Available online at www.sciencedirect.com
SCIENCE DIRECT®
ELSEVIER
Computers & Operations Research 32 (2005) 219–228
computers & operations research
www.elsevier.com/locate/dsw

A heuristic for the periodic rural postman problem

Gianpaolo Ghiani[a,*], Roberto Musmanno[b], Giuseppe Paletta[c], Chefi Triki[d]

[a]Dipartimento di Ingegneria dell'Innovazione, Università degli Studi di Lecce, Lecce 73100, Italy
[b]Dipartimento di Elettronica, Informatica e Sistemistica, Università degli Studi della Calabria, Rende (CS) 87030, Italy
[c]Dipartimento di Economia e Statistica, Università degli Studi della Calabria, Rende (CS) 87030, Italy
[d]Dipartimento di Matematica, Università degli Studi di Lecce, Lecce 73100, Italy

Abstract

The periodic rural postman problem (PRPP) is variant of the classical rural postman problem whose applications arise in garbage collection and street sweeping. In the PRPP each required arc/edge of a graph must be visited a given number of times over an m-day planning period in such a way that service days are equally spaced. The PRPP amounts to select a service day combination for each required arc/edge and to determine a postman tour for each day of the planning period. The objective is to minimize the total distance travelled. In this paper a simple but effective heuristic for the undirected PRPP is presented. Extensive computational results indicate that the algorithm is capable of providing high quality solutions. To our knowledge this is the first methodological paper devoted to a periodic arc routing problem.

Keywords: Arc routing problems; Periodic routing problems

1. Introduction

The purpose of this article is to present a heuristic for the *undirected periodic rural postman problem* (PRPP) defined as follows: Let $G=(V,E)$ be an undirected graph, where V is the vertex set, E is the edge set, c_{ij} is the cost of traversing edge $(v_i,v_j)\in E$, and $R\subseteq E$ is a set of *required* edges. Each required edge $e\in R$ must be serviced n_e times over an m-day planning period in such a way that service days are equally spaced. The PRPP amounts to deciding on which days each required edge has to be serviced and to design a postman tour for each day of the planning period. The objective is to minimize the total distance travelled over the m-day period.

The PRPP is NP-hard as it contains the rural postman problem (RPP) as a special case. Indeed, once a service day combination is chosen for each required edge, a PRPP solution can be obtained

* Corresponding author.
E-mail addresses: gianpaolo.ghiani@unile.it (G. Ghiani), musmanno@unical.it (R. Musmanno), g.paletta@unical.it (G. Paletta), chefi.triki@unile.it (C. Triki).

doi:10.1016/S0305-0548(03)00213-2

by determining an RPP route for each day of the planning period. Exact algorithms and heuristics for the RPP have been presented recently in [1–3].

The PRPP arises in the design of garbage collection and street sweeping routes whenever streets do not require to be serviced every day (see, e.g., [4–6]). At present the literature devoted to periodic arc routing problems is quite poor and disorganized, in spite of the economic importance of these problems [7–9]. This is in contrast to periodic node routing problems which have been intensively studied (see, e.g., [10–13]). To our knowledge this is the first methodological paper devoted to a periodic arc routing problem.

The remainder of this article is organized as follows. In Section 2, we describe a heuristic for the PRPP. This is followed by a computational assessment of the performance of the algorithm in Section 3 and by conclusions in Section 4.

2. A heuristic

We have developed a heuristic that first selects the same service day combination for all the edges e having a given service frequency n_e and then performs a local search in the attempt to obtain a cost saving. The local search phase makes use of an innovative neighborhood structure.

Let S be the set of divisors of m and let $N \subseteq S$ be the set of feasible service frequencies. Also denote, for each $k \in N$, $R_k = \{e \in R\text{: } n_e = k\}$ as the set of edges that are required to be serviced k times and V_k the set of vertices such that an edge e exists in R_k. Given a PRPP solution, $R^{[t]}$ denotes the edges serviced on day t and $RPP(R^{[t]})$ a rural postman tour including edges $R^{[t]}$. At each step k the heuristic tentatively assigns required edges R_k to service days $t = 1, m/k + 1, \ldots, (k-1)m/k + 1$. Then the algorithm tries to achieve a cost reduction by servicing a subset of edges $F \subseteq R_k$ on days t $(t = u, m/k + u, \ldots, (k-1)m/k + u)$ for some $u \in 2, \ldots, m/k$. Such a step is performed through three procedures: *Path Transfers*, *Cycle Transfers* and *Component Transfers*. A formal description of our heuristic is presented in Algorithm 1.

