Publié par :	Faculté des sciences de l'administration
Published by :	Université Laval
Publicación de la :	Québec (Québec) Canada G1K 7P4
	Tél. Ph. Tel. : (418) 656-3644
	Fax : (418) 656-2624
Édition électronique :	Céline Frenette
Electronic publishing :	Vice-décanat à la recherche et au développement
Edición electrónica :	Faculté des sciences de l'administration
Disponible sur Internet :	http ://www.fsa.ulaval.ca/rd
Available on Internet	rd@fsa.ulaval.ca
Disponible por Internet :	

DOCUMENT DE TRAVAIL 2000-015

A SWEEP BASED ALGORITHM FOR THE FLEET SIZE AND MIX VEHICLE ROUTING PROBLEM

Jacques Renaud, Fayez Boctor

Centre de recherche sur les technologies de l'organisation réseau (CENTOR)

Version originale :	ISBN – 2-89524-108-2
Original manuscript :	ISBN -
Version original :	ISBN -
Série électronique mise à jour :	04-2000
One-line publication updated :	
Seria electrónica, puesta al dia	

A SWEEP BASED ALGORITHM FOR THE FLEET SIZE AND MIX VEHICLE ROUTING PROBLEM

Jacques Renaud[1,2] and Fayez F. Boctor[2,3]

1 Télé-Université, 2600 boulevard Laurier, Case postale 10700, Sainte-Foy, Canada, G1V 4V9. Email : jrenaud@teluq.uquebec.ca, Tel : (418) 657-2262, Fax : (418) 657-2094.
2 Centre de recherche sur les technologies de l'organisation réseau (CENTOR), Université Laval, Sainte-Foy, Canada, G1K 7P4
3 Faculté des Sciences de l'administration, Université Laval, Québec, Canada, G1K 7P4. Email: fayez.boctor@fsa.ulaval.ca

April, 2000

A SWEEP BASED HEURISTIC FOR THE FLEET SIZE AND MIX VEHICLE ROUTING PROBLEM

Abstract

This paper presents a new sweep based heuristic for the fleet size and mix vehicle routing problem. This problem involves two kinds of decisions: the selection of a mix of vehicles among the available vehicle types and the routing of the selected fleet. The proposed algorithm first generates a large number of routes which are serviced by one or two vehicles. The selection of routes and vehicles to be used is then made by solving to optimality, in polynomial time, a set-partitioning problem having a special structure. Results on a set of benchmark test problems show that the proposed heuristic produces excellent solutions in short computing times. Having a fast but good solution method is needed for transportation companies that rent a significant part of their fleet and consequently can take advantage of frequent changes in fleet composition. Finally, the proposed heuristic produced the best-known solution for some of the test problems. These new best-known solutions are reported.

Key words*:* Vehicle routing problem, fleet selection, sweep based heuristic

Résumé

Ce papier présente une nouvelle heuristique de balayage pour l'élaboration de tournées de véhicules en présence d'une flotte hétérogène. Ce problème fait intervenir simultanément deux types de décision: le choix du nombre de véhicules de chaque type à utiliser puis l'élaboration des tournées individuelles des véhicules choisis. L'heuristique proposée commence par générer un grand nombre de routes devant être desservies par un ou deux véhicules. La sélection des routes et des véhicules à utiliser se fait ensuite à l'aide d'un problème de partition dont la structure particulière en permet la résolution optimale en temps polynomial. Les résultats sur des problèmes tests montrent que l'heuristique proposée produit d'excellentes solutions dans des temps relativement courts. Pouvoir résoudre rapidement ce genre de problèmes est particulièrement avantageux pour les compagnies qui louent une partie significative de leur flotte et par conséquent peuvent tirer profit des changements fréquents de la composition de leurs flottes. Nous rapportons également des nouvelles meilleures solutions pour certains problèmes tests.

Mots clés : Problème de tournées de véhicules, composition de flotte, heuristique de balayage

1. Introduction

The *fleet size and mix vehicle routing problem* (FSMVRP), also called *vehicle fleet mix problem* or *fleet size and composition vehicle routing problem* (Taillard 1999), involves two basic decisions: the composition of a heterogeneous vehicle fleet and the routing of this fleet. The vehicle fleet can be composed of vehicles having different capacities as well as different fixed and variable costs. The objective is to minimize the total cost which is composed of vehicle fixed utilization costs and of variable traveling costs. This objective can be achieved by finding the optimal mix of vehicles and by determining the associated routes while satisfying the problem constraints.

