CIRRELT

Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation

A Hybrid Genetic Algorithm for Multi-Depot and Periodic Vehicle Routing Problems

Thibaut Vidal
Teodor Gabriel Crainic
Michel Gendreau
Nadia Lahrichi
Walter Rei

July 2010

CIRRELT-2010-34

Bureaux de Montréal :
Université de Montréal
C.P. 6128, succ. Centre-ville
Montréal (Québec)
Canada H3C 3J7
Téléphone : 514 343-7575
Télécopie : 514 343-7121

Bureaux de Québec :
Université Laval
2325, de la Terrasse, bureau 2642
Québec (Québec)
Canada G1V 0A6
Téléphone : 418 656-2073
Télécopie : 418 656-2624

www.cirrelt.ca

Université Laval · UQÀM · HEC Montréal · École Polytechnique Montréal · Université de Montréal

A Hybrid Genetic Algorithm for Multi-Depot and Periodic Vehicle Routing Problems

Thibaut Vidal[1,2], Teodor Gabriel Crainic[1,3,*], Michel Gendreau[1,4], Nadia Lahrichi[1,3], Walter Rei[1,3]

[1] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT)

[2] Department of Computer Science and Operations Research, Université de Montréal, P.O. Box 6128, Station Centre-ville, Montréal, Canada H3C 3J7

[3] Department of Management and Technology, Université du Québec à Montréal, P.O. Box 8888, Station Centre-Ville, Montréal, Canada H3C 3P8

[4] Department of Mathematics and Industrial Engineering, École Polytechnique de Montréal, P.O. Box 6079, Station Centre-ville, Montréal, Canada H3C 3A7

Abstract*.* We propose an algorithmic framework that successfully addresses three vehicle routing problems: the multi-depot VRP, the periodic VRP, and the multi-depot periodic VRP with heterogeneous capacitated vehicles and constrained route duration. The meta-heuristic combines the exploration breadth of population-based evolutionary search, the aggressive-improvement capabilities of neighborhood-based meta-heuristics, and advanced population-diversity management schemes. Extensive computational experiments show that, the method performs impressively, in terms of both solution quality and computational efficiency. It particular, it either identifies the best known solutions, including the optimal ones, or identifies new best solutions for all currently available benchmark instances for the three problem classes.

Keywords. Multi-depot multi-period vehicle routing problems, hybrid populations-based meta-heuristics, adaptive population diversity management.

Acknowledgements. Partial funding for this project has been provided by the Natural Sciences and Engineering Research Council of Canada (NSERC), through its Industrial Research Chair and Discovery Grants programs, and by the partners of the Chair, CN, Rona, Alimentation Couche-Tard and the Ministry of Transportation of Québec, and by the Fonds québécois de la recherche sur la nature et les technologies (FQRNT) through its Team Research Project Program.

---

* Corresponding author: Teodor.Gabriel@cirrelt.ca

Dépôt légal – Bibliothèque nationale du Québec,
Bibliothèque nationale du Canada, 2010

1 Introduction

Vehicle Routing Problem (VRP) formulations are used to model an extremely broad range of issues in many application fields, transportation, supply chain management, production planning, and telecommunications, to name but a few (Golden et al., 2002; Crainic, 2008; Hoff et al., 2010). The mathematical structure of most VRP formulations is generally simple but only deceivingly so, most problems of interest being NP-Hard and yielding integer programming (IP) combinatorial formulations presenting significant methodological challenges. Not surprisingly, starting with the initial introduction of the VRP by Dantzig and Ramser (1959), routing problems make up an extensively and continuously studied field, as illustrated by numerous conferences, survey articles (e.g., Christofides et al., 1979; Bodin et al., 1983; Fisher, 1995; Desrosiers et al., 1995; Powell et al., 1995; Gendreau et al., 2002; Laporte and Semet, 2002; Bräysy et al., 2004; Bräysy and Gendreau, 2005a,b; Cordeau et al., 2005, 2007; Bräysy et al., 2008a,b; Francis et al., 2008; Laporte, 2009), and books (Toth and Vigo, 2002; Golden et al., 2008).

Surveying the literature one notices, however, that not all problem classes have received an equal nor adequate degree of attention. This is the case for the problems with multiple depots and periods, the recent contributions to the periodic variant not altering the statement. A second general observation is that most methodological developments target a particular problem variant, the capacitated VRP or the VRP with time windows, for example, very few contributions aiming to address a broader set of problems, the Unified Tabu Search (Cordeau et al., 1997, 2001) being a notable exception. This also applies to the classes of problems targeted in this paper.

Our objective is to contribute toward addressing these two challenges. We propose an algorithmic framework that successfully addresses three VRP variants: the multi-depot VRP, *MDVRP*, the periodic VRP, *PVRP*, and the multi-depot periodic VRP, *MDPVRP*, with heterogeneous capacitated vehicles and constrained route duration. The literature on these problems is relatively scarce despite their relevance to many applications, e.g., raw material supply (Alegre et al., 2007), refuse collection (Beltrami and Bodin, 1974; Russell and Igo, 1979; Teixeira et al., 2004), food collection or distribution (Carter et al., 1996; Golden and Wasil, 1987; Parthanadee and Logendran, 2006), and maintenance operations (Blakeley et al., 2003; Hadjiconstantinou and Baldacci, 1998).

We propose a meta-heuristic that combines the exploration breadth of population-based evolutionary search, the aggressive-improvement capabilities of neighborhood-based meta-heuristics, and advanced population-diversity management schemes. The method, that we name *Hybrid Genetic Search with Adaptive Diversity Control* (*HGSADC*), performs impressively, in terms of both solution quality and computational efficiency. It particular, HGSADC either identifies the best known solutions, including the optimal ones, or identifies new best solutions for all currently available benchmark instances for the three problem classes

To sum up, the main contributions of this article are: 1) A new meta-heuristic that is highly effective for three classes of important vehicle routing problem classes, the MDVRP, the PVRP, and the MDPVRP. It equals or outperforms the current best methods proposed for each particular class and requires a limited computational effort. 2) New population-diversity management mechanisms to allow a broader access to reproduction, while preserving the memory of what characterizes good solutions represented by the elite individuals of the population. In this respect, we revisit the traditional *survival-of-the-fittest* paradigm to enhance the evaluation of individuals by making it rely on both fitness and diversity (distance-to-the-others) measures. Our empirical studies show this mechanism not only efficiently avoids premature population convergence, but also outperforms traditional diversity management methods relative to the general behavior of the solution method. 3) An efficient offspring education scheme that integrates key features from efficient neighborhood search procedures, e.g., memories and granular tabu search concepts.

The paper is organized as follows. Section 2 states the notation and formal definition of the three classes of VRPs we address, while the relevant literature is surveyed in Section 3. The proposed meta-heuristic is detailed in Section 4, its performances are analyzed in Section 5, and we conclude in Section 6.

2 Problem Statement

We present the problem description for the MDVRP, PVRP, and MDPVRP problem classes, together with the notation we use in this paper, and the problem transformation, MDPVRP to a generalized PVRP, which supports the following developments.

The basic Capacitated Vehicle Routing Problem (CVRP) can be defined as follows. Let $G = (\mathcal{V}, \mathcal{A})$ be a complete graph, vertex $v_0 \in \mathcal{V}$ representing the depot where vehicles and the product to be distributed are kept, each other vertex in $\mathcal{V}\backslash\{v_0\}$ standing for one of the n customers to be served from the depot. Each customer $i,\ i = 1, \dots,\ |\mathcal{V}|$ is characterized by a non-negative demand q_i and a service duration d_i. Arcs $a_{ij} \in \mathcal{A},\ i, j \in \mathcal{V}$ represent the possibility to travel from vertex i to vertex j at a duration cost of c_{ij}. A fleet of m identical vehicles with capacity Q is located at the depot, the total duration of the route performed by each vehicle, computed as the total travel and service time required to serve the customers on the route, being limited at D. The goal is to design a set of m vehicle routes servicing all customers, such that vehicle and route constraints are respected, and the total route duration is minimized.

Several depots, d, are available to service customers in the Multi-Depot VRP, m representing the number of vehicles available at each depot. A time dimension is introduced in the Periodic VRP as route planning is to be performed over a horizon of t periods.

Each customer i is characterized by a service frequency f_i, representing the number of visits to be performed during the t periods, and a list L_i of possible visit-period combinations, called *patterns*. The PVRP aims to select a pattern for each customer and construct the associated routes to minimize the total cost over all periods. Finally, the Multi-Depot Periodic VRP extends the two previous problem settings, the selection of a depot and visit pattern for each customer being required to construct optimal routes. It is important to note that a linking constraint is implicitly defined, services in different periods to the same customer being required to originate at the same depot. The MDPVRP reduces to a PVRP when $d = 1$ and to a MDVRP when $t = 1$.

Cordeau et al. (1997) proposed a formulation of a generalized PVRP with vehicle and period-dependent routing costs. The authors showed that this formulation includes the MDVRP as a special case, by associating a different period to each depot, such that each customer i has a frequency $f_i = 1$ and may be served any period. We follow this lead and show that a MDPVRP with t days and d depots can be transformed into a generalized PVRP, by associating a period in the new problem to each (period, depot) pair in the MDPVRP.

Let P be a MDPVRP instance, with periods $\{0, \dots, t-1\}$, depots $\{0, \dots, d-1\}$ and m vehicles per depot. Let i be a customer with frequency f_i and pattern list L_i containing p patterns $L_i = \{\{p_{11}, \dots, p_{1f_i}\}, \dots, \{p_{p1}, \dots, p_{pf_i}\}\}$. We now define the problem P' which has $t' = t \times d$ periods and m vehicles available at each period. For each customer i in the new problem, $f'_i = f_i$ and the pattern list L'_i contains $d \times p$ patterns defined by Equation 1.

$$L'_i = \bigcup_{\substack{a \in \{0, \dots, d-1\} \\ b \in \{1, \dots, p\}}} \{p_{b1} + a \times t, \dots, p_{bf_i} + a \times t\} \tag{1}$$

where the first t periods are associated with depot 0, the next are associated with depot 1, and so on. Travel costs (durations) are adjusted to take into account that vehicles operating in period l in the new general PVRP originate from depot l/d.

We rely on this transformation in the development of the proposed methodology and work on (depot, pattern) couples. We thus transform a problem with several attributes into a problem with less attributes, which thus becomes somewhat easier to address. Of course, the method must be computationally efficient to deal with the increased number of periods and the corresponding increase in problem dimension. As the computational results displayed in Section 5 show, we achieve both these goals.

3 Literature Review

This section provides a brief literature review of contributions for the PVRP, the MDVRP, and the MDPVRP. The purpose of this review is twofold. First, to present the most recently proposed meta-heuristic algorithms, particularly population-based ones, for the considered problems. Second, to distinguish the leading solution approaches for the three problem settings.