Algorithm 1. Outline of the heuristic.

```
Sort N in non ascending order. Let succ(k) be the successor of k in N;
Set R^[t] = R_m, t = 1,...,m;
Determine RPP(R^[t]), t = 1,...,m;
Set k = m;
while (k > 1) do begin
k := succ(k);
R^[t] := R^[t] ∪ R_k, t = 1, m/k + 1,...,(k − 1)m/k + 1;
Determine RPP(R^[t]), ∪R_k, t = 1, m/k + 1,...,(k − 1)m/k + 1;
repeat
Call Path Transfers(k ∈ N) procedure;
Call Cycle Transfers(k ∈ N) procedure;
Call Component Transfers(k ∈ N) procedure;
until (no additional cost saving can be obtained)
end.
end.
```

Both *Path Transfers*($k \in N$) and *Cycle Transfers*($k \in N$) procedures attempt to remove deadheaded edges by suitably moving paths of serviced edges $e \in R_k$ from a service combination to another (Algorithms 2 and 3). On the other hand, *Component Transfers*($k \in N$) subroutine verifies whether a cost saving can be obtained by assigning a different service combination to a connected component induced by R_k (Algorithm 4).

Algorithm 2. Outline of *Path Transfers* procedure.

Procedure Path Transfers ($k \in N$)
begin
for $i \in V_k,\ j \in V_k$ *do begin*
Determine whether there exists a path P_{ij} *in* $RPP(R^{[t]})$, $t = 1, m/k + 1, \ldots, (k-1)m/k + 1$, *between i and j made up of serviced edges* $e \in R_k$.
Determine whether there exists a path $\bar{P}^t_{ij}$ *in* $RPP(R^{[t]})$, $t = 1, \ldots, m$, *between i and j made up of deadheaded edges.*
if $\exists P_{ij}$
if $\exists \bar{P}^t_{ij}$, $t = 1, u, m/k + 1, m/k + u, \ldots, (k-1)m/k + 1, (k-1)m/k + u$ *for some* $u \in \{2, \ldots, m/k\}$ *then*
begin
Remove paths $\bar{P}^t_{ij}$, $t = 1, u, m/k + 1, m/k + u, \ldots, (k-1)m/k + 1, (k-1)m/k + u$.
Move paths P_{ij} *from days* $t = 1, m/k + 1, \ldots, (k-1)m/k + 1$ *to days* $t = u, m/k + u, \ldots, (k-1)m/k + u$.
end. end.

Algorithm 3. Outline of *Cycle Transfers* procedure.

Procedure Cycle Transfers ($k \in N$)
begin
for $i \in V_k, jV_k$ *do begin*
Determine whether there exist two edge-disjoint paths P_{ij} *and* $\bar{P}_{ij}$ *in* $RPP(R^{[t]})$, $t = 1, m/k + 1, \ldots, (k\text{-}1)m/k + 1$, *between i and j made up of serviced edges* eR_k.
Determine whether there exists a path $\hat{P}^t_{ij}$ *in* $RPP(R^{[t]})$, $t = u, m/k + u, \ldots, (k\text{-}1)m/k + u$ *for some* $u2,, m/k$, *between i and j made up of deadheaded edges.*
Determine whether there exists a path $\underline{P}^t_{ij}$ *in* $RPP(R^{[t]})$, $t = u', m/k + u', \ldots, (k\text{-}1)m/k + u'$ *for some* $u'\{2, \ldots, m/k\}$, uu', *between i and j made up of deadheaded edges.*
if P_{ij} *and* $\bar{P}_{ij}$
if $\hat{P}^t_{ij}$, $t = u, m/k + u, \ldots, (k\text{-}1)m/k + u$
if $\underline{P}^t_{ij}$, $t = u', m/k + u', \ldots, (k\text{-}1)m/k + u'$ *for some* $u, u'\{2, \ldots, m/k\}, uu'$ *then begin*
Remove paths $\hat{P}^t_{ij}$, $t = u, m/k + u, \ldots, (k\text{-}1)m/k + u$.
Remove paths $\underline{P}^t_{ij}$, $t = u', m/k + u', \ldots, (k\text{-}1)m/k + u'$.
Move paths P_{ij} *and* $\bar{P}_{ij}$ *from days* $t = 1, m/k + 1, \ldots, (k\text{-}1)m/k + 1$ *to days* $t = u, m/k + u, \ldots, (k\text{-}1)m/k + u$ *and* $t = u', m/k + u', \ldots, (k\text{-}1)m/k + u'$, *respectively.*
end. end.