Mathematically, the problem may be defined as follows. Let $G = (V, A)$ be a graph where $V = \{v_0, ..., v_n\}$ is the vertex set and $A = \{(v_i,v_j): v_i, v_j \in V, i \neq j\}$ is the arc set. Vertex v_0 represents a *depot* where M different vehicle types are based. Each vertex $v_i \in V \backslash \{v_0\}$ corresponds to a *customer* and is associated with a non-negative demand q_i and a service time s_i. In the version of the problem under consideration, all arcs are undirected, i. e. they are *edges*. Each edge (v_i,v_j) is associated with a non-negative cost, c_{ij}, representing its *travel cost* and a non-negative time, t_{ij}, representing its *travel time*. In addition, F_k, Q_k and T_k represent respectively the *vehicle fixed cost*, the *vehicle capacity* and the *maximum travel time* for vehicle type $k = 1, ..., M$. We assume that $F_{k_1} < F_{k_2}$ implies $Q_{k_1} < Q_{k_2}$, that vehicle types are numbered in ascending order of F_k and that we can use any number of vehicles of type k. The FSMVRP is to determine a mix of vehicles as well as their routes such that: (1) routes start and end at the depot; (2) each customer is visited exactly once; (3) the total demand of a route does not exceed the capacity of the vehicle type used; (4) the total duration of each route (including travel and service times) does not exceed the maximal traveling time T_k of the vehicle type used; and (5) the sum of fixed and variable costs are minimized.

In this paper we propose a new sweep based heuristic that has produced highly competitive results on a set of benchmark problems. This composite heuristic can solve both Euclidean planar problems and non Euclidean problems as well which is not the case of some of the recently published heuristics. In addition, the proposed heuristic produces better results than comparable composite heuristics and results very close to those obtained by the best-published, Tabu-based heuristic algorithms.

The remainder of this paper is organized as follows. In Section 2, we review the main algorithms for the FSMVRP. Our heuristic is described in Section 3, followed by our computational results in Section 4 and conclusions in Section 5.

2. Literature review

The FSMVRP is clearly NP-hard as it reduces to the *vehicle routing problem* (VRP) when $M = 1$. This latter problem has generated a considerable amount of research over the last three decades (see the review by Laporte (1992b) and the bibliography by Laporte and Osman (1995)). The best existing optimal algorithms for the VRP appear to be those of Cornuéjols and Harche (1993), and of Hadjiconstantinou, Christofides and Mingozzi (1995), and can rarely solve problem instances involving more than 50 customers. The best heuristics for the VRP appear to be the tabu search based algorithms of Taillard (1993), Osman (1993) and Gendreau, Hertz and Laporte (1994), and the improved petal heuristic of Renaud, Boctor and Laporte (1996a).

In spite of its practical importance, the FSMVRP has attracted less research effort. Gould (1969) developed a linear program for a problem version where only round trips between the depot and each customer are considered. Woods and Harris (1979) also addressed this problem by using a simulation approach. Etezadi and Beasley (1983) presented a formulation where vehicles may visit many customers. Golden, Assad, Levy and Gheysens (1984) presented a mathematical formulation of the FSMVRP.

Because of its complexity, the research effort dealing with the FSMVRP has focused on heuristics. One of the most important contributions to this field is that of Golden, Assad, Levy and Gheysens (1984) who suggested five adaptations of Clarke and Wright's (1964) savings algorithm. The first one, denoted CW, is a straightforward adaptation of the original algorithm to the FSMVRP. The second, called *combined savings*, or CS, extends the concept of savings to include fixed vehicle costs. Three variants of the CS approach were also developed. In the *optimistic opportunity savings* algorithm, OOS, the opportunity savings associated with an additional vehicle is defined as the cost of the smallest vehicle that can service the entire unused capacity of the new vehicle. The fourth method presented, called the *realistic opportunity savings* heuristic, ROS, suggests that opportunity savings encourage the use of larger vehicles when it seems profitable to do so. Thus, the opportunity savings should not be included in the savings formula unless combining two subtours requires the use of a larger vehicle. Finally, the last algorithm, called ROS-γ, is a variant of the ROS algorithm where γ is used as a shape parameter in the savings calculation.

In their paper, Golden, Assad, Levy and Gheysens also present some two-step procedures. First, a good method is used to generate a traveling salesman tour that visits all customers (for such methods see Laporte (1992a)). This tour is then partitioned into subtours, each satisfying the problem constraints. The first of these algorithms, called the *single partition giant tour* algorithm (SGT), starts with a tour that begins at the central depot, visits each customer exactly once and then returns to the depot. The second algorithm is called *multiple partition giant tour* algorithm (MGT).

In this version, the depot is not included in the initial tour, which allows more flexibility and produces a more powerful partitioning. Finally, other variants attempt improving the giant tour solution by using some traveling salesman improvement procedures. The 2-opt of Lin (1965) and the Or-opt of Or (1976) were used. Results presented on a set of twenty test problems show that the ROS-γ is the best savings based heuristic while the (MGT+Or-opt)[5] procedure, a combination of the MGT and the Or-opt procedures, is the best overall combination.

Gheysens, Golden and Assad (1984,1986) present two heuristics for the FSMVRP. The first one incorporates the vehicle capacity constraints into the objective function together with the fixed vehicle costs and the variable traveling costs, by using penalty multipliers. The resulting problem is then solved by using the (MGT+Or-opt) algorithm for different values of the penalty multipliers. The second heuristic is a two-stage algorithm. In the first stage, a lower bounding procedure is used to determine a vehicle mix, i.e., the number of vehicles of each type to be used. In the second stage, the generalized assignment procedure of Fisher and Jaikumar (1981) is used to solve the remaining routing problem for the obtained vehicles.

