Available online at www.sciencedirect.com
ScienceDirect
European Journal of Operational Research 176 (2007) 756–773
EUROPEAN JOURNAL OF OPERATIONAL RESEARCH
www.elsevier.com/locate/ejor

Discrete Optimization

The multi-depot vehicle routing problem with inter-depot routes

Benoit Crevier, Jean-François Cordeau, Gilbert Laporte *

Canada Research Chair in Distribution Management, HEC Montréal, 3000 Chemin de la Côte-Sainte-Catherine, Montréal, Canada H3T 2A7

Received 19 March 2004; accepted 19 August 2005
Available online 18 November 2005

Abstract

This article addresses an extension of the multi-depot vehicle routing problem in which vehicles may be replenished at intermediate depots along their route. It proposes a heuristic combining the adaptative memory principle, a tabu search method for the solution of subproblems, and integer programming. Tests are conducted on randomly generated instances.

Keywords: Multi-depot vehicle routing problem; Replenishment; Adaptative memory; Tabu search; Integer programming

1. Introduction

We study a variant of the multi-depot vehicle routing problem where depots can act as intermediate replenishment facilities along the route of a vehicle. This problem is a generalization of the *Vehicle Routing Problem* (VRP). The classical version of the VRP is defined on a graph $G = (V_c \cup V_d, A)$, where $V_c = \{v_1, v_2, \ldots, v_n\}$ is the customer set, $V_d = \{v_{n+1}\}$ is the depot set and $A = \{(v_i, v_j)\colon\ v_i, v_j \in V_c \cup V_d,\ i \neq j\}$ is the arc set of G. A fleet of m vehicles of capacity Q is located at v_{n+1}. Each customer has a demand q_i and a service duration d_i. A cost or travel time c_{ij} is associated with every arc of the graph. The VRP consists of determining m routes of minimal cost satisfying the following conditions: (i) every customer appears on exactly one route; (ii) every route starts and ends at the depot; (iii) the total demand of the

* Corresponding author. Tel.: +1 514 343 7575; fax: +1 514 343 7121.
E-mail address: gilbert@crt.umontreal.ca (G. Laporte).

0377-2217/$ - see front matter
doi:10.1016/j.ejor.2005.08.015

customers on any route does not exceed Q; (iv) the total duration of a route does not exceed a preset value D.

Several algorithms are available for the VRP. Because this is a hard combinatorial problem, exact methods tend to perform poorly on large size instances, which is why numerous heuristics have been developed. These include *classical heuristics* such as construction and improvement procedures or two-phase approaches, and *metaheuristics* like simulated annealing, tabu search, variable neighborhood search and evolutionary algorithms. For surveys, see Laporte and Semet [22], Gendreau et al. [15] and Cordeau et al. [8].

In some contexts, one can assign more than one route to a vehicle. The *Vehicle Routing Problem with Multiple Use of Vehicles* (VRPM) is encountered, for example, when the vehicle fleet is small or when the length of the day is large with respect to the average duration of a route. Fleischmann [12] was probably the first to propose a heuristic for this problem. It is based on the savings principle for route construction combined with a bin packing procedure for the assignment of routes to vehicles. Taillard et al. [29] have developed an adaptative memory and a tabu search heuristic, again using a bin packing procedure for assigning routes to vehicles. Other heuristics have been proposed for the VRPM, such as those of Brandão and Mercer [3,4] or Zhao et al. [34] based on tabu search or, in the ship routing context, the methods proposed by Suprayogi et al. [28] and Fagerholt [11] which create routes by solving traveling salesman problems (TSPs) and solve an integer program (Suprayogi et al. proposed a set partitioning problem) for the assignment part.

Another well-known generalization of the VRP is the *Multi-Depot Vehicle Routing Problem* (MDVRP). In this extension every customer is visited by a vehicle based at one of several depots. In the standard MDVRP every vehicle route must start and end at the same depot. There exist only a few exact algorithms for this problem. Laporte et al. [20] as well as Laporte et al. [21] have developed exact branch-and-bound algorithms but, as mentioned earlier, these only work well on relatively small instances. Several heuristics have been put forward for the MDVRP. Early heuristics based on simple construction and improvement procedures have been developed by Tillman [30], Tillman and Hering [32], Tillman and Cain [31], Wren and Holliday [33], Gillett and Johnson [16], Golden et al. [17], and Raft [24]. More recently, Chao et al. [6] have proposed a search procedure combining Dueck's [10] record-to-record local method for the reassignment of customers to different vehicle routes, followed by Lin's 2-opt procedure [23] for the improvement of individual routes. Renaud et al. [26] described a tabu search heuristic in which an initial solution is built by first assigning every customer to its nearest depot. A petal algorithm developed by the same authors [25] is then used for the solution of the VRP associated with each depot. It finally applies an improvement phase using either a subset of the 4-opt exchanges to improve individual routes, swapping customers between routes from the same or different depots, or exchanging customers between three routes. The tabu search approach of Cordeau et al. [7] is probably the best known algorithm for the MDVRP. An initial solution is obtained by assigning each customer to its nearest depot and a VRP solution is generated for each depot by means of a sweep algorithm. Improvements are performed by transferring a customer between two routes incident to the same depot, or by relocating a customer in a route incident to another depot. Reinsertions are performed by means of the GENI heuristic [13]. One of the main characteristics of this algorithm is that infeasible solutions are allowed throughout the search. Continuous diversification is achieved through the penalization of frequent moves.

The *Multi-Depot Vehicle Routing Problem with Inter-Depot Routes* (MDVRPI) has not received much attention from researchers. A simplified version of the problem is discussed by Jordan and Burns [19] and Jordan [18] who assumed that customer demands are all equal to Q and that inter-depot routes consist of back-and-forth routes between two depots. The authors transform the problem into a matching problem which is solved by a greedy algorithm. Angelelli and Speranza [2] have developed a heuristic for a version of the *Periodic Vehicle Routing Problem* (PVRP) in which replenishments at intermediate facilities are allowed. Their algorithm is based on the tabu search heuristic of Cordeau et al. [7]. A version of the problem where time windows are considered is proposed by Cano Sevilla and Simón de Blas [5]. The algorithm is based on neural networks and on an ant colony system.

Our interest in the MDVRPI arises from a real-life grocery distribution problem in the Montreal area. Several similar applications are encountered in the context where the route of a vehicle can be composed of multiple stops at intermediate depots in order for the vehicle to be replenished. When trucks and trailers are used, the replenishment can be done by a switch of trailers. Angelelli and Speranza [1] present an application of a similar problem in the context of waste collection.

Our aim is to develop a heuristic for the MDVRPI and to introduce a set of benchmark instances for this problem. The remainder of this article is organized as follows. The problem is formulated in Section 2 and the heuristic is described in Section 3. Computational results are presented in Section 4, followed by the conclusion in Section 5.