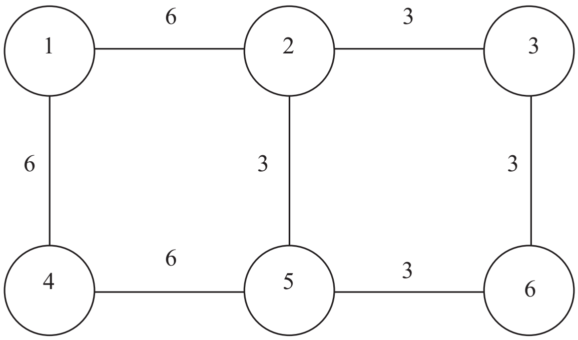

Fig. 1. Sample graph showing edge service frequencies. Edge travel costs are equal to one.

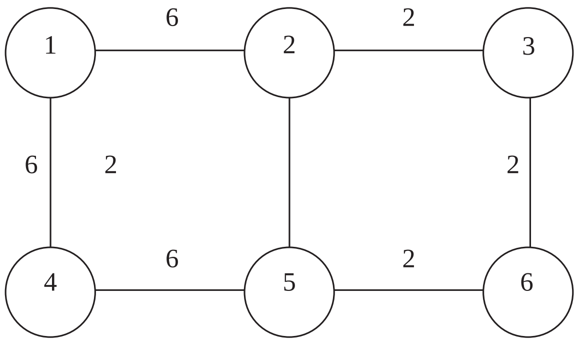

Fig. 2. Sample graph showing edge service frequencies. Edge travel costs are equal to one.

Algorithm 4. Outline of *Component Transfers* procedure.

Procedure *Component Transfer* ($k \in N$)
begin
Verify whether a cost saving can be obtained by moving a connected component induced by R_k *from* $RPP(R^{[t]})$ $(t = 1, m/k + 1, \ldots, (k-1)m/k + 1)$ *to* $RPP(R^{[t']})$ $(t' = u, m/k + u, \ldots, (k-1)m/k + u)$ *for some* $u \in 2, \ldots, m/k$.
end.

We now illustrate *Path Transfers* and *Cycle Transfers* procedures through the following two examples (Figs. 1 and 2). In both examples the value of m is six. Consider the first sample problem depicted in Fig. 1. At the first iteration our procedure determines an RPP route servicing edges e with $n_e = 6$ (Fig. 3). Then, at the second iteration, edges e with $n_e = 3$ are allocated to the first service combination and the RPP routes are updated (Fig. 4). Finally, *Path Transfers* procedure is applied (Fig. 5). The second example (Figs. 2) is similar, except that at the second iteration *Cycle Transfers* procedure is applied (Figs. 6, 7, 8).

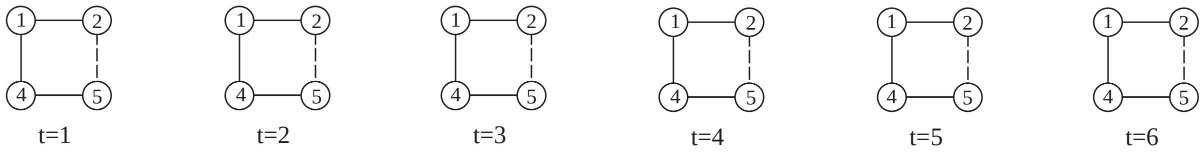

Fig. 3. Partial solution after the insertion of edges having service frequency equal to six. Serviced edges are shown in bold lines.