Other saving based algorithms have been proposed by Desrochers and Verhoog (1991). Their *matching based savings algorithm* (MBSA) is based on successive route fusions where the best fusion is selected by solving a weighted matching problem. A number of variants of this algorithm are proposed where each variant uses a different savings formula.

Salhi and Rand (1993) presented a route perturbation (RPERT) procedure which extends the previous work of Salhi and Rand (1987) for the VRP. The algorithm applies some perturbation procedures to the routes in order to improve the vehicle use of the whole fleet. Their solutions of the benchmark problems of Golden, Assad, Levy and Gheysens (1984) show that Salhi and Rand's method performs slightly better than (MGT+Or-opt)[5]. Salhi, Sari, Saidi and Touati (1992) showed how to extend existing heuristics to integrate variable traveling costs in the decision process. Osman and Salhi (1996) presented a modified version of RPERT, called MRPERT, allowing the search process to restart several times producing several solutions. The best solution is then retained.

A different strategy to solve the FSMVRP is to use a *neighborhood search procedures*, like the *Tabu search method* to repeatedly improve the obtained solution. Three tabu search based procedures are presented in Osman and Salhi (1996), in Taillard (1999) and in Gendreau, Laporte, Musaraganyi and Taillard (1999). For Osman and Salhi's (1996) tabu search algorithm, the initial solution is obtained by Salhi and Rand's (1993) algorithm and is improved by using the λ-interchange mechanism of Osman (1993). Taillard (1999) has presented a different approach which uses tabu search to generate a set of interesting solutions (see Rochat and Taillard, 1995), and solves a set-partitioning problem to select the best combination of routes. Although this heuristic produced excellent results, we notice that the set-partitioning problem employed is a NP-hard problem, and as such, could be very difficult to solve and highly time-consuming for large problems. Gendreau *et al* (1999) proposed a Tabu-based heuristic using the heuristic called GENIUS developed to solve the traveling salesman problem by Gendreau, Hertz and Laporte (1992) within an adaptive memory procedure. Both Taillard (1999) and Gendreau *et al* (1999) have not attempted to solve non-Euclidean problems and consequently, they provide no indications about their performance in this case.

The heuristic we propose hereafter can handle Euclidean and non-Euclidean problems. We also provide an evaluation of its performance for both problem classes. As will be shown, the proposed heuristic outperforms other composite heuristics and produces almost as good solutions as the best tabu search heuristics proposed earlier, but is much faster. Having a fast good solution method is very useful for transportation companies that rent a significant part of their fleet and consequently need to solve this type of problem quite frequently in order to take advantage of the savings that can be reached by frequent changes in fleet composition.

3. The proposed sweep based algorithm

The heuristic proposed here uses different procedures to generate a large set of good routes and then chooses those which satisfy the problem constraints at the lowest cost using a polynomial set partitioning algorithm. Specifically, the proposed heuristic uses five subordinate procedures called: *Order*, *1-petal*, *2-petal*, *Petals Selection* and *Improve*. We will describe these procedures before providing the general framework of the heuristic itself.

Order

As the heuristic proposed here is a sweep based method (like the sweep heuristic by Wren and Holliday (1972), Gillett and Miller (1974) and the petal heuristic introduced by Foster and Ryan (1976) later modified by Ryan, Hjorring and Glover (1993)), we need to order the vertices. If the problem is Euclidean, an ordering can easily be obtained by numbering the vertices in increasing order of the angle between the line linking the vertex to a reference point (e.g. the depot) and an arbitrary axis passing through the chosen reference point. Ties are broken by selecting the vertex having the smallest distance to the reference point. The heuristic can be executed several times while a different reference point is used at each execution. For Euclidean problems, our implementation uses two different reference points: the depot location and the geometrical center of all customer locations.

If the problem is not Euclidean, we also define two different orders by applying the following procedure. First we choose a set of three different vertices. In our implementation of the sweep heuristic, to define the first order we choose the three vertices v_i, v_j and v_k satisfying the following:

$$\underset{\substack{i,j,k \in S \setminus \{v_0\} \\ i \neq j, i \neq k, j \neq k}}{Max} \left\{ d_{0i} + d_{0j} + d_{0k} + d_{ij} + d_{jk} + d_{ki} \right\}$$

The second order uses the three vertices v_i, v_j and v_k that maximize the sum (d_{ij}+d_{jk}+d_{ki}).

These vertices are called *attraction vertices* and to each couple of attraction vertices (v_i , v_j), we associate a set of vertices, denoted A_{ij}, called *attraction set*. The three attraction sets are mutually exclusive and exhaustive where a vertex v_l is assigned to A_{ij} if $d_{li} \leq d_{lk}$ and $d_{lj} \leq d_{lk}$. Finally, the vertices of each attraction set are inserted between their attraction vertices by using the cheapest insertion procedure. The vertices are then numbered according to the order in which they appear in the resulting tour.

