OPERATIONS RESEARCH
Vol. 60, No. 3, May–June 2012, pp. 611–624
ISSN 0030-364X (print) | ISSN 1526-5463 (online)

http://dx.doi.org/10.1287/opre.1120.1048

A Hybrid Genetic Algorithm for Multidepot and Periodic Vehicle Routing Problems

Thibaut Vidal
CIRRELT and Département d'informatique et de recherche opérationnelle, Université de Montréal, Montréal, Quebec H3C 3J7, Canada, thibaut.vidal@cirrelt.ca

Teodor Gabriel Crainic
CIRRELT and Département de management et technologie, École des sciences de la gestion, University of Quebec in Montréal, Montréal, Quebec H3C 3P8, Canada, teodorgabriel.crainic@cirrelt.ca

Michel Gendreau, Nadia Lahrichi
CIRRELT and Département de mathématiques et génie industriel, École Polytechnique, Montréal, Quebec H3C 3A7, Canada {michel.gendreau@cirrelt.ca, nadia.lahrichi@cirrelt.ca}

Walter Rei
CIRRELT and Département de management et technologie, École des sciences de la gestion, University of Quebec in Montréal, Montréal, Quebec H3C 3P8, Canada, walter.rei@cirrelt.ca

We propose an algorithmic framework that successfully addresses three vehicle routing problems: the multidepot VRP, the periodic VRP, and the multidepot periodic VRP with capacitated vehicles and constrained route duration. The metaheuristic combines the exploration breadth of population-based evolutionary search, the aggressive-improvement capabilities of neighborhood-based metaheuristics, and advanced population-diversity management schemes. Extensive computational experiments show that the method performs impressively in terms of computational efficiency and solution quality, identifying either the best known solutions, including the optimal ones, or new best solutions for all currently available benchmark instances for the three problem classes. The proposed method also proves extremely competitive for the capacitated VRP.

Subject classifications: multidepot, multiperiod vehicle routing problems; hybrid population-based metaheuristics; adaptive population diversity management.
Area of review: Transportation.
History: Received July 2010, revisions received January 2011, August 2011; accepted October 2011.

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 (Toth and Vigo 2002, Hoff et al. 2010). Not surprisingly, starting with the seminal work of Dantzig and Ramser (1959), routing problems make up an extensively and continuously studied field, as illustrated by numerous conferences, survey articles (e.g., Cordeau et al. 2007, 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 or adequate degree of attention. This is the case for the problems with multiple depots and periods. A second general observation is that most methodological developments target a particular problem variant, the capacitated VRP (*CVRP*) or the VRP with time windows (*VRPTW*), for example, very few contributions aiming to address a broader set of problem settings. This also applies to the problem classes 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 multidepot VRP, *MDVRP*; the periodic VRP, *PVRP*; and the multidepot periodic VRP, *MDPVRP*, with capacitated vehicles and constrained route duration. The literature on these problems is relatively scarce (Francis et al. 2008) 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, Coene et al. 2010), food collection or distribution (Golden and Wasil 1987, Parthanadee and Logendran 2006), and maintenance operations (Hadjiconstantinou and Baldacci 1998, Blakeley et al. 2003).

We propose a metaheuristic that combines the exploration breadth of population-based evolutionary search, the aggressive-improvement capabilities of neighborhood-based metaheuristics, 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. HGSADC outperforms the state-of-the-art methods for the three problem classes on all currently available benchmark instances, identifying either the best known solutions, including the optimal ones, or new best solutions. Moreover, with very limited adaptation, HGSADC proves to be extremely competitive for the CVRP.

The two main contributions of this article are as follows: (1) A new metaheuristic, which is highly effective for four important vehicle routing problem classes, the MDVRP, the PVRP, the MDPVRP, and the CVRP. The method is built on general components that are applicable, provided minor adjustments, to an even wider range of VRP variants. (2) New population-diversity management mechanisms to allow a broader access to reproduction, while preserving the characteristics of elite solutions. In this respect, we revisit the traditional *survival of the fittest* paradigm to enhance the evaluation of individuals by making it rely on both solution cost and diversity (distance to the others) measures. Thus, diversity appears as an integral part of the objective, which contrasts with classical diversity-management procedures that have traditionally imposed diversity constraints for the acceptance of solutions in the population. Empirical studies show that these new mechanisms not only avoid premature population convergence efficiently, but also lead to higher-quality solutions in reduced computation time when compared to traditional approaches. It must be emphasized that the proposed population-diversity management mechanisms could be applied to almost any problem that one can solve using population-based evolutionary search methods.

The paper is organized as follows. Section 2 states the notation and formal definition of the three classes of VRPs we address, whereas the relevant literature is surveyed in §3. The proposed metaheuristic is detailed in §4, its performances are analyzed in §5, and we conclude in §6. An electronic companion to this paper is available as part of the online version at http://dx.doi.org/10.1287/opre.1120.1048.