Population and neighborhood-based meta-heuristics have been proposed for the PVRP. With respect to the former, Drummond et al. (2001) proposed an island-based parallel evolutionary method, which evolves individuals representing schedules (patterns), the fitness of each individual being obtained by constructing routes for each period with the Clarke and Wright (1964) savings heuristic. Alegre et al. (2007) proposed a scatter search procedure designed especially for PVRPs with a large numbers of periods. As in Drummond et al. (2001), the core of the method is dedicated to the improvement of visit schedules, while a neighborhood-based improvement procedure is used to design routes for each period. Contrasting with the two previous methods, Matos and Oliveira (2004) proposed an ant colony optimization (ACO) approach that first optimizes routes, then schedules. The PVRP is first transformed into a a large VRP containing each customer as many times as given by its frequency and addressed by an ACO method. The problem of distributiing the resulting routes among periods is then solved as a graph coloring problem, with occasional changes in customer patterns to progress toward a feasible PVRP solution. In a final step, an ACO method is used to optimize the plan for each period separately.

Until recently, however, the most successful contributions to this problem were based on the serial exploration of neighborhoods. The local search approach of Chao et al. (1995) was the first to use deteriorating moves to escape from poor local optima. It also to temporarily allowed relaxation of vehicle-capacity limits to enhance the exploration of the solution space. The tabu search proposed by Cordeau et al. (1997) introduced an innovative guidance scheme, which collects statistics on customer assignments to periods and vehicle routes in order to penalize recurring assignments within the solutions obtained and, thus, gradually diversify the search. For a long period of time, this method stood as the state of the art solution approach for both the PVRP and the MDVRP, as well as, in its Unified Tabu Search version (Cordeau et al., 2001), for a number of other VRP variants. It has only been outperformed recently by the Variable Neighborhood Search of Hemmelmayr et al. (2009), which is built upon various well-known VRP neighborhoods, e.g., the string relocate, the swap, and the 3-opt. Finally, one should notice the VNS algorithm with multilevel refinement strategy of Pirkwieser and Raidl (2010), particularly tailored for large-size instances.

We are aware of only two evolutionary approaches for the MDVRP, both taking advantage of geometric aspects within the problem. Thus, Thangiah and Salhi (2001) represented solutions as circles in the 2D space of depot and customers, which is closely related to the idea of customer clustering into routes, whereas Ombuki-Berman and Hanshar (2009) introduced a mutation operator that specifically targeted the depot assignment to "borderline" customers, i.e., customers that are close to several depots. In this case also, however, neighborhood-based methods, such as the record-to-record local search of Chao et al. (1993), the tabu search algorithms of Cordeau et al. (1997) and Renaud et al. (1996), and the simulated annealing method of Lim and Zhu (2006),proved to be more efficient. To date, the most successful approach for the MDVRP remains the adaptive large neighborhood search (ALNS) method of Pisinger and Ropke (2007), which implements the ruin-and-recreate paradigm with an adaptive selection of its destruction and reparation operators.

In the case of the MDPVRP, most proposed algorithms do not consider all attributes simultaneously, but rather apply a successive-optimization approach. Thus, the method developed by Hadjiconstantinou and Baldacci (1998) starts by first assigning all customers to a particular depot. Given these a priori assignments, customer visits are then successively inserted among available periods to obtain feasible visit combinations. The depot-period VRP subproblems obtained are then separately solved using a tabu search algorithm. Finally, a phase that attempts to improve the solution by modifying the period or depot assignments through customer interchanges is applied. The overall solution strategy then repeats this sequence of heuristics for a fixed number of iterations. Other such approaches were proposed by Kang et al. (2005) and Yang and Chu (2000), where schedules for each depot and period are first determined, followed by the design of the corresponding routes.

We are aware of only two methods that aim to address problems similar to the MDPVRP as a whole. Parthanadee and Logendran (2006) implemented a tabu search method for a complex variant of the MDPVRP with backorders. The authors also study the impact of interdependent operations between depots, where the depot assignment of a customer may vary according to the periods considered. Significant gains are reported on small test instances when such operations are applied. Crainic et al. (2009b,a) introduced the Integrative Cooperative Search (ICS) framework, which relies on the problem decomposition by attributes, concurrent resolution of subproblems, integration of the elite partial solutions yielded by the subproblems, and adaptive search-guidance mechanisms. The authors used the MDPVRP with time windows to illustrate the methodology with very promising results, but no results are available for the problems addressed in this paper. Moreover, ICS targets complex problem settings and we provide a simpler way to treat the MDPVRP.

A number of exact methods were also proposed for one or another of the problems we address. Noteworthy are the recent contributions of Baldacci and Mingozzi (2009) and Baldacci et al. (2010) addressing the MDVRP and the PVRP. Exact methods are limited in the size of instances they may handle, but these particular approaches have proven quite successful in solving to optimality several instances that are used as a test bed for the algorithm we propose.

This brief review supports the general statement made previously that no satisfactory method has yet been proposed for the three problem settings. Furthermore, the contributions to the MDPVRP literature are very scarce, those addressing all the problem characteristics simultaneously being scarcer still. Most solution methods proposed address the periodic and multi-depot VRP settings, with neighborhood-based methods yielding, until now, the best results on standard benchmark instances. However, evolutionary methods have proven recently to be efficient on the standard VRP (Prins, 2004; Mester and Braysy, 2007; Nagata and Bräysy, 2009) and on a number of other VRP variants, as underlined in a review on evolutionary methods for the VRPTW (Bräysy et al., 2004). Noteworthy is the contribution of Prins (2004) for the VRP, which introduced an important methodological element, namely the solution representation for the VRP as a TSP tour without delimiters along with a polynomial time algorithm to partition the sequence of customers into separate routes. This approach was later applied by (Lacomme et al., 2005; Chu et al., 2006) to the periodic capacitated arc routing problem, which shares a number of common characteristics with the PVRP. We adopt this solution representation for the population-based method we propose to efficiently address the periodic and multi-depot problems, as well as the MDPVRP as a whole. This methodology is described in the next section.

4 The Hybrid Genetic Search with Adaptive Diversity Control Meta-heuristic

The hybrid meta-heuristic we propose is based on the Genetic Algorithm (GA) paradigm introduced by Holland (1975), but includes a number of advanced features, particularly in terms of generation of new individuals, offspring education, and population management, which contribute to its originality and high performance level.

The method evolves a population of individuals, representing feasible and unfeasible solutions, through successive application of a number of operators to select two parent solutions, combine them, yielding a new individual, and enhance this offspring. We identify the latter operator as offspring *education*. It includes a series of solution transformations proved to be efficient in neighborhood-based meta-heuristics for routing problems, as well as a feasibility *repair* procedure.

Feasible and unfeasible solutions are kept in two separate subpopulations, managed to both evolve toward high-quality solutions and maintain a high level of diversity among individuals. This is performed not only when selecting the surviving individuals for the next "generation", which is rather standard GA methodology, but also through an evaluation mechanism of individuals combining traditional fitness and a representation of the particular contribution an individual makes to the diversity of the gene pool. This fitness-diversity evaluation mechanism is shown to play an important role in the overall performance of the proposed methodology.

The general structure of the proposed *Hybrid Genetic Search with Adaptive Diversity Control* (*HGSADC*) meta-heuristic therefore is

- Initialization (Section 4.6);
- Repeat
  - Selection of parent solutions and generation of offspring (Section 4.4);
  - Offspring education; If unfeasible, repair with probability P_{rep} (Section 4.5);
  - Adjustment of parameters enforcing penalties for violating feasibility conditions (if necessary) (Section 4.6);
  - Diversification if best solution not improved for It_{div} consecutive iterations (Section 4.6);
  - Culling of any sub-population that reaches its maximum number of individuals (Section 4.6);
- Until stopping conditions are satisfied, i.e., either It_{NI} iterations without improvement of the best feasible solution, or the time limit T_{max} is reached.

We initiate the description with the definition of the search space (Section 4.1) and the representation of the individuals (Section 4.2). We then proceed with detailed discussions of the individual evaluation procedure (Section 4.3), the selection, crossover, and education operators, and the population management mechanism.

4.1 Search space

It is well-known in meta-heuristic literature that allowing a controlled exploration of unfeasible solutions may enhance the performance of the search, which may more easily transition between structurally different feasible solutions.

A number of constraint relaxations have been proposed in the VRP literature. Thus, for example, Prins (2004) and Lacomme et al. (2005) relax the fleet-size constraints in their GA-based methods, while Gendreau et al. (1994) and Cordeau et al. (1997) relax vehicle capacity and maximum travel time limits in their tabu search algorithms. We favor the second approach even though our solution representation is closer to that of the first authors. Indeed, a solution with too high a number of vehicles may require sophisticated and computationally costly route-reduction methods. Moreover, it is much easier in the second case to introduce dynamically self-adjusting penalties that gradually guide the search into and out of the unfeasible-solution zone, while still allowing the use of most local search moves used in the VRP literature to modify and enhance routes. We therefore define the search space as the set of feasible and unfeasible solutions, the latter being obtained by relaxing the limits on vehicle capacities and maximum route travel time.

Two penalty parameters are defined, σ^{Q} for vehicle load excess and σ^{C} for route travel time excess. Given a route r with load $q(r)$ and travel time $t(r)$, Equation 2 defines its penalized cost $\phi(r)$. The total cost of a solution is then computed as the sum of the penalized costs of all routes and is used in the computation of the fitness of the individuals relative to the current population.

$$\phi(r) = t(r) + \sigma^{\mathrm{C}} \times max(0, t(r) - D) + \sigma^{\mathrm{Q}} \times max(0, q(r) - Q) \tag{2}$$

The penalty parameters σ^{Q} and σ^{C} are dynamically adjusted during the execution of the algorithm, to favor the generation of "naturally" feasible individuals, that is, individuals that are feasible following education without requiring to call on the repair procedure. Let ξ^{REF} be the given desired proportion of naturally feasible individuals, and ξ^{Q} and ξ^{C} the proportion in the last 100 generated individuals of naturally feasible solutions with respect to vehicle capacity and route duration, respectively. The adjustment is then performed every 100 iterations (the rules are identical for the duration-constraints parameter ξ^{C}):

- if $\xi^{\mathrm{Q}} \leq \xi^{\mathrm{REF}} - 0.05$, then $\sigma^{\mathrm{Q}} = \sigma^{\mathrm{Q}} * 1.2$;
- if $\xi^{\mathrm{Q}} \geq \xi^{\mathrm{REF}} + 0.05$, then $\sigma^{\mathrm{Q}} = \sigma^{\mathrm{Q}} * 0.85$.

We do not consider the initial penalty parameter values as particularly critical, because the dynamic adjustment drives them towards suitable values, as long as the initial penalties are of an order of magnitude comparable to the objective function. We therefore set the initial values for these parameters to $\sigma^{\mathrm{C}} = 1$ and $\sigma^{\mathrm{Q}} = \bar{c}/\bar{q}$, where $\bar{c}$ represents the average distance between two customers and $\bar{q}$ is the average demand.