2. Formulation

The MDVRPI can be formulated as follows. Let $G = (V_c \cup V_d, A)$ be a directed graph where $V_c = \{v_1, \ldots, v_n\}$ is the customer set, $V_d = \{v_{n+1}, v_{n+2}, \ldots, v_{n+r}\}$ is a set of r depots, and $A = \{(v_i, v_j) : v_i, v_j \in V_c \cup V_d,\ i \neq j\}$ is the arc set. A demand q_i and a service duration d_i are assigned to customer i, and a cost or travel time c_{ij} is associated with the arc (v_i, v_j). Here we use the terms cost, travel time, and distance interchangeably. A homogeneous fleet of m vehicles of capacity Q is available. Let τ be the fixed duration representing the time needed for a vehicle to dock at a depot. The set of all routes assigned to a vehicle is called a *rotation* whose total duration cannot exceed a preset value D. A *single-depot* route starts and ends at the same depot while an *inter-depot* route connects two different depots.

A route h is characterized by the set of customers it contains. Hence define e_{ih} and f_{hl} coefficients as follows:

$$e_{ih} = \begin{cases} 1 & \text{if customer } i \text{ is on route } h, \\ 0 & \text{otherwise,} \end{cases}$$

$$f_{hl} = \begin{cases} 2 & \text{if route } h \text{ starts and ends at depot } l, \\ 1 & \text{if route } h \text{ starts or ends at} \\ & \quad \text{depot } l, \text{ but not both,} \\ 0 & \text{otherwise.} \end{cases}$$

Let T denote the set of all routes h satisfying

$$\sum_{i=1}^{n} e_{ih} q_i \leqslant Q, \tag{1}$$

and

$$\sum_{l=1}^{r} f_{hl} = 2. \tag{2}$$

Our formulation for the MDVRPI uses binary variables x_h^k equal to 1 if and only if route $h \in T$ is assigned to vehicle k. Also define binary variables y_l^k equal to 1 if and only if the rotation of vehicle k starts at depot l, and integer variables z_l^k equal to the number of times vehicle k arrives and leaves depot l on an inter-depot route. Define the parameter π_h as the travel duration of route h. If route h starts and ends at the same depot, then π_h is obtained by solving a TSP on the vertices of h; if h is an inter-depot route, then π_h is obtained by determining a shortest Hamiltonian path linking the two depots. In addition, define the parameter μ_h corresponding to the total duration of route h as follows:

$$\mu_h = \tau + \pi_h + \sum_{i=1}^{n} e_{ih} d_i,$$

and define the sets:

$I \subseteq T$: the set of inter-depot routes;
$\Delta(S) \subseteq T$: the set of routes starting and ending in S, where $S \subseteq V_{\mathrm{d}}$;
$\Psi(S) \subseteq T$: the set of routes with one depot in S and the other depot outside S, where $S \subseteq V_{\mathrm{d}}$.

The formulation is then

$$\text{Minimize} \quad \sum_{k=1}^{m} \sum_{h=1}^{|T|} \pi_h x_h^k, \tag{3}$$

$$\text{subject to} \quad \sum_{k=1}^{m} \sum_{h=1}^{|T|} e_{ih} x_h^k = 1, \quad i = 1, \ldots, n, \tag{4}$$

$$\sum_{l=1}^{r} y_l^k \leqslant 1, \quad k = 1, \ldots, m, \tag{5}$$

$$\sum_{h \in I} f_{hl} x_h^k - 2 z_l^k = 0, \quad k = 1, \ldots, m; \;\; l = 1, \ldots, r, \tag{6}$$

$$\sum_{h=1}^{|T|} \mu_h x_h^k \leqslant D, \quad k = 1, \ldots, m, \tag{7}$$

$$\sum_{h \in \Delta(S)} x_h^k \leqslant |\Delta(S)| \left(\sum_{h \in \Psi(S)} x_h^k + \sum_{l \in S} y_l^k \right), \quad \forall S \subseteq V_{\mathrm{d}}; \;\; k = 1, \ldots, m, \tag{8}$$

$$x_h^k \in \{0, 1\}, \quad \forall h, \forall k, \tag{9}$$

$$y_l^k \in \{0, 1\}, \quad \forall k, \forall l, \tag{10}$$

$$z_l^k \text{integer}, \quad \forall k, \forall l. \tag{11}$$

Constraints (4) guarantee that each customer will be visited exactly once, while constraints (5) state that at most one rotation will be assigned to every vehicle. Constraints (6) ensure that when a vehicle goes to an intermediate depot, it also leaves it. Constraints (7) impose a limit on the total duration of a rotation. Finally, (8) are subtour elimination constraints: given $S \subseteq V_{\mathrm{d}}$, if at least one route of vehicle k belongs to $\Delta(S)$ (in which case the left-hand side of the inequality is positive), then there must exist at least one route of that rotation in $\Psi(S)$, or else one of the depots of S has to be the starting depot of that vehicle's rotation (since otherwise the right-hand side of the inequality is equal to zero).

3. Algorithm

Because the MDVRPI is an extension of the VRP and only small instances of the VRP can be solved exactly, it is clear that one cannot expect to solve the MDVRPI with the above formulation. We have therefore opted for the development of a tabu search (TS) heuristic. This choice is motivated by the success of TS for the classical VRP and the MDVRP (see, for example, Cordeau et al. [8]).

In this section, we will describe our algorithmic approach for the MDVRPI. It is based in part on the adaptative memory principle proposed by Rochat and Taillard [27] where solutions are created by combining elements of previously obtained solutions. Here single-depot and inter-depot routes will be combined. These routes will be generated by means of a tabu search heuristic applied to three types of problems resulting from the decomposition of the MDVRPI into an MDVRP, VRPs and inter-depot subproblems. In an inter-depot problem, a set of minimal cost routes are generated on a network composed of customers and two depots, where each route starts at one depot and ends at the other. From the solutions to the subproblems just described, the underlying single-depot and inter-depot routes will be extracted and inserted in a solution pool T. Next, an MDVRPI solution will be created by the execution of a set partitioning algorithm based on the above formulation. Routes of T will therefore be selected so as to generate a set of feasible rotations in which every customer is visited. Finally, a post-optimization phase will be performed in an attempt to improve the solution.

This section successively describes the five components of our methodology: (1) the tabu search heuristic; (2) the procedure applied for the generation of a solution pool; (3) the route generation algorithm; (4) the set partitioning algorithm, and (5) the post-optimization phase. This description is followed by a pseudocode of the algorithm.

3.1. Tabu search heuristic

Our tabu search heuristic is based on the TS heuristic proposed by Cordeau et al. [7] which has proved highly effective for the solution of a wide range of classical vehicle routing problems, namely the PVRP, the MDVRP as well as extensions of these problems containing time windows [9]. We now recall the main features of this heuristic.

Neighbor solutions are obtained by removing a customer from its current route and reinserting it in another route by means of the GENI procedure [13]. In the MDVRP, insertions can be made in a vehicle associated with the same depot or with another depot. To implement tabu tenures for the VRP a set of attributes (i, k) indicating that customer i is on the route of vehicle k is first defined. Whenever a customer i is removed from route k, attribute (i, k) is declared tabu, and reinserting customer i in route k is forbidden for a fixed number of iterations θ. The MDVRP works with attributes (i, k, l), meaning that customer i is on the route of vehicle k from depot l. The tabu status of an attribute is revoked if the new solution is feasible and of lesser cost than the best known solution having this attribute.