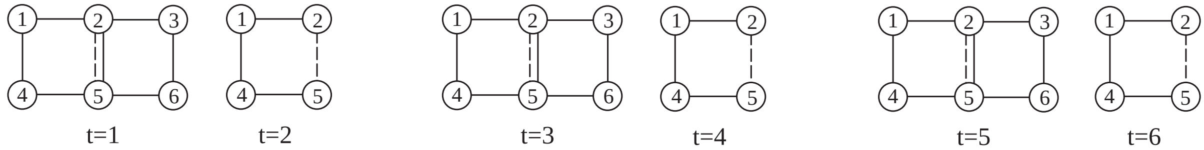

Fig. 4. Partial solution after the insertion of edges having service frequency equal to three. Serviced edges are shown in bold lines.

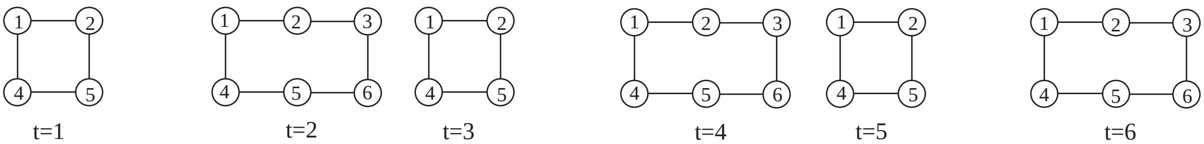

Fig. 5. Partial solution provided by *Path Transfer* procedure. Serviced edges are shown in bold lines.

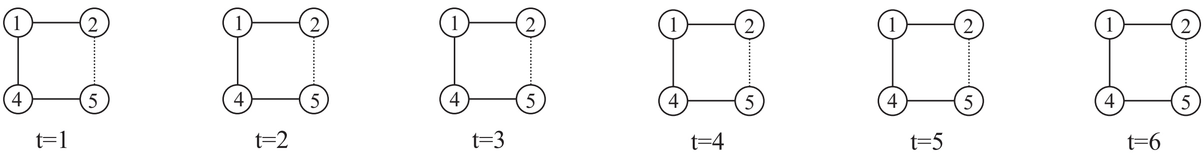

Fig. 6. Partial solution after the insertion of edges having service frequency equal to six. Serviced edges are shown in bold lines.

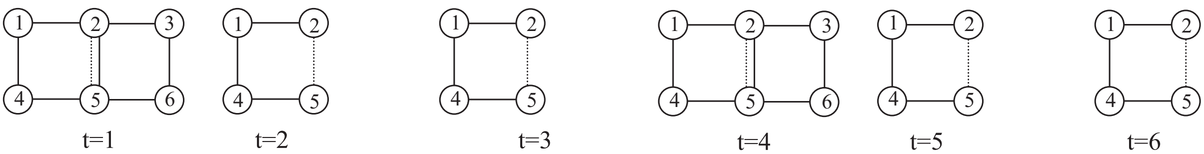

Fig. 7. Partial solution after the insertion of edges having service frequency equal to two. Serviced edges are shown in bold lines.

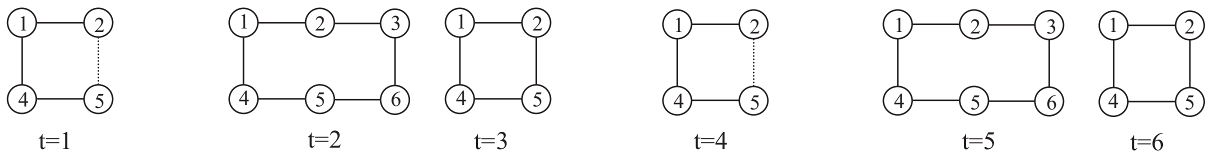

Fig. 8. Partial solution provided by *Cycle Transfer* procedure. Serviced edges are shown in bold lines.