4.2 Solution representation

An individual I in HGSADC is defined by three "chromosomes" related to its customer schedule, depot allocation, and corresponding routes. The *pattern chromosome*, P_I, thus registers for each customer i the list of days of visit $\lambda_I[i]$, which must correspond to one of its pre-defined patterns. The *depot chromosome*, D_I, contains the depot allocation $d_I[i]$ of each customer i. Finally, the *route chromosome*, R_I, contains $t \times d$ sequences of customers without trip delimiters, the sequence $s_I[k,l]$ corresponding to the customers served from depot k during period l. The route chromosome corresponds to a giant TSP tour representation as introduced by Prins (2004) and illustrated in Figure 1 for an instance with two periods and two depots. The corresponding solution is encoded as four sequences of visits without delimiters, one for each couple (period, depot). The *Split* algorithm is then used on each sequence to find the optimal segmentation into routes and compute the fitness of the individual.

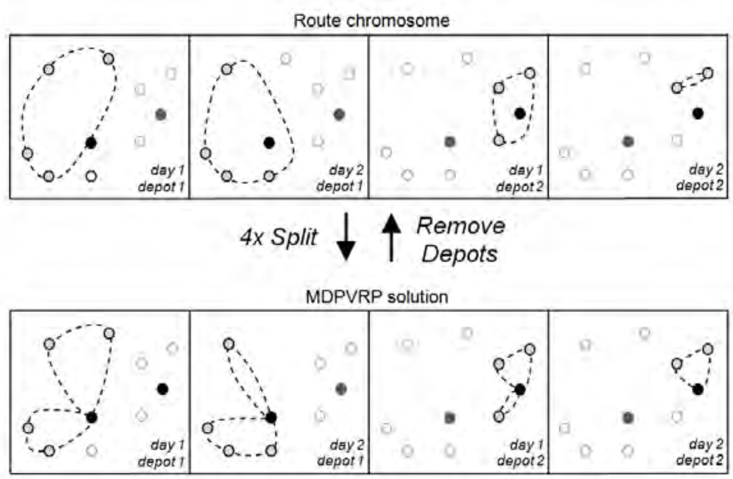

Figure 1: Giant TSP representation of the routes of a MDPVRP solution

The *Split* algorithm described in Prins (2004) reduces the problem of finding the route delimiters to a shortest path problem on a particular graph and operates in $O(n^2)$, where n is the number of customers in the sequence. Chu et al. (2006) extended it to account for fleet-size limitations. We further generalize the algorithm to work with penalized unfeasible solutions.

Let S_i be the customer in position i of the visiting sequence. Define an auxiliary graph $\mathcal{H} = (\mathcal{V}, \mathcal{A})$, where $\mathcal{V}$ contains $n+1$ nodes indexed from 0 to n. For each pair $i < j$, an arc (i,j) represents the trip $r_{i+1,j}$ starting from the depot, visiting customers S_{i+1} to S_j and coming back to the depot. Its associated travel time and load are given by Equations (3) and (4). The arc (i,j) is included in $\mathcal{A}$ only if the total load of the corresponding trip does not exceed a value Q_{max}, which we set at $Q_{max} = 2Q$ to avoid solutions that are too far from feasibility and reduce the number of arcs. The associated arc cost is noted $\phi(r_{i+1,j})$.

$$q(r_{i+1,j}) = \sum_{l=i+1,j} q_{S_l} \quad (3)$$

$$t(r_{i+1,j}) = c_{0,S_{i+1}} + \sum_{l=i+1,j-1} (d_{S_l} + c_{S_l,S_{l+1}}) + d_{S_j} + c_{S_j,0} \quad (4)$$

An optimal segmentation of the giant tour consists in identifying a minimum-cost path from 0 to n in $\mathcal{H}$ containing less than m edges, where m is the number of vehicles available per period. This minimum-cost path can be computed in m iterations of the Bellman-Ford algorithm (see Cormen et al., 2001, for an implementation), each iteration executing in $O(n^2)$. When the demand or the distance between customers is "large", it is possible to impose a bound b on the number of valid trips ending at a given customer i. Thus the complexity of an iteration becomes $O(n \times b)$, and the Split algorithm works in $O(m \times n \times b)$.

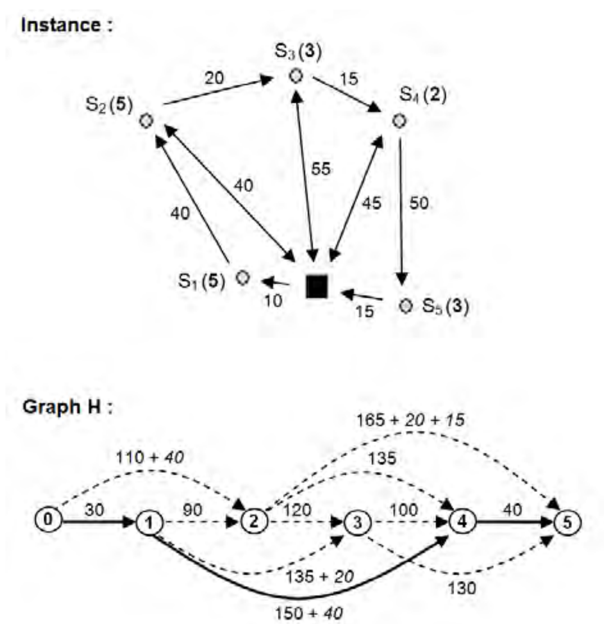

Figure 2: Illustration of a Split graph and shortest-path solution

Figure 2 illustrates the Split algorithm on a sequence of 5 customer visits S_1 to S_5. The first graph presents the problem data: cost of each arc and the customer demands (in bold). In this example, the vehicle capacity is set to $Q = 6$, thus $Q_{max} = 12$, the maximum route duration to $D = 150$, each customer i has an identical service time $d_i = 10$, and the penalty parameters are $\sigma^{\mathrm{Q}} = 10$ and $\sigma^{\mathrm{C}} = 1$. The corresponding graph $\mathcal{H}$ displays on each arc the associated route cost including penalties. Thus, for example, the route servicing customers S_3, S_4, and S_5 has a cost of $165 + 20 + 15$, the penalties of 20 and 15 corresponding to its load excess (two units) and extra duration, respectively. The optimal solution of the minimum-cost path problem, with a cost of 260, is made up of the three following routes: route 1 visits S_1, route 2 visits S_2, S_3, and $S_4\}$, and route 3 visits S_5. Notice that in the actual implementation of the algorithm, there is no need to explicitly build the graph $\mathcal{H}$ (Chu et al., 2006).

4.3 Evaluation of individuals

A major difficulty in population-based algorithms is to avoid premature convergence of the population. The danger of premature convergence is more acute in hybrid methods, where the double impact of education and natural selection heavily favors individuals with good characteristics. Several methods have been proposed to address this issue, e.g., eliminating identical solutions, the so-called clones (Prins, 2004), or imposing inter-individual distance constraints when inserting individuals into the population (Sörensen and Sevaux, 2006; Lozano et al., 2008).

We include some of these ideas into our population-management mechanism (Section 4.6). We also propose, however, a mechanism that addresses the population-diversity issue continuously, during parent and survivor selection and, thus, right at the level of the admittance to the offspring-creation process. Indeed, our very *evaluation function* accounts not only for the fitness of an individual, but also for its contribution to the population diversity, aiming to equilibrate the drive for the best individual (elitism), and the possible loss of information usually associated to this drive. It is thus an adaptive mechanism to control the diversity of the population while still aiming for an elitist behavior of the meta-heuristic.

We define the *diversity contribution* $\Delta(I)$ of an individual I as the average distance to its n_{close} closest neighbors, computed according to Equation (5). Several distances measures were tested in the experiments leading to this final algorithm. A normalized Hamming distance $\delta_H(I_1, I_2)$, based on the differences between the service schedules and depot assignments of two customers I_1 and I_2 appeared the most adequate for the multi-depot, period routing problems we address. This distance is computed according to Equation (6), where $\mathbf{1}(cond)$ is a valuation function that returns 1 if the condition *cond* is true, 0, otherwise.

$$\Delta(I) = \frac{1}{n_{close}} \sum_{J \in N_{close}} \delta_H(I, J) \tag{5}$$

$$\delta_H(I_1, I_2) = \frac{1}{2n} \sum_{i=1}^{n} (\mathbf{1}(\lambda_{I_1}[i] \neq \lambda_{I_2}[i]) + \mathbf{1}(d_{I_1}[i] \neq d_{I_2}[i]) \tag{6}$$

Individuals are sorted by fitness and diversity contribution. Let $fit(I) \in [0, 1]$ and

$dc(I) \in [0, 1]$ be the normalized ranks of individual I relative to the current population according to each criterion, respectively, where the best individual has rank 0 and the worst has rank 1. The *Biased Fitness* function $BF(I)$ given by Equation (7) establishes a trade-off between fitness and diversity, and it is used to evaluate the quality of an individual during parent (Section 4.4) and survivor (Section 4.6) selections. It depends upon the actual number of individuals in the population $nbIndiv$, and a parameter $nbElit$ that corresponds to the desired number of elite solutions for which survival is guaranteed from one generation to the next.

$$BF(I) = fit(I) + \left(1 - \frac{nbElit}{nbIndiv - 1}\right) \times dc(I) \tag{7}$$

4.4 Selection and Crossover

We propose a new *periodic crossover with insertions* (PIX) for the MDPVRP. This operator produces a single child C, which inherits characteristics from its two parents P_1 and P_2. Each parent is selected by a binary tournament. The binary tournament selection consists in picking from the union of the two subpopulations two individuals with uniform probability, and keeping the one with the best Biased Fitness. Both feasible and unfeasible individuals can be selected to undergo crossover in order to lead the research close to the borders of feasibility, where we expect to find optimal solutions.

The crossover operator was designed to transmit good sequences of visits, while enabling pattern, depot and route recombinations. Another very important property of the operator is its versatility, as it allows both a wide exploration of the search space, by combining genetic material from the parents in nearly equal proportions, and small refinements of a "good" solution, by copying almost the totality of a parent, along with small parts of the other one. The whole process takes place in four steps.

Step 1. Determine period and depot inheritance. The first task of the crossover is to determine for each couple (depot, period) whether the genetic material of P_1, P_2 or both is transmitted. Choosing randomly for each depot and period among the three previous cases would not be suitable for the last property stated previously, because when the number of couples becomes large, the amount of information taken from each parent may tend to be balanced because of the law of large numbers. To keep the possibility to focus on one of the parents, we proceed as follows. Two random numbers are first picked between 0 and td according to a uniform distribution; let n_1 and n_2 be the smallest and the largest of these numbers, respectively. n_1 couples (depot, period) are then selected at random to form the set Λ_{P1} and $n_2 - n_1$ of the remaining couples are again randomly picked to form the set Λ_{P2}; the remaining $td - n_2$ couples make up the set Λ_{mix}.