To broaden the search, infeasible intermediate solutions are allowed by associating a penalized objective $f(s)$ to each solution s. This function is a weighted sum of three terms: the actual solution cost $c(s)$, the violation $q(s)$ of the capacity constraints, and the violation $d(s)$ of the duration constraints. The global cost function is then $f(s) = c(s) + \alpha q(s) + \beta d(s)$, where α and β are positive parameters dynamically updated throughout the search, as will be explained in Section 3.3. If the solution is feasible the two functions $c(s)$ and $f(s)$ coincide. This mechanism was first proposed by Gendreau et al. [14] in the context of the VRP. It allows the search to oscillate between feasible and infeasible solutions and enables the use of simple neighborhoods which do not have to preserve feasibility. The parameter adjustment procedure described in Section 3.3 leads to the examination of several good feasible solutions throughout the search.

A continuous diversification mechanism is applied to penalize frequent vertex moves. When a neighbor solution s' is obtained from the current solution s by adding attribute (i, k, l) and $f(s') > f(s)$, a penalty $g(s') = \gamma\sqrt{nmr}c(s)\rho_{ikl}/\lambda$ is then added to $f(s')$. In $g(s')$, the factor γ is used to calibrate the intensity of the diversification, $\sqrt{nmr}$ is a factor associated with problem size, λ is the current number of iterations, and ρ_{ikl} is a counter increased by one each time attribute (i, k, l) is added to the solution.

3.2. Generation of the solution pool

To create the solution pool T one must solve three types of subproblem: an MDVRP, a VRP, and an inter-depot subproblem. The VRP is solved for each of the r depots, and the inter-depot subproblem is solved for every pair of depots. In these two cases we only consider customers that could lead to the generation of routes likely to belong to the solution of the original problem. The VRP associated with depot B contains the following customers: (1) the $\xi n/r$ customers closest to B, where $0 < \xi \leqslant 1$ (in a planar problem this defines a circle centered at B), and (2) the customers having B as their closest depot. There are two main reasons for selecting customers defined by (2): first, there is a high probability that in an optimal solution a customer will be served by its nearest depot; also this procedure includes some remote customers that might not be frequently considered by (1) otherwise. The control parameter $\xi n/r$ limits the number of customers in each subproblem. Fig. 1 shows the circular domain of a given subproblem. Customers lying outside the domain but for which the considered depot is their closest one are represented by $\oplus$, while depots are identified by ■.

For the inter-depot subproblems, we propose a similar approach where, for a subproblem associated with depots B and C, only customers sufficiently close to both B and C are considered for inclusion in the inter-depot routes. In planar problems, a customer with coordinates (x, y) is selected if it satisfies the following inequality:

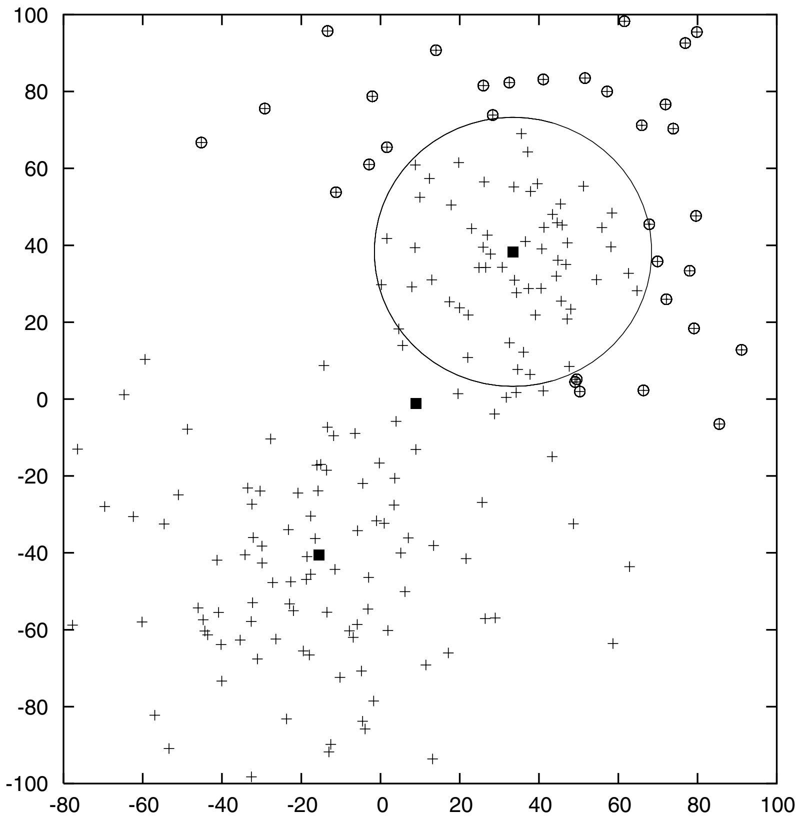

Fig. 1. Example of a circular domain.

$$\frac{((x-a)\cos(\phi)+(y-b)\sin(\phi))^2}{\varphi_1^2}+\frac{((a-x)\sin(\phi)+(y-b)\cos(\phi))^2}{\varphi_2^2}\leqslant 1.$$

This inequality defines the interior of an ellipse centered at (a,b), the midpoint of segment $\overline{BC}$. The two depots B and C will define the foci of the ellipse, while parameters φ_1 and φ_2 represent half the length of the major and minor axis, respectively. The parameter ϕ defines the angle of rotation of the ellipse. We only consider the customers lying in the elliptical domain and those, represented by $\oplus$ in Fig. 2, for which the two depots located at the foci of the ellipse are their two nearest ones.

Each VRP and inter-depot subproblem is solved σ times, each time with a new domain in order to diversify the solution pool, i.e., the set of solutions generated should cover a broad spectrum of characteristics likely to arise in an MDVRPI solution. We apply to the inter-depot subproblems the same TS procedure as for the VRP, with the exception that the distance between a customer and depot B is replaced by the distance between that customer and depot C.

3.3. Route generation algorithm

To generate an initial solution s_0 for each subproblem we apply the sweep algorithm of Cordeau et al. [7]. In the VRP, at most m routes are generated in such a way that all routes, except possibly the last one, are feasible. For the MDVRP, customers are first assigned to their closest depot and the VRP procedure is then