2. Problem Statement

We formally state the MDVRP, PVRP, and MDPVRP, introducing the notation used in this paper and the transformation of the MDPVRP into a PVRP, which supports the algorithmic developments.

The CVRP can be defined as follows. Let $G = (\mathcal{V}, \mathcal{A})$ be a complete graph with $|\mathcal{V}| = n + 1$ vertices, divided in two sets $\mathcal{V} = \mathcal{V}^{\mathrm{DEP}} \cup \mathcal{V}^{\mathrm{CST}}$. The unique vertex $v_0 \in \mathcal{V}^{\mathrm{DEP}}$ represents the depot where the product to be distributed is kept and a fleet of m identical vehicles with capacity Q is based. Vertices $v_i \in \mathcal{V}^{\mathrm{CST}}$ stand for customers i, $i = 1, \ldots, n$, requiring service and characterized by a nonnegative demand q_i and a service duration τ_i. Arcs $a_{ij} \in \mathcal{A}$, $i, j \in \mathcal{V}$ represent the direct-travel possibility from v_i to v_j with travel time equal to c_{ij}. The duration of a vehicle route is computed as the total travel and service time required to serve the customers and is limited to T. The goal is to design a set of vehicle routes servicing all customers, such that vehicle capacity and route duration constraints are respected, and the total travel time is minimized.

Several depots, d, are available to service customers in the multidepot VRP, m representing the number of vehicles available at each depot. In this case, vertices $v_0, \ldots, v_d$ make up the set $\mathcal{V}^{\mathrm{DEP}}$, whereas the remaining vertices $\mathcal{V}^{\mathrm{CST}}$ stand for customers. A time dimension is introduced in the periodic VRP as route planning 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 multidepot periodic VRP extends the two previous problem settings, asking for the selection of a depot and a visit pattern for each customer, with services in different periods to the same customer being required to originate at the same depot. The CVRP is NP-hard, and so are the three problem classes that generalize it and are addressed in this paper.

The MDPVRP reduces to a PVRP when $d = 1$ and to a MDVRP when $t = 1$. Furthermore, the three problem settings share similar mathematical structures. We take advantage of this property and, in the spirit of the problem transformation from MDVRP to PVRP of Cordeau et al. (1997), we transform an MDPVRP with d depots and t periods into an equivalent PVRP with $d \times t$ periods corresponding to (depot, period) couples. (The e-companion EC.1 details these mathematical structures and problem transformations.) This transformation provides the means to address the three problem classes with the same solution method and reduces the number of problem characteristics. 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 §5 show, we achieve this goal.

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 metaheuristic algorithms, particularly population-based ones, for the considered problems. Second, to distinguish the leading solution approaches for the three problem settings.

Some population and neighborhood-based metaheuristics already exist in the PVRP literature. 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 a savings heuristic. Alegre et al. (2007) proposed a scatter search procedure designed especially for PVRPs with a large number 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. This ACO method addresses the PVRP as a VRP, where each customer is duplicated as many times as indicated by its frequency, the resulting routes being then distributed to periods by solving a graph-coloring problem.

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, and also allow 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 (UTS) version (Cordeau et al. 2001), for a number of other VRP variants. To date, however, the best PVRP results have been produced by the variable neighborhood search (VNS) of Hemmelmayr et al. (2009) and the record-to-record travel approach of Gulczynski et al. (2011), the latter combining local search and an integer-programming based large neighborhood search. Finally, one should notice the VNS algorithm with multilevel refinement strategy of Pirkwieser and Raidl (2010), specifically tailored for large-size instances.

Population-based metaheuristics proposed for the MDVRP often took advantage of geometric aspects of the problem. Thus, Thangiah and Salhi (2001) represented solutions as circles in the 2D space, whereas Ombuki-Berman and Hanshar (2009) introduced a mutation operator that targeted the depot assignment to "borderline" customers (i.e., which are close to several depots). Lau et al. (2010) explored a different genetic algorithm idea by using diversity measures and fuzzy logic to adapt the mutation and crossover rates during the search. Finally, a parallel ACO approach was proposed by Yu and Yang (2011).

As underlined in Ombuki-Berman and Hanshar (2009), various neighborhood-based single-trajectory searches have been proposed in the MDVRP literature. The most successful approach 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 operators.

In the case of the MDPVRP, most proposed algorithms do not consider all characteristics 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 last phase attempts to improve the solution by modifying some period or depot assignments. 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. (2009) introduced the integrative cooperative search (ICS) framework, which relies on 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 did not report results 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. (2011), 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 multidepot 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, Nagata and Bräysy 2009) and on a number of other variants, e.g., the VRPTW (Bräysy et al. 2004). Noteworthy is the contribution of Prins (2004), who 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

Operations Research 60(3), pp. 611–624, © 2012 INFORMS

et al. (2005) and 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 multidepot 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 Metaheuristic

The Hybrid Genetic Search with Adaptive Diversity Control (HGSADC) metaheuristic we propose is based on the genetic algorithm (GA) paradigm introduced by Holland (1975) but includes a number of advanced features, in terms of solution evaluation, offspring generation and improvement, and population management, which contribute to its originality and high performance level.