1-petal

To visit a given set of consecutive customers, say S, the 1-petal procedure generates a good route (called 1-petal route) that can be served by one vehicle and selects a vehicle-type to serve these customers. How to enumerate the different sets S will be given later. The 1-petal procedure is applied if and only if the sum of demands of the customers considered, denoted $Q(S)$, is less than or equal to Q_M, the capacity of the vehicle type having the largest capacity. The procedure selects the vehicle type having the smallest capacity larger than or equal to $Q(S)$, let h be this vehicle type.

In order to reduce the number of generated routes, the 1-petal procedure is only applied if $Q(S) \geq \Psi Q_h$ where $\Psi \in [0,1]$. Setting Ψ=0 implies that the 1-petal procedure will be always applied, otherwise, we require that at least a portion Ψ of the vehicle capacity be used by the customers of the set S in order to generate a 1-petal route over S.

The 1-petal route can be determined by applying any good TSP algorithm. In our implementation, we used the I^3 algorithm of Renaud, Boctor and Laporte (1996b) which is a three-phase algorithm. The first phase is a sub-tour construction procedure, called CLOCK, which constructs an envelope (not necessarily convex) of the vertices. The second phase inserts the remaining vertices into the sub-tour. A local optimization procedure is executed after each insertion to keep the tour cost as low as possible. Finally, a restricted 4-opt improvement procedure, called 4-Opt*, is used to improve the final tour. The same heuristic can be applied if the problem is non-Euclidean except that the initial sub-tour in this case, will be the triangle composed of the two most distant vertices, say v_i and $v_j \in S$, and a third vertex $v_k \in S$ yielding $\underset{\substack{k \neq i \\ k \neq j}}{Max} \left\{ c_{ik} + c_{kj} \right\}$.

2-petal

This procedure generates a set of two routes, called a 2-petal route, to serve a given set of customers S using exactly two vehicles and selects vehicle types to serve these two routes. It is applied if and only if the sum of demands of S is larger than Q_1, the capacity of the vehicle type having the smallest capacity, and less than or equal to two times Q_M, the largest vehicle capacity.

We examine all the vehicle type couples (g , h) satisfying the following three conditions: (1) $Q_g \leq Q_h$, (2) h is the vehicle type having the smallest capacity larger than or equal to $Q(S)$-Q_g, and (3) g is the vehicle type having the smallest capacity that can carry a demand equal to $Q(S)$-Q_h. Again, in order to reduce the number of generated routes, the 2-petal procedure is not applied if $Q(S)<\Psi(Q_g+Q_h)$ where $\Psi \in [0,1]$. Notice that some of the couples studied could be composed of identical vehicle types. Also notice that every type-couple not considered here will have a fixed cost higher than one of the couples studied and as such it will be considered as a dominated choice.

Among the vehicle-type couples considered, only the couple yielding the feasible solution with the lowest total cost (fixed cost plus travel cost), and the corresponding 2-petal route will be retained and used by the petals-selection procedure. In addition, if the set of customers S can be visited by only one vehicle then either the corresponding 1-petal solution or the best 2-petal solution is retained for the petals-selection procedure depending on which alternative yields the lowest cost.

For a given vehicle-type couple (g , h) and a given set of customers S, to construct a 2-petal route over S we first determine the two farthest vertices in S and create two back and forth routes between the depot and these vertices. Each remaining customer is then inserted into the route yielding the cheapest insertion cost while maintaining feasibility.

If some vertices of S remain uninserted after applying the above procedure, an improvement mechanism is applied to increase the chance of inserting more vertices. Let (i_1, j_1, k_1, l_1) be a sequence of four vertices (any of which may be the depot) on the first route and (i_2, j_2, k_2, l_2) be a sequence of four vertices (any of which may be the depot) on the second route. The following *eleven moves* are then attempted, but the depot is never moved: 1) insert j_1 between i_2 and j_2; 2) insert j_2 between i_1 and j_1; 3) swap j_1 and j_2; 4) insert (j_1, k_1) between i_2 and j_2, considering the two possible orientations; 5) insert (j_2, k_2) between i_1 and j_1, considering the two possible orientations; 6) swap (j_1, k_1) with (j_2, k_2), considering all four combinations. Only moves yielding better feasible solutions are considered and the *first* improving move is implemented. If some of the vertices of S are still uninserted, replace the vehicle type g by the type g+1 and repeat the route construction procedure from the beginning. Then, the 4-Opt* is used to improve each of the modified route.

Once a two-vehicle solution is obtained and if the total capacity of the two vehicles is tight, we re-apply the improvement procedure just described. To be precise, the procedure is applied if $Q(S) \geq \beta(Q_g + Q_h)$ where $\beta \in [0,1]$. Furthermore, re-apply the improvement procedure a second time if $g \neq h$ and $Q(S) \geq \delta (Q_g + Q_h)$ where $\delta \in [\beta,1]$, and this time the *best* (not the first) improving move is implemented. Our computational experiment showed that re-applying this improvement procedure this way improves the final solution significantly.

Petals Selection

Once all 1-petals and 2-petals have been generated, an optimal combination can be determined by solving a set partitioning problem where there is a column for each retained petal and a row for each vertex (customer) to be visited. Since each petal corresponds to a contiguous sector of vertices within the defined order, the set-partitioning matrix possesses the column circular property and thus the set-partitioning problem can be solved in polynomial time. A detailed description of the set-partitioning algorithm used is given in Boctor and Renaud (2000).