3. Computational results

The heuristic was coded in C and run on a PC with a Pentium III processor clocked at 700 MHz. Rural postman problem tours $RPP(R^{[t]})$ were obtained through the exact algorithm presented in [2].

The main goal of our computational tests was to assess the quality of solutions produced by the heuristic. For this purpose we generated a set of PRPP instances (with $m = 6$, as is customary in garbage collection applications) whose optimal solution is known a priori. Such problems were

Table 1
$p = 0.1$

$\|V\|$	Instance	Gap (%)	Time (s)
100	1	2.46	62
	2	2.06	25
	3	0	23
	4	0	22
	5	0	24
	Average	0.90	31.2
150	1	0	114
	2	0	115
	3	0	122
	4	0	114
	5	0	114
	Average	0	115.8
200	1	1.27	407
	2	0	388
	3	0.62	398
	4	0	385
	5	0.63	406
	Average	0.50	396.8
250	1	2.93	952
	2	0	910
	3	0	1048
	4	0.68	1065
	5	1.03	1066
	Average	0.92	1008.2

Table 2
$p = 0.2$

$\|V\|$	Instance	Gap (%)	Time (s)
100	1	1.99	56
	2	2.04	27
	3	0.53	30
	4	0	25
	5	0.64	26
	Average	1.04	32.8
150	1	0	116
	2	0	118
	3	0.78	122
	4	0	115
	5	0	118
	Average	0.15	117.8
200	1	0.92	411
	2	0.57	402
	3	0.61	409
	4	0.39	404
	5	0.63	401
	Average	0.62	405.4
250	1	2.80	1086
	2	0.58	1079
	3	0	1061
	4	1.61	1072
	5	2.62	1100
	Average	1.52	1079.6

obtained as follows: First, we generated a set of type 3 Eulerian random graph as in [2,14]. Then we set $n_e = 6$ for each edge. Finally, we randomly selected a subset F of edges with probability p and each edge $e \in F$ was substituted for a subset of parallel edges according to the following scheme:

(a) edge e is substituted for a pair of parallel edges e^1, e^2 with $n_{e^1} = n_{e^2} = 3$;
(b) edge e is substituted for a triple of parallel edges e^1, e^2, e^3 with $n_{e^1} = n_{e^2} = n_{e^3} = 2$;
(c) edge e is substituted for four parallel edges e^1, e^2, e^3, e^4 with $n_{e^1} = 3$ and $n_{e^2} = n_{e^3} = n_{e^4} = 1$;
(d) edge e is substituted for four parallel edges e^1, e^2, e^3, e^4 with $n_{e^1} = n_{e^2} = 2$ and $n_{e^3} = n_{e^4} = 1$;
(e) edge e is substituted for five parallel edges e^1, e^2, e^3, e^4, e^5 with $n_{e^1} = 2$ and $n_{e^2} = n_{e^3} = n_{e^4} = n_{e^5} = 1$;
(f) edge e is substituted for six parallel edges $e^1, e^2, e^3, e^4, e^5, e^6$ with $n_{e^1} = n_{e^2} = n_{e^3} = n_{e^4} = n_{e^5} = n_{e^6} = 1$.

The resulting PRPP instance is easily proved to have an optimal solution with no deadheaded edge, i.e. an optimal solution cost equal to $\sum_{(i,j) \in \bar{E}} c_e n_e$, where $\bar{E}$ is the edge set of the multigraph produced by the above procedure.

Five graphs were considered for $|V| = 100, 150, 200, 250$ and for $p = 0.1, 0.2, 0.3, 0.4$. Computational results and average statistics over all instances are reported in Tables 1–4. The meanings of the

Table 3
$p = 0.3$

$\lvert V \rvert$	Instance	Gap (%)	Time (s)
100	1	2.03	40
	2	3.91	27
	3	1.70	28
	4	0.94	27
	5	2.68	28
	Average	2.25	30
150	1	0	121
	2	1.54	129
	3	0.50	124
	4	0	121
	5	2.13	126
	Average	0.83	124.2
200	1	2.38	422
	2	2.29	417
	3	2.93	418
	4	1.21	419
	5	2.38	491
	Average	2.23	433.4
250	1	3.18	1091
	2	1.17	1082
	3	1.03	1082
	4	2.71	1090
	5	3.65	1090
	Average	2.34	1069

column headings are as follows:

- $\lvert V \rvert$: number of vertices of the graph;
- Instance: instance number;
- Gap: gap between the heuristic solution value and the optimal solution value;
- Time: CPU time in seconds.

