ISCTE IUL
REPOSITÓRIO
INSTITUTO UNIVERSITÁRIO DE LISBOA

Repositório ISCTE-IUL

Deposited in *Repositório ISCTE-IUL*:
2022-07-20

Deposited version:
Accepted Version

Peer-review status of attached file:
Peer-reviewed

Citation for published item:
Nunes, A. C., Mourão, M. C. & Prins, C. (2009). Heuristic Methods for the sectoring arc routing problem. European Journal of Operational Research . 196 (3), 856-868

Further information on publisher's website:
10.1016/j.ejor.2008.04.025

Publisher's copyright statement:
This is the peer reviewed version of the following article: Nunes, A. C., Mourão, M. C. & Prins, C. (2009). Heuristic Methods for the sectoring arc routing problem. European Journal of Operational Research . 196 (3), 856-868, which has been published in final form at https://dx.doi.org/10.1016/j.ejor.2008.04.025. This article may be used for non-commercial purposes in accordance with the Publisher's Terms and Conditions for self-archiving.

Use policy

Creative Commons CC BY 4.0
The full-text may be used and/or reproduced, and given to third parties in any format or medium, without prior permission or charge, for personal research or study, educational, or not-for-profit purposes provided that:

- a full bibliographic reference is made to the original source
- a link is made to the metadata record in the Repository
- the full-text is not changed in any way

The full-text must not be sold in any format or medium without the formal permission of the copyright holders.

Serviços de Informação e Documentação, Instituto Universitário de Lisboa (ISCTE-IUL)
Av. das Forças Armadas, Edifício II, 1649-026 Lisboa Portugal
Phone: +(351) 217 903 024 | e-mail: administrador.repositorio@iscte-iul.pt
https://repositorio.iscte-iul.pt

Accepted Manuscript

Heuristic methods for the sectoring arc routing problem

Maria Cândida Mourão, Ana Catarina Nunes, Christian Prins

PII:	S0377-2217(08)00383-4
DOI:	10.1016/j.ejor.2008.04.025
Reference:	EOR 9065
To appear in:	*European Journal of Operational Research*
Received Date:	16 May 2007
Revised Date:	28 January 2008
Accepted Date:	18 April 2008

Please cite this article as: Mourão, M.C., Nunes, A.C., Prins, C., Heuristic methods for the sectoring arc routing problem, *European Journal of Operational Research* (2008), doi: 10.1016/j.ejor.2008.04.025

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

ACCEPTED MANUSCRIPT

Heuristic methods for the sectoring arc routing problem

Maria Cândida Mourão [a,d], Ana Catarina Nunes [b,d,*], Christian Prins [c]

[a] *Instituto Superior de Economia e Gestão, Universidade Técnica de Lisboa, Rua do Quelhas 6, 1200-781 Lisboa, Portugal*

[b] *ISCTE Business School, Av. das Forças Armadas, 1649-026 Lisboa, Portugal*

[c] *Université de Technologie de Troyes, Institut Charles Delaunay (FRE CNRS 2848), Industrial Systems Optimization Group, BP 2060, 10010 Troyes Cedex, France*

[d] *Centro de Investigação Operacional, Universidade de Lisboa, Portugal*

Abstract

The Sectoring–Arc Routing Problem (SARP) is introduced to model activities associated with the streets of large urban areas, like municipal waste collection. The aim is to partition the street network into a given number of sectors and to build a set of vehicle trips in each sector, to minimize the total duration of the trips. Two two-phase heuristics and one best insertion method are proposed. In the two-phase methods, phase 1 constructs the sectors using two possible heuristics, while phase 2 solves a Mixed Capacitated Arc Routing Problem (MCARP) to compute the trips in each sector. The best insertion method determines sectors and trips simultaneously. In addition to solution cost, some evaluation criteria such as imbalance, diameter and dispersion measures are used to compare algorithms. Numerical results on large instances with up to 401 nodes and 1056 links (arcs or edges) are reported and analysed.

Key words: Routing, Heuristics, Districting, Capacitated arc routing problem, Mixed graph

* Corresponding author.
Email addresses: cmourao@iseg.utl.pt (Maria Cândida Mourão), catarina.nunes@iscte.pt (Ana Catarina Nunes), christian.prins@utt.fr (Christian Prins).

Preprint submitted to European Journal of Operational Research 26th April 2008

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

1 Introduction

The *Sectoring–Arc Routing Problem* (SARP) is defined on a mixed multigraph that models the street network of a large city. A fleet of identical vehicles is based at a depot node. Each *link* (edge or arc) needs a certain time to be traversed without service by a vehicle, called *deadheading time*. Some links require service by a vehicle and have known demands and service durations. The goal is to partition these required links into a given number of sectors, each sector being assigned to one vehicle, and to solve a Mixed Capacitated Arc Routing Problem (MCARP) in each sector, in order to minimize the total duration of trips over all sectors. The SARP combines two families of classical problems: sectoring problems and arc routing problems. It is obviously NP-hard because it reduces to an NP-hard MCARP for one single sector.

Sectoring (or districting) problems consist of partitioning a large region into smaller sub-regions (sectors or districts), to facilitate the management of some activities. The definition of sectors is a strategic decision over a long-term horizon, often associated with resource allocation or facility location decisions in each sector. The resulting sectors must remain stable during a few years, to avoid costly rearrangements. Sectoring problems have many practical applications, such as political districting (Bozkaya et al., 2003; Hess et al., 1965; Hojati, 1996), sales territory design (Drexl & Haase, 1999; Hess & Samuels, 1971; Zoltners & Sinha, 1983), waste collection (Hanafi et al., 1999; Male & Liebman, 1978; Silva Gomes, 1983), postal delivery (Bodin & Levy, 1991; Levy & Bodin, 1989), meter reading (Wunderlich et al., 1992), winter maintenance operations (Kandula & Wright, 1995, 1997; Labelle et al., 2002; Muyldermans et al., 2003; Perrier et al., 2007). An interesting survey can be found in Muyldermans (2003).

In arc routing problems (ARPs), the activities of vehicles correspond to some links of a network. The aim is to build one or several trips to cover required links and minimize some measure like the total service cost. Applications include winter gritting, postal distribution, meter reading and street sweeping, as reported in Assad & Golden (1995), Eiselt et al. (1995*a*) and Eiselt et al. (1995*b*). Dror edited a book (Dror, 2000) entirely devoted to ARPs.

In the SARP, the demand of a sector often obliges the vehicle assigned to this sector to do several trips. Moreover, most street sides are modelled as arcs, but some streets whose both sides can be serviced simultaneously and in any direction are modelled as edges. Hence, the underlying ARP in each sector is a mixed CARP. To the best of our knowledge, the MCARP has been studied only by Belenguer et al. (2006) and by Mourão & Amado (2005).

In this paper, the SARP is illustrated by waste collection, but any application

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

involving the partition of the streets of a town into sectors and the definition of vehicle trips in each sector could be handled as well. The SARP is more realistic than the MCARP because waste management is easier with sectors: production statistics can be gathered for each area, drivers have to memorize a smaller subset of streets, the workload among different crews can be better balanced, etc.

Section 2 presents the SARP and introduces the required notation. The general principles of proposed solution methods are described in Section 3. Two 2-phase heuristics and one best insertion heuristic are respectively detailed in Sections 4 and 5. Some criteria to evaluate the resulting partitions are defined in Section 6. Section 7 reports computational experiments and, finally, Section 8 outlines some conclusions and remarks.

2 Problem modelling and notation

This section explains the notation summarized in Table 1 (page 6).

2.1 Mixed multigraph

The SARP can be modelled by a mixed multigraph Γ with m links, in which a subset of τ links, called *required links* or *tasks*, must be serviced. The nodes of Γ correspond to crossroads or street ends, while the links model street segments. The crew base, the dump site and a fleet of identical vehicles are located at the same depot node s. All vehicles have a limited capacity W and a maximum working time L.

