the number of iterations, the solution times and the optimality gap when the subtour-merging algorithm is employed.

The most important conclusion from these data is that the subtour-merging procedure allows to get optimal solutions (with respect to the complete formulation (1)-(11)) in most instances. Indeed, with the exception of Castelar7, the subtour-merging procedure could construct a feasible solution from the integer solution of the relaxed model defined by (1)-(7) and (9)-(11). The so-constructed solution has the same objective value than the relaxed solution, and is optimal for all the instances except Castelar3 and Castelar6, which attained the time limit and thus are subject to the gaps reported in the last column. On the other hand, just one subtour-eliminating constraint (12) was added in Castelar7, and the subtour-merging

procedure could construct a feasible solution from the resulting solution.

Sector	$ V $	$ E $	$ A_M $	$ A_{AUX} $	Without Step 5		With Step 5		
					Iterations	Time	Iterations	Time	Gap
Castelar1	1372	658	28	1570	2	5.2 sec	1	2.4 sec	0.00%
Castelar2	650	100	225	559	5	19 sec	1	3 sec	0.00%
Castelar3	852	200	226	782	59	9.7 hrs	1	10 min	0.47%
Castelar4	878	354	85	876	6	52.4 sec	1	5.6 sec	0.00%
Castelar5	1090	508	37	1416	2	5.7 sec	1	3 sec	0.00%
Castelar6	738	166	203	650	22	3.7 hrs	1	10 min	1.05%
Castelar7	1082	524	17	1341	35	5.9 hrs	2	20 min	0.31%
Castelar8	1276	622	46	1574	24	3.7 hrs	1	17.9 sec	0.00%

Table 3: Results of the experiments on subtour-merging algorithm effectiveness for Castelar District with and without the subtour-merging procedure (Step 5 in Table 1).

To compare the route solutions generated by our methodology with the manually-designed routes previously used by the truck drivers, we were able to obtain data on the latter from the municipality for a limited number of sectors, shown here in Table 4. This information was gathered using GPS devices installed in the trucks. As mentioned in the introduction, these manual routes had been improvised by the drivers, so the inevitable result was that in some sectors a considerable number of street segments were often left unserved. As the table indicates, the worst case in absolute terms was Castelar4 where 60 street blocks were skipped, though in some other sectors the situation in percentage terms was worse still.

Sector	Distance travelled	Street blocks skipped	Collection time	% of sector serviced
Castelar4	46 km	60	4:10hs	88,5%
Castelar7	43,1 km	6	3:40hs	98,6%
Haedo2	17,5 km	20	2:26hs	89,7%
Palomar1	25 km	46	3:38hs	84,5%
Palomar2	24,5 km	35	4:10hs	87,5%
Palomar3	31,2 km	17	3:12hs	94,8%
Palomar4	35,2 km	35	4:46hs	91,0%
VillaSarmiento1	24,6 km	10	3:09hs	96,1%
VillaSarmiento2	23,7 km	9	2:44hs	96,3%
VillaSarmiento3	23,2 km	17	3:13hs	93,2%

Table 4: Available GPS data supplied by the municipality for certain sectors.

This tendency to skip street segments under the manual routes inevitably impacted the comparisons between them and our model-generated routes for these sectors given that the latter were defined using constraints ensuring that all street segments be traversed. The effect

can be seen in the difference between the “Distance” column in Table 5 and the distances travelled shown in Table 4. In some sectors where the numbers of segments skipped was high, the model-generated route was longer/more costly than the manual route. On the other hand, in three sectors where the number of omitted segments was relatively small (Castelar7, Palomar3 and VillaSarmiento1), the model found a route that was shorter than the manual one. In most of the remaining sectors (Haedo2, Palomar2, VillaSarmiento2, VillaSarmiento3) the distance of the model-generated route was slightly higher but the difference never exceeded 2 km. And, of course, whereas the model leaves no segments unserved, those left out by the improvised routes added up to as much as 6 km for a single sector.