The general scheme of the metaheuristic we propose is displayed in Algorithm 1. The method evolves a population of individuals, managing feasible and infeasible solutions, which are kept in two separate groups (*subpopulations*). It applies successively a number of operators to select two parent individuals and combine them, yielding a new individual (*offspring*), which is first enhanced using local search procedures (*education* and *repair*), and then included in the appropriate subpopulation in relation to its feasibility. Of particular interest is the evaluation mechanism we propose, which is used to select both parents for mating (Line 3 of Algorithm 1) and individuals to survive to the next generation (Line 8). The mechanism takes into account not only the solution cost (§4.1), which is often the norm, but also the contribution the individual makes to the diversity of the gene pool. Thus it contributes to maintaining a high level of diversity among individuals, and plays an important role in the overall performance of the proposed methodology.

Algorithm 1 (HGSADC)
1: Initialize population
2: **while** *number of iterations without improvement* $< It_{NI}$, and *time* $< T_{max}$
3: Select parent solutions P_1 and P_2
4: Generate offspring C from P_1 and P_2 (crossover)
5: Educate offspring C (local search procedure)
6: **if** C is infeasible, **then** insert C into infeasible subpopulation; repair with probability P_{rep}
7: **if** C is feasible, **then** insert C into feasible subpopulation
8: **if** *maximum subpopulation size reached*, **then** select survivors
9: Adjust penalty parameters for violating feasibility conditions
10: **if** *best solution not improved for* It_{div} *iterations*, **then** diversify population
11: Return best feasible solution

We initiate the description of HGSADC with the definition of the search space, §4.1, followed by the representation and evaluation of individuals, §§4.2 and 4.3, respectively. We then proceed with detailed discussions of parent selection and crossover, §4.4, education and repair, §4.5, and population management, §4.6.

4.1. Search Space

The metaheuristic literature indicates that allowing a controlled exploration of infeasible solutions may enhance the performance of the search, which may more easily transition between structurally different feasible solutions (Glover and Hao 2011). We thus define the search space $\mathcal{S}$ as a set of feasible and infeasible solutions $s \in \mathcal{S}$, the latter being obtained by relaxing the limits on vehicle capacities and maximum route travel time (as in Gendreau et al. 1994, Cordeau et al. 1997).

Let $\mathcal{R}(s)$ represent the set of routes making up solution s. Each route $r \in \mathcal{R}(s)$ starts from a depot $\sigma_0^r \in \mathcal{V}^{DEP}$, visits a sequence of n_r customers $\sigma_1^r, \dots, \sigma_{n_r}^r \in \mathcal{V}^{CST}$, and returns to the same depot $\sigma_{n_r+1}^r = \sigma_0^r$. It is characterized by load $q(r) = \sum_{i=1}^{n_r} q_{\sigma_i^r}$, driving time $c(r) = \sum_{i=0}^{n_r} c_{\sigma_i^r \sigma_{i+1}^r}$, and total duration $\tau(r) = c(r) + \sum_{i=1}^{n_r} \tau_{\sigma_i^r}$.

Let ω^Q and ω^D represent the penalties for exceeding the vehicle capacity and the route maximum duration, respectively. The *penalized cost* of a route r is then defined in Equation (1) as its driving time plus, when the route is infeasible, the weighted sum of its excess duration and/or load.

$$\phi(r) = c(r) + \omega^D \max\{0, \tau(r) - T\} + \omega^Q \max\{0, q(r) - Q\}. \quad (1)$$