A required street segment (producing some waste) is modelled either i) by one edge, if it is a two way street whose both sides can be collected simultaneously and in any direction (*zigzag collection*); ii) by two opposite arcs, if it is a two-way street with sides collected separately; iii) or as one arc, if it is an one-way street. A non-required street segment is represented as one arc or two opposite arcs, depending on its possible traversal directions. A multigraph is obtained when parallel arcs are allowed, for instance to model a wide one-way street requiring two separate traversals, when zigzag collection is too dangerous. Γ contains $\tau = \alpha + \epsilon$ required links, where α and ϵ respectively denote the number of required arcs (arc-tasks) and required edges (edge-tasks).

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

2.2 Internal network model

To simplify implementations, the mixed multigraph Γ is transformed into a fully directed multigraph $G = (N, A)$ in which each required edge is replaced by two opposite arcs, as proposed in Lacomme et al. (2001). The arcs in A are identified by indexes from 1 to $|A|$ instead of pairs of nodes, to distinguish between parallel arcs. Each arc u of A is defined by one begin-node $b(u)$, one end-node $e(u)$ and a deadheading time $d_u \geq 0$. Hence, the $\alpha + \epsilon$ required links of Γ correspond in G to a set of required arcs $R \subseteq A$, with $|R| = \alpha + 2 \cdot \epsilon$ arcs. Each arc $u \in R$ has additional attributes: a demand $q_u > 0$, a collecting time $t_u > d_u$ and a pointer $inv(u)$. Each arc-task of Γ gives one arc u in R with $inv(u) = 0$. Each edge-task is coded in R as two opposite arcs u and v, such that $inv(u) = v$, $inv(v) = u$, $q_u = q_v$, $t_u = t_v$ and $d_u = d_v$.

The pointers are used by algorithms to mark both u and v when the edge is serviced. By extension, the arcs of R are also called tasks, knowing that two arcs u and v with $inv(u) = v$ represent in fact the same edge-task of Γ. The required multigraph $G_R = (N_R, R)$ is the partial subgraph induced by N_R and R, $N_R \subseteq N$ being the set of nodes spanned by required arcs. Note that G_R is not necessarily connected.

It is assumed that all data are integer, G is strongly connected, no split collection is allowed and no demand exceeds vehicle capacity. Moreover, the maximum working time L is sufficient to allow any vehicle to leave the depot, reach any required link, collect it and come back to the depot.

2.3 Forbidden turns and proximity measures

Two measures taking forbidden turns into account are defined to appraise the proximity of any two arcs u and v of G. The first one D_{uv}, also called *D-distance*, is the minimum travel time from u to v, d_u and d_v not counted, using allowed turns only. It does not always correspond to a quickest path μ from node $i = e(u)$ to node $j = b(v)$, because of possible forbidden turns between u and the first arc of μ or between its last arc and v. To be coherent with this arc-to-arc notation, a dummy loop $\sigma = (s, s)$, called *depot-loop*, is added to A. All D-distances can be pre-computed, using an adaptation of Dijkstra's shortest path algorithm (Cormen et al., 2001).

The second measure U_{uv}, called *U-distance*, is an undirected version of the D-distance defined by equation (1), in which the inverse of an arc u is considered only when it exists, i.e., if $inv(u) \neq 0$.

ACCEPTED MANUSCRIPT

$$U_{uv} = \min\{D_{uv}, D_{u,inv(v)}, D_{inv(u),v}, D_{inv(u),inv(v)}, D_{vu}, D_{v,inv(u)}, D_{inv(v),u}, D_{inv(v),inv(u)}\} \quad (1)$$

Note that these two measures are not true metrics in the mathematical sense, even if both satisfy the triangle inequality. Indeed, the D-distance is not symmetric, $D_{uv} = 0$ if and only if u and v are consecutive arcs ($e(u) = b(v)$) and, in general, $D_{uu} \neq 0$. The U-distance is symmetric but $U_{uv} = 0$ if and only if u and v are consecutive or represent the same edge. However, these measures are well adapted to mixed networks and lead to partitions approved by decision makers. The Euclidean distance could have been used, but the actual path between two nodes or arcs is much longer when real networks with one-way streets and forbidden turns are present.

2.4 Sectors, trips and solutions

In our version of the SARP, the number K of sectors is imposed. A valid sectoring is a partition $S = \{R_1, R_2, \ldots, R_K\}$ of the tasks of R, in which R_k is the set of tasks assigned to sector k, $k = 1, 2, \ldots, K$. N_k denotes the set of nodes spanned by the tasks of R_k, called *inner nodes* of sector k. For two sectors k and l, note that $N_k \cap N_l$ may be non-empty, when there exist two adjacent or parallel arcs $u \in R_k$ and $v \in R_l$.

For a given circuit c, R_c is the set of tasks serviced and q_c the total demand of these tasks. A trip c is a special case of circuit. It can be coded as a sequence of distinct task indexes, between two copies of the depot-loop σ, and connected by implicit shortest paths. Its cost (duration) $cost(c)$ includes i) the collecting times of the tasks, ii) the deadheading times from the depot to the first task, between two consecutive tasks, and from the last task to the depot, and iii) a fixed dumping time λ to unload the waste. The deadheading times are easily obtained using the pre-computed D-distances. The trip load must not exceed vehicle capacity W.

One vehicle is assigned to each sector. It may perform several trips, subject to the time limit L. The cost of a sector k, $cost(k)$, is the total duration of its trips. A SARP solution X is defined by a partition of R into K sectors and by a set of trips for each sector. Its cost is the total cost of its sectors, i.e., $cost(X) = \sum_{k=1}^{K} cost(k)$.

ACCEPTED MANUSCRIPT

Table 1
Glossary of mathematical symbols

Network	**Arc-related symbols**	**Miscellaneous**
Γ mixed multigraph	u, v arc indexes	s, σ depot-node and depot-loop
m nb of links	b_u begin-node	W, L vehicle capacity and range
τ nb of required links	e_u end-node	λ vehicle dump time
α nb of required arcs	d_u deadheading time	k, K sector index and nb of sectors
ϵ nb of required edges	t_u service time	R_k tasks of sector k
$G = (N, A)$ directed version of Γ	q_u demand	N_k inner nodes of sector k
$R \subseteq A$ arcs coding the tasks of Γ	$inv(u)$ opposite arc	$S = \{R_1, ..., R_K\}$ partition in sectors
N_R nodes spanned by arcs of R	D_{uv} D-distance	R_c set of tasks on circuit c
$G_R = (N_R, R)$ required multigraph	U_{uv} U-distance	q_c total demand of tasks on circuit c

3 Principles of SARP algorithms

3.1 The two kinds of heuristics

The SARP is very hard since it combines one NP-hard partitioning problem and K NP-hard mixed CARPs. Moreover, the multigraph G for big cities contains hundreds of arcs and such instances are out of range for exact algorithms. This is why two heuristic methods are proposed. The first one, presented in Section 4 works in two phases: phase 1 determines the sectors while phase 2 computes vehicle routes in each sector. Two variants are obtained by selecting one of two possible heuristics for the sectoring phase. The second method, detailed in Section 5, uses a best insertion principle to build sectors and trips simultaneously.

The two-phase heuristics (TPH) and the best insertion heuristic (BIH) share common components, explained in the remainder of this section. Each sector is initialized with a seed-task, using the selection rule explained in Subsection 3.2. Then, a main loop is executed until all tasks are assigned to sectors. In each iteration, a sector is selected and receives one or several tasks. In this way, sectors are built simultaneously and balanced sectors are promoted. Sector selection is described in Subsection 3.3. The selection rule for the new tasks depends on each heuristic and will be detailed later, but the idea is to favour tasks with small demands or small collecting times.

3.2 Initialization of sectors

In both heuristics, each of the K sectors is initialized with one required arc of R, called *seed-task*. The rule used to select the seeds, called MaxDist, tries to maximize the U-distance (equation (1)) between the K seed-tasks, to favour a

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

better spread over the entire network. The first seed is the task farthest from the depot-loop. The $K-1$ other seeds are selected to maximize the minimum distance to the seeds already chosen.