Step 2. Copy data from the first parent. For each depot k and period l, if $(k, l) \in \Lambda_{P1}$ all visits are copied from $s_{P_1}[k, l]$ into $s_C[k, l]$. If $(k, l) \in \Lambda_{mix}$, two cutting points α_{kl} and β_{kl} are generated in $s_{P_1}[k, l]$, and the corresponding substring is copied into $s_C[k, l]$.

Step 3. Copy data from the second parent. For each depot k and period l selected **in random order**, if $(k, l) \in \Lambda_{P1}$, no visit is copied from $s_{P_2}[k, l]$. If $(k, l) \in \Lambda_{P2}$, all the visits of $s_{P_2}[k, l]$ are successively considered, starting from the beginning. A customer visit $i \in s_{P_2}[k, l]$ is copied at the end of $s_C[k, l]$ if the depot choice $d_C[i]$ is undefined or equal to k, and if there exists at least one visit pattern of customer i containing the sub-list $\lambda_C[i] \cup l$. Finally if $(k, l) \in \Lambda_{mix}$, we do the same as when $(k, l) \in \Lambda_{P2}$, but all the visits of $s_{P_2}[k, l]$ are considered circularly from β_{kl}.

Step 4. Fill the remaining services. For each customer whose frequency is not satisfied, additional visits are placed in a random fashion as many times as needed in C via a least cost insertion in the sequence, such that the resulting visit list corresponds to an existing pattern, and the same depot is used for all the visits to a given customer.

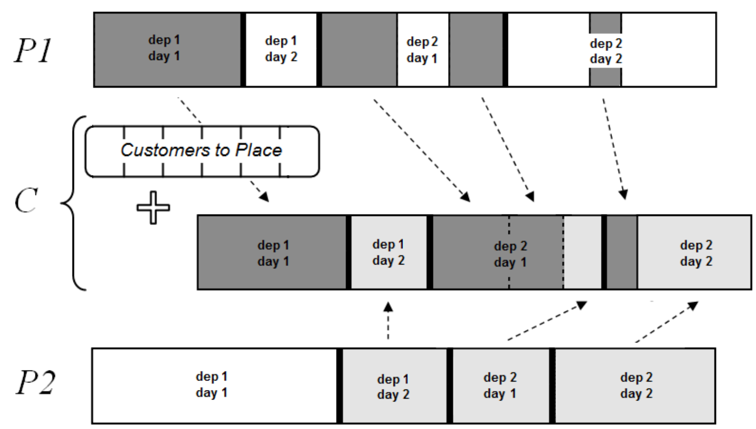

Figure 3: The PIX crossover after Step 3

An illustration of the crossover operator PIX is given in Figure 3. In this exemple $\Lambda_{P1} = \{(1, 1)\}$, hence (depot 1, period 1) of P_1 is copied totally. $\Lambda_{P2} = \{(1, 2)\}$, and for other combinations of periods and depots, C inherits a partial sequence of P_1, with additional visits ($\Lambda_{mix} = \{(2, 1); (2, 2)\}$).

4.5 Education

The offspring resulting from the crossover operation undergoes the *Split* procedure to extract its routes. An *Education* operator is then applied with probability P_m to improve the quality of the new individual. Education goes beyond the classical GA concepts of random mutation and enhancement through hill-climbing techniques, as it includes several local-search procedures based on well-known neighborhoods for the VRP. A *Repair* phase completes the Education operator when the educated offspring is unfeasible.

Two sets of local-search procedures are defined. The nine *route improvement* (RI) procedures are dedicated to optimize each VRP subproblem separately, whereas the *pattern improvement* (PI) procedure relies on a quick and simple move to improve the period assignment of the customers by changing their patterns and depots. These local searches are called in the RI, PI, RI sequence.

Route Improvement. Define the neighborhood of a vertex, customer or depot, u as the $h \times n$ closest vertexes to u, where $h \in [0, 1]$ is a *granularity threshold* restricting the search to nearby vertexes (Toth and Vigo, 2003). Let v be a neighbor of u, x and y the successors of u and v in their respective routes $T(u)$ and $T(v)$, respectively. The notation (u_1, u_2) identifies the route from u_1 to u_2. The Route Improvement phase is performed by browsing in a random order each vertex u and each of its neighbors v. For each couple, we evaluate the following moves:

- (M1) If u is a customer visit, remove u and place it after v.
- (M2) If u and x are customer visits, remove them, then place u and x after v.
- (M3) If u and x are customer visits, remove them, then place x and u after v.
- (M4) If u and v are customer visits, swap u and v.
- (M5) If u, x, and v are customer visits, swap u and x with v.
- (M6) If u, x, v, and y are customer visits, swap u and x with v and y.
- (M7) If $T(u) = T(v)$, replace (u, x) and (v, y) by (u, v) and (x, y).
- (M8) If $T(u) \neq T(v)$, replace (u, x) and (v, y) by (u, v) and (x, y).
- (M9) If $T(u) \neq T(v)$, replace (u, x) and (v, y) by (u, y) and (x, v).

Moves M1 to M6 are applied indifferently on the same or on different routes. M7 is an intra-route move, while the last two are inter-route swaps. The first three moves correspond to *insertions*, the moves M4 to M6 are *swaps*, while moves M7 to M9 are known under the names of *2-opt* and *2-opt**. It is important to browse the nodes in random order as always visiting a given node before another could lead to bias. As soon as an improvement is found the move is performed. The Route-Improvement phase stops when all possible moves have been successively tried without success.

Pattern Improvement. Let $\psi(i, k, l)$ be the minimum insertion cost of customer i in a route from depot k in period l. Let k^* and p^* be the depot and pattern, respectively, of customer i in the current solution. The Pattern-Improvement procedure iterates on each customer i in a random order and computes $\Psi(i,k,p) = \sum_{l \in p} \psi(i,k,l)$ for each depot k and pattern $p \in L_i$. If $\Psi(i,k,p) < \Psi(i,k^*,p^*)$, then all visits to customer i are removed, and a new visit is inserted in the best location in each sequence corresponding to depot k and period $l \in p$. The procedure stops when all customers have been successively considered without success.

The Pattern-Improvement procedure is significantly faster when the optimal position and insertion cost of each customer is stored for each route. The proportion of routes changed by a move, and the resulting loss of information, is generally small compared to the total number of routes for all depots and periods of the problem. It is also worth noting that, sometimes, the current pattern and depot choices are kept, but a better insertion of the customers is found. The resulting move is then, in fact, a combination of intra-day M1 insertions. This may prove particularly interesting for the exceptional case when the move was not attempted in RI because of proximity conditions. The Pattern-Improvement phase thus fulfills the double role of changing the patterns, and attempting moves between distant vertexes.

A feasible individual yielded by the RI, PI, RI sequence is, of course, inserted into the feasible subpopulation. An unfeasible one is either subjected to the Repair procedure with probability P_{rep}, or inserted into the unfeasible subpopulation. If Repair is successful, the resulting individual is added to the feasible subpopulation.

Repair consists in temporarily multiplying the penalty parameters by 10 and restarting the RI, PI, RI sequence. When the resulting individual is still unfeasible, penalty parameters are multiplied by 100 and the sequence is started again. This significant increase in the value of the penalties aims to redirect the search toward feasible solutions. The individual is discarded in case of failure. Computational experiments showed that Repair is crucial to attain good solutions on many tightly constrained instances, e.g., when total vehicle capacity is close to the total demand or the route-duration limit is too tight for the number of available vehicles.

4.6 Population management

The population management mechanism complements the selection-crossover-education operators in identifying the characteristics of good solutions and providing the means for a thorough and efficient search. We propose thus a mechanism with the dual purpose of preserving the accumulated memory of successive selections for the most promising solution characteristics and the diversity of the individuals in the population.

The mechanism manages both the feasible and the unfeasible subpopulations and is made up of three components: the initialization of the populations, a diversification scheme applied when the search does not seem to advance at a purposeful pace, and the procedure to select the individuals that will survive to the next "generation" when a particular subpopulation reaches its maximum size.

Each subpopulation contains between μ and $\mu + \lambda$ individuals. To initialize them, 4μ individuals are created by randomly choosing a pattern for each customer and producing for each period the associated service sequence in random order. Each one of these initial individuals undergoes education, and half of those are repaired (if needed). The resulting solutions are stocked in the appropriate subpopulation. If either one of the two subpopulations reaches the maximum size $\mu + \lambda$, λ individuals are discarded as explained later, in the survivor-selection part. At the end of the Initialization phase, one of the two subpopulations can be incomplete, having less than μ individuals.

During the search, diversification is called whenever It_{div} iterations are performed without improving the best solution. Diversification is performed by retaining only the best $\mu/3$ individuals of each subpopulation, and completing with 4μ new individuals as in the Initialization phase. This process introduces a significant amount of new genetic material, which helps to pursue the search further, even when the population has lost most of its diversity.

The proposed meta-heuristic proceeds by keeping the parents following crossover and having the offspring start to reproduce immediately after education. The populations therefore grow continuously. As the maximum size of $\mu + \lambda$ individuals in one subpopulation is reached, it is reduced by discarding λ individuals. The remaining μ survivors provide the next generation. The selection of the survivors aims to preserve the population diversity in terms of visit patterns, profit from new individuals, and protect an elite. We thus aim to eliminate "clones" and solutions that are bad both in fitness and contribution to diversity, as measured by their Biased Fitness function value.

Let a *clone* be an individual I_2 with the same attributes as another individual I_1, i.e., $\delta_H(I_1, I_2) = 0$, or the same fitness. The selection of the survivors is then performed as the successive elimination of, first, the clones, and, then, bad solutions, until λ individuals are discarded:

- Let X be the set of all individuals, different from the current best solution, having a "clone";
- If $X \neq \emptyset$, remove $I \in X$ with maximum Biased Fitness;
- Otherwise, remove I in the subpopulation with maximum Biased Fitness;
- Update the distance measures and X, and repeat.

We can now state an important elitism property. PROPOSITION *Using the Biased*

Fitness function, if an individual $I \notin X$ is part of the nbElit best individuals of the subpopulation in terms of fitness, then I will not be removed from the population by the previous survivor-selection procedure.

Proof. *Let I be the individual with the worst fitness, thus $fit(I) = 1$ and $BF(I) = fit(I) + (1 - \frac{nbElit}{nbIndiv-1}) \times dc(I) \geq 1$. Let J be an individual among the best nbElit solutions in terms of fitness, thus $BF(J) \leq \frac{nbElit-1}{nbIndiv-1} + 1 - \frac{nbElit}{nbIndiv-1} < 1$. Individual J will not be removed as I has a worst biased fitness.*

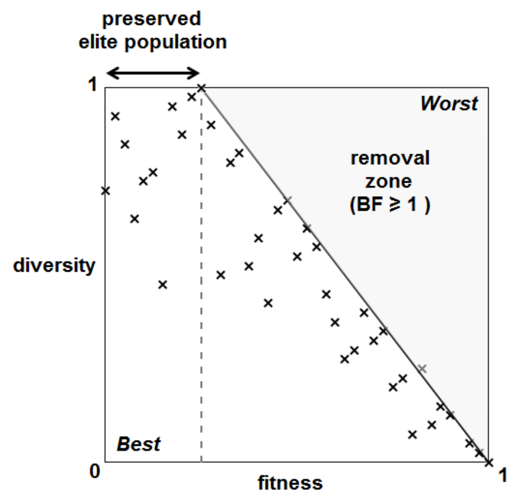

Figure 4: Illustration of the survivor selection property

Figure 4 represents the fitness and diversity measures of a subpopulation taken from our experimentations after the survivor-selection procedure was run. As stated by the previous proposition, this removal policy calls for *nbElit* individuals to be considered an elite. The figure also displays the *removal zone*, where individuals with with $BF \geq 1$ can be eliminated according to the Biased Fitness criteria. The procedure favors individuals with excellent fitness and bad diversity over solutions with bad fitness and excellent diversity, as the former are difficult to find. This proved to be a success factor in the experiments leading to the final algorithm.