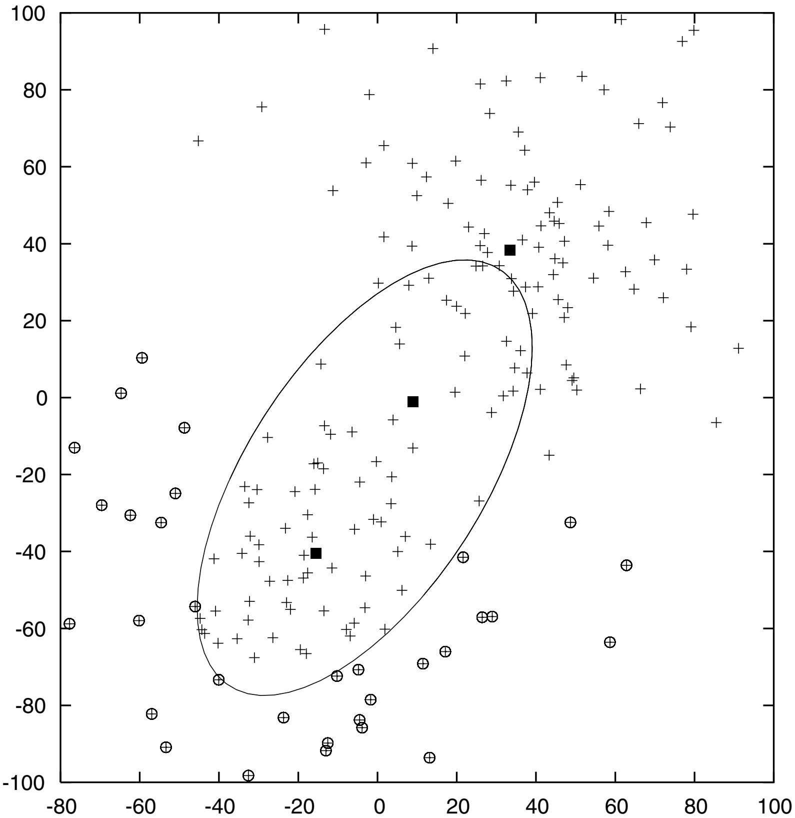

Fig. 2. Example of an elliptical domain.

executed on each depot. In this case, up to r infeasible routes can possibly be created. For the inter-depot subproblems, a similar procedure is proposed where we take into account the particular structure of the domain. In a planar problem, customers are first ordered according to the angle they make with the major axis of the ellipse. The insertion of customers in the routes is performed as for the VRP.

Because infeasible solutions are considered during the search, a mechanism is put in place in order to recover feasibility. Consider the function $f(s) = c(s) + \alpha q(s) + \beta d(s)$ and define $T(s)$ as the set of routes in solution s. The value of $f(s)$ is obtained by computing

$$c(s) = \sum_{h \in T(s)} \pi_h,$$

$$q(s) = \sum_{h \in T(s)} \left[\left(\sum_{i=1}^{n} e_{ih} q_i \right) - Q \right]^+,$$

$$d(s) = \sum_{h \in T(s)} [\mu_h - D]^+,$$

where $[x]^+ = \max\{0, x\}$. Initially set equal to 1, parameters α and β are dynamically updated throughout the search. When a solution is feasible with respect to capacity constraints, α is divided by $1 + \delta$ (where $\delta > 0$); otherwise it is multiplied by $1 + \delta$. The same applies to β with respect to duration constraints.

In order to control the cardinality of T, after solving a particular subproblem we only keep the routes associated with a feasible solution whose cost does not exceed $(1 + \epsilon)c(s^*)$, where $c(s^*)$ is the value of the best solution identified, and ϵ is a positive parameter controlling the proportion of solutions that should be kept.

3.4. Set partitioning algorithm

We now need to create a feasible solution to the MDVRPI from the pool of routes generated. We propose a set partitioning algorithm based on the mathematical formulation described in Section 2. Because T does not include all feasible routes it would be overly restrictive to impose that each customer should be visited only once. We will therefore transform the set partitioning problem presented earlier into a set covering problem. Also, to eliminate symmetric solutions, we impose that whenever $i \leqslant m$, customer i be served by vehicle k, where $k \leqslant i$. Constraints (4) can now be transformed into

$$\sum_{k=1}^{\min\{i,m\}} \sum_{h=1}^{|T|} e_{ih} x_h^k \geqslant 1, \quad i = 1, \ldots, n.$$

We ensure that each customer is covered only once in the final solution. The criterion applied to remove a customer is the largest saving obtained with the GENI heuristic.

Finally, we tighten the subtour elimination constraints by making use of the information on the duration of the routes in $\Delta(S)$ and on the maximal duration of a rotation. We first define the set $\Delta'(S)$ as follows:

1. set $\Delta'(S) := \emptyset$;
2. while $\sum_{h \in \Delta'(S)} \mu_h < D$, set $\Delta'(S) := \Delta'(S) \cup \operatorname{argmin}_{h \in \Delta(S) \setminus \Delta'(S)} \{\mu_h\}$,

and we rewrite constraints (8) as

$$\sum_{h \in \Delta(S)} x_h^k \leqslant |\Delta'(S)| \left(\sum_{h \in \Psi(S)} x_h^k + \sum_{l \in S} y_l^k \right), \quad \forall S \subseteq V_{\mathrm{d}}; \;\; k = 1, \ldots, m.$$

The idea is to use as a bound the maximal number of routes in $\Delta(S)$ that can be assigned to a vehicle without violating the duration constraint of a rotation. The effect of this is to increase the value of the lower bound by the linear programming relaxation of the set partitioning problem.

3.5. Post-optimization

In the post-optimization phase, we attempt to improve the solution composed of the routes of the set T' defined as the routes from the original pool T selected by the set partitioning algorithm. We use the tabu search heuristic previously described with three slight modifications. First, we now consider the attributes (i, h, k) indicating that customer i is on route h in the rotation of vehicle k. Next, the component $d(s)$ of $f(s)$ is modified to take into account the assignment of more than one route to a vehicle. The new definition is

$$d(s) = \sum_{k=1}^{m} \left[\left(\sum_{h \in T'} \mu_h x_h^k \right) - D \right]^+.$$

Finally, the factor used in the diversification procedure to compensate for problem size when evaluating $g(s')$ is now $\sqrt{n|T'|}$, the square root of the number of possible attributes.

The tabu search procedure is first applied to the solution corresponding to the set T' of routes. Rotations containing empty routes in the best solution are identified and empty routes are eliminated. Note that removing inter-depot routes from a rotation may lead to an infeasible solution due, for example, to the creation of subtours, or to violations of the constraints stating that a rotation has to start and end at the same depot. We have therefore devised an enumerative procedure to restore feasibility. The first step consists of modifying the remaining routes of the rotation by eliminating the edges incident to the depots. All feasible sequences of routes and depots are enumerated and the least cost rotation is identified. Obviously, the first and last depot of the sequence must be identical. Tabu search is then reapplied to the feasible solution and the post-optimization process is repeated until no empty route is generated by the tabu search.

3.6. Pseudo-codes of the algorithm

Our notation is summarized in Tables 1–4 and the main steps of the algorithm are described in the following pseudo-codes. The MDVRPI heuristic performs, in sequence, Algorithm 1, the set partitioning algorithm (Cplex based) and Algorithm 2.

Algorithm 1. Route generation

$T := \emptyset$ and $\Upsilon := \emptyset$.