3.3 Sector selection and workload estimates

Recall that the workload in a sector should not exceed the maximum working time L. When a specific task can not be added to a sector without violating this constraint, this sector is *closed*, otherwise it is *open*. The sectors to be expanded are always selected among open sectors.

In the best insertion method, the sectors and trips are built simultaneously and the exact workload of a sector is the total durations of its trips. In the two-phase methods, such an exact workload is not yet available in phase 1, because trips are built only in phase 2. In this case, an estimate must be used.

WE (*workload estimate*) is used to estimate the workload. Based on imaginary trips built in a best insertion manner, it is not a lower bound. Each time a task is added to a sector k, the workload estimate is updated according to the cost variation derived from the insertion of that task into the best position on an imaginary trip of k. These are called imaginary trips as they are valid only during phase 1. This insertion cost is computed as described in Section 5, using expressions (7) and (8).

To promote well-balanced sectors, sectors are built simultaneously. In each iteration, the sector with the smallest workload estimate (two-phase methods) or smallest exact workload (best insertion heuristic) is selected for expansion.

4 Two-phase methods

4.1 Principle

Two-phase heuristics for the SARP (TPH) mimic the behavior of most waste collection network managers. In a first phase, the required arcs are partitioned into K sectors with a workload estimate not greater than L. In a second phase, the routing in each sector is modelled as a mixed CARP, solved by fast heuristics described in Lacomme et al. (2004).

Two sectoring heuristics may be called in phase 1; they must be explained first to better understand the whole algorithm. The first one, called the *Circuit of Tasks Heuristic* (CTH), is described in 4.2. It adds to the selected sector the

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

tasks of a small demand circuit computed in a balanced graph. The rationale for this is the existence of at least one circuit in balanced graphs: once this circuit is removed, the remaining graph stays balanced, which guarantees the existence of circuits in subsequent iterations. The second sectoring heuristic, named the *Single Task Heuristic* or STH, is introduced in 4.3. Contrary to CTH, it adds one task at a time. A general algorithm for TPH is given in 4.4, with a quick description of the MCARP heuristics.

4.2 Circuit of tasks heuristic (CTH)

The principle of CTH is to compute a minimum demand circuit in a balanced graph (Mourão & Amado, 2005) and to assign all the tasks of this circuit to a sector. The aim is to build sectors in which the tasks can be linked with a small set of deadheading arcs during the routing phase. This is particularly important in networks with a lot of one-way streets.

Whenever possible, a sector k is first expanded with the tasks of a circuit including one of its inner nodes, i.e., a node spanned by the tasks already in the sector. This strategy is called *expansion from inside*. It is expected to favour sector compactness and contiguity. If circuits based on an inner node do not exist, the *expansion from outside* consists of growing the sector using a circuit built from the unassigned task closest to the sector. The tasks traversed by the circuit are added to the sector and removed from the balanced graph. The remainder of this subsection details the steps of CTH and gives the resulting algorithm structure.

Balanced graph

The minimum demand circuits are identified in a balanced graph $G_B = (N_B, A_B)$ deduced from the required graph G_R. A digraph is *balanced* if the in-degree $d^-(i)$ of each node i equals its out-degree $d^+(i)$. In such a graph, every task belongs to one circuit at least. G_B is obtained by adding deadheading copies of arcs from G to G_R, to balance it at minimum cost. This process is equivalent to a Transportation Problem (TP) (Beltrami & Bodin, 1974) which can be solved using a primal-dual algorithm (Syslo et al., 1983). Like G_R, G_B is not necessarily connected.

In the TP, each node i in G_R with $d^-(i) > d^+(i)$ is an origin with supply $d^-(i) - d^+(i)$, and each node j such that $d^+(j) > d^-(j)$ becomes a destination with demand $d^+(j) - d^-(j)$. The cost per unit transported from origin i to destination j is the duration of a shortest deadheading path from i to j in graph G, which is defined by:

ACCEPTED MANUSCRIPT

$$D^{ij} = \min\{D_{uv} | e(u) = i \wedge b(v) = j\} \quad (2)$$

Let x^*_{ij} be the optimal number of units to be transported from i to j. The balanced graph G_B includes the graph G_R and x^*_{ij} copies of each arc traversed by the shortest path from i to j in G. All these copies have no demand and keep their original deadheading time. $N_B \subseteq N$ is thus the set of nodes spanned by the arcs in A_B.

Minimum demand circuits

A minimum demand circuit in G_B with one given arc $u \in A_B$, denoted by $c(u)$, can be obtained by computing a path with minimum total demand from u to u. The modified Dijkstra's algorithm, already cited in 2.3 for the D-distances, can be used for this purpose, considering the demands as costs. Due to the use of arc-to-arc distances, a minimum demand circuit $c(i)$ with one given node $i \in N_B$ can be identified too, but this requires the computation of one circuit for each arc leaving i, i.e., if $q_{c(i)}$ denotes the total demand on $c(i)$:

$$q_{c(i)} = \min\{q_{c(u)} | u \in A_B \wedge b(u) = i\} \quad (3)$$

When determining such circuits, each task v with copies in G_B is first used as a required arc, i.e., the first circuit including v considers it as a task. Only then, non-required replicas of v might be chosen.

Sector expansion from inside – Inner node selection

When possible (if $N_k \cap N_B \neq \emptyset$), a sector k is expanded from inside, by computing a minimum demand circuit in G_B from one inner node i of N_k. Two selection rules are proposed for i. The MDC rule (*maximum demand circuit*) computes one minimum demand circuit in G_B for each inner node of k and returns the node i and the circuit $c(i)$ with maximum demand:

$$i = \arg\max\{q_{c(j)} | j \in N_k \cap N_B\} \quad (4)$$

The CST rule (*closest to seed-task*) selects in $N_k \cap N_B$ the inner node i closest to the seed-task a of sector k. Its goal is to favour the construction of compact sectors. Using the U-distances between two arcs, the distance between any arc $u \in A$ and any node $j \in N$ can be defined as:

$$U^j_u = \min\{U_{uv} | v \in A \wedge b(v) = j\} \quad (5)$$

And the node i closest to the seed-task a can be determined as follows:

$$i = \arg\min\{U^j_a | j \in N_k \cap N_B\} \quad (6)$$

ACCEPTED MANUSCRIPT

Sectors expansion from outside – Task selection

If sector k is open (expandable) and $N_k \cap N_B = \emptyset$, it is expanded from outside by determining the unassigned task b closest to its seed-task a, using again the U-distance, and by computing in G_B a minimum demand circuit containing b. The sector expansion from outside is considered as an attempt to obtain a well balanced set of sectors in terms of workload, and also to avoid deadlocks as much as possible. Workload balance turns out to be a more crucial issue than the compactness and contiguity ones in a waste collection application.

Balanced graph and sector updates

Once a minimum demand circuit c is obtained for a target-sector k, all its arcs, required or not, are deleted in G_B and we say that c is removed from G_B. In this way, the residual graph stays balanced. Nodes in G_B with no incident links are also deleted.

Then, every task u traversed by circuit c is added to sector k. We say that c is added to k, although only the tasks are added. If u represents one required edge, recall that there exists one arc v such that $inv(u) = v$ and $inv(v) = u$ for the other edge direction. Whether v is traversed by c or not, it is also added to the sector. This feature will defer to the routing phase the choice of the best service direction for the edge. However, the balanced graph must be cautiously updated if v is not on c: v must stay in G_B, but as a non-required arc.

Algorithm of CTH

CTH is summarized by Algorithm 1. Parameter *NSR* must be selected. $NSR \in \{\text{MDC}, \text{CST}\}$ is the *node selection rule* used in case of sector expansion from inside (page 9).

The balanced graph G_B is computed first. Each of the K required sectors is initialized with one seed-task and the tasks of one minimum demand circuit c containing the seed, computed in G_B. After this initialization step, a loop tries to grow the sectors until all tasks are allocated. In each iteration, the open sector k with the smallest workload estimate is first selected. If $N_k \cap N_B \neq \emptyset$, a candidate minimum demand circuit based on one inner node i is selected for an expansion from inside. Otherwise, a minimum demand circuit including the task b closest to k is computed for an expansion from outside. In both kinds of expansions, the circuit found is added to sector k and removed from G_B, if the resulting workload estimate does not exceed L. Otherwise, the expansion has failed and sector k is closed.