Computational results show that our heuristic provides a solution whose cost exceeds the optimal solution cost by 0.58%, 0.83%, 1.91% and 2.92% on the average for $p = 0.1, 0.2, 0.3$ and 0.4, respectively. Instances and solutions are available on the web site http://persone.dii.unile.it/ghiani/.

4. Conclusion

We have developed a simple but effective heuristic for the undirected PRPP. Extensive computational results indicate that the algorithm is capable of providing high quality solutions. To our knowledge this is the first methodological paper devoted to a periodic arc routing problem.

Table 4
$p = 0.4$

$\|V\|$	Instance	Gap (%)	Time (s)
100	1	4.03	28
	2	5.09	29
	3	2.62	30
	4	2.42	28
	5	3.08	29
	Average	3.44	28.8
150	1	0.35	125
	2	1.54	129
	3	3.34	134
	4	0.50	126
	5	3.81	132
	Average	1.90	129.2
200	1	3.50	426
	2	3.87	425
	3	3.48	417
	4	3.31	429
	5	3.33	432
	Average	3.50	428
250	1	4.00	1095
	2	2.2	1088
	3	1.30	1102
	4	2.78	1089
	5	4.18	1096
	Average	2.85	1095.2

Acknowledgements

This research was partially supported by MIUR (Ministero dell'Istruzione, dell'Università e della Ricerca), by the Italian National Research Council, and by the Center of Excellence on High-Performance Computing at the University of Calabria, Italy. This support is gratefully acknowledged.

References

[1] Fernández E, Grafinkel R, Meza O, Ortega M. On the undirected rural postman problem: tight bounds based on a new formulation. Operations Research, 2003;51(2):281–91.
[2] Ghiani G, Laporte G. A branch-and-cut algorithm for the undirected rural postman problem. Mathematical Programming 2000;87(3):467–81.
[3] Hertz A, Laporte G, Nanchen-Hugo P. Improvement procedures for the undirected rural postman problem. INFORMS Journal on Computing 1999;11:53–62.
[4] Clark RM, Gillean JI. Analysis of solid waste management operations in Cleveland. Interfaces 1975;6(2):32–42.

[5] Negreiros M. O Problema de Planejamento e Percuso de Veculos na Coleta do Lixo Urbano Domiciliar. M.Sc. Dissertation, Sistemas/COPPE, Federal University of Rio de Janeiro, Brazil, 1974.
[6] Wyskida R, Gupta J. IE's improve city's solid waste collection. Industrial Engineering 1973;46:12–5.
[7] Dror M, editors. Arc routing: theory, solutions and applications. New York: Kluwer Academic Publishers; 2002.
[8] Eiselt HA, Gendreau M, Laporte G. Arc routing problems. Part I: the Chinese postman problem. Operations Research 1995;43:231–42.
[9] Eiselt HA, Gendreau M, Laporte G. Arc routing problems. Part II: the rural postman problem. Operations Research 1995;43:399–414.
[10] Christofides N, Beasley JE. The period routing problem. Networks 1984;14:237–56.
[11] Cordeau J-F, Gendreau M, Laporte G. A Tabu search heuristic for periodic and multi-depot vehicle routing problems. Networks 1997;30:105–19.
[12] Gaudioso M, Paletta G. A heuristic for the periodic vehicle routing problem. Transportation Science 1992;26:86–92.
[13] Paletta G. The period traveling salesman problem: a new heuristic algorithm. Computers and Operations Research, 2003, forthcoming.
[14] Corberán A, Sanchis JM. A polyhedral approach to the rural postman problem. European Journal of Operational Research 1994;79:95–114.