Improve

The petals-selection procedure produces a feasible solution to the FSMVRP composed of p routes, r_1, r_2, ... r_p , where each route serves a set of vertices S_i, i = 1, ..., p. This solution is then improved by applying the following two procedures until no further improvement can be reached. First, consider each couple of routes (r_i , r_j), check if a single vehicle can carry all the loads of the clients of $S_{ri} \cup S_{rj}$ and if possible, solve a TSP using I^3 over $S_{ri} \cup S_{rj}$. If the resulting cost is lower, the routes are concatenated. Second, consider each couple of routes (r_i , r_j) and improve them using the *eleven-move* improvement procedure described within the 2-petal procedure.

The proposed heuristic

The proposed heuristic generates all the possible sets of consecutive customers S that can be visited by one or two vehicles, it constructs the corresponding 1-petal and 2-petal routes, selects the best combination of routes to visit all customers and attempts to improve the obtained solution. In details, the heuristic is composed of the following steps.

Step 1, Orders determination

Use the procedure "*Order*" to select the required number of orders on the set of all vertices.

Step 2, Order selection

If all the selected orders are used, stop. Otherwise, consider one of the unused orders (in the following steps we assume that the vertices are renumbered according to the order being considered). Set $i := 0$.

Step 3, Route initialization

Set $i := i + 1$. If $i > n$, the petal generation process is finished; go to Step 7. Otherwise add to the set of possible routes (petals) the route (v_0, v_i, v_0), and store its cost $C_{ii} := 2c_{0i} + F_h$ where h is the smallest vehicle type that can carry a load of q_i. Set $j := i$.

Step 4, 1-Petal route

Set $j := j + 1$ and $S := \{v_i, ..., v_j\}$. If $Q(S) \leq Q_M$ and $Q(S) \geq \Psi Q_h$ where h is the smallest vehicle type that can carry a load of $Q(S)$, then apply the 1-petal procedure to determine a 1-petal route over S. If the total duration of the route obtained is less than or equal to T_h, the maximum allowable route time for the selected vehicle type, store the obtained route and its total cost, denoted C_{ij}.

Step 5, 2-petal route

If $Q(S) \geq 2Q_M$ then go to step 6. Otherwise, and if $Q(S) \geq Q_1$, apply the 2-petal procedure to determine a 2-petal route over *S*. If a feasible solution is obtained then store it and its total cost. Go to Step 4.

Step 6, Dominance test

Some of the petals just created may be dominated. For $k = j, j-1, ..., i+2$, consider the vertex sets $\{v_i, ..., v_k\}$ and $\{v_i, ..., v_{k-1}\}$. If $C_{i,k-1} \geq C_{ik}$, the petal defined over $\{v_i, ..., v_{k-1}\}$ is dominated and should be removed. It is replaced by the petal obtained from the one defined over $\{v_i, ..., v_k\}$ by removing v_k and adjusting the cost consequently. Go to Step 3.

Step 7, Petals selection

Find the optimal combination of the generated routes that constitute a solution to the FSMVRP by applying the "*Petals-selection*" procedure.

Step 8, Improvement of the initial solution

Use the "*Improve*" procedure to improve the obtained solution. Store the improved solution if it is the best obtained so far. Go back to step 2.

4. Computational results

This section reports on the performance of the proposed sweep heuristic over the twenty benchmark test problems of Golden, Assad, Levy and Gheysens (1984). First, a preliminary analysis is made to study the effect of the heuristic parameters on its performance. Comparisons are then made against the best composite heuristics of the literature. The results of the proposed sweep heuristic are also compared to the best available tabu search heuristics. In the following, computing times are in seconds on a Pentium II 233Mhz. All tested heuristics were coded in Delphi 3.0.

Tables 1 and 2 present the aggregated average results over the twenty test problems for different parameter combinations. The column *Average deviation* presents the average deviation above the best-known solutions given in the literature (see Table 4). The *standard sweep heuristic* corresponds to the complete algorithm presented in Section 3. The *truncated sweep heuristic* is similar to the standard one but, in order to save time, the improvement procedure in the *2-petal* algorithm is restricted to the following five moves (instead of eleven moves): 1) insert j_1 between i_2 and j_2; 2) insert j_2 between i_1 and j_1; 3) swap j_1 and j_2; 4) swap (j_1, k_1) with (j_2, k_2) while reversing (j_1, k_1); 5) swap (j_1, k_1) with (j_2, k_2) while reversing (j_2, k_2).

Table 1 shows that the standard sweep heuristic, with β= 0.25, δ = 0.80 and Ψ= 0 produces an average increase over the best-known solutions of only 0.49% for an average computing time of 179 seconds. With the same parameters, the truncated version produces an average deviation of 0.63% but the average time is reduced to 154 seconds. These results slightly change if Ψ, the lower bound on the route load, is increased to 0.90. The standard version using Ψ=0.90 brings up the average percentage increase over the best-known solution from 0.49% to 0.61%, but saves 63 seconds of computing time on average. The truncated version also performs well and its computing time can be largely reduced without significantly affecting the solution quality. Using the truncated version with Ψ=0.90 produced an average deviation of 0.64% in only 96 seconds, which represents a deterioration of less than 0.01% with respect to the truncated algorithm with Ψ=0.