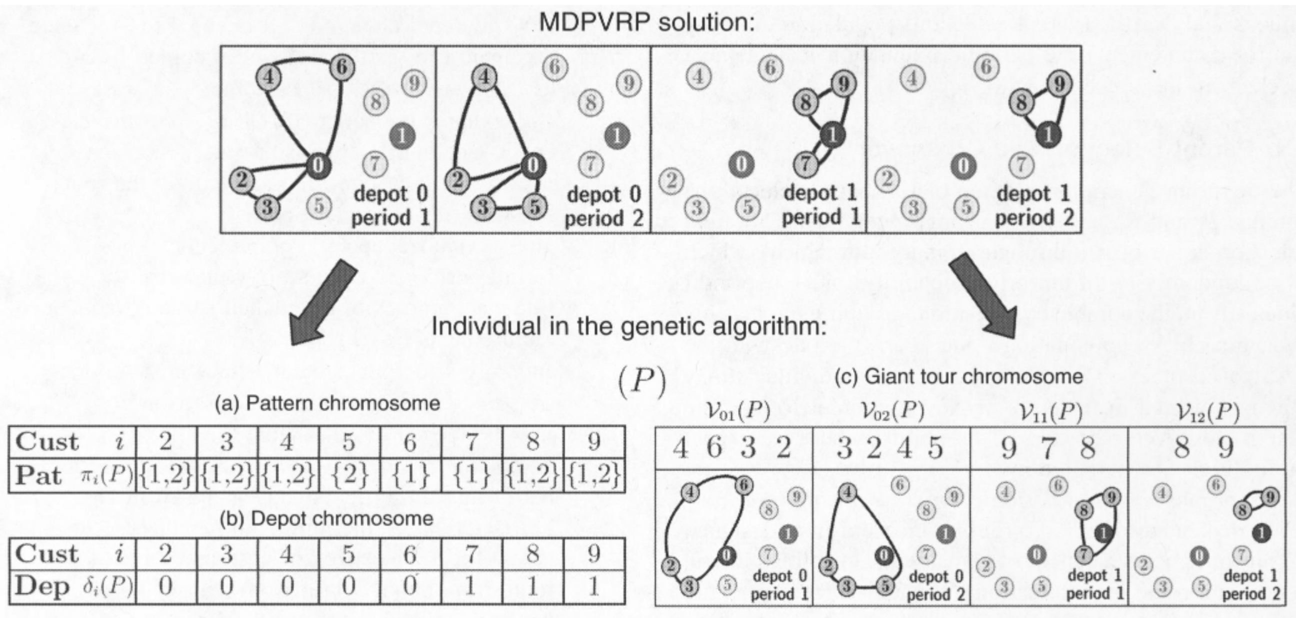
The penalized cost $\phi(s)$ of a solution s is then computed as the sum of the penalized costs of all its routes $\phi(s) = \sum_{r \in \mathcal{R}(s)} \phi(r)$ and is used to compute the fitness of the individuals.

4.2. Solution Representation

Solutions $s \in \mathcal{S}$ are characterized by their customer schedules, depot assignments, and routes. The individuals representing them in the HGSADC population are thus represented as a set of three chromosomes: (1) the *pattern chromosome*, which registers for each customer i its pattern $\pi_i(P)$; (2) the *depot chromosome*, containing the depot assignment $\delta_i(P)$ of each customer i; and (3) the *giant tour chromosome*, containing for each combination (depot o, period l), a sequence $\mathcal{V}_{ol}(P)$ of customers *without trip delimiters*, obtained by concatenating all routes from depot o during period l, in an arbitrary order, and removing visits to depots. Figure 1 illustrates this representation scheme for a small MDPVRP problem with two periods, two depots, and eight customers.

The representation of the routes out of the same period and depot as a giant tour provides the means to use simple

Figure 1. From an MDPVRP solution to the individual chromosome representation.



and efficient crossover procedures working on permutations, but requires an algorithm to find the optimal segmentation of the tour into routes and, thus, retrieve both the solution and its cost. The first successful utilization of a giant tour representation within a genetic algorithm was reported by Prins (2004), who also introduced an efficient algorithm to optimally extract the routes from the tour. This algorithm, named *Split*, reduces the problem of finding the route delimiters to a shortest-path problem on an auxiliary acyclic graph. It is straightforward to adapt to the setting with penalized costs and limited fleet size and can be implemented in polynomial time $O(mn^2)$, as explained in the e-companion EC.2.

4.3. Evaluation of Individuals

The individual-evaluation function in population-based metaheuristics aims to determine for each individual a relative value with respect to the entire population. Often based on the value of the objective function of the problem at hand (e.g., the value of the individual compared to the average value of the population), this so-called *fitness* measure is then used to perform various selections, e.g., parents for mating or individuals to advance to the next generation, the latter being named *survivors* in the following. Such an approach is, however, generally myopic with respect to the possible impact of the evaluation and selection processes on the diversity of the population, a critical performance factor for this class of metaheuristics. We therefore propose a mechanism that addresses both objectives, the *evaluation function* accounting for the cost of an individual and its contribution to the population diversity.

We define the *diversity contribution* $\Delta(P)$ of an individual P as the average distance to its n_{close} closest neighbors, grouped in set $\mathcal{N}_{\text{close}}$, computed according to Equation (2). Several distance measures were tested in the experiments leading to the final algorithm. A normalized Hamming distance $\delta^{\text{H}}(P_1, P_2)$, based on the differences between the service patterns and depot assignments of two individuals P_1 and P_2, appeared the most adequate for the multidepot, multiperiod routing problems we address. This distance is computed according to Equation (3), where $\mathbf{1}$ (*cond*) is a valuation function that returns 1 if the condition *cond* is true, 0, otherwise.

$$\Delta(P) = \frac{1}{n_{\text{close}}} \sum_{P_2 \in \mathcal{N}_{\text{close}}} \delta^{\text{H}}(P, P_2) \tag{2}$$

$$\delta^{\text{H}}(P_1, P_2) = \frac{1}{2n} \sum_{i=1,\ldots,n} (\mathbf{1}(\pi_i(P_1) \neq \pi_i(P_2))) + \mathbf{1}(\delta_i(P_1) \neq \delta_i(P_2)). \tag{3}$$