5 Computational Experiments

Previous work on the PVRP and the MDVRP has led to the development of a large collection of test problems that have been used extensively to provide comparisons between algorithms. Cordeau et al. (1997) describes two sets of instances. The first gathers several problems proposed in previous article, while the second was generated by the authors. The size of these instances range from 50 to 417 customers. Some instances in the PVRP set are particularly large, involving up to 900 visits to customers over the different periods. Detailed information on these instances is provided in Subsection 5.2.

No benchmark was available for the MDPVRP. We therefore built a set of 10 MDPVRP instances by merging the PVRP and the MDVRP instances of the second set provided by Cordeau et al. (1997). Each of the 10 MDVRP instances was combined with the PVRP instance with the same number of customers. The number of periods and the patterns were taken from the PVRP instance, the depots from the MDVRP one, and the number of vehicles was fixed to the smallest number such that a feasible solution may be found by HGSADC (and probably by most any other algorithm). The full data sets can be obtained from the authors.

The proposed meta-heuristic was coded in C++, and run on a 2.4 Ghz AMD Opteron 250 CPU workstation. To allow for easier comparisons with previous work, all CPU times reported in this section were converted into their equivalent Pentium IV 3.0 Ghz run times using Dongarra (2009) factors. Before providing detailed computational results, we first explain how we calibrated the algorithm.

5.1 Calibration of the HGSADC algorithm

Similarly to most meta-heuristics, evolutionary ones in particular, HGSADC relies on a set of correlated parameters and configuration choices for its key operators. able 1 provides a summary of the HGSADC parameters, together with the range of values we estimate as appropriate due to either the parameter definition (e.g., probabilities and proportions), conceptual requirements (a local distance measure is assumed to implicate not more than 25% of the population), or values found in the literature (e.g., subpopulations sizes). Calibration aims to identify values for these parameters to ensure good performance of the algorithm over a varied set of instances. It is thus often performed by running the algorithm with a number of parameter-value combinations on a limited, hopefully representative, set of instances.

A different approach is offered by the principle of *meta-calibration*, which aims to optimize the set of parameters of a given algorithm by means of meta-heuristics. Early studies on meta-calibration can be found in Mercer and Sampson (1978). More recently, Smit and Eiben (2009) provide a comparison of parameter-tuning methods, and emphasize the efficiency of calibrating genetic algorithms by means of a meta-evolutionary algorithm.

We adopted this approach and performed the calibration of the HGSADC algorithm with a meta-evolutionary method, namely, the Evolutionary Strategy with Covariance Matrix Adaptation (CMA-ES) of Hansen and Ostermeier (2001). The source code is available at (http://www.lri.fr/˜hansen/cmaes_inmatlab.html). Retrieving the fitness of a given set of parameters implicates to launch automatically HGSADC on a restricted set of *training instances* and measure its efficiency. The calibration was thus performed independently for each problem class, with the dual objective of measuring the dependency of the best parameter set upon the problem class, and identifyuing an eventual set of parameters suitable for all problem classes considered. The calibration results for each class, along with the final choice of parameter values for HGSADC, are presented in Table 2.

Except for the generation size λ, the optimum set of parameters appears independent of the problem type. We therefore averaged these results to get the final parameter valuess of Table 2, with the exception of the probability to educate a new individual (the education rate P_m). Calibrated education rates are generally very high, with an average value of 0.8. Additional tests indicated similarly good performance as long as $P_m \geq 0.7$. Hence we selected the value $P_m = 1$, which corresponds to a systematic education of all individuals, and reduces the number of parameters in use. The only parameter that is problem dependent is λ, which is set to 40 for the PVRP, 70 for MDVRP, and 100 for the MDPVRP.

5.2 Results on PVRP and MDVRP instances

For comparison with other algorithms on Cordeau et al. (1997) instances, we report the results of HGSADC with $It_{NI} = 10000$, $It_{div} = 4000$, and $T_{max} = 10min$. The benchmark contains some very large problems with more than 450 visits to customers, for which the population size was reduced by two, and the computation time limit was increased.

Results on PVRP benchmarks are presented in Table 3. The first four columns display the instance identifier and the numbers of customers, vehicles, and periods, respectively. The results of of HGSADC are shown in Columns 8 and 9 as the average results and CPU times on 10 runs. We compare the performance of HGSADC to that of state-of-the-art methods for the PVRP: the tabu search of Cordeau et al. (1997) (CGL in Column 5), the scatter search of Alegre et al. (2007) (ALP, Column 6), and the average results of 19 runs of the variable neighborhood search of Hemmelmayr et al. (2009) (HDH, Column 7). We indicate in boldface the best average result among the four algorithms for each instance, as well as, in the last two columns, the previous best-known solution (BKS), and the best solution obtained by HGSADC during all our experiments. Optimality has been proved for 14 instances by Baldacci et al. (2010). (Due to numerical approximations, optimality was proved within a 0.02% precision and, hence, it is sometimes possible to find a slightly better solution.) These solutions are marked with *. Several upper bounds have been improved, and the corresponding new state-of-the-art solutions are underlined.

The last two lines provide average measures over the 42 instances: the average per-

Table 1: Parameters of HGSADC

Population and distance measures parameters		
μ	Population size	[5,200]
λ	Number of offspring in a generation	[1,200]
el	Proportion of elite individuals, such that $nbElit = el \times \mu$	[0,1]
nc	Proportion of close individuals considered for distance evaluation, such that $n_{close} = nc \times \mu$	[0,0.25]
Genetic operators parameters		
P_m	Education rate	[0,1]
P_{rep}	Repair rate	[0,1]
h	Granularity threshold in RI	[0,1]
Adaptive penalty parameters		
ξ^{REF}	Reference proportion of feasible individuals	[0,1]

Table 2: Calibration Results

Parameter	PVRP Set	MDVRP Set	MDPVRP Set	**Final parameter values**
μ	18	24	30	**25**
λ	33	87	146	**40 / 70 / 100**
el	0.38	0.45	0.36	**0.4**
nc	0.24	0.18	0.15	**0.2**
P_m	0.86	0.86	0.70	**1.0**
P_{rep}	0.57	0.61	0.33	**0.5**
h	0.53	0.36	0.35	**0.4**
ξ^{REF}	0.10	0.30	0.20	**0.2**

centage of error relative to the previous BKS, and the average computation time of each method. Because CGL was tested on a rather old computer, it was hard to make the conversion of the run time using Dongarra (2009) factors. However, Hemmelmayr et al. (2009) ran the CGL code on a Pentium IV 3.2 Ghz and provided updated computation times.

HGSADC produces high quality solutions, with an average overall deviation of +0.20% relative to the previous BKS compared to more than +1.40% for the other approaches. The method we propose yields the best average results on every instance, but one (p06). It should be noted that, in the case of problem p06, every customer has a frequency of one and can be served on any period. The problem may therefore be addressed as a CVRP. This induces symmetry, which results in many different representations in terms of periods for the same solution. These may then proliferate in the population and can thus explain why the algorithm is less efficient for this particular instance. The average computation time is reasonably short (5.56 min), and is suitable for many operational decisions. Although higher than for the other methods, the run times remain comparable (i.e., in a similar range of values).

It is noteworthy that the average standard deviation per instance obtained by HGSADC is 0.15%, meaning that the algorithm is very reliable. It also performs at a high level in terms of solution quality. All the previous best-known solutions were found, and new reference results were obtained for 20 instances out of 42. The proposed algorithm almost always retrieves the optimal values for the 14 solutions for which the optimal values are known. An average deviation of only 0.09% to the optimum may be observed for these instances.

To observe the behavior of HGSADC when the number of iterations increases, we provide additional results with three termination criteria allowing more computation time: $(It_{NI}, T_{max}) = (2.10^4, 30min)$, and $(It_{NI}, T_{max}) = (5.10^4, 1h)$. Table 4 compares the average results of HGSADC to those obtained by Hemmelmayr et al. (2009) using various numbers of iterations. No running times were reported by Hemmelmayr et al. (2009) for these experiments, but it seems reasonable to expect them to grow linearly with the number of iterations.

The figures of Table 4 indicate that the proposed algorithm scales well when the available computation time is increased. It should also be pointed out that even long runs of 10^9 iterations of the HDH algorithm (100 times the number of iterations in the standard HDH method and probably around 300 minutes of computing time) produce inferior average results compared to those obtained by HGSADC in a few minutes.

Results on MDVRP benchmarks of both sets of Cordeau et al. (1997) are displayed in Table 5, where Columns 2 to 4 indicate the numbers of customers, vehicles, and depots, respectively. The results of HGSADC are compared to those of the state-of-the-