The added length of certain of the model-optimized routes also raised the technical question of whether the 5-hour period allotted for the waste collection rounds was still sufficient. A simple estimation shows that it was. To take one example, the manually-designed route for the Castelar4 sector was 46km long and required 4:10 hours to cover whereas the model defined a 50km route. Though it was impossible to know a priori exactly how long the latter route (or any other defined by the model) would take to complete, the average speed implied by the manual route data (approx. 11km/hr) suggests we could safely assume the extra 4km could be traversed within the aforementioned time window. This and the a priori estimates for the other cases where the model solution route was longer than the manual route proved to be reasonable when the former were applied in practice.

Sector	$ V $	$ E $	A_M	A_{AUX}	Solution time	Gap	Iterations	Distance
Castelar4	878	354	85	876	5.6 sec	0.00 %	1	50.45 km
Castelar7	1082	524	17	1341	1200 sec	0.31 %	2	40.45 km
Haedo2	1154	396	181	1256	0.3 sec	0.00 %	1	18.31 km
Palomar1	1046	502	21	1251	30 sec	0.00 %	1	33.51 km
Palomar2	822	316	95	875	512 sec	0.00 %	1	26.90 km
Palomar3	1254	608	19	1596	1200 sec	0.30 %	2	30.53 km
Palomar4	1090	500	45	1353	1200 sec	0.30 %	2	30.53 km
VillaSarmiento1	304	12	140	258	1 sec	0.00 %	1	24.17 km
VillaSarmiento2	600	134	166	494	1 sec	0.00 %	1	24.25 km
VillaSarmiento3	384	76	116	322	1 sec	0.00 %	1	25.08 km

Table 5: Results for routes calculated by model for sectors in Table 4, covering all mandatory street blocks from each sector.

As for solution times of the Table 4 sectors, the longest was 20 minutes. This is quite reasonable for the requirements of the recycling system given that once a definitive route has been established, only occasional changes will later be needed and in such cases an instantaneous response is unlikely to be required. The optimality gaps were very small if not zero within the time limit proposed for model execution.

6 Conclusions

A methodology was proposed using integer programming techniques to optimize the routes of recycling waste collection vehicles in Morón, an important suburban Buenos Aires municipality. A number of conditions peculiar to this case required the inclusion of certain restrictions and options such that the problem was best formulated as a particular case of the generalized directed open rural postman problem on a directed graph constructed from the city map. These conditions included the Argentinean traffic regulation prohibitions on left turns at traffic lights and U-turns, the exception to this rule made for U-turns in cul-de-sacs, the inclusion of perimeter bands for detours outside the serviced area, and street directionality (one-way and two-way).

From a theoretical standpoint, the main contribution of this study is the development of the subtour-merging procedure, that enables running times to be reduced. This is possible because mandatory arcs are connected by zero-cost auxiliary arcs, hence no additional cost results from the subtour merging. This hypothesis does not hold in general, however, so the proposed procedure cannot be applied for solving general instances of the generalized directed rural postman problem. Since the model is run with a time limit of 10 minutes, the quality of the obtained routes with and without applying the subtour-merging procedure cannot be compared if the time limit is attained. Furthermore, the addition of the standard subtour-eliminating inequalities (12) instead of the model constraints (8) simplifies the code but may hinder optimality. Nevertheless, the obtained results for the instances coming from the municipality of Morón allow us to safely assume that the obtained routes are of good quality for practical purposes.

The recycling vehicle routes generated by the proposed methodology and implemented by the municipality perform significantly better than the manually defined routes used previously. The main advance is in the coverage provided by the collection service, which now fully covers every street within the municipal limits. Before implementation, up to 60 street blocks (or, equivalently 16% of street blocks) were skipped in certain sectors. This improvement was achieved while in some cases actually reducing total vehicle distance compared to the old manual routes. In a few cases distance did increase, but only slightly.

The routes designed by the model were well received by the municipality. Field analyses were conducted before the routes were implemented to validate their feasibility and the reactions of the municipal authorities were very positive. After several months of testing the routes defined by the integer programming-based methodology, the municipality began implementation in mid-2014. The result was better coverage of all streets and the virtual disappearance of complaints from the public regarding the recyclable waste collection service, which previous to implementation had been running at about dozens of calls or messages a week. The success of the project has been such that the authors have been invited by Argentina's Ministry of the Interior and Transport to replicate the experience in a number of other municipalities around the country.