Let $\text{fit}(P)$ and $\text{dc}(P)$ in $\{1, \ldots, nbIndiv\}$ stand for the *rank* of an individual P in a subpopulation of size *nbIndiv*, with respect to its penalized cost $\phi(P)$ and diversity contribution $\Delta(P)$, respectively. The *biased fitness* function $\text{BF}(P)$ we propose combines the cost and diversity ranks, and is given by Equation (4), where *nbElit* is the number of elite individuals one desires to survive to the next generation.

$$\text{BF}(P) = \text{fit}(P) + \left(1 - \frac{nbElit}{nbIndiv}\right)\text{dc}(P). \tag{4}$$

The ranks and biased fitness measures are continuously updated for the two subpopulations and are used to evaluate the quality of an individual during parent (§4.4) and survivor (§4.6) selections. The biased fitness is thus an adaptive mechanism aiming to balance the drive for the best individual (elitism) and the possible loss of information usually associated with this drive. This concern for continuous and "early" (parent selection) population-diversity control complements the periodic population management mechanism introduced in §4.6.

4.4. Parent Selection and Crossover

The offspring generation scheme of HGSADC selects two parents, P_1 and P_2, and yields a single individual C. Parent selection is performed through a binary tournament, which twice randomly (with uniform probability) picks two individuals from the complete population, grouping the feasible and infeasible subpopulations, and keeps the one with the best biased fitness. Feasible and infeasible individuals may thus be selected to undergo crossover in order to lead the search close to the borders of feasibility, where we expect to find high-quality solutions.

We propose a new *periodic crossover with insertions* (*PIX*) dedicated to periodic routing problems and designed to transmit good sequences of visits, while enabling pattern, depot, and route recombinations. We aimed for a versatile crossover, which would allow for both a wide exploration of the search space and small refinements of "good" solutions. The possibility for the offspring to inherit genetic material from its parents in nearly equal proportions is required to provide the crossover with the former capability, while copying most of one parent along with small parts of the other provides the latter. To ensure that PIX has both capabilities meant avoiding a priori determined rules on how much genetic material the offspring inherits from each parent, as well as rules based on simple random selection of individual characteristics.

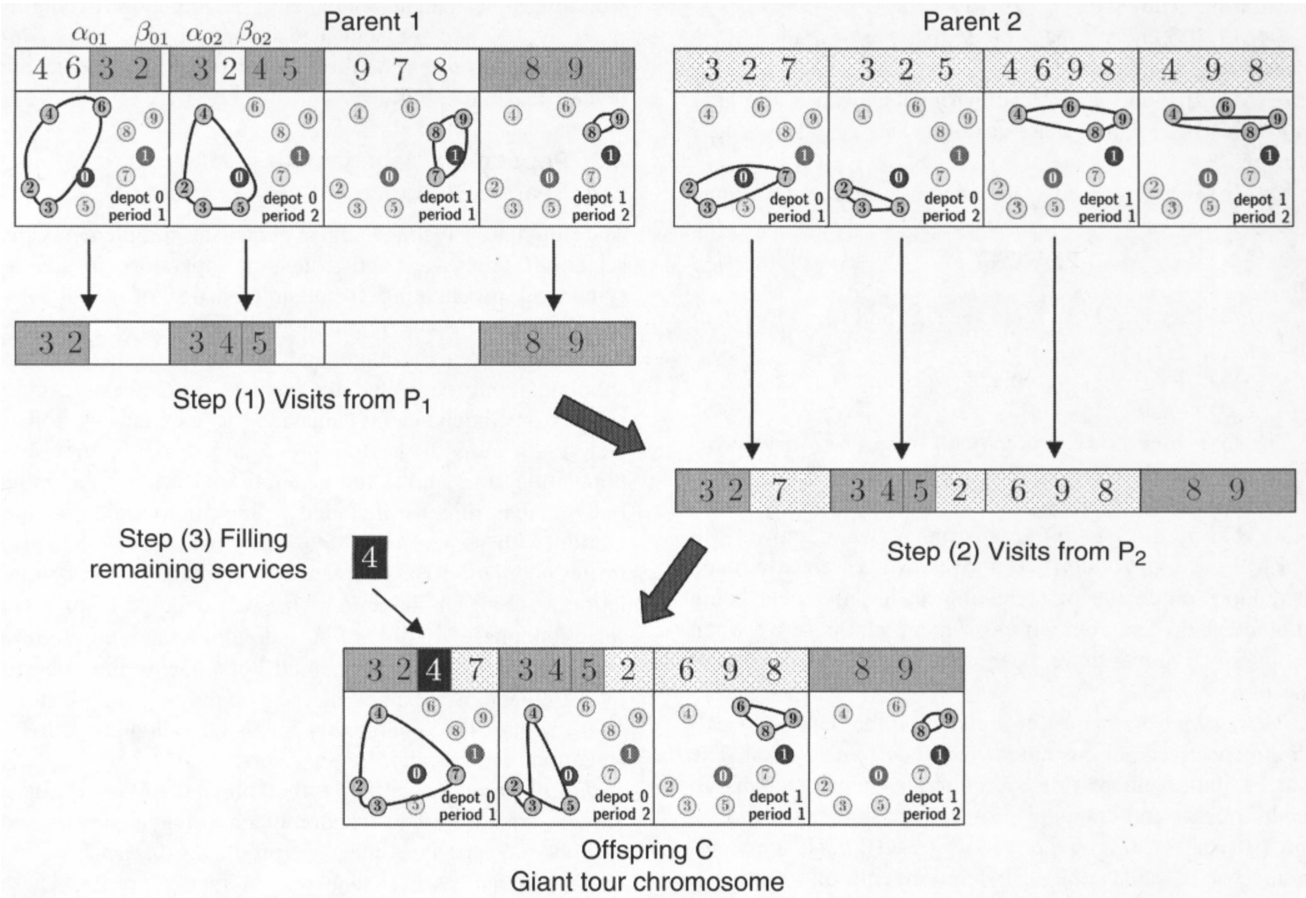
Algorithm 2 (PIX)