Table 3: Results on Cordeau et al. (1997) PVRP instances

				Average					BKS	
Inst	**n**	**m**	**p**	**CGL**	**ALP**	**HDH**	**HGSADC**	**T(min)**	**prev BKS**	**HGSADC**
				(1 run)	—	(10 runs)	(10 runs)		—	(all exp.)
p01	50	3	2	**524.61**	531.02	**524.61**	**524.61**	0.22	524.61*	524.61*
p02	50	3	5	1330.09	1324.74	1332.01	**1322.87**	0.44	1322.87	1322.87
p03	50	1	5	**524.61**	537.37	528.97	**524.61**	0.18	524.61*	524.61*
p04	75	6	5	837.94	845.97	847.48	**836.59**	1.05	835.26*	835.26*
p05	75	1	10	2061.36	2043.74	2059.74	**2033.72**	2.27	2027.99	2024.96
p06	75	1	10	840.30	**840.10**	884.69	842.48	0.89	835.26*	835.26*
p07	100	4	2	829.37	829.65	829.92	**827.02**	0.88	826.14	826.14
p08	100	5	5	2054.90	2052.51	2058.36	**2022.85**	2.54	2034.15	2022.47
p09	100	1	8	829.45	829.65	834.92	**826.94**	1.01	826.14	826.14
p10	100	4	5	1629.96	1621.21	1629.76	**1605.22**	1.80	1593.45	1593.43
p11	126	4	5	817.56	782.17	791.18	**775.84**	4.60	779.06	770.89
p12	163	3	5	1239.58	1230.95	1258.46	**1195.29**	5.34	1195.88	1186.47
p13	417	9	7	3602.76	—	3835.90	**3599.86**	40.00	3511.62	3492.89
p14	20	2	4	**954.81**	**954.81**	**954.81**	**954.81**	0.08	954.81*	954.81*
p15	38	2	4	**1862.63**	**1862.63**	**1862.63**	**1862.63**	0.17	1862.63*	1862.63*
p16	56	2	4	**2875.24**	**2875.24**	**2875.24**	**2875.24**	0.32	2875.24*	2875.24*
p17	40	4	4	**1597.75**	**1597.75**	1601.75	**1597.75**	0.27	1597.75*	1597.75*
p18	76	4	4	3159.22	3157.00	3147.91	**3131.09**	0.89	3136.69	3131.09
p19	112	4	4	4902.64	4846.49	4851.41	**4834.50**	2.26	4834.34	4834.34
p20	184	4	4	**8367.40**	8412.02	**8367.40**	**8367.40**	4.01	8367.40	8367.40
p21	60	6	4	2184.04	2173.58	2180.33	**2170.61**	0.90	2170.61*	2170.61*
p22	114	6	4	4307.19	4330.59	4218.46	**4194.23**	4.27	4193.95	4193.95
p23	168	6	4	6620.50	6813.45	6644.93	**6434.10**	4.29	6420.71*	6420.71*
p24	51	3	6	3704.11	3702.02	3704.60	**3687.46**	0.32	3687.46*	3687.46*
p25	51	3	6	3781.38	3781.38	3781.38	**3777.15**	0.59	3777.15*	3777.15*
p26	51	3	6	**3795.32**	3795.33	**3795.32**	**3795.32**	0.33	3795.32*	3795.32*
p27	102	6	6	23017.45	22561.33	22153.31	**21885.70**	3.52	21912.85	21833.87
p28	102	6	6	22569.40	22562.44	22418.52	**22272.60**	4.67	22246.69*	22242.51
p29	102	6	6	24012.92	23752.15	22864.23	**22564.05**	3.86	22543.75*	22543.75*
p30	153	9	6	77179.33	76793.99	75579.23	**74534.38**	9.99	74464.26	73875.19
p31	153	9	6	79382.35	77944.79	77459.14	**76686.65**	10.00	76322.04	76001.57
p32	153	9	6	80908.95	81055.52	79487.97	**78168.82**	10.00	78072.88	77598.00
pr01	48	2	4	2234.23	—	2209.11	**2209.02**	0.29	2209.02	2209.02
pr02	96	4	4	3836.49	—	3787.51	**3768.86**	2.49	3774.09	3767.50
pr03	144	6	4	5277.62	—	5243.09	**5174.80**	7.32	5175.15	5153.54
pr04	192	8	4	6072.67	—	6011.39	**5936.16**	10.00	5914.93	5877.37
pr05	240	10	4	6769.80	—	6778.00	**6651.76**	20.00	6618.95	6581.86
pr06	288	12	4	8462.37	—	8461.45	**8284.94**	20.00	8258.08	8207.21
pr07	72	3	6	5000.90	—	5007.01	**4996.14**	1.49	4996.14	4996.14
pr08	144	6	6	7183.39	—	7119.61	**7035.52**	10.00	6989.81	6970.68
pr09	216	9	6	10507.34	—	10259.09	**10162.22**	20.00	10075.40	10038.43
pr10	288	12	6	13629.25	—	13342.41	**13091.00**	20.00	12924.66	12897.01
Avg Gap to BKS				+1.82%	+1.40%	+1.45%	+0.20%			
Avg Time				4.28 min	3.64 min	3.34 min	5.56 min			

art methods for this problem: the tabu search of Cordeau et al. (1997) (CGL, Column 5) and the Adaptive Large Neighborhood Search of Pisinger and Ropke (2007) (PR, Column 6). The table format is identical to the one of Table 3.

The main conclusions are similar to those stated above for the PVRP. HGSADC is by far the most effective method overall, but it requires somewhat longer computing times than the two others to achieve its superior results. An average error gap of -0.01% is obtained, meaning that the new method is on average better than the previous best known solutions of each instance. Only 2.15 min are required, on average, to find the final solution, but then, an equal amount of time is spent to reach the time-limit termination criterion, thus resulting in an average total run time of 4.24 min. Computation times can therefore be drastically reduced by reducing the values of the termination criteria.

The solutions obtained by HGSADC are usually very close to the best-known solutions. For 23 instances out of 33, all 10 runs converged to the best-known solution. The average standard deviation per instance is again small, 0.05%, which highlights the reliability of the method for this problem class as well.

Baldacci and Mingozzi (2009) recently solved several of these instances to optimality, but without taking into account the limited fleet constraint. Hence, these solutions can be seen as lower bounds to the optimal values. For five instances (p1, p2, p6, p7 and p12), these values coincide with the best available upper bound for the problem with the constraint on the number of vehicles, which indicates that the considered solutions are optimal for the MDVRP with limited fleet. HGSADC always reached the optimum solution for four out of these five instances.

5.3 Results on MDPVRP instances

We tested HGSADC on the new set of MDPVRP instances. As these problems are harder to solve, the maximum running time was increased to 30 minutes (It_{NI} remains set to 10^4). Table 6 reports the average and best results on 10 runs for each instance. To assess the performance of the algorithm, these results are compared to the best solutions ever found during all our experiments.

The average error gap compared to the BKS is +0.42%, which is reasonable given the increased problem difficulty. Clearly, HGSADC results vary in accordance with the problem difficulty. The average gap to the BKS ranges from 0.00% to 1.50% (for the larger problem pr10). On small instances like pr01 and pr02, the algorithm seems to always converge toward the same solution, while problems pr08 to pr10, with a larger number of depots and periods, seem particularly difficult to address. The average standard deviation per instance is 0.26%, which is higher than on previous problem classes, and may illustrate the increased irregularity of the search space.

Table 4: Behaviour of PVRP methods on Cordeau et al. (1997) instances, when the run time increases

	CGL	**HDH**	**HDH**	**HDH**	**HGSADC**	**HGSADC**	**HGSADC**
	15.10^3 it	10^7 it	10^8 it	10^9 it	(10^4,10min)	(2.10^4,30min)	(5.10^4,1h)
T	3.96 min	3.09 min	—	—	5.56 min	13.74 min	28.21 min
%	+1.82%	+1.45%	+0.76%	+0.39%	+0.20%	+0.12%	+0.07%

Table 5: Results on Cordeau et al. (1997) MDVRP instances

				Average				BKS	
Inst	**n**	**m**	**d**	**CGL**	**PR**	**HGSADC**	**T(min)**	**prev BKS**	**HGSADC**
				(1 run)	(5-10 runs)	(10 runs)		—	(all exp.)
p01	50	4	4	**576.87**	**576.87**	**576.87**	0.23	576.87*	576.87*
p02	50	2	4	**473.87**	**473.53**	**473.53**	0.21	473.53*	473.53*
p03	75	3	2	645.15	**641.19**	**641.19**	0.43	641.19	641.19
p04	100	8	2	1006.66	1006.09	**1001.23**	1.94	1001.04	1001.04
p05	100	5	2	753.34	752.34	**750.03**	1.06	750.03	750.03
p06	100	6	3	877.84	883.01	**876.50**	1.14	876.50*	876.50*
p07	100	4	4	891.95	889.36	**884.43**	1.55	881.97*	881.97*
p08	249	14	2	4482.44	4421.03	**4397.42**	10.00	4387.38	<u>4372.78</u>
p09	249	12	3	3920.85	3892.50	**3868.59**	9.50	3873.64	<u>3858.66</u>
p10	249	8	4	3714.65	3666.85	**3636.08**	9.82	3650.04	<u>3631.11</u>
p11	249	6	5	3580.84	3573.23	**3548.25**	7.14	3546.06	3546.06
p12	80	5	2	**1318.95**	1319.13	**1318.95**	0.52	1318.95*	1318.95*
p13	80	5	2	**1318.95**	**1318.95**	**1318.95**	0.57	1318.95	1318.95
p14	80	5	2	**1360.12**	**1360.12**	**1360.12**	0.55	1360.12	1360.12
p15	160	5	4	2534.13	2519.64	**2505.42**	1.92	2505.42	2505.42
p16	160	5	4	**2572.23**	2573.95	**2572.23**	1.97	2572.23	2572.23
p17	160	5	4	2720.23	**2709.09**	**2709.09**	2.14	2709.09	2709.09
p18	240	5	6	3710.49	3736.53	**3702.85**	4.52	3702.85	3702.85
p19	240	5	6	**3827.06**	3838.76	**3827.06**	4.20	3827.06	3827.06
p20	240	5	6	**4058.07**	4064.76	**4058.07**	4.37	4058.07	4058.07
p21	360	5	9	5535.99	5501.58	**5476.41**	10.00	5474.84	5474.84
p22	360	5	9	5716.01	5722.19	**5702.16**	10.00	5702.16	5702.16
p23	360	5	9	6139.73	6092.66	**6078.75**	10.00	6078.75	6078.75
pr01	48	2	4	**861.32**	**861.32**	**861.32**	0.17	861.32	861.32
pr02	96	4	4	1314.99	1308.17	**1307.34**	0.76	1307.34	1307.34
pr03	144	6	4	1815.62	1810.66	**1803.80**	1.91	1806.60	<u>1803.80</u>
pr04	192	8	4	2094.24	2073.16	**2059.36**	5.22	2060.93	<u>2058.31</u>
pr05	240	10	4	2408.10	2350.31	**2340.29**	9.56	2337.84	<u>2331.20</u>
pr06	288	12	4	2768.13	2695.74	**2681.93**	10.00	2685.35	<u>2676.30</u>
pr07	72	3	6	1092.12	**1089.56**	**1089.56**	0.34	1089.56	1089.56
pr08	144	6	6	1676.26	1675.74	**1665.05**	2.05	1664.85	1664.85
pr09	216	9	6	2176.79	2144.84	**2134.17**	6.10	2136.42	<u>2133.20</u>
pr10	288	12	6	3089.62	2905.43	**2886.59**	10.00	2889.49	<u>2868.26</u>
Avg Gap to BKS				+0.96%	+0.34%	-0.01%			
Avg Time				small	3.54 min	4.24 min			

Notice that keeping the best solution of 10 separate runs leads to significantly better solutions, with an average error gap of +0.13%, but also requires more computational resources. This approach corresponds to the well-known independent-search strategy for parallel meta-heuristics (Crainic and Nourredine, 2005; Crainic and Toulouse, 2010). More sophisticated strategies, based on cooperation, in particular, could certainly be used to improve the exploration of the search space and reach better results.

5.4 Sensitivity analysis on diversity management methods

The adaptive population diversity control mechanism we propose is a cornerstone of the proposed methodology. We therefore compared its performance to those of two mechanisms proposed in the literature, mechanisms that proved their worth in their respective contexts. Two new algorithms were thus derived from HGSADC to conform exactly to each of these two rules, as well as a variant without any diversity control (identified as *HGS0*).