ACCEPTED MANUSCRIPT

This process is nested in a main loop – *repeat* – which increments K when all sectors are closed before a complete assignment of tasks, indicating that the sectoring is infeasible for the previously fixed number of sectors. However, such increments are rare if K is correctly estimated. One circuit being removed in each iteration, G_B stays balanced, which guarantees the existence of circuits until the end. Note that the sectoring process may end with a non-empty graph G_B, but containing only non-required arcs.

Algorithm 1 – Circuit of Tasks Heuristic: CTH (*NSR*)

```
compute balanced graph G'_B
repeat
   G_B = G'_B
   //Initialize K open sectors
   for k = 1 to K do
      WE(k) := 0 (initialize workload estimate for sector k)
      R_k := ∅, N_k := ∅
      select one seed-task a ∈ R using the MaxDist rule
      find in G_B a minimum demand circuit c containing a
      update R_k and N_k (add tasks from c to k, and their inverses if not null)
      update G_B (remove c from G_B) and WE(k)
   end for
   //Expand sectors
   S_o := {1, 2, ..., K} (set of open sectors)
   repeat
      select k ∈ S_o with the minimum workload estimate
      if N_k ∩ N_B ≠ ∅ then
         select i ∈ N_k using the NSR rule (expansion from inside)
         find in G_B a minimum demand circuit c containing node i
      else
         select b ∈ R ∩ A_B closest to k (expansion from outside)
         find in G_B a minimum demand circuit c containing task b
      end if
      if WE(R_k ∪ R_c) ≤ L then
         update R_k and N_k (add tasks from c to k, and their inverses if not null)
         update G_B (remove c from G_B) and WE(k)
      else
         S_o := S_o \ {k} (close sector k)
      end if
   until all tasks are assigned to sectors or S_o = ∅
   increment K if some tasks are left unassigned
until all tasks are assigned to sectors
```

ACCEPTED MANUSCRIPT

Example (CTH)

Figure 1 gives an example for three sectors with the directed graph represented in (a). Each arc has a deadheading time equal to 1 and, if it is required, its

ACCEPTED MANUSCRIPT

collecting time is 5 and its demand 1. The vehicles capacity is 3, $L = 34$ and dump time is $\lambda = 1$. The balanced graph, resulting from the solution of a transportation problem, is represented in (b), where the copies of an arc have the same arc number for simplecity. In (c), sectors 1, 2 and 3 are respectively initialized with seed-tasks 17, 1 and 19, and the corresponding minimum demand circuits. (d) shows, after a previous expansion of sectors 3 and 2, the expansion of sector 1 from inside, from node 7 (the node closest to the seed-task 17): its minimum demand circuit with arcs 15, 16, 11 and 9 is considered, and tasks 11 and 16 are assigned to sector 1. The expansion from outside is illustrated for sector 3 in (e): task 8 is selected, leading to the minimum demand circuit built with arcs 8 and 2, and task 8 is added to sector 3.

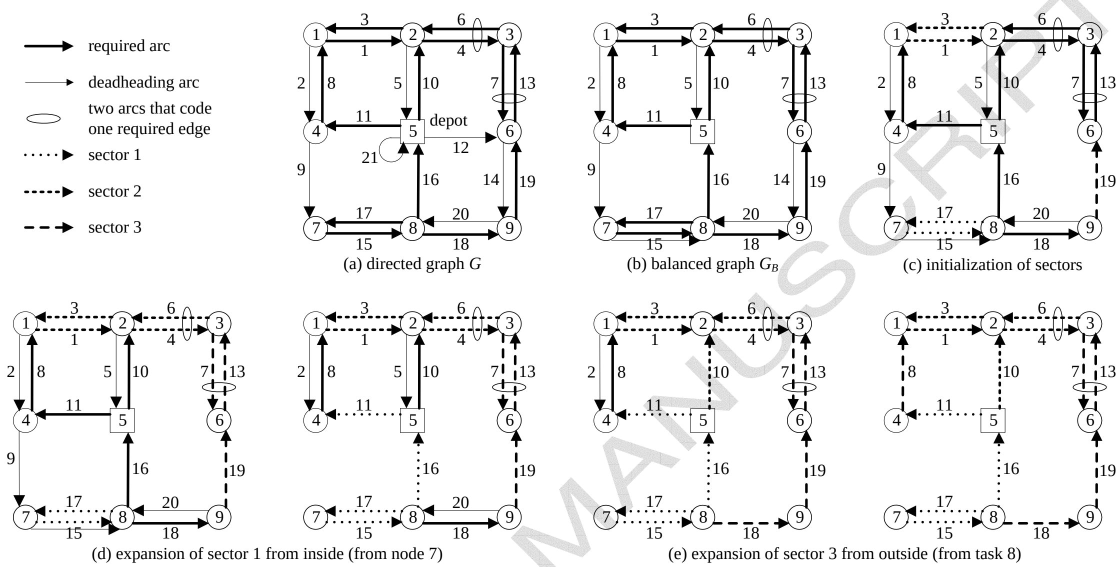

Figure 1. Example for CTH

4.3 Single task heuristic (STH)

Contrary to CTH, this simpler sectoring heuristic grows sectors by adding one single task a time. To promote sectors compactness and contiguity, the task b added to a sector in each iteration is the one that minimizes the distance to the seed-task. Two distances may be applied.

The first one is the U-distance (Sec. 2.3), already used by CTH to expand a sector from outside. However, a task looking geographically close to a sector may require a long travelling time to reach it, for instance when the network contains many one-way streets or forbidden turns. One may observe that the walking distance is small, while the deadheading distance is larger. This suggests the use of an *Euclidean-based distance*, or *E-distance*, provided node coordinates are available, which is the case in our instances and in reality.

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

Given the Euclidean distance δ^{pq} between two nodes $p, q \in V$, the E-distance between a task $u = (i, j)$, with $i, j \in V$, and a seed-task $a = (k, l)$, with $k, l \in V$, is defined as $E_{ua} = \min\{\delta^{ik}, \delta^{il}, \delta^{jk}, \delta^{jl}\}$.

STH is outlined in Algorithm 2. Parameter DK (*distance kind*) defines the distance used to select task b in each iteration. First, each sector is initialized with a seed-task. Then, sectors are expanded until all tasks are allocated. In each iteration, the open sector k with smallest workload estimate is selected. If the resulting workload satisfies L, the unassigned task closest to the seed-task is added to k, otherwise the sector is closed. Like in CTH, this process may fail if K is too small, this is why it is nested in a main loop which increments K in case of failure.

Algorithm 2 – Single Task Heuristic: STH (DK)

repeat
  //Initialize K open sectors
  $R' := R$ (set of required arcs not yet assigned to sectors)
  for $k = 1$ to K **do**
    $WE(k) := 0$ (initialize workload estimate for sector k)
    $R_k := \emptyset$, $N_k := \emptyset$
    select one seed-task $a \in R$ using the MaxDist rule
    update R_k and N_k (add a to R_k, with $inv(a)$ if not null)
    update R' (remove a from R', and $inv(a)$ if not null) and $WE(k)$
  end for
  //Expand sectors
  $S_o := \{1, 2, \ldots, K\}$ (set of open sectors)
  repeat
    select $k \in S_o$ with the minimum workload estimate
    select $b \in R'$ closest to the seed-task of k, using distance DK
    if $WE(R_k \cup \{b\}) \leq L$ **then**
      update R_k and N_k (add b to R_k, and $inv(b)$ if not null)
      update R' (remove b from R', and $inv(b)$ if not null) and $WE(k)$
    else
      $S_o := S_o \setminus \{k\}$ (close sector k)
    end if
  until $R' = \emptyset$ **or** $S_o = \emptyset$
  increment K if some tasks are left unassigned
until all tasks are assigned to sectors

4.4 General structure of TPH

The general structure of the two-phase heuristics is depicted in Algorithm 3. The two phases are nested in a main loop because they may fail for the given number of sectors K. In that case, K is incremented and the process repeated. In addition to parameters transmitted to sectoring heuristics, TPH requires

ACCEPTED MANUSCRIPT

two specific parameters *SH* and *RH*. *SH* indicates the sectoring heuristic to be used in phase 1. If *SH* = CTH, the CTH is called with its parameter *NSR* explained before. Otherwise (*SH* = STH), STH is executed with its parameter *DK*. *RH* specifies the MCARP heuristic applied to each sector in phase 2: EM, EPS or EU.