Acknowledgements

This study was partly funded by UBACyT grant no. 20020130100808BA (Argentina), AN-PCyT PICT grant no. 2012-1324 (Argentina) and FONDECyT grant no. 1140787 (Chile) as well as by the Millennium Institute Complex Engineering Systems (Chile) and by the consulting firm papyro.com (Argentina). The authors would like to thank the municipality of Morón for their support in carrying out this project, and Kenneth Rivkin for his useful comments.

References

- [1] ARRIBAS, C., BLAZQUEZ, C., AND LAMAS, A. Urban solid waste collection systems using mathematical modelling and tools of geographic information systems. *Waste Management & Research* 24, 4 (2010), 355–363.
- [2] ÁVILA, T., CORBERÁN, A., PLANA, I., AND SANCHIS, J. A new branch-and-cut algorithm for the generalized directed rural postman problem. *Transportation Science, in press* 50, 8 (2014), 750–761.
- [3] BONOMO, F., DURÁN, G., LARUMBE, F., AND MARENCO, J. A method for optimizing waste collection using mathematical programming: A Buenos Aires case study. *Waste Management & Research* 30, 3 (2012), 311–324.
- [4] CHANG, N., LU, H., AND WEI, L. GIS technology for vehicle routing and scheduling in solid waste collection systems. *Journal of Environmental Engineering* 123 (1997), 901–933.
- [5] CHRISTOFIDES, N., CAMPOS, V., CORBERÁN, A., AND MOTA, R. An algorithm for the rural postman problem on a directed graph. *Mathematical Programming Study* 26 (1986), 155–166.
- [6] CORBERÁN, A., MARTÍ, R., MARTÍNEZ, E., AND SOLER, D. The rural postman problem on mixed graphs with turn penalties. *Computers and Operations Research* 29, 7 (2002), 887–903.
- [7] CORBERÁN, A., MARTÍ, R., AND ROMERO, A. Heuristics for the mixed rural postman problem. *Computers and Operations Research* 27, 2 (2000), 183–203.
- [8] CORBERÁN, A., MOTTA, E., AND SANCHIS, J. A comparison of two different formulations for arc routing problems on mixed graphs. *Computers and Operations Research* 33, 12 (2006), 3384–3402.
- [9] CORBERÁN, A., PLANA, I., AND SANCHIS, J. The rural postman problem on directed, mixed, and windy graphs. *In: Arc Routing: Problems, Methods, and Applications, MOS-SIAM Series on Optimization, MO20* (2014), 101–127.

- [10] DREXL, M. On the generalized directed rural postman problem. *Journal of the Operational Research Society* 65, 8 (2014), 1143–1154.
- [11] EISELT, H., GENDREAU, M., AND LAPORTE, G. Arc routing problems, part ii: The rural postman problem. *Operations Research* 43, 3 (1995), 399–414.
- [12] EISENSTEIN, D., AND IYER, A. Garbage collection in Chicago: a dynamic scheduling model. *Management Science* 43 (1997), 922–933.
- [13] INSTITUTO NACIONAL DE ESTADÍSTICAS Y CENSOS. Censo Nacional 2010 (retrieved July, 2016). <http://www.censo2010.indec.gov.ar> (2010).
- [14] KIM, B., KIM, S., AND SAHOO, S. Waste collection vehicle routing problem with time windows. *Computers and Operations Research* 33 (2006), 3624–3642.
- [15] MOURAO, M., AND ALMEIDA, M. Lower-bounding and heuristic methods for a refuse collection vehicle routing problem. *European Journal of Operational Research* 121 (2000), 420–434.
- [16] MUNICIPALITY OF MORÓN. “Día verde” institutional video (retrieved July, 2016). <https://www.youtube.com/watch?v=V2939nImAR0> (2013).
- [17] PEARN, W., AND WU, T. Algorithms for the rural postman problem. *Computers and Operations Research* 22, 8 (1995), 819–828.
- [18] RODRIGUES, A., AND FERREIRA, J. Solving the rural postman problem by memetic algorithms. *Proceedings of MIC’2001 - 4th Metaheuristics International Conference* (2001), 679–683.
- [19] YEOMANS, J., HUANG, G., AND YOOGALINGAM, R. Combining simulation with evolutionary algorithms for optimal planning under uncertainty: An application to municipal solid waste management planning in the regional municipality of Hamilton-Wentworth. *Journal of Environmental Informatics* 2, 1 (2003), 11–30.