The *HGS1* variant involves a dispersal rule in the objective space as in Prins (2004). Let F be the fitness function and Δ_F a fitness spacing parameter. A new individual I is then added to the population only if $|F(I) - F(C)| \geq \Delta_F$ for all individuals C already in the population. An incremental population management scheme is used and only individuals with a fitness below the median of the population can be discarded.

The second variant, named *HGS2*, relies on the population management framework of Sörensen and Sevaux (2006). The insertion of a new individual in the population is accepted only if a dispersal rule in the solution space is fulfilled. Let Δ_D be a spacing parameter, and δ_H be the distance measure presented in Section 4.3. In order to be added to the population, an individual I must verify $\delta_H(I, C) \geq \Delta_D$ for all C already in the population. In our implementation, the value of Δ_D changes during run time: strong distance constraints are imposed at the beginning of the search to encourage exploration, whereas as the method approaches the termination criteria, the value of Δ_D decreases progressively toward zero to encourage the exploitation of the good solutions. As in the previous method, we use an incremental population management and only individuals with a fitness below the median of the population can be discarded.

The three algorithms were tested on the benchmark instances presented before, and Table 7 reports the average error gap to the best-known solutions obtained by each algorithm for each group of instances, as well as the average run time of each method. One observes that the results verify that applying the dispersal rule with respect to the solution space (HGS2) is more effective than using the dispersal rule with respect to the objective space(HGS1). This is an indication that the hybrid evolution strategy of HGSADC performs as expected. One also observes that proceeding without diversity management yields rather poor results compared to all other strategies. The best results

Table 6: Results on new MDPVRP instances

Inst	**n**	**m**	**d**	**t**	**Average**	**T(min)** (10 runs)	**Best**	**BKS** (all exp.)
pr01	48	1	4	4	**2019.07**	0.35	**2019.07**	2019.07
pr02	96	1	4	4	**3547.45**	1.49	**3547.45**	3547.45
pr03	144	2	4	4	4491.08	7.72	**4480.87**	4480.87
pr04	192	2	4	4	5151.73	22.10	5144.41	5134.17
pr05	240	3	4	4	5605.60	30.00	5581.10	5570.45
pr06	288	3	4	4	6570.28	30.00	6549.57	6524.92
pr07	72	1	6	6	4502.06	2.18	**4502.02**	4502.02
pr08	144	1	6	6	6029.58	7.96	**6023.98**	6023.98
pr09	216	2	6	6	8310.19	27.79	8271.66	8257.80
pr10	288	3	6	6	9972.35	30.00	9852.87	9818.42
Avg Gap to BKS					+0.42%		+0.13%	
Avg Time					15.96 min		*159.6 min*	

Table 7: Comparison of population-diversity management mechanisms

Benchmark		**HGS0**	**HGS1**	**HGS2**	**HGSADC**
PVRP	T	4.68 min	5.15 min	5.37 min	5.56 min
	%	+0.70%	+0.62%	+0.39%	+0.20%
MDVRP	T	3.37 min	3.55 min	4.49 min	4.24 min
	%	+0.80%	+0.61%	+0.10%	-0.01%
MDPVRP	T	13.16 min	14.00 min	15.94 min	15.96 min
	%	+2.95%	+2.95%	+2.37%	+0.42%

are definitely obtained with the adaptive diversity management method we propose, which yields the best average error gap for an equivalent computational effort.

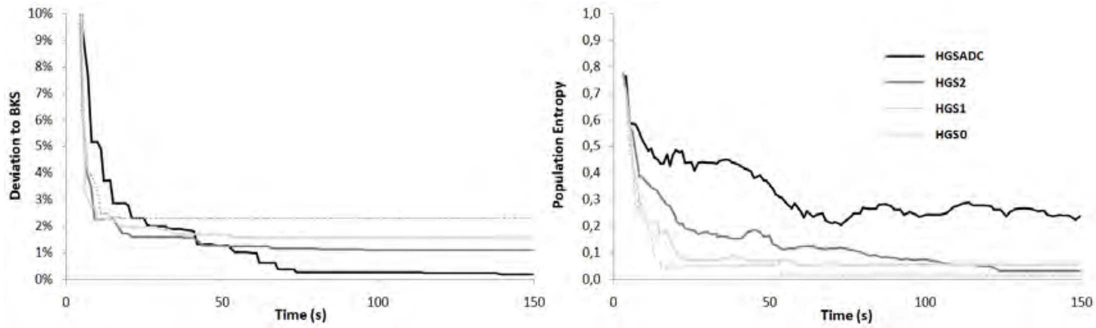

Figure 5: Population entropy and error gap to the BKS for the diversity management strategies on instance p03

Figure 5 illustration the behavior of the four population-diversity management strategies during one of the runs (150 seconds) for instance p03, as measured by the population entropy and the error gap to the BKS. The population entropy is computed as the average distance from one individual to another. All algorithms close the error gap to less than 2.50% in a few seconds. The methods that use diversity management are able, however, to efficiently continue searching and, thus, to reach better solutions. The proposed HGSADC meta-heuristic is still regularly improving its best found solution as the time limit approaches, despite being already very close to the best-known solution (the error gap is 0.19% only). The no-diversity management strategy, HGS0, provides a perfect example of premature convergence. In less than one minute, one observes no additional improvement of the best solution, very low entropy, and quite likely very little evolution in the population. HGSADC maintains a healthy diversity in the population, as illustrated by rather high level of entropy at 0.3. In comparison, the two alternate strategies, HGS1 and HGS2, display lower entropy levels, around 0.1.

We conclude that the proposed diversity management mechanism is particularly effective for the problem classes considered in this paper. In the experiments we conducted, it allowed to avoid premature convergence and to reach high quality solutions.

6 Conclusions and Research Perspectives

We proposed in this paper a new hybrid genetic search meta-heuristic to efficiently address several classes of multi-depot and periodic vehicle routing problems, for which few efficient algorithms are currently available. Given the great practical interest of the problem considered, the proposed methodology opens the way to significant progress in the optimization of distribution networks.

The papers introduces several methodological contributions, in particular, in the crossover and education operators, the management of unfeasible solutions, the individual evaluation procedure driven both by fitness and the contribution to population diversity and, more generally, the adaptive population management mechanism that enhances diversity, allows a broader access to reproduction, and preserving the memory of what characterizes good solutions represented by the elite individuals. The combination of these concepts provides the capability of the proposed *Hybrid Genetic Search with Adaptive Diversity Control* meta-heuristic to reach high quality solutions on the literature benchmarks. The method actually identifies either the best known solutions, including the optimal ones, or new best solutions for all benchmark instances, thus outperforming the current state-of-the-art meta-heuristics for each particular problem class.

Among the many avenues of research that remain open, we mention in particular the interest to explore the impact of the adaptive diversity control mechanism for other classes of problems, and to validate its good performance using theoretical models. We also plan to generalize the methodology to problems with additional attributes, and thus progress toward addressing *rich VRP* problem settings as well as real world applications.

Acknowledgments

While working on this project, T.G. Crainic was the NSERC Industrial Research Chair on Logistics Management, ESG UQAM, and Adjunct Professor with the Department of Computer Science and Operations Research, Université de Montréal, and the Department of Economics and Business Administration, Molde University College, Norway, while N. Lahrichi was postdoctoral scientist with the Chair. M. Gendreau was the NSERC/Hydro-Québec Industrial Research Chair on the Stochastic Optimization of Electricity Generation, MAGI, École Polytechnique, and Adjunct Professor with the Department of Computer Science and Operations Research, Université de Montréal.

Partial funding for this project has been provided by the Natural Sciences and Engineering Council of Canada (NSERC), through its Industrial Research Chair and Discovery Grant programs, by the partners of the Chair, CN, Rona, Alimentation Couche-Tard and the Ministry of Transportation of Québec, and by the Fonds québécois de la recherche sur la nature et les technologies (FQRNT) through its Team Research Project program.

References

J. Alegre, M. Laguna, and J. Pacheco. Optimizing the periodic pick-up of raw materials for a manufacturer of auto parts. *European Journal of Operational Research*, 179(3): 736–746, 2007.

R. Baldacci and A. Mingozzi. A unified exact method for solving different classes of vehicle routing problems. *Mathematical Programming A*, 120(2):347–380, 2009.

R. Baldacci, E. Bartolini, A. Mingozzi, and A. Valletta. An exact algorithm for the periodic routing problem. *Operations Research*, 2010. to appear.

E. J. Beltrami and L. D. Bodin. Networks and vehicle routing for municipal waste collection. *Networks*, 4:65–94, 1974.

F. Blakeley, B. Bozkaya, B. Cao, W. Hall, and J. Knolmajer. Optimizing periodic maintenance operations for schindler elevator corporation. *Interfaces*, 33(1):67–79, 2003. ISSN 0092-2102.

L. Bodin, B. Golden, A. Assad, and M. Ball. Routing and scheduling of vehicles and crews: The state of the art. *Computers & Operations Research*, 10(2):63–211, 1983.

O. Bräysy and M. Gendreau. Vehicle Routing Problem with Time Windows, Part I: Route Construction and Local Search Algorithms. *Transportation Science*, 39(1):104–118, 2005a.

O. Bräysy and M. Gendreau. Vehicle Routing Problem with Time Windows, Part II: Metaheuristics. *Transportation Science*, 39(1):19–139, 2005b.

O. Bräysy, W. Dullaert, and M. Gendreau. Evolutionary algorithms for the vehicle routing problem with time windows. *Journal of Heuristics*, 10(6):587–611, 2004. ISSN 1381-1231.

O. Bräysy, W. Dullaert, and M. Gendreau. Evolutionary algorithms for the vehicle routing problem with time windows. *Journal of Heuristics*, 10(6):587–611, 2004.

O. Bräysy, M. Gendreau, G. Hasle, and A. Løkketangen. A Survey of Heuristics for the Vehicle Routing Problem Part I: Basic Problems and Supply Side Extensions. Technical report, SINTEF Report, 2008a.

O. Bräysy, M. Gendreau, G. Hasle, and A. Løkketangen. A Survey of Heuristics for the Vehicle Routing Problem Part II: Demand Side Extensions. Technical report, SINTEF Report, 2008b.

M. W. Carter, J. Farolden, G. Laporte, and J. Xu. Solving an integrated logistics problem arising in grocery distribution. *INFOR*, 34:290–306, 1996.

I. M. Chao, B. L. Golden, and E. Wasil. A new heuristic for the multi-depot vehicle routing problem that improves upon best-known solutions. *Am. J. Math. Manage. Sci.*, 13(3-4):371–406, 1993. ISSN 0196-6324.

I. M. Chao, B. L. Golden, and E. Wasil. An improved heuristic for the period vehicle routing problem. *Networks*, 26(1):25–44, 1995.

N. Christofides, M. A., and P. Toth. The Vehicle Routing Problem. In N. Christofides, M. A., P. Toth, and C. Sandi, editors, *Combinatorial Optimization*, pages 315–338. John Wiley, New York, 1979.