Table 1
Notation used in the description of the instances

n	Number of customers
m	Number of vehicles
r	Number of depots
d_i	Service duration of customer i
q_i	Demand of customer i
s_i	Service time of customer i
D	Maximum duration of a rotation
Q	Capacity of a vehicle
τ	Fixed duration for a vehicle to dock at a depot
ω	Time required for one unit of goods to be loaded in a vehicle

Table 2
Notation used in the description of the model

Sets	
V_{c}	Customer set
V_{d}	Depot set
A	Arc set
T, T'	Route sets
I	Set of inter-depot routes
$\Delta(S)$, $\Delta'(S)$	Sets of routes starting and ending in S, where $S \subseteq V_{\mathrm{d}}$
$\Psi(S)$	Set of routes with one depot in S and the other depot outside S, where $S \subseteq V_{\mathrm{d}}$
Indices	
h	Route index
i	Customer index
k	Vehicle index
l	Depot index
Coefficients	
e_{ih}	Indicates if customer i is on route h
f_{hl}	Indicates if route h starts and/or ends at depot l
Parameters	
π_h	Travel duration of route h
μ_h	Total duration of route h
Variables	
x_h^k	Indicates if route h is assigned to vehicle k
y_l^k	Indicates if the rotation of vehicle k starts at depot l
z_l^k	Indicates the number of times vehicle k arrives and leaves depot l on an inter-depot route

Table 3
Notation used in the description of the TS algorithm

$c(s)$	Routing cost of solution s
$d(s)$	Excess duration of solution s
$f(s)$	Cost of solution s
$g(s)$	Penalty cost of solution s
$q(s)$	Excess quantity of solution s
p	Neighborhood size in GENI
s_0	Initial solution
s, s'	Solutions
s^*	Best solution identified
$T(s)$	Set of routes in solution s
$[x]^+$	$\max\{0, x\}$
α	Penalty factor for overcapacity
β	Penalty factor for overduration
γ	Factor used to adjust the intensity of the diversification
δ	Parameter used to update α and β
η	Total number of iterations to be performed
θ	Tabu duration
λ	Iteration counter
$\rho_{ik\ell}$	Number of times attribute (i, k, ℓ) has been added to the solution
σ	Number of times each VRP and inter-depot subproblem is solved
ϵ	Proportion of solutions kept when solving a VRP or an inter-depot subproblem

Table 4
Notation used in the description of the domains for VRP and inter-depot subproblems

VRP domain associated with depot B	
ξ	Percentage of customers selected in a domain
$\bar{\xi}$	Maximal percentage of customers not having B as closest depot that can be selected
Inter-depot domain associated with depots B and C	
φ_1	Half the length of the major axis
φ_2	Half the length of the minor axis
$\overline{\varphi}_1$	Maximal percentage of the distance between B and C added in the evaluation of φ_1
$[\overline{\varphi}_2, \overline{\varphi}_2']$	Percentage interval of the length of φ_1 used to determine φ_2
ϕ	Angle of rotation

Solve the MDVRP and insert the sequence of best solutions in Υ. Let s^* be the best found solution.
for all $s \in \Upsilon$ such that $c(s) \leqslant (1+\epsilon)c(s^*)$ **do**
  Extract the single-depot routes from s and insert them in T.
end for
for $l := 1,\ldots,r$ **do**
  <u>*VRP subproblems*</u>
  for $h := 1,\ldots,\sigma$ **do**
    $\Upsilon := \emptyset$ and $\Gamma := \emptyset$.
    Create a domain around depot l.
    Insert in Γ the customers in the domain and those, lying outside, for which l is their closest depot.
    Solve the VRP at depot l over customer set Γ and insert the sequence of best solutions in Υ. Let s^* be the best found solution.
    for all $s \in \Upsilon$ such that $c(s) \leqslant (1+\epsilon)c(s^*)$ **do**
      Extract the single-depot routes from s and insert them in T.
    end for
  end for

  if $l < r$ **then**
    for $l' := l+1,\ldots,r$ **do**
      <u>*Inter-depot subproblems*</u>
      for $h := 1,\ldots,\sigma$ **do**
        $\Upsilon := \emptyset$ and $\Gamma := \emptyset$.
        Create a domain around depots l and l'.
        Insert in Γ the customers in the domain and those, lying outside, for which l and l' are their two closest depots.
        Solve the inter-depot problem for depots l and l' over customer set Γ and insert the sequence of best solutions in Υ. Let s^* be the best found solution.
        for all $s \in \Upsilon$ such that $c(s) \leqslant (1+\epsilon)c(s^*)$ **do**
          Extract the inter-depot routes from s and insert them in T.
        end for
      end for
    end for
  end if
end for

Algorithm 2. Post-optimization

Apply tabu search on the solution defined by T'.
Let R be the set composed of the indexes of the vehicles having at least one empty route in their rotation.
while $R \neq \emptyset$ **do**
  for all $k \in R$ **do**
    for all non-empty routes in the rotation of k **do**
      Eliminate the edges incident to the depots.
    end for
    Identify the least-cost rotation.
  end for
  Apply tabu search on the new feasible solution.
  Update R.
end while

4. Computational results

This section presents a sensitivity analysis of the parameters as well as computational results. We first describe how the benchmark instances were generated. The sensitivity analyses follow. Finally, numerical results are presented on the randomly generated instances as well as MDVRP instances proposed by Cordeau et al. [7], adapted to the context of the MDVRPI.

4.1. MDVRPI: general case

The aim of the MDVRPI is the creation of at most m least cost feasible rotations such that each customer is visited once by a route belonging to the rotation of one of the m vehicles. Preliminary tests conducted on randomly generated instances with vehicles based at each depot have shown that inter-depot routes do not occur very often in solutions since it is rarely economical to use such routes in this type of instance. In order to create instances in which inter-depot routes will be more likely, we base all vehicles at a *central depot* and we use the remaining depots as intermediate replenishment facilities.

4.2. MDVRPI with a single location of the vehicle fleet

Instances were created as for the MDVRP studied in Cordeau et al. [7]. These instances will be used as benchmark for the calibration of the parameters.

1. Randomly generate $r - 1$ depots in the $[-50; 50] \times [-50; 50]$ domain.
2. Set $i := 1$.
3. While $i \leqslant n$, do
   (a) Randomly generate the coordinates of customer i in the $[-100; 100] \times [-100; 100]$ domain.
   (b) Let u be a random number selected in the $[0; 1]$ interval and Φ the distance between customer i and its closest depot. If $u < e^{-b\Phi}$, set $i := i + 1$.

All preceding random selections are made according to a continuous uniform distribution. In this procedure, b controls the compactness of the customer clusters. This parameter was fixed at 0.05. The

Table 5
Characteristics of MDVRPI instances

Instance	r	n	m	D	Q
$a1$	3	48	6	550	60
$b1$	3	96	4	1200	210
$c1$	3	192	5	1850	360
$d1$	4	48	5	600	80
$e1$	4	96	5	1300	230
$f1$	4	192	4	2000	380
$g1$	5	72	5	750	80
$h1$	5	144	4	1550	230
$i1$	5	216	4	2350	380
$j1$	6	72	4	800	100
$k1$	6	144	4	1650	250
$l1$	6	216	4	2500	400

coordinates of the central depot were set equal to the average coordinates of the other depots. Furthermore, let ω be the time required for one unit of goods to be loaded in the vehicle and define the total service duration of customer i as $d_i = s_i + \omega q_i$ where the service time s_i and the demand q_i are selected in the $[1; 25]$ interval according to a discrete uniform distribution. We set $\tau = 15$ and $\omega = 0.25$. Finally, D and Q were determined experimentally in order to guarantee the feasibility of the instance. Table 5 summarizes the characteristics of the instances on which the algorithm was tested.

4.3. Sensitivity analyses

Several parameters must be calibrated in order to obtain a good balance between solution quality and computational time.