The fast heuristics EM, EPS and EU used in phase 2 are classical CARP heuristics extended to mixed networks with forbidden turns (Lacomme et al., 2004).

EM is the MCARP version of the Augment-Merge algorithm (Golden & Wong, 1981), without the augment phase. Starting from a trivial solution in which each task is served by one trip, each iteration of EM evaluates the merger (concatenation) of any two trips, subject to W and L, and merges the two trips with the largest positive saving. The process is repeated while feasible mergers are found. A tie-breaking rule is added in Belenguer et al. (2006) to improve EM: when several mergers have the same saving, priority is given to the one whose trips have the maximum discrepancy in load. This trick favours the creation of a few large trips that absorb much smaller trips. Without it, the

Algorithm 3 – Two-Phase Heuristic: TPH (*SH*, *RH*, *NSR*, *DK*)

```
input K
repeat
  //Phase 1: sectoring heuristic detailed in 4.2 and 4.3
  if SH = CTH then
    call sectoring heuristic CTH (NSR)
  else
    call sectoring heuristic STH (DK)
  end if
  //Phase 2: trip construction in each sector
  k := 1
  repeat
    case RH of
      EM: call MCARP heuristic EM on sector k
      EPS: call MCARP heuristic EPS on sector k
      EU: call MCARP heuristic EU on sector k
    end case
    if cost(k) > L then
      failure:=true
    end if
    if not failure then
      k := k + 1
    end if
  until k = K + 1 or failure
  increment K in case of failure
until not failure
```


ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

merging process stops earlier because most trips become more than half-full and cannot be merged.

EPS corresponds to the Path-Scanning CARP heuristic (Golden et al., 1983), a kind of Nearest Neighbour method completed by five rules to break ties. Five solutions are computed (one per rule) and the best one is returned at the end.

EU is an adaptation of Ulusoy's heuristic (Ulusoy, 1985). Vehicle capacity is first relaxed to generate a small set of greedy randomized giant tours (typically, 20 tours). Using a tour splitting algorithm, these circuits are converted into MCARP solutions whose the best is returned.

The following example shows that extending CARP heuristics to the MCARP is not trivial. In the CARP, defined on an undirected network, a trip A has the same cost as its inverse $\overline{A}$. Therefore, four ways of merging two trips A and B need to be tested: (A, B), $(A, \overline{B})$, $(\overline{A}, B)$ and $(\overline{A}, \overline{B})$. Indeed, a combination like $(\overline{B}, \overline{A})$, for instance, can be discarded as it is equivalent to (A, B). In the MCARP, the costs of A and $\overline{A}$ are different in general, and eight combinations must be evaluated. More implementation details can be found in Lacomme et al. (2004).

5 Best insertion heuristic (BIH)

The Best Insertion Heuristic (BIH) for the SARP builds sectors and trips simultaneously. It shares the following features with the two-phase methods: each sector is initialized with a different seed-task; the sector with minimum workload is selected for expansion in each iteration, to favour sectors balance; one task close to the sector is added to it, in order to limit the increase in workload and to keep the sector compact and contiguous, as much as possible. Here, workload estimates are no longer needed, since we know exactly the total duration of the trips in each sector.

BIH is summarized by Algorithm 4. It requires on input one parameter already used by STH: a distance kind DK (U-distance or E-distance, see 4.3). At the beginning, one seed-task a is selected to initialize each sector, using the MaxDist rule, and the best possible trip to serve a is created, i.e., the other service direction $inv(a)$ is also considered when it exists. Then, in each iteration of the expansion loop, the sector k with minimum workload is selected.

The unassigned task b closest to the seed-task of k for the distance DK is determined. The minimum insertion cost IC^* of b (or $inv(b)$ if not null) into an existing trip of k or into a new trip is evaluated. If the working time limit

ACCEPTED MANUSCRIPT

L is not exceeded, b or $inv(b)$ is inserted into the best trip found, otherwise sector k is closed.

As mentioned in 2.4, a trip r can be encoded as a sequence of task indexes, between two copies of the depot-loop σ, with implicit shortest paths computed using the D-distance. Hence, it can be denoted as $r = (v_0 = \sigma, v_1, v_2, \ldots, v_p, v_{p+1} = \sigma)$, where p is the number of tasks or *length* of r. The insertion cost or cost variation $IC(r, i, u)$ if task u is inserted in trip r after element i, $0 \leq i \leq p$, is given by equation (7).

$$IC(r,i,u) = \begin{cases} D_{\sigma,u} + t_u + D_{u,\sigma} + \lambda, & \text{if } v_i = v_{i+1} \\ D_{v_i,u} + t_u + D_{u,v_{i+1}} - D_{v_i,v_{i+1}}, & \text{otherwise} \end{cases} \quad (7)$$

Assume that T_k denotes the current set of trips in sector k and that it always includes one empty trip $(v_0 = \sigma, v_1 = \sigma)$ with length 0 and cost 0. Thanks to this trick, the best insertion cost $IC^*(k, b)$ of b in the non-empty trips of

Algorithm 4 – Best-insertion heuristic: BIH (DK)

input K
repeat
  //Initialize K open sectors
  $R' := R$ (set of required arcs not yet assigned to sectors)
  for $k = 1$ to K **do**
    $cost(k) := 0$ (initialize total duration of trips for sector k)
    select one seed-task $a \in R$ using the MaxDist rule
    create a minimum duration trip with a or $inv(a)$ if not null
    update R' (remove a from R', and $inv(a)$ if not null) and $cost(k)$
  end for
  $S_o := \{1, 2, \ldots, K\}$ (set of open sectors)
  repeat
    select the minimum cost sector $k \in S_o$
    select $b \in R'$ closest to the seed-task of k, using distance DK
    $BIC := IC^*(k, b)$; $b^* := b$
    if $inv(b) \neq 0$ **and** $IC^*(k, inv(b)) < BIC$ **then**
      $BIC := IC^*(k, inv(b))$; $b^* := inv(b)$
    end if
    if $cost(k) + BIC \leq L$ **then**
      let r^* be the trip associated with BIC and i^* the insertion position
      insert b^* in r^*, after position i^*
      update R' (remove b^* from R', and $inv(b^*)$ if not null) and $cost(k)$
    else
      $S_o := S_o \setminus \{k\}$ (close sector k)
    end if
  until $R' = \emptyset$ **or** $S_o = \emptyset$
  increment K if some tasks are left unassigned
until all tasks are inserted

ACCEPTED MANUSCRIPT

sector k or in a new trip can be found by equation (8).

$$IC^*(k, b) = \min\{IC(r, i, b)|\ r \in T_k \wedge q_r + q_b \leq W \wedge 0 \leq i \leq length(r)\} \quad (8)$$

Similarities between BIH and TPH(STH)

As described, BIH and STH have in common i) the seed-task selection rule, (ii) the selection of the sector to be expanded at each iteration (BIH trips coincide with STH trips of phase 1), and (iii) the two task selection rules (depending on DK). Indeed, for each DK option, BIH and STH produce identical sectors partition. Only the trips may differ when phase 2 of TPH(STH) is applied.

6 Partitions evaluation

Different criteria can be used to evaluate a partition (S) into sectors and SARP solutions (X). The main one is the total duration of trips over the K sectors, which is the selected objective function. The secondary criteria described below are useful complements.