F. Chu, N. Labadi, and C. Prins. A scatter search for the periodic capacitated arc routing problem. *European Journal of Operational Research*, 169(2):586–605, March 2006.

G. Clarke and J. W. Wright. Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, 12(4):568–581, 1964.

J. F. Cordeau, M. Gendreau, and G. Laporte. A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks*, 30:105–119, 1997.

J. F. Cordeau, G. Laporte, and A. Mercier. A unified tabu search heuristic for vehicle routing problems with time windows. *Journal of the Operational Research Society*, 52 (8):928–936, 2001.

J.-F. Cordeau, M. Gendreau, A. Hertz, G. Laporte, and J.-S. Sormany. New heuristics for the vehicle routing problem. In A. Langevin and D. Riopel, editors, *Logistics Systems: Design and Optimization*, pages 279–297. Springer, New York, NY, 2005.

J.-F. Cordeau, G. Laporte, M.W.F. Savelsbergh, and D. Vigo. Vehicle Routing. In Barnhart, C. and Laporte, G., editors, *Transportation*, Handbooks in Operations Research and Management Science, pages 367–428. North-Holland, Amsterdam, 2007.

T. H. Cormen, C. Stein, R. L. Rivest, and C. E. Leiserson. *Introduction to Algorithms*. McGraw-Hill Higher Education, 2001.

T. G. Crainic. Parallel solution methods for vehicle routing problems. In Golden, B., Raghavan, S., and Wasil, E., editors, *The Vehicle Routing Problem: Latest Advances and New Challenges*, pages 171–198. Springer, New York, 2008.

T. G. Crainic and H. Nourredine. Parallel Meta-Heuristics Applications. In Alba, E., editor, *Parallel Metaheuristics: A New Class of Algorithms*, pages 447–494. John Wiley & Sons, Hoboken, NJ, 2005.

T. G. Crainic and M. Toulouse. Parallel Strategies for Meta-Heuristics. In Gendreau, M. and Potvin, J.-Y., editors, *Handbook of Metaheuristics*. Springer, 2010. to appear.

T. G. Crainic, G. C. Crisan, M. Gendreau, N. Lahrichi, and W. Rei. A Concurent Evolutionary Approach for Cooperative Rich Combinatorial Optimization. In *Genetic and Evolutionary Computation Conference - GECCO 2009, July 8-12, Montréal, Canada.* ACM, 2009a. CD-ROM.

T. G. Crainic, G. C. Crisan, M. Gendreau, N. Lahrichi, and W. Rei. Multi-thread Integrative Cooperative Optimization for Rich Combinatorial Problems. In *The 12th International Workshop on Nature Inspired Distributed Computing - NIDISC'09, 25-29 May, Rome*, 2009b. CD-ROM.

G. B. Dantzig and J. H. Ramser. The truck dispatching problem. *Management Science*, 6:80–91, 1959.

J. Desrosiers, Y. Dumas, M. M. Solomon, and F. Soumis. Time constrained routing and scheduling. In M. Ball, Magnanti, T.L., Monma, C.L., and Nemhauser, G.L., editors, *Network Routing*, volume 8 of *Handbooks in Operations Research and Management Science*, pages 35–139. North-Holland, Amsterdam, 1995.

J. Dongarra. Performance of various computers using standard linear equations software. Technical report, University of Tennessee, 2009.

L. M. A. Drummond, L. S. Ochi, and D. S. Vianna. An asynchronous parallel metaheuristic for the period vehicle routing problem. *Future Generation Computer Systems*, 17 (4):379–386, 2001.

M. Fisher. Vehicle Routing. In M. Ball, Magnanti, T.L., Monma, C.L., and Nemhauser, G.L., editors, *Network Routing*, volume 8 of *Handbooks in Operations Research and Management Science*, pages 1–33. North-Holland, Amsterdam, 1995.

P. M. Francis, K. R. Smilowitz, and M. Tzur. The period vehicle routing problem and its extensions. In B. L. Golden, S. Raghavan, and E. A. Wasil, editors, *The Vehicle Routing Problem: Latest Advances and New Challenges*, pages 73–102. Society for Industrial and Applied Mathematics, 2008.

M. Gendreau, A. Hertz, and G. Laporte. A tabu search heuristic for the vehicle routing problem. *Manage. Sci.*, 40(10):1276–1290, 1994.

M. Gendreau, G. Laporte, and J.-Y. Potvin. Metaheuristics for the Vehicle Routing Problem. In P. Toth and D. Vigo, editors, *The Vehicle Routing Problem*, SIAM Monographs on Discrete Mathematics and Applications, pages 129–154. SIAM, Philadelphia, PA, 2002.

B. L. Golden and E. A. Wasil. Computerized vehicle routing in the soft drink industry. *Operations Research*, 35(1):6–17, 1987.

B. L. Golden, A. A. Assad, and E. A. Wasil. Routing vehicles in the real world: applications in the solid waste, beverage, food, dairy, and newspaper industries. In P. Toth and D. Vigo, editors, *The Vehicle Routing Problem*, SIAM Monographs on Discrete Mathematics and Applications, pages 245–286. SIAM, Philadelphia, PA, 2002.

B. L. Golden, S. Raghavan, and E. A. Wasil. *The vehicle routing problem: latest advances and new challenges*. Springer, 2008.

E. Hadjiconstantinou and R. Baldacci. A multi-depot period vehicle routing problem arising in the utilities sector. *Journal of the Operational Research Society*, 49(12): 1239–1248, 1998.

N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evol. Comput.*, 9(2):159–195, 2001.

V. C. Hemmelmayr, K. F. Doerner, and R. F. Hartl. A variable neighborhood search heuristic for periodic routing problems. *European Journal of Operational Research*, 195 (3):791–802, 2009.

A. Hoff, H. Andersson, M. Christiansen, G. Hasle, and A. Løkketangen. Industrial aspects and literature survey: Fleet composition and routing. *Computers & Operations Research*, 37(12):2041–2061, 2010. ISSN 0305-0548.

J. H. Holland. *Adaptation in natural and artificial systems*. Ann Arbor, MI: The University of Michigan Press, 1975.

K. H. Kang, Y. H. Lee, and B. K. Lee. An exact algorithm for multi depot and multi period vehicle scheduling problem. In *Computational Science and Its Applications - ICCSA 2005*, Lecture Notes in Computer Science, pages 350–359. Springer Berlin / Heidelberg, 2005.

P. Lacomme, C. Prins, and W. Ramdane-Cherif. Evolutionary algorithms for periodic arc routing problems. *European Journal of Operational Research*, 165(2):535–553, 2005.

G. Laporte. Fifty years of vehicle routing. *Transportation Science*, 43(4):408–416, 2009.

G. Laporte and F. Semet. Classical Heuristics for the Vehicle Routing Problem. In Toth, P. and Vigo, D., editors, *The Vehicle Routing Problem*, SIAM Monographs on Discrete Mathematics and Applications, pages 109–128. SIAM, Philadelphia, PA, 2002.

A. Lim and W. Zhu. A fast and effective insertion algorithm for multi-depot vehicle routing problem with fixed distribution of vehicles and a new simulated annealing approach. In *IEA/AIE*, pages 282–291, 2006.

M. Lozano, F. Herrera, and J.-R. Cano. Replacement strategies to preserve useful diversity in steady-state genetic algorithms. *Inf. Sci.*, 178(23):4421–4433, 2008. ISSN 0020-0255.

A. C. Matos and R. C. Oliveira. An experimental study of the ant colony system for the period vehicle routing problem. In *Ant Colony, Optimization and Swarm Intelligence*, Lecture Notes in Computer Science, pages 1–29. Springer Berlin / Heidelberg, 2004.

R. Mercer and J. Sampson. Adaptive search using a reproductive metaplan. *Kybernetes*, 7:215–228, 1978.

D. Mester and O. Braysy. Active-guided evolution strategies for large-scale capacitated vehicle routing problems. *Computers & Operations Research*, 34(10):2964–2975, 2007.

Y. Nagata and O. Bräysy. Edge assembly-based memetic algorithm for the capacitated vehicle routing problem. *Networks*, 54(4):205–215, 2009.

B. Ombuki-Berman and T. Hanshar. Using genetic algorithms for multi-depot vehicle routing. In F. B. Pereira and J. Tavares, editors, *Bio-inspired Algorithms for the Vehicle Routing Problem*, pages 77–99. Springer, 2009.

P. Parthanadee and R. Logendran. Periodic product distribution from multi-depots under limited supplies. *IIE Transactions*, 38(11):1009–1026, 2006.

S. Pirkwieser and G. R. Raidl. Multilevel variable neighborhood search for periodic routing problems. In P. Cowling and P. Merz, editors, *Evolutionary Computation in Combinatorial Optimisation - EvoCOP 2010*, volume 6022 of *LNCS*, pages 226–238. Springer, 2010.

D. Pisinger and S. Ropke. A general heuristic for vehicle routing problems. *Computers & Operations Research*, 34(8):2403–2435, 2007.

W. Powell, P. Jaillet, and A. Odoni. Stochastic and dynamic networks and routing. In Ball, M., Magnanti, T.L., Monma, C.L., and Nemhauser, G.L., editors, *Network Routing*, volume 8 of *Handbooks in Operations Research and Management Science*, pages 141–295. North-Holland, Amsterdam, 1995.

C. Prins. A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers & Operations Research*, 31(12):1985–2002, 2004.

J. Renaud, G. Laporte, and F. F. Boctor. A tabu search heuristic for the multi-depot vehicle routing problem. *Comput. Oper. Res.*, 23(3):229–235, 1996.

R. Russell and W. Igo. An assignment routing problem. *Networks*, 9(1):1–17, 1979.

S. Smit and A. Eiben. Comparing parameter tuning methods for evolutionary algorithms. In *Proceedings of the 2009 IEEE Congress on Evolutionary Computation*, 2009.

K. Sörensen and M. Sevaux. Mapm: memetic algorithms with population management. *Computers & Operations Research*, 33(5):1214 – 1225, 2006.

J. Teixeira, A. P. Antunes, and J. P. De Sousa. Recyclable waste collection planning–a case study. *European Journal of Operational Research*, 158(3):543 – 554, 2004.

S. Thangiah and S. Salhi. Genetic clustering: An adaptive heuristic for the multi depot vehicle routing problem. *Applied Artificial Intelligence*, 15(4):361–383, 2001.

P. Toth and D. Vigo, editors. *The Vehicle Routing Problem*. SIAM Monographs on Discrete Mathematics and Applications. SIAM, Philadelphia, PA, 2002.

P. Toth and D. Vigo. The granular tabu search and its application to the vehicle-routing problem. *INFORMS Journal on Computing*, 15(4):333–346, 2003.

W. T. Yang and L. C. Chu. A heuristic algorithm for the multi-depot periodic vehicle routing problem. *Journal of Information & Optimization Sciences*, 22:359–367, 2000.