1: STEP 0: INHERITANCE RULE.
2: Pick two random numbers between 0 and td according to a uniform distribution. Let n_1 and n_2 be, respectively, the smallest and the largest of these numbers;
3: Randomly select n_1 (depot, period) couples to form the set Λ_1;
4: Randomly select $n_2 - n_1$ remaining couples to form the set Λ_2;
5: The remaining $td - n_2$ couples make up the set Λ_{mix}.
6: STEP 1: INHERIT DATA FROM P_1.
7: **for** each (depot, period) (o, l), belonging to set
8: Λ_1: Copy the sequence of customer visits from $\mathcal{V}_{o,l}(P_1)$ to $\mathcal{V}_{o,l}(C)$;
9: Λ_{mix}: Randomly (uniform distribution) select two chromosome-cutting points α_{kl} and β_{kl}; copy the α_{kl} to β_{kl} substring of $\mathcal{V}_{o,l}(P_1)$ to $\mathcal{V}_{o,l}(C)$.
10: STEP 2: INHERIT DATA FROM P_2.
11: **for** each (depot, period) $(o, l) \in \Lambda_2 \cup \Lambda_{\text{mix}}$ selected in random order
12: Consider each customer visit i in $\mathcal{V}_{o,l}(P_2)$ and copy it at the end of $\mathcal{V}_{o,l}(C)$ when
(1) The depot choice $\delta_i(C)$ is equal to o or undefined (no visit to i has been copied to C yet);
(2) One visit pattern of customer i, at least, contains the set $\pi_i(C) \cup l$ of visit periods.
13: STEP 3. COMPLETE CUSTOMER SERVICES.
14: Perform the *Split* algorithm and extract the routes for each (depot, period) pair;
15: **if** the service-frequency requirements are satisfied for all customers **then stop**; Otherwise,
16: **while** customers with unsatisfied service frequency requirements exist, **repeat**:
17: Randomly select a customer i for which service frequency requirements are not satisfied;
18: Let $\mathcal{F}$ be the set of admissible (depot, period) combinations (o, l) with respect to its pattern list L_i and the visits already included in C. Let $\psi(i, o, l)$ be the minimum penalized cost (§4.1) for the insertion of customer i into a route from depot o in period l. Insert i into $(o^*, l^*) = \arg\min_{(o,l)\in\mathcal{F}} \psi(i, o, l)$.

Algorithm 2 displays the detailed pseudo code for the PIX crossover procedure, which is illustrated in Figure 2 on a problem with two periods and two depots, and described in the rest of this section. The crossover begins in Step 0 by determining the period and depot inheritance rule. Let Λ_1, Λ_2, and Λ_{mix} be the sets of (depot, period) couples corresponding to inheriting material from the first parent, P_1, the second parent, P_2, or both parents, respectively. The procedure then first determines the cardinality of each set and then fills them up sequentially by randomly selecting the appropriate number of (depot, period) couples. We assume for the example of Figure 2 that $\Lambda_1 = \{(d1, p2)\}$, $\Lambda_2 = \{(d1, p1)\}$, and $\Lambda_{\text{mix}} = \{(d0, p1), (d0, p2)\}$.