Cost gap

A cutting-plane algorithm yielding a high-quality lower bound for the MCARP is described in Belenguer et al. (2006). The MCARP being a relaxation of the SARP in which no sectors are required, this bound LB is still valid for the SARP, although it might be weaker. The cost gap or deviation to the lower bound in percent for a SARP solution X is defined as $gap(X) = ((cost(X) - LB)/LB) \cdot 100$.

Imbalance

The imbalance of a solution X, over a partition S, is defined as the difference between the maximum and the minimum sector costs, i.e. $imbal(X) = \max\{cost(k)|k \in S\} - \min\{cost(k)|k \in S\}$. Knowing that a sector is allocated to a vehicle crew, small imbalance reduces the risk of conflicts between crews, since they have similar amounts of work.

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

Diameter

The diameter, together with dispersion measures, concerns the shape of sectors. The diameter of a sector k is measured by the maximum U-distance between two of its tasks: $diam(k) = \max\{U_{uv}|u, v \in k\}$. The diameter of one partition S is defined as the maximum diameter of its sectors, i.e. $diam(S) = \max\{diam(k)|k \in S\}$.

Dispersion measures

Two dispersion measures are proposed to evaluate sectors compactness, based on the mean distance to the seed-task in each sector and its standard deviation. Given a sector k and its seed-task a, the mean distance to the seed-task is $\mu_k = (1/|R_k|) \cdot \sum_{u \in R_k} U_{ua}$ and the standard deviation $\sigma_k = (1/|R_k|) \cdot \sum_{u \in R_k} (U_{ua} - \mu_k)^2$. For a given partition S, we suggest to use the mean value M_μ of the μ_k and the mean value M_σ of the σ_k, both computed over all sectors $k \in S$.

7 Computational results

7.1 Implementation, instances and parameters used

All SARP heuristics were coded in Delphi 7, a Pascal-like language, and executed on a 2.4 GHz Intel CORE 2 E6600 PC with 2 GB of RAM and Windows XP. They were evaluated on 15 MCARP instances, called *lpr* files, defined in Belenguer et al. (2006). These planar instances have $28 - 401$ nodes, $52 - 1056$ links (edges or arcs), $0 - 387$ required edges and $11 - 764$ required arcs. They comprise three groups a, b and c with five instances each. These groups respectively mimic modern towns (majority of wide two-way streets, with two sides collected independently), old historical town centres (with a majority of one-way streets) and low-traffic suburban areas (with a majority of two-way street with zigzag collection).

In all instances, the depot is either central (C) or peripheral (P), the demands are amounts of waste in kg, the deadheading and collecting times are given in seconds, and vehicle capacity and dump time are always 10 000 kg and 300 seconds, respectively. To obtain SARP instances, we just added a maximum working time $L = 21\,600\,\text{s} = 6\,\text{h}$, common to all instances, and a fixed number K of sectors, specific to each instance. The features of resulting instances are listed in Table 2: instance name, number of nodes n, links m, required links τ, required edges ϵ and required arcs α, number of sectors K and depot position.

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

Table 2
Instance features

File	n	m	τ	ϵ	α	K	Depot
lpr-a-01	28	94	52	0	52	2	C
lpr-a-02	53	169	104	5	99	2	P
lpr-a-03	146	469	304	33	271	4	C
lpr-a-04	195	651	503	34	469	7	P
lpr-a-05	321	1056	806	58	748	12	P
lpr-b-01	28	63	50	5	45	2	C
lpr-b-02	53	117	101	9	92	2	C
lpr-b-03	163	361	305	26	279	5	C
lpr-b-04	248	582	501	8	493	8	P
lpr-b-05	401	876	801	37	764	13	P
lpr-c-01	28	52	50	39	11	2	P
lpr-c-02	53	101	100	77	23	2	P
lpr-c-03	163	316	302	241	61	6	P
lpr-c-04	277	604	504	362	142	9	C
lpr-c-05	369	841	803	387	416	14	C

Three algorithms were evaluated: the two-phase heuristic (TPH) with the sectoring heuristic CTH (version called TPH(CTH)), TPH with the sectoring method STH (called TPH(STH)) and the best insertion heuristic BIH. Some components in these methods are selected using parameters, recalled in Table 3. According to the results reported in Belenguer et al. (2006) for the MCARP without sectoring (i.e., the whole network is considered as one single sector), EM outperformed the two other heuristics EPS and EU for the *lpr* files. Hence, tests were performed setting parameter RH to EM.

Table 3
Parameters used by heuristics

			Heuristic method		
Parameter	Section	Values	TPH(CTH)	TPH(STH)	BIH
RH, MCARP heuristic	4.4	EM, EPS, EU	×	×	
NSR, inner node selection	4.2	MDC, CST	×		
DK, distance for task selection	4.3	U-distance, E-distance		×	×

The combinations evaluated and analysed are listed in Table 4: combinations C1 and C2 are different versions of TPH(CTH), S1 and S2 correspond to TPH(STH), while B1 and B2 are the settings tested for BIH.

All versions were compared using partition evaluation criteria presented in Section 6: gap cost, imbalance, diameter and dispersion measures. As mentioned before, MCARP lower bounds are used to compute solution gaps. The best-known lower bounds for *lpr* instances are reported in Belenguer et al. (2006). Only average and worst values over the 15 *lpr* instances are given for most tests, because tables with results per instance would need too much

ACCEPTED MANUSCRIPT

Table 4
Combinations of parameters tested

Parameter	Value	C1	C2	S1	S2	B1	B2
RH, MCARP heuristic	EM	×	×	×	×		
NSR, inner-node selection	MDC	×					
	CST		×				
DK, distance for task selection	U-distance			×		×	
	E-distance				×		×

space. In the next three subsections, computational results are reported for the two-phase and best insertion heuristics. A global comparison is provided in Section 7.5.

7.2 Evaluation of two-phase heuristic with CTH – TPH(CTH)

The versions of TPH(CTH) with the two inner node selection rules MDC and CST presented in Subsection 4.2 correspond to combinations C1 and C2 in Table 4. Table 5 presents the mean and worst values of solution gaps, imbalance, diameter and dispersion measures.

Table 5
Impact of the inner node selection rule on TPH(CTH)

	Cost gap (%)		Imbalance		Diameter		Dispersion measures M_μ		M_σ	
	MDC	CST	MDC	CST	MDC	CST	MDC	CST	MDC	CST
Average	5,0	4,7	481,3	706,1	381,7	353,9	91,4	77,3	56,6	48,9
Worst	10,1	9,8	1149,0	2978,0	636,0	624,0	128,4	116,0	78,0	75,1

Average and worst cost gaps are slightly greater for option MDC. Imbalance decreases for MDC (−32% for the mean), while diameter increases (8%). Dispersion measures are about 16% bigger for MDC. Then, for the comparisons between the three heuristics, parameter NSR is set to CST, on the basis of its better evaluation criteria on average, and combination C2 is used.

7.3 Evaluation of two-phase heuristic with STH – TPH(STH)

The two versions of TPH(STH) obtained varying the distance rule used to select the tasks added to sectors, U-distance and E-distance presented in 4.3, correspond to S1 and S2 in Table 4. The results listed in Table 6 show that mean and worst gaps are slightly smaller for the E-distance (Edist in the table). Compared to the U-distance (Udist in the table), increases are observed

ACCEPTED MANUSCRIPT

for imbalance (7% for the mean and 52% for the worst value) and diameter maintains. Concerning dispersion measures, they are at least 2% bigger for the E-distance.

The evaluation criteria are not clearly in favour of one version. Then, U-distance is selected for comparisons with other heuristics, since the U-distance is more related with the true distance covered by the vehicles. Combination S1 is then used.