Table 1: Average deviation above the best known solutions with β = 0.25, δ = 0.80

Ψ	Standard sweep heuristic		Truncated sweep heuristic	
	Average deviation	Time (seconds)	Average deviation	Time (seconds)
0.00	0.49%	179	0.63%	154
0.50	0.61%	154	0.63%	126
0.75	0.61%	147	0.63%	120
0.85	0.61%	133	0.64%	111
0.90	0.61%	116	0.64%	96

Table 2 presents the same results but for δ = 0.90 (δ is the parameter controlling the second execution of the improvement procedure, see section 3). This table shows that the solution quality is sensitive to changes in the value of δ.

Table 2: Average deviation above the best known solutions with β = 0.25, δ = 0.90

Ψ	Standard sweep heuristic		Truncated sweep heuristic	
	Average deviation	Time (seconds)	Average deviation	Time (seconds)
0.00	0.85%	151	0.90%	124
0.50	0.85%	147	0.90%	123
0.75	0.87%	143	0.90%	120
0.85	0.87%	130	0.90%	109
0.90	0.89%	109	0.90%	96

Many other computations were performed but are not reported here. For example, compared to δ = 0.80, using δ = 0.70 produced no improvement while requiring more computing time. Thus, δ = 0.80 seems to be a good setting. Similarly several values of β were tested and our results led us to choose β=0.25. Consequently, we chose to use the following parameter values with both the standard and the truncated sweep heuristic: β = 0.25, δ = 0.80 and Ψ = 0.00 (first line of Table 1).

Table 3 compares the two versions of the proposed sweep heuristic to some simple heuristics and some of the best composite heuristics presented in the literature. In the *Problem* column, the problem number is followed by an S if the problem is symmetric and by an E if the problem is Euclidean. Columns two and three present respectively the number of vehicle types (*K*) and the number of vertices (*n*). Other columns give the solution values of the heuristics tested. The results for ROS-γ (4.52%) are the best over 31 variants with γ between 0.0 and 3.0, while the results for (MGT+Or-Opt)[5] (1.92%) are the best over five repetitions. The last row of the table gives the average percentage increase above the best-known solutions. This table shows that the proposed standard sweep heuristic clearly outperforms the other heuristics tested, with an average deviation of only 0.49% above the best-known solutions. The adaptation of the Clarke and Wright method (CW) is clearly the least efficient method with an average increase over the best known solutions of 15.13%.

Table 3: Results against some other reported heuristics

Problem	K	n	CW	ROS-γ	(MGT+ Or-opt)[5]	RPERT	MPERT	Sweep (standard)	Sweep (truncated)
1S	3	12	640	618	622	614	606	602	602
2S	3	12	796	768	722	722	722	722	722
3E	5	20	1119	1009	966	1003	971.95	965.6	965.6
4E	3	20	7822	6937	6930	6447	6447.8	6467.01	6467.01
5E	5	20	1061	1048	1013	1015	1015.13	1009.14	1009.14
6E	3	20	9343	6522	6974	6516	6516.56	6588.93	6588.93
7S	5	30	7988	7452	7389	7402	7377	7346	7346
8S	4	30	2430	2468	2367	2367	2352	2347	2347
9S	5	30	2228	2266	2220	2209	2209	2211	2211
10S	4	30	2434	2424	2370	2377	2377	2362	2362
11S	4	30	5428	4953	4763	4819	4787	4765	4765
12S	6	30	4276	4221	4136	4092	4092	4092	4092
13E	6	50	2550	2566	2438	2493	2462.01	2412.51	2412.51
14E	3	50	12000	9178	9132	9153	9141.69	9128.98	9128.98
15E	3	50	2885	2763	2640	2623	2600.31	2618.03	2639.60
16E	3	50	3026	2894	2822	2765	2745.04	2767.29	2778.71
17E	4	75	1968	1958	1783	1767	1766.81	1757.72	1768.46
18E	6	75	3447	2520	2432	2439	2439.4	2419.92	2429.92
19E	3	100	11319	8741	8721	8751	8704.2	8687.31	8699.43
20E	3	100	4689	4293	4195	4187	4166.03	4102.82	4120.41
Average deviation			**15.13%**	**4.52%**	**1.92%**	**1.30%**	**0.79%**	**0.49%**	**0.63%**

Table 4 compares the standard version of the proposed heuristic to the tabu search heuristics of Osman and Salhi (1994), of Taillard (1999) and of Gendreau *et al* (1999). The results for Taillard's heuristic are the average results of five runs. Taillard presents also the best solutions he obtained during his computations but does not give any indications about their computational times; these best results are given in Table 5. The results of Gendreau *et al* (1999) are the average solutions of ten runs. The corresponding computation times are not given as they were not given by the authors in their paper. The best results over the ten runs were reported and are given in Table 5.