Steps 1 and 2 are dedicated to taking genetic material from the two parents and combining their giant-tour chromosomes. Step 1 targets the first parent and copies for each selected (depot, period) couple either the complete material, if it belongs to Λ_1, or a random subsequence, if it belongs to Λ_{mix}. Thus, in Figure 2, all customers serviced from depot 1 in period 2 ($\Lambda_1 = \{(d1, p2)\}$) are inherited from P_1, whereas only subsets are inherited for depot 0 at periods 1 and 2 ($\Lambda_{\text{mix}} = \{(d0, p1), (d0, p2)\}$). Step 2 targets the second parent and, thus, the material from (depot, period) couples in $\Lambda_2 \cup \Lambda_{\text{mix}}$. The inheritance is restricted by the selections performed in Step 1, however, and thus customers are inserted at the end of the corresponding sequence when the depot and pattern compatibility conditions specified at line 12 of Algorithm 2 are satisfied. Given the random order $(d0, p2)$, $(d1, p1)$, $(d0, p1)$ of $\Lambda_2 \cup \Lambda_{\text{mix}}$ in the illustration, the visits from P_2 that conform to this test and are transmitted to C are marked through dots on white background in Figure 2. Thus, for example, the $(d1, p1)$ pair of P_2 yields only the subsequence $[6, 9, 8]$

Figure 2. The PIX crossover.



out of [4, 6, 9, 8], because a visit to customer 4 was copied during Step 1 from ($d0$, $p2$) of P_1.

The offspring built at the end of Step 2 might not be feasible, however, because of customers with unsatisfied service frequency requirements. The goal of Step 3 is then to perform the necessary insertions of additional visits. Most insertion mechanisms used in vehicle routing and traveling salesman problems could be used. However, to enhance the precision of the insertion, and because the routes must be extracted in all cases, before undertaking the next phase of the metaheuristic (§4.5), we anticipate this extraction, using the Split algorithm, and perform best-insertion directly into the actual routes based on the corresponding biased fitness measure. In Figure 2, the necessary services to customer 4 are not fulfilled in C following the first crossover steps, and a possible result of least-cost insertion is illustrated.

4.5. Education

An *Education* operator is applied with probability P_m to improve the quality of the offspring solution (the routes were extracted in Step 3 of the PIX procedure). Education goes beyond the classical genetic-algorithm concepts of random mutation and enhancement through hill-climbing techniques, because it includes several local-search procedures based on neighborhoods for the VRP. A *Repair* phase eventually completes the Education operator when the educated offspring is infeasible.

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 visit assignments of customers by changing their patterns and depots. These local searches are called in the RI, PI, RI sequence.

Route Improvement. Let $r(u)$ stand for the route containing vertex u in the given (depot, period) routing subproblem, and (u_1, u_2) identify the partial route from u_1 to u_2. Define the neighborhood of vertex u, customer or depot, as the hn closest vertices, where $h \in [0, 1]$ is a *granularity threshold* restricting the search to nearby vertices (Toth and Vigo 2003). Let v be a neighbor of u, and x and y the successors of u in $r(u)$ and v in $r(v)$, respectively. The Route improvement phase iterates, in random order, over each vertex u and each of its neighbors v, and evaluates 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 $r(u) = r(v)$, replace (u, x) and (v, y) by (u, v) and (x, y);
- (M8) If $r(u) \neq r(v)$, replace (u, x) and (v, y) by (u, v) and (x, y);
- (M9) If $r(u) \neq r(v)$, replace (u, x) and (v, y) by (u, y) and (x, v).

The first three moves correspond to *insertions*, whereas moves M4 to M6 are generally called *swaps*. These moves can be applied indifferently on the same or different routes. Move M7 is a 2-opt intraroute move, while moves M8 and M9 are 2-opt* interroute moves. Moves are examined in random order, the first yielding an improvement being implemented. The route improvement phase stops when all possible moves have been successively tried without success.

Pattern Improvement. Let $\bar{o}$ and $\bar{p}$ be the depot and pattern, respectively, of customer i in the current solution. The pattern improvement procedure iterates on customers in random order and computes, for each customer i, depot o, and pattern $p \in L_i$, $\Psi(i, o, p) = \sum_{l \in p} \psi(i, o, l)$, the minimum cost to satisfy the visit requirements of i from the depot o according to the visit pattern p. If a (i, o, p) combination exists such that $\Psi(i, o, p) < \Psi(i, \bar{o}, \bar{p})$, then all visits to customer i are removed, and a new visit is inserted in the best location in each sequence corresponding to depot o and period $l \in p$. The procedure stops when all customers have been successively considered without a modification.

The pattern improvement procedure is significantly faster when the optimal position and insertion cost of each customer is stored for each route. It is also worth noting that, sometimes, the current pattern and depot choices are kept, but a better insertion of customers is found. The resulting move is then, in fact, a combination of intraperiod 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 vertices.

The individual yielded by the RI, PI, RI education sequence may be feasible, in which case, we call it *naturally feasible*, or infeasible, and it is inserted into the appropriate subpopulation. Infeasible individuals are subject to the Repair procedure with probability P_{rep}. When Repair is successful, the resulting individual is added to the feasible subpopulation (the infeasible one is not deleted from the infeasible subpopulation). Repair consists in temporarily multiplying the penalty parameters by 10 and restarting the RI, PI, RI sequence. When the resulting individual is still infeasible, penalty parameters are temporarily multiplied by 100 and the sequence is started again. This significant increase of penalties aims at redirecting the search toward feasible solutions.