Table 6
Impact of distance for task selection on TPH(STH)

							Dispersion measures			
	Cost gap (%)		Imbalance		Diameter		M_μ		M_σ	
	Udist	Edist	Udist	Edist	Udist	Edist	Udist	Edist	Udist	Edist
Average	5,5	5,2	491,0	523,1	349,8	349,7	72,7	74,5	44,6	45,7
Worst	13,8	12,8	1013,0	1535,0	584,0	589,0	106,3	115,2	71,0	74,5

7.4 Evaluation of best insertion heuristic – BIH

As in the previous section, the two versions of BIH were obtained varying the distance for task selection (U-distance and E-distance, see Subsection 4.3), and correspond to combinations B1 and B2 in Table 4. The impact on the evaluation criteria is shown in Table 7. As referred in Section 5 (page 17), BIH defines the same sectors as TPH(STH). As a consequence, for the same *DK*, BIH and TPH(STH) produce the same values for sectors evaluation criteria: diameter and dispersion measures.

Table 7
Impact of distance for task selection on BIH

							Dispersion measures			
	Cost gap (%)		Imbalance		Diameter		M_μ		M_σ	
	Udist	Edist	Udist	Edist	Udist	Edist	Udist	Edist	Udist	Edist
Average	6.6	6.2	331.8	362.4	349.8	349.7	72.7	74.5	44.6	45.7
Worst	15.4	14.7	730.0	879.0	584.0	589.0	106.3	115.2	71.0	74.5

Cost gaps are slightly smaller for the E-distance, while imbalance and dispersion measures are bigger. U-distance is then selected for comparisons with other heuristics, and combination B1 is used.

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

Heuristics TPH(CTH), TPH(STH) and BIH are compared from combinations C2, S1 and B1, as explained. Solution costs and running times in seconds are shown in Table 8 for each instance. The three last rows indicate the mean, best and worst values of solution gaps (in %) and times (in seconds). Table 9 reports the partition evaluation criteria (imbalance, diameter and dispersion measures), with their respective mean, best and worst value at the end. In both tables, the best value among the three heuristics is emphasized in boldface.

Table 8
Solution costs and running times for the three heuristics

File	MCARP	TPH(CTH)		TPH(STH)		BIH	
	LB	Cost	Time	Cost	Time	Cost	Time
lpr-a-01	13484	**13681**	0.02	13761	0.02	13745	0.00
lpr-a-02	28052	29273	0.02	**29249**	0.02	29743	0.00
lpr-a-03	76108	78624	0.04	**78511**	0.02	80078	0.02
lpr-a-04	126941	**134310**	0.04	135635	0.03	137331	0.02
lpr-a-05	202735	**220461**	0.09	222398	0.04	224716	0.03
lpr-b-01	14835	**15187**	0.00	15320	0.02	15207	0.00
lpr-b-02	28654	**29636**	0.02	29744	0.00	30241	0.00
lpr-b-03	77837	**82545**	0.04	83449	0.02	84097	0.00
lpr-b-04	126932	**135854**	0.05	139234	0.04	140741	0.02
lpr-b-05	209791	**230330**	0.08	238692	0.05	242047	0.02
lpr-c-01	18639	**18879**	0.00	19025	0.00	19014	0.00
lpr-c-02	36339	37279	0.02	**37131**	0.02	37458	0.02
lpr-c-03	111117	116673	0.02	**116560**	0.02	118021	0.02
lpr-c-04	168441	**175807**	0.07	175943	0.02	178225	0.03
lpr-c-05	257890	**272307**	0.11	272530	0.05	276827	0.03
Mean*		**4.7**	0.04	5.5	0.02	6.6	0.01
Best*		**1.3**	0.00	2.1	0.00	1.9	0.00
Worst*		**9.8**	0.11	13.8	0.05	15.4	0.03

* Cost columns: cost gaps to *LB* in %.

The mean and worst solutions gaps are smallest for the two-phase heuristic with the CTH sectoring method: 4.7% and 9.8%. Slightly greater deviations to *LB* are achieved by STH: 5.5% and 13.8%. The less effective heuristic is BIH, with an 6.6% and 15.4%. In particular, there is no instance for which BIH outperforms the two-phase heuristics. For all heuristics, the running times are negligible: they never exceed 0.11 seconds, even on the largest graph with 1056 links.

Among the two-phase heuristics, if partition criteria are considered (Table 9), TPH(CTH) has the worst imbalance: using STH in phase 1 decreases the mean and worst value by 30% and 66%, respectively. TPH with STH has some slight

ACCEPTED MANUSCRIPT

Table 9
Imbalance, diameter and dispersion measures for heuristics CTH, STH and BIH

File	Imbalance			Diameter			Dispersion measures M_μ			Dispersion measures M_σ		
	CTH	STH	BIH	CTH	STH	BIH	CTH	STH	BIH	CTH	STH	BIH
lpr-a-01	**33**	179	145	143	**131**	**131**	46,8	**43,4**	**43,4**	27,7	**23,0**	**23,0**
lpr-a-02	141	285	**97**	269	**265**	**265**	85,5	**82,1**	**82,1**	45,3	**41,1**	**41,1**
lpr-a-03	465	711	**396**	380	**304**	**304**	89,6	**83,1**	**83,1**	49,0	**38,8**	**38,8**
lpr-a-04	679	884	**518**	429	**399**	**399**	75,1	**72,7**	**72,7**	58,9	**56,7**	**56,7**
lpr-a-05	942	753	**508**	**480**	573	573	76,0	**71,4**	**71,4**	54,3	**49,6**	**49,6**
lpr-b-01	153	194	**107**	**163**	194	194	63,4	**59,7**	**59,7**	43,0	**41,5**	**41,5**
lpr-b-02	792	**32**	141	290	**284**	**284**	109,3	**106,3**	**106,3**	48,8	**45,2**	**45,2**
lpr-b-03	714	**179**	235	385	**349**	**349**	97,4	**86,5**	**86,5**	59,5	**47,5**	**47,5**
lpr-b-04	931	861	**412**	465	**422**	**422**	97,7	**81,8**	**81,8**	60,0	**42,3**	**42,3**
lpr-b-05	2978	841	**504**	624	**584**	**584**	116,0	**104,2**	**104,2**	75,1	**71,0**	**71,0**
lpr-c-01	239	213	**108**	**111**	122	122	41,6	**40,8**	**40,8**	26,3	**25,0**	**25,0**
lpr-c-02	265	**213**	262	**258**	**258**	**258**	**75,9**	**75,9**	**75,9**	40,5	**40,2**	**40,2**
lpr-c-03	645	**350**	496	377	**368**	**368**	65,3	**63,1**	**63,1**	49,2	**47,3**	**47,3**
lpr-c-04	**713**	1013	730	**378**	428	428	**60,2**	**60,2**	**60,2**	**43,2**	45,6	45,6
lpr-c-05	901	657	**318**	**556**	566	566	59,3	**59,2**	**59,2**	**52,5**	54,6	54,6
Mean	706,1	491,0	**331,8**	353,9	**349,8**	**349,8**	77,3	**72,7**	**72,7**	48,9	**44,6**	**44,6**
Best	33,0	**32,0**	97,0	**111,0**	122,0	122,0	41,6	**40,8**	**40,8**	26,3	**23,0**	**23,0**
Worst	2978,0	1013,0	**730,0**	624,0	**584,0**	**584,0**	116,0	**106,3**	**106,3**	75,1	**71,0**	**71,0**

advantages regarding the diameter (−1% and −6%, respectively for the mean and worst values). STH also favours the dispersion measures, for which it reduces the mean and worst values of M_μ (−6% and −8%) and M_σ (−9% and −5%). Apart from solution gaps, clearly smaller with TPH(CTH), TPH(STH) is generally better for sectors quality: STH seems capable of producing more balanced and compact sectors.

If BIH is now compared to the two-phase heuristics, its strong point is a much better imbalance, with a 51% reduction for the mean and 79% for the worst value, compared to TPH(CTH). Concerning diameter and dispersion measures, the gains of BIH to TPH(CTH) are the same as with TPH(STH), resulting from the above mentioned relations between BIH and TPH(STH). Once again, it is difficult to make a decision on which one of these two heuristics gives the best sectoring, since it seems that BIH leads to better sectors balance and dispersion, while CTH favours sectors cost.