Table 4: Results compared to tabu search heuristics

Problem	Best known	Osman & Salhi		Taillard		Gendreau *et al*		Standard Sweep	
		Cost	Time[1]	Cost	Time[2]	Cost	Time	Cost	Time[3]
1S	602	602	3	--	--	--	--	602	0
2S	722	722	2	--	--	--	--	722	0
3E	961.03	971.24	5	--	--	961.03	--	965.6	4
4E	6437.33	6445.1	6	--	--	6441.01	--	6467.01	6
5E	1007.05	1009.15	5	--	--	1008.72	--	1009.14	5
6E	6516.47	6516.56	4	--	--	6517.98	--	6588.93	9
7S	7298	7310	15	--	--	--	--	7346	28
8S	2348	2348	17	--	--	--	--	2347	44
9S	2209	2209	14	--	--	--	--	2211	32
10S	2363	2363	14	--	--	--	--	2362	10
11S	4755	4755	19	--	--	--	--	4765	34
12S	4092	4092	10	--	--	--	--	4092	19
13E	2408.41	2471.07	62	2436.78	470	2424.88	--	2412.51	50
14E	9119.03	9125.65	71	9123.60	570	9121.98	--	9128.98	160
15E	2586.37	2606.72	46	2593.61	334	2590.68	--	2618.03	45
16E	2741.50	2745.01	35	2744.25	349	2743.96	--	2767.29	28
17E	1747.24	1762.05	85	1753.74	2072	1752.29	--	1757.72	652
18E	2373.63	2412.56	116	2382.80	2744	2392.57	--	2419.92	1037
19E	8661.81	8685.71	289	8665.40	12528	8682.50	--	8687.31	1110
20E	4047.55	4188.73	306	4063.18	2117	4100.20	--	4102.82	307
Average deviation		**0.57**%	56.2	**0.35**%	2648	**0.32**%	--	**0.49**%	179

[1]In seconds on VAX 4500
[2]In seconds on Sparc workstation, processor TMS390Z55
[3] In seconds on Pentium II 233 Mhz.

Table 4 shows that the sweep heuristic produces results better than those of Osman and Salhi's (1994) tabu search heuristic. Results of Taillard (1999) are the best for 7 of the 8 problems solved by this author. The major drawback of Taillard's algorithm is its computational time, which increases rapidly. The results obtained by Gendreau *et al* are good results always better than or equivalent to those obtained by Osman and Salhi, and almost as good as those given by Taillard while requiring less computation time.

Table 5 compares the best results we obtained through our numerous runs, to the best results reported by other researchers. In this table, boldface figures indicate the best known solutions. The Table also shows new best results for problems 8, 10 and 13 that were obtained by our heuristic.

Table 5: Best results obtained by different researches

Problem	Osman & Salhi	Taillard	Gendreau *et al*	Standard Sweep
1S	**602**	--	--	**602**
2S	**722**	--	--	**722**
3E	971.24	--	**961.03**	963.61
4E	6445.1	--	**6437.33**	**6437.33**
5E	1009.15	--	**1007.05**	1007.96
6E	6516.56	--	**6516.47**	6537.74
7S*	7310	--	--	7346
8S	2348	--	--	**2347**
9S	**2209**	--	--	2211
10S	2363	--	--	**2362**
11S	**4755**	--	--	4761
12S	**4092**	--	--	**4092**
13E	2471.07	2413.78	2408.41	**2406.43**
14E	9125.65	**9119.03**	**9119.03**	9122.01
15E	2606.72	**2586.37**	**2586.37**	2618.03
16E	2745.01	**2741.50**	2742.50	2761.96
17E	1762.05	**1747.24**	1749.50	1757.21
18E	2412.56	**2373.63**	2381.43	2413.39
19E	8685.71	**8661.81**	8675.16	8687.31
20E	4188.73	**4047.55**	4086.76	4094.54
Average deviation	0.57%	0.03%	0.14%	0.35%

* The best-known solution for this problem is 7298 found by Golden *et al* (1984).

5. Conclusions

This paper presents a new sweep based heuristic for solving the fleet size and mix vehicle routing problem. To the best of our knowledge, it is the first time that a sweep procedure has been used to solve this problem. We also demonstrate how to adapt the sweep procedure to deal with non-Euclidean problems. It has been shown that the proposed heuristic outperforms existing composite algorithms of the literature and is very close to, and some times better than, the best known tabu search based algorithms. For twenty benchmark test problems, the standard version of the proposed sweep heuristic produces an average deviation above the best-known solution of 0.49% with an average computing time of 179 seconds. It produced seven best-known solutions, three of which are new best-known solutions. A faster and truncated version of the algorithm is also proposed. The latter produces an average deviation above the best-known solution of 0.63% for a computing time of only 154 seconds. In light of these results, the algorithms presented in this paper can be viewed as an interesting alternative to more time consuming procedures.

Acknowledgments

This work was partially supported by grants OGP0172633 and OGP0036509 from the Canadian Natural Sciences and Engineering Research Council (NSERC). This support is gratefully acknowledged.

References

Boctor, F. F. and Renaud J. (2000), The column-circular sub-sets selection problem: Complexity and solutions, *Computers and Operation Research*, **27**, 383-398.