4.3.1. Parameters δ, γ, θ, p and η

The tabu search heuristic requires the tuning of the five following parameters: (1) a parameter δ controlling the dynamic update of α and β; (2) a parameter γ controlling the diversification intensity; (3) θ, the number of iterations during which an attribute is considered tabu; (4) p, the neighborhood size in GENI; (5) η, the total number of iterations to be performed by the algorithm.

The best values of these parameters were extensively studied by Cordeau et al. [7,9]. A conclusion of these authors is that the best parameter values are remarkably stable over a wide range of problems such as the VRP, the MDVRP, the PVRP and variants of these problems with time windows. Setting $\delta = 0.5$, $\gamma = 0.015$, $\theta = [7.5 \log_{10} n]$ (where $[x]$ represents the integer closest to x), and $p = 3$ is recommended. Because our problem has a structure similar to that of the VRP and the MDVRP we have used the same settings for δ, γ, θ and p. The number η of iterations is left to the user.

4.3.2. Parameters ξ, $\overline{\xi}$, φ_1, $\overline{\varphi}_1$, φ_2, $(\overline{\varphi}_2, \overline{\varphi}'_2)$, σ and ϵ

Having fixed most of the tabu search parameters, we now discuss those defining the VRP and inter-depot domains. The parameter ξ shaping the boundary of the VRPs is selected, as mentioned earlier, in the]0;1] interval according to a continuous uniform distribution. However, for a VRP associated with depot B, since customers for which the depot is their closest one are considered, the value of ξ is chosen so that up to $\overline{\xi}$

percent of the customers closest to B, but not having B as their closest depot, can be selected. The parameters φ_1 and φ_2 defining the region in the inter-depot subproblems are determined as follows. Consider the problem associated with depots B and C and let c_{BC} be the distance between these two depots. Since in a planar instance φ_1 corresponds to half the length of the major axis going through the depots, at least half the distance between the two depots will be assigned to it. Also, we add a random value selected in the $[0; \overline{\varphi_1} c_{BC}]$ interval, where $0 \leqslant \overline{\varphi_1} \leqslant 1$, so that $\overline{\varphi_1} c_{BC}$ represent a certain percentage of c_{BC}. Consequently $\varphi_1 := c_{BC}/2 + u\overline{\varphi_1} c_{BC}$, where $u \in [0; 1]$ is a randomly chosen number according to a continuous uniform distribution. Finally, once φ_1 is set, the length of the half-axis φ_2 will be randomly selected in the $[\overline{\varphi}_2 \varphi_1; \overline{\varphi}'_2 \varphi_1]$ interval with $0 \leqslant \overline{\varphi}_2 < \overline{\varphi}'_2 \leqslant 1$ which defines φ_2 as a certain percentage of φ_1. Therefore, $\varphi_2 := u\varphi_1$, where $u \in [\overline{\varphi}_2; \overline{\varphi}'_2]$. The values of $\overline{\varphi}_1$, $\overline{\varphi}_2$ and $\overline{\varphi}'_2$ will be discussed next. The parameters were analyzed jointly since tight relations can be expected among them.

To determine the value assignment to the parameters just described, a preliminary version of the algorithm was tested on the instances described in Section 4.2. The parameters σ and ϵ, defining respectively the number of times each VRP or inter-depot subproblem is solved, and the proportion of solutions to keep when solving a subproblem, were set equal to 10 and 0.01. These values were selected so as to create a diversity of solutions possessing characteristics that might arise in good MDVRPI solutions. The value of η was set equal to 7500 for the VRP and inter-depot subproblems, and to 15,000 for the MDVRP. In the post-optimization phase, preliminary tests have shown that recursive calls to the tabu search heuristic require a decreasing number of iterations to adequately explore the solution space. That is why η was fixed at 45,000 in the first call and at 30,000 in the subsequent calls.

Tests have shown that allowing the selection of large inter-depot domains leads, on average, to better solutions. Larger domains generate larger clusters of customers, resulting in the creation of superior route structures. However, because more customers are considered each time a subproblem is solved, more routes are generated and the size of T increases. Two combinations of parameters stand out: $\overline{\xi} = 0.2$, $\overline{\varphi}_1 = 1$, $(\overline{\varphi}_2, \overline{\varphi}'_2) = (0.5, 1)$ and $\overline{\xi} = 0.6$, $\overline{\varphi}_1 = 1$, $(\overline{\varphi}_2, \overline{\varphi}'_2) = (0.25, 0.5)$. The first parameter combination generates better results on average but requires more computational time on large size instances. The second one produces slightly worse results on average but yields more stable computational times. Further tests have shown that the second parameter combination, combined with $\epsilon = 0.01$ and $\sigma = 12$ yields the best results.

4.4. Results on benchmark instances

We now present the results of tests conducted on the 12 MDVRPI instances of Table 5 and on a set of instances derived from the MDVRP instances of Cordeau et al. [7]. These instances and the best known solutions are available at http://www.hec.ca/chairedistributique/data/. They vary in size from $n = 48$ to $n = 288$, which is consistent with the size of the benchmark instances commonly used for the MDVRP (see [7]). The algorithm was coded in C and the set covering problem was solved with CPLEX 7.1. Tests were run on a Prosys, 2 GHz computer. All computations were performed in double precision arithmetic and the final results are reported with two significant digits after the decimal point.

4.4.1. Results on randomly generated instances

The algorithm was executed with the following parameter values: $\overline{\xi} = 0.6$, $\overline{\varphi}_1 = 1$, $(\overline{\varphi}_2, \overline{\varphi}'_2) = (0.25, 0.5)$, $\sigma = 12$, $\epsilon = 0.01$. The number of iterations η is set equal to 15,000 in the solution of the MDVRP, and to 5000 in the VRP and inter-depot subproblems since tests have shown that a sufficient exploration of the solution space is performed with these values. Furthermore, the first two executions of the tabu search heuristic in the post-optimization phase seemed to be the most crucial since it is during those calls that the structure of the solution is mostly modified. We have therefore set $\eta = 35{,}000$ in the first call, 25,000 in

Table 6
Solutions obtained on MDVRPI instances

Instance	$\overline{c(s^*)}$	$c(s_b)$	%	$\%_b$	$\%_w$	$\overline{\lvert T\rvert}$	$\bar{t}_{gen}$	$\bar{t}_{spa}$	$\bar{t}_{po}$	$\bar{t}_{tot}$
*a*1	1211.28	1179.79	2.67	2.00	3.10	317.90	2.44	1.40	0.74	4.58
*b*1	1232.67	1217.07	1.28	0.00	2.43	555.30	6.36	0.23	2.58	9.17
*c*1	1893.01	1886.15	0.36	0.11	0.66	1340.50	23.08	3.04	10.10	36.22
*d*1	1076.31	1059.43	1.59	0.00	3.48	350.90	2.57	4.93	1.05	8.55
*e*1	1311.60	1309.12	0.19	0.00	1.89	469.10	7.46	0.49	5.57	13.52
*f*1	1601.54	1576.33	1.60	1.01	2.18	1107.50	21.18	4.27	15.95	41.41
*g*1	1202.00	1181.13	1.77	0.83	2.79	641.40	5.01	48.33	1.88	55.22
*h*1	1598.51	1547.25	3.31	1.26	5.83	866.30	13.97	8.60	9.49	32.07
*i*1	1976.11	1927.99	2.50	0.92	3.27	1334.100	25.81	6.09	19.11	51.01
*j*1	1161.77	1120.65	3.67	2.12	6.56	916.10	7.39	48.99	2.53	58.90
*k*1	1618.45	1586.92	1.99	0.00	3.79	1568.50	17.37	39.55	7.70	64.61
*l*1	1917.08	1884.92	1.71	0.68	3.22	2002.80	34.20	48.38	21.69	104.27
Average	1483.36	1456.40	1.89	0.74	3.27	955.87	13.90	17.86	8.20	39.96

the second one and 15,000 in the following calls. The heuristic was executed ten times on each of the randomly generated instances. Table 6 provides results. The column headings are defined as follows:

- $\overline{c(s^*)}$, the average value of the solutions over the ten runs;
- $c(s_b)$, the value of the best solution identified throughout the sensitivity analyses;
- %, the gap in percentage between the average value of the solutions and the best known solution;
- $\%_b$, the percentage gap between $c(s_b)$ and the best solution found among the ten solutions generated;
- $\%_w$, the percentage gap between $c(s_b)$ and the worst solution identified among the ten solutions obtained;
- $\overline{|T|}$, the average cardinality of the set T;
- $\bar{t}_{gen}$, the average time, in minutes, spent on route generation;
- $\bar{t}_{spa}$, the average time required by the set partitioning algorithm;
- $\bar{t}_{po}$, the average time of the post-optimization phase;
- $\bar{t}_{tot}$, the average total computational time.

The average is computed and presented in bold at the end of the corresponding column.

We observe that the average percentage gap between the average and best solution values for each instance is 1.89% and the average percentage gap between the overall best and worst values over the ten runs on each instance is 3.27%, which is reasonable given the many random components of the algorithm. More stability can be reached through the use of higher values for σ or ϵ and at the expense of longer computational times. Computational times are closely related to instance size and to the tightness of constraints (1) and (7). For a given instance, the variability in computational time is mostly explained by the time spent in the solution of the set covering problem.

Fig. 3 depicts the best solution obtained for instance *c*1. The central depot is identified by ● and the other depots by ■. Every vehicle route is represented by a different line type. We can distinguish four rotations, each composed of two inter-depot routes.

4.4.2. Results on the Cordeau, Gendreau and Laporte instances

Ten new instances were generated from those proposed by Cordeau et al. [7] for the MDVRP. These instances contain between 48 and 288 customers as well as four or six depots. In order to adapt the instances to the MDVRPI, a central depot was added at the centroïd of the other depots. The resulting instances contain either five or seven depots. The values of D and Q were determined experimentally to

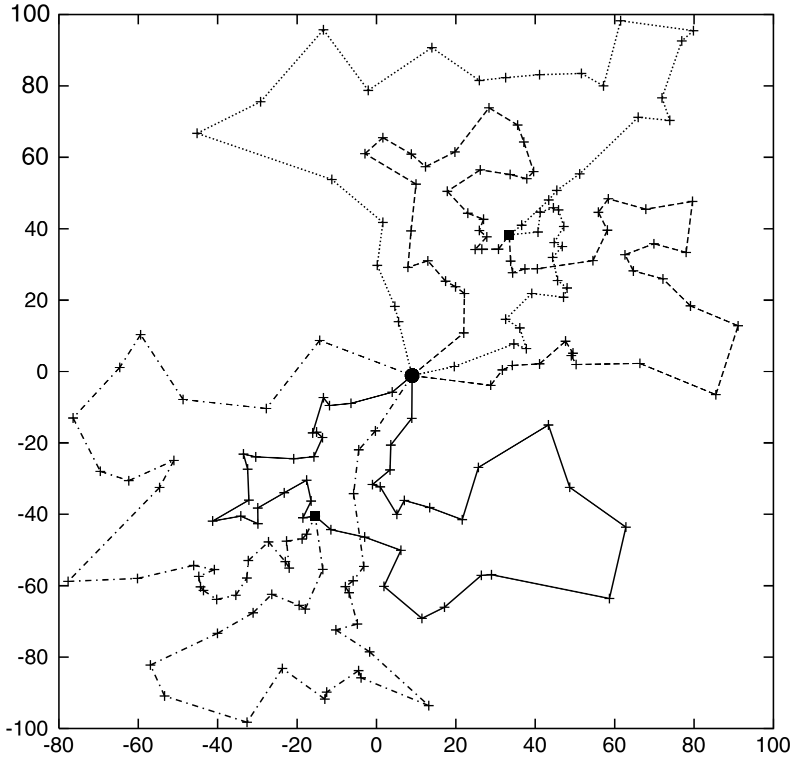

Fig. 3. Best solution for instance *c*1.

Table 7
New MDVRPI instances

Instance	r	n	m	D	Q
*a*2	5	48	4	600	150
*b*2	5	96	4	1150	200
*c*2	5	144	4	1700	250
*d*2	5	192	3	2250	300
*e*2	5	240	3	2800	350
*f*2	5	288	3	3350	400
*g*2	7	72	4	950	175
*h*2	7	144	4	1800	250
*i*2	7	216	3	2650	325
*j*2	7	288	3	3500	400

guarantee feasibility. Table 7 summarizes the main characteristics of the modified instances and Table 8 presents the results. Again, the algorithm was executed ten times on each instance. The behaviour of the heuristic is similar to that observed on the randomly generated instances. We can, however, note that a substantial computational time is required for the last two instances.

Table 8
Solutions obtained on new MDVRPI instances

Instance	$\overline{c(s^*)}$	$c(s_b)$	%	$\%_b$	$\%_w$	$\overline{\lvert T \rvert}$	$\bar{t}_{gen}$	$\bar{t}_{spa}$	$\bar{t}_{po}$	$\bar{t}_{tot}$
*a*2	1005.16	997.94	0.72	0.23	1.30	259.90	3.26	1.71	1.41	6.39
*b*2	1333.20	1307.28	1.98	0.00	2.95	683.10	7.89	2.64	4.19	14.72
*c*2	1792.46	1747.61	2.57	0.22	4.41	1100.70	16.98	35.60	9.10	61.68
*d*2	1898.21	1871.42	1.43	0.30	2.41	1494.90	22.62	6.12	11.80	40.54
*e*2	1995.75	1942.85	2.72	1.61	3.35	1539.20	27.45	26.38	19.95	73.78
*f*2	2312.15	2284.35	1.22	0.62	1.67	2637.20	44.81	92.22	25.19	162.22
*g*2	1185.93	1162.58	2.01	0.00	5.36	714.10	8.10	18.53	2.89	29.51
*h*2	1611.75	1587.37	1.54	0.38	3.11	1787.20	21.08	130.01	9.70	160.79
*i*2	1998.20	1972.00	1.33	0.34	2.51	3950.90	58.39	247.13	16.89	322.41
*j*2	2325.18	2294.06	1.36	0.39	1.92	3277.40	61.99	162.17	32.70	256.85
Average	1745.80	1716.75	1.69	0.41	2.90	1744.46	27.26	72.25	13.38	112.89

5. Conclusion

We have considered the MDVRPI, an extension of the MDVRP in which vehicles can be replenished at intermediate depots along their route. This problem has applications notably in distribution and collection management. The presence of intermediate depots adds considerable difficulties to the standard MDVRP. The MDVRPI has received relatively little attention in the past. We have proposed a three-phase methodology based on adaptative memory and tabu search for the generation of a set of routes, and on integer programming in the execution of a set partitioning algorithm for the determination of least cost feasible rotations. Finally, a post-optimization phase was applied to the routes. The algorithm was tested on randomly generated instances and on benchmark instances derived from those proposed for the MDVRP by Cordeau et al. [7]. The heuristic exhibits a robust behaviour and reasonably fast running times. Because the MDVRPI is a new problem, no previous statistics are available but we hope our results will enable other researchers to produce comparative results.