From the comparisons made between TPH(CTH), TPH(STH) and BIH, each of these heuristics has advantages and disadvantages for the criteria considered in the analysis. Even if TPH(CTH) provides better working times, it is not possible to establish an absolute hierarchy if sectoring quality is considered.

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

8 Conclusions and further remarks

In this paper, two-phase heuristics and one best insertion heuristic were proposed to integrate sector definition and trip construction in urban refuse collection networks. Classical sectoring techniques like node partitions using Voronoi diagrams or dissections of undirected planar graphs into facets cannot be used since a mixed multigraph with forbidden turns is considered.

The impact of several parameters was investigated. All heuristics are very fast, with running times less than 0.11 seconds for the bigger instances. The best heuristic, TPH(CTH), provides reasonable average deviations to MCARP lower bounds (4.7%), which is remarkable because there is no sectoring in the MCARP: the actual deviations to optimal SARP solutions are perhaps twice smaller. Although MCARP solutions have a slightly better total mileage, SARP solutions are more attractive from a practical point of view, because waste collection management is simplified.

A SARP-specific lower bound is being studied to evaluate the margin left to improvement, for a metaheuristic for instance. From the results reported, one may notice that the criteria used to evaluate partitions are frequently conflicting. This suggests a multi-objective approach based on Pareto optimality. A multi-objective genetic algorithm is already being developed. Such algorithms are usually time-consuming, because they work in parallel on a population of solutions. However, they may be envisaged since the heuristics described in this paper are finally much faster than expected on very large networks.

The presented heuristics were designed to promote workload balance, compactness, and contiguity. In general, the three heuristics produce solutions with different characteristics, as can be seen from the comparison between the evaluation criteria. Solution costs can be reduced, for instance, with local search methods that move tasks from one trip or sector to another one. The same applies to improve sectors workload balance and contiguity. Since the main research goal was to propose simple and fast heuristics which produce different types of solutions, such refinements are not described herein. The idea is to use these heuristics to generate populations of diversified solutions for a population management algorithm.

Acknowledgements

The contributions by M.C. Mourão and A.C. Nunes were supported by FCT (POCTI-ISFL-1-152) and GRICES (PESSOA program). C. Prins was founded by the Champagne-Ardenne Regional Council and the European Social Fund.

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

References

Assad, A. A. & Golden, B. L. (1995), Arc routing methods and applications, *in* M. O. Ball, T. L. Magnanti, C. L. Monma & G. L. Nemhauser, eds, 'Network Routing', Vol. 8 of *Handbooks in Operations Research and Management Science*, North-Holland, Amsterdam, pp. 375–483.

Belenguer, J. M., Benavent, E., Lacomme, P. & Prins, C. (2006), 'Lower and upper bounds for the mixed capacitated arc routing problem', *Computers & Operations Research* **33**(12), 3363–3383.

Beltrami, E. & Bodin, L. (1974), 'Networks and vehicle routing for municipal waste collection', *Networks* **4**(1), 65–94.

Bodin, L. & Levy, L. (1991), 'The arc partitioning problem', *European Journal Of Operational Research* **53**(3), 393–401.

Bozkaya, B., Erkut, E. & Laporte, G. (2003), 'A tabu search heuristic and adaptive memory procedure for political districting', *European Journal of Operational Research* **144**(1), 12–26.

Cormen, T. H., Leiserson, C. E., Rivest, R. L. & Stein, C. (2001), *Introduction to algorithms*, 2nd edn, The MIT Press, Cambridge, MA.

Drexl, A. & Haase, K. (1999), 'Fast approximation methods for sales force deployment', *Management Science* **45**(10), 1307–1323.

Dror, M., ed. (2000), *Arc Routing: Theory, Solutions and Applications*, Kluwer Academic Publishers, Dordrecht.

Eiselt, H. A., Gendreau, M. & Laporte, G. (1995*a*), 'Arc routing-problems, Part I: The chinese postman problem', *Operations Research* **43**(2), 231–242.

Eiselt, H. A., Gendreau, M. & Laporte, G. (1995*b*), 'Arc routing-problems, Part II: The rural postman problem', *Operations Research* **43**(3), 399–414.

Golden, B. L. & Wong, R. T. (1981), 'Capacitated arc routing-problems', *Networks* **11**(3), 305–315.

Golden, B. L., DeArmon, J. S. & Baker, E. K. (1983), 'Computational experiments with algorithms for a class of routing-problems', *Computers & Operations Research* **10**(1), 47–59.

Hanafi, S., Freville, A. & Vaca, P. (1999), 'Municipal solid waste collection: An effective data structure for solving the sectorization problem with local search methods', *Infor* **37**(3), 236–254.

Hess, S. W. & Samuels, S. A. (1971), 'Experiences with a sales districting model - criteria and implementation', *Management Science Series B-Application* **18**(4), 41–54.

Hess, S. W., Weaver, J. B., Siegfeldt, J. N., Whelan, J. N. & Zitlau, P. A. (1965), 'Nonpartisan political redistricting by computer', *Operations Research* **13**(6), 998–1006.

Hojati, M. (1996), 'Optimal political districting', *Computers & Operations Research* **23**(12), 1147–1161.

Kandula, P. & Wright, J. R. (1995), 'Optimal design of maintenance districts', *Transportation Research Board* **1509**, 6–14. This paper appears in Transportation Research Record No. 1509, Maintenance Management and Safety.

Kandula, P. & Wright, J. R. (1997), 'Designing network partitions to improve maintenance routing', *Journal of Infrastructure Systems* **3**(4), 160–168.

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

Labelle, A., Langevin, A. & Campbell, J. F. (2002), 'Sector design for snow removal and disposal in urban areas', *Socio-Economic Planning Sciences* **36**(3), 183–202.

Lacomme, P., Prins, C. & Ramdane-Chérif, W. (2001), A genetic algorithm for the capacitated arc routing problem and its extensions, *in* E. Boers & et al., eds, 'Applications of evolutionary computing', Vol. 2037 of *Lecture Notes in Computer Science*, Springer, Berlin, pp. 473–483.

Lacomme, P., Prins, C. & Ramdane-Cherif, W. (2004), 'Competitive memetic algorithms for arc routing problems', *Annals of Operations Research* **131**(1-4), 159–185.

Levy, L. & Bodin, L. (1989), 'The arc oriented location routing problem.', *INFOR* **27**(1), 74–94.

Male, J. W. & Liebman, J. C. (1978), 'Districting and routing for solid-waste collection', *Journal of the Environmental Engineering Division-ASCE* **104**(1), 1–14.

Mourão, M. C. & Amado, L. (2005), 'Heuristic method for a mixed capacitated arc routing problem: A refuse collection application', *European Journal of Operational Research* **160**(1), 139–153.

Muyldermans, L. (2003), Routing, districting and location for arc traversal problems, PhD thesis, Katholieke Universiteit Leuven, Belgium.

Muyldermans, L., Cattrysse, D. & Van Oudheusden, D. (2003), 'District design for arc-routing applications', *Journal of the Operational Research Society* **54**(11), 1209–1221.

Perrier, N., Langevin, A. & Campbell, J. F. (2007), 'The sector design and assignment problem for snow disposal operations', *European Journal of Operational Research*.

Silva Gomes, A. C. (1983), Sur la collecte des ordures ménagères: le problème de la sectorisation, PhD thesis, Institut National Polytechnique de Grenoble, France (in French).

Syslo, M., Deo, N. & Kowalik, J. S. (1983), *Discrete optimization algorithms with Pascal programs*, Prentice Hall, Englewood Cliffs, NJ.

Ulusoy, G. (1985), 'The fleet size and mix problem for capacitated arc routing', *European Journal of Operational Research* **22**(3), 329–337.

Wunderlich, J., Collette, M., Levy, L. & Bodin, L. (1992), 'Scheduling meter readers for southern-california-gas-company', *Interfaces* **22**(3), 22–30.

Zoltners, A. A. & Sinha, P. (1983), 'Sales territory alignment - a review and model', *Management Science* **29**(11), 1237–1256.

ACCEPTED MANUSCRIPT