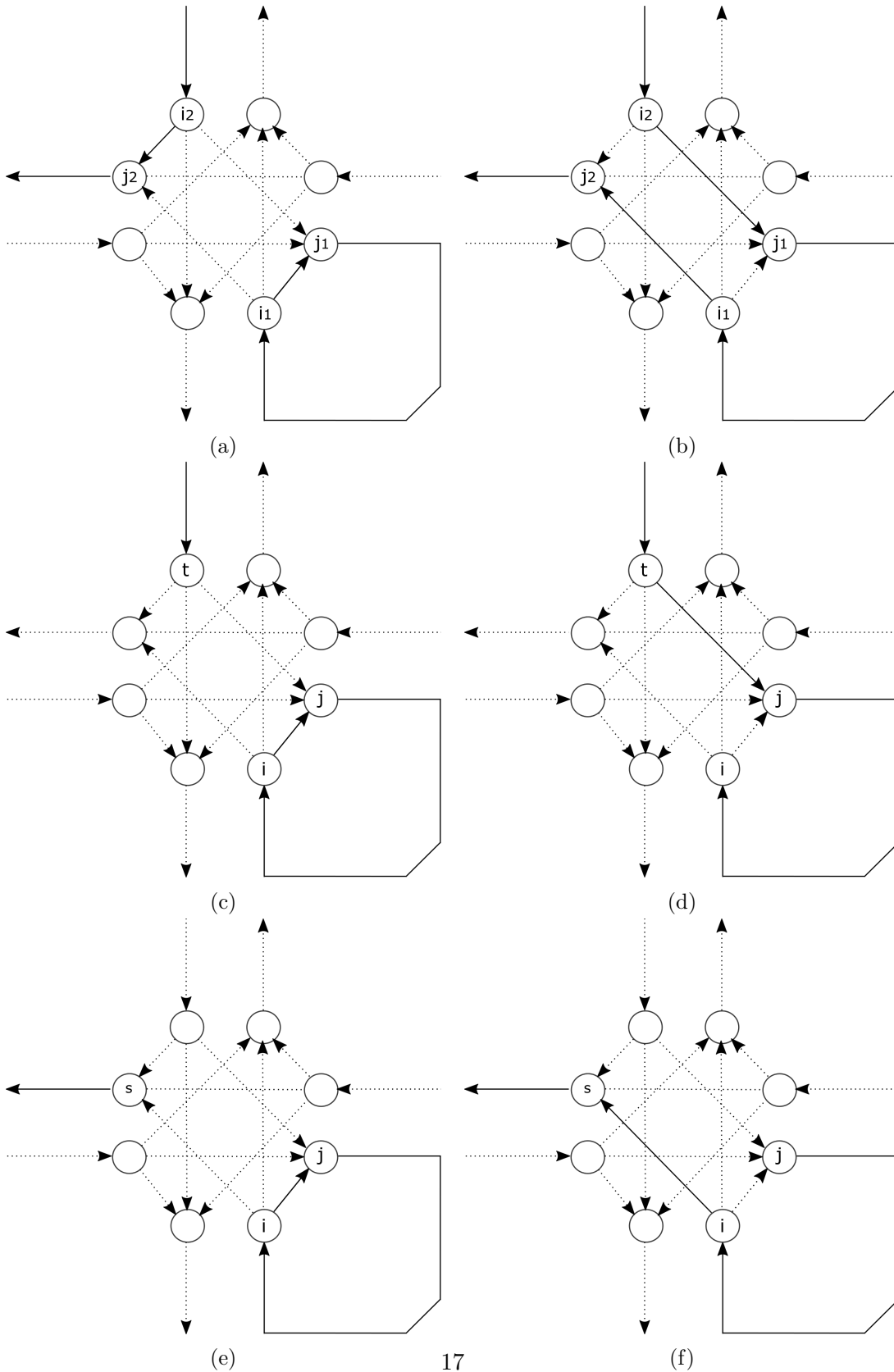


Figure 5: Possible configurations in the subtour-merging procedure.