Acknowledgements

This work was partly supported by the Natural Sciences and Engineering Research Council of Canada under grants 227837-00 and OGP0039682. This support is gratefully acknowledged. Thanks are due to three referees for their valuable comments.

References

[1] E. Angelelli, M.G. Speranza, The application of a vehicle routing model to a waste-collection problem: Two case studies, Journal of the Operational Research Society 53 (2002) 944–952.
[2] E. Angelelli, M.G. Speranza, The periodic vehicle routing problem with intermediate facilities, European Journal of Operational Research 137 (2002) 233–247.
[3] J.C.S. Brandão, A. Mercer, A tabu search algorithm for the multi-trip vehicle routing and scheduling problem, European Journal of Operational Research 100 (1997) 180–191.
[4] J.C.S. Brandão, A. Mercer, The multi-trip vehicle routing problem, Journal of the Operational Research Society 49 (1998) 799–805.
[5] F. Cano Sevilla, C. Simón de Blas, Vehicle routing problem with time windows and intermediate facilities, in: S.E.I.O.'03 Edicions de la Universitat de Lleida, 2003, pp. 3088–3096.

[6] I.-M. Chao, B.L. Golden, E.A. Wasil, A new heuristic for the multi-depot vehicle routing problem that improves upon best-known solutions, American Journal of Mathematical and Management Sciences 13 (1993) 371–406.
[7] J.-F. Cordeau, M. Gendreau, G. Laporte, A tabu search heuristic for periodic and multi-depot vehicle routing problems, Networks 30 (1997) 105–119.
[8] J.-F. Cordeau, M. Gendreau, G. Laporte, J.-Y. Potvin, F. Semet, A guide to vehicle routing heuristics, Journal of the Operational Research Society 53 (2002) 512–522.
[9] J.-F. Cordeau, G. Laporte, A. Mercier, A unified tabu search heuristic for vehicle routing problems with time windows, Journal of the Operational Research Society 52 (2001) 928–936.
[10] G. Dueck, New optimization heuristics: The great deluge algorithm and the record-to-record travel, Journal of Computational Physics 104 (1993) 86–92.
[11] K. Fagerholt, Designing optimal routes in a liner shipping problem, Maritime Policy & Management 31 (2004) 259–268.
[12] B. Fleischmann, The vehicle routing problem with multiple use of vehicles, Working paper, Fachbereich Wirtschaftswissens-chaften, Universität Hamburg, 1990.
[13] M. Gendreau, A. Hertz, G. Laporte, New insertion and postoptimization procedures for the traveling salesman problem, Operations Research 40 (1992) 1086–1094.
[14] M. Gendreau, A. Hertz, G. Laporte, A tabu search heuristic for the vehicle routing problem, Management Science 40 (1994) 1276–1290.
[15] M. Gendreau, G. Laporte, J.-Y. Potvin, Metaheuristics for the capacitated vehicle routing problem, in: P. Toth, D. Vigo (Eds.), The Vehicle Routing Problem, Monographs on Discrete Mathematics and Applications, SIAM, Philadelphia, 2002, pp. 129–154.
[16] B.E. Gillett, J.G. Johnson, Multi-terminal vehicle-dispatch algorithm, Omega 4 (1976) 711–718.
[17] B.L. Golden, T.L. Magnanti, H.Q. Nguyen, Implementing vehicle routing algorithms, Networks 7 (1977) 113–148.
[18] W.C. Jordan, Truck backhauling on networks with many terminals, Transportation Research 21B (1987) 183–193.
[19] W.C. Jordan, L.D. Burns, Truck backhauling on two terminal networks, Transportation Research 18B (1984) 487–503.
[20] G. Laporte, Y. Nobert, D. Arpin, Optimal solutions to capacitated multidepot vehicle routing problems, Congressus Numerantium 44 (1984) 283–292.
[21] G. Laporte, Y. Nobert, S. Taillefer, Solving a family of multi-depot vehicle routing and location-routing problems, Transportation Science 22 (1988) 161–172.
[22] G. Laporte, F. Semet, Classical heuristics for the capacitated vehicle routing problem, in: P. Toth, D. Vigo (Eds.), The Vehicle Routing Problem, Monographs on Discrete Mathematics and Applications, SIAM, Philadelphia, 2002, pp. 109–128.
[23] S. Lin, Computer solutions of the traveling salesman problem, Bell System Technical Journal 44 (1965) 2245–2269.
[24] O.M. Raft, A modular algorithm for an extended vehicle scheduling problem, European Journal of Operational Research 11 (1982) 67–76.
[25] J. Renaud, G. Laporte, F.F. Boctor, An improved petal heuristic for the vehicle routing problem, Journal of the Operational Research Society 47 (1996) 329–336.
[26] J. Renaud, G. Laporte, F.F. Boctor, A tabu search heuristic for the multi-depot vehicle routing problem, Computers & Operations Research 23 (1996) 229–235.
[27] Y. Rochat, É.D. Taillard, Probabilistic diversification and intensification in local search for vehicle routing, Journal of Heuristics 1 (1995) 147–167.
[28] Suprayogi, H. Yamato, Iskendar, Ship routing design for the oily liquid waste collection, Journal of the Society of Naval Architects of Japan 190 (2001) 325–335.
[29] É.D. Taillard, G. Laporte, M. Gendreau, Vehicle routing with multiple use of vehicles, Journal of the Operational Research Society 47 (1996) 1065–1070.
[30] F.A. Tillman, The multiple terminal delivery problem with probabilistic demands, Transportation Science 3 (1969) 192–204.
[31] F.A. Tillman, T.M. Cain, An upperbound algorithm for the single and multiple terminal delivery problem, Management Science 18 (1972) 664–682.
[32] F.A. Tillman, R.W. Hering, A study of look-ahead procedure for solving the multiterminal delivery problem, Transportation Research 5 (1971) 225–229.
[33] A. Wren, A. Holliday, Computer scheduling of vehicles from one or more depots to a number of delivery points, Operational Research Quarterly 23 (1972) 333–344.
[34] Q.-H. Zhao, S.-Y. Wang, K.-K. Lai, G.-P. Xia, A vehicle routing problem with multiple use of vehicles, Advanced Modeling and Optimization 4 (2002) 21–40.