Clarke, G. and Wright J. (1964), Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, **12**, 568-581.

Cornuéjols, G. and Harche F. (1993), Polyhedral study of the capacitated vehicle routing problem. *Mathematical Programming*, **60**, 21-52.

Desrochers, M. and Verhoog T. W. (1991), A new heuristic for the fleet size and mix vehicle routing problem. *Computers and Operations Research*, **18**, 263-274.

Etazadi, T. and Beasley J. E. (1983), Vehicle fleet composition. *Journal of the Operational Research Society*, **34**, 87-91.

Fisher, M. and Jaikumar M. (1981), A generalized assignment heuristic for vehicle routing. *Networks*, **11**, 109-124.

Foster, B. A. and Ryan D. M. (1976), An integer programming approach to the vehicle scheduling problem. *Operational Research Quarterly*, **27**, 377-384.

Gendreau, M., Hertz A. and Laporte G. (1992), New insertion and postoptimization procedures for the traveling salesman problem. *Operations Research*, **40**, 1086-1094.

Gendreau, M., Hertz A. and Laporte G. (1994), A tabu search heuristic for the vehicle routing problem. *Management Science*, **40**, 1276-1290.

Gendreau, M, Laporte G., Musaraganyi Ch. and Taillard E. (1999), A tabu search heuristic for the heterogeneous fleet vehicle routing problem, *Computers and Operations Research*, **26**, 1153-1173.

Gheysens, F., Golden B. and Assad A. (1984), A comparison of techniques for solving the fleet size and mix vehicle routing problem. *OR Spektrum*, **6**, 207-216.

Gheysens, F., Golden B. and Assad A. (1986), A new heuristic for determining fleet size and composition. *Mathematical Programming Study*, **26**, 233-236.

Gillett, B. and Miller L. (1974), A heuristic for the vehicle dispatching problem. *Operations Research*, **22**, 340-349.

Golden, B., Assad A., Levy L. and Gheysens F. (1984), The fleet size and mix vehicle routing problem. *Computers and Operations Research*, **11**, 49-66.

Gould, J. (1969), The size and composition of a road transport fleet. *Operational Research Quarterly*, **20**, 81-92.

Hadjiconstantinou, E., Christofides N. and Mingozzi A. (1995), A new exact algorithm for the vehicle routing problem based on *q*-path and *k*-shortest paths relaxations. *Annals of Operations Research*, **61**, 21-43.

Laporte, G. (1992a), The traveling salesman problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, **59**, 231-247.

Laporte, G. (1992b), The vehicle routing problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, **59**, 345-358.

Laporte, G. and Osman I. H. (1995), Routing problems: A bibliography. *Annals of Operations Research*, **61**, 227-262.

Lin, S. (1965), Computer solution of the traveling salesman problem. *Bell System Computer Journal*, **44**, 2245-2269.

Or, I. (1976), Traveling salesman-type combinatorial problems and their relation to the logistics of regional blood banking. Doctoral dissertation, Northwestern University.

Osman, I. H. (1993), Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Annals of Operations Research*, **41**, 421-451.

Osman, I. and Salhi S. (1996), Local search strategies for the vehicle fleet mix problem. In V.J. Rayward-Smith, I.H. Osman, C.R. Reeves and G.D. Smith ed. *Modern Heuristic Search Methods*, Wiley, 131-153.

Renaud, J., Boctor F. F. and Laporte G. (1996a), An improved petal heuristic for the vehicle routing problem. *Journal of the Operational Research Society*, **47**, 329-336.

Renaud, J., Boctor F. F. and Laporte G. (1996b), A fast composite heuristic for the symmetric traveling salesman problem. *INFORMS Journal on Computing*, **8**, 134-143.

Rochat, Y. and Taillard É. D. (1995), Probabilistic diversification and intensification in local search for vehicle routing. *Journal of Heuristics*, **1**, 147-176.

Ryan, D. M., Hjorring C. and Glover F. (1993), Extension of the petal method for vehicle routing. *Journal of the Operational Research Society*, **44**, 289-296.

Salhi, S. and Rand G. K. (1987), Improvement to vehicle routing heuristics. *Journal of the Operational Research Society*, **38**, 293-295.

Salhi, S. and Rand G. K. (1993), Incorporating vehicle routing into the vehicle fleet composition problem. *European Journal of Operational Research*, **66**, 313-330.

Salhi, S., Sari M., Saidi D. and Touati N. A. C. (1992), Adaptation of some vehicle fleet mix heuristics. *OMEGA International Journal of Management Science*, **20**, 653-660.

Taillard, É. (1993), Parallel iteration search methods for vehicle routing. *Networks*, **23**, 661-673.

Taillard, É. D. (1999), A heuristic column generation method for the heterogeneous fleet VRP, *RAIRO*, **33**, 1-34.

Woods, D. G. and Harris F. C. (1979), Truck fleet size composition for concrete distribution. *International Journal of Physical Distribution*, **10**, 3-14.

Wren, A. and A. Holliday (1972), Computer scheduling of vehicle from one or more depots to a number of delivery points. *Operational Research Quarterly*, **23**, 333-344.