4.6. Population Management and Search Guidance

The population management mechanism complements the selection, crossover, and education operators in identifying and propagating the characteristics of good solutions, enhancing the population diversity, and providing the means for a thorough and efficient search. The two subpopulations dedicated to feasible and infeasible individuals are independently managed to contain between μ and $\mu + \lambda$ individuals, μ representing the minimum subpopulation size, and λ the generation size. Any incoming individual is directly included in the appropriate subpopulation with respect to its feasibility, and thus acceptance in the population is systematically granted. Any subpopulation reaching its maximum size will undergo a survivors selection phase to discard λ individuals and thus return to its minimum size. Four main components thus constitute the general behavior of the population: initialization, adjustment of the penalties for infeasible individuals, diversification, and selection of survivors.

Initialization. To initialize the subpopulations, 4μ individuals are created by randomly choosing a pattern and a depot for each customer and producing for each period the associated service sequence in random order. These initial individuals undergo education, repair with probability 0.5, and are inserted into the appropriate subpopulation in relation to their feasibility. Survivor selection is activated, as described later on, when a subpopulation reaches the maximum size. At the end of initialization, one of the two subpopulations may be incomplete, with less than μ individuals.

Penalty parameter adjustment. The penalty parameters are initially set to $\omega^{\text{D}} = 1$ and $\omega^{\text{Q}} = \bar{c}/\bar{q}$, where $\bar{c}$ represents the average distance between two customers and $\bar{q}$ is the average demand. The parameters are then dynamically adjusted during the execution of the algorithm to favor the generation of naturally feasible individuals as defined in §4.5. Let ξ^{REF} be a target proportion of naturally feasible individuals, and ξ^{Q} and ξ^{D} the proportion in the last 100 generated individuals of naturally feasible one with respect to vehicle capacity and route duration, respectively. The following adjustment is then performed every 100 iterations, where PAR = Q, D:

- if $\xi^{\text{PAR}} \leqslant \xi^{\text{REF}} - 0.05$, then $\omega^{\text{PAR}} = \omega^{\text{PAR}} \times 1.2$;
- if $\xi^{\text{PAR}} \geqslant \xi^{\text{REF}} + 0.05$, then $\omega^{\text{PAR}} = \omega^{\text{PAR}} \times 0.85$.

Diversification is called when It_{div} iterations occurs without improving the best solution. It is performed by eliminating all but the best $\mu/3$ individuals of each subpopulation and creating 4μ new individuals as in the initialization phase. This process introduces a significant amount of new genetic material, which revives the search further, even when the population has lost most of its diversity.

Diversity and selection of survivors. A major challenge in population-based algorithms is avoiding premature convergence of the population. The issue is even more challenging when, as in our case, education compounds the parent selection tendency to favor individuals with good characteristics, thus reducing the genetic material diversity in the population. The mechanism we propose aims to address this challenge by simultaneously identifying and preserving the most promising solution characteristics, and ensuring the diversity of both subpopulations.

The first component of this mechanism is made up of the definition of the biased fitness function and the explicit consideration of diversity during parents selection (§4.3). The second takes place whenever one of the two subpopulations reaches the maximum size $\mu + \lambda$. Named *Survivor selection*, the procedure determines the μ individuals that will go on to the next generation, such that the population diversity, in terms of visit patterns, is preserved and elite individuals in terms of cost are protected. The λ discarded individuals are thus either clones (Prins 2004) or bad with respect to cost and contribution to diversity as measured by their biased fitness.

Let a *clone* be an individual P_2 with either the same pattern and depot assignments as another individual P_1, i.e., $\delta^{\mathrm{H}}(P_1, P_2) = 0$, or the same solution cost. The procedure successively eliminates, first, clones, and then bad individuals, as described in Algorithm 3. Proposition 1 formalizes the elitism property of the survivor selection procedure.

Algorithm 3 (Survivor selection)
1. **for** $i = 1 \ldots \lambda$
2. $X \leftarrow$ all individuals having a clone
3. **if** $X \neq \varnothing$ **then** remove $P \in X$ with maximum Biased Fitness
4. **else** remove P in the subpopulation with maximum Biased Fitness
5. Update distance and Biased Fitness measures

Proposition 1. *An individual $P \notin X$, among the nbElit best individuals of the subpopulation in terms of cost, will not be removed from the subpopulation by the survivor selection procedure.*

Proof. Let J be the individual with the worst cost in the subpopulation, i.e., $\mathrm{fit}(J) = \mathit{nbIndiv}$, and thus $\mathrm{BF}(J) \geqslant \mathit{nbIndiv} + 1$. P belongs to the best *nbElit* solutions in terms of cost, thus $\mathrm{BF}(P) \leqslant \mathit{nbElit} + (1 - \mathit{nbElit}/\mathit{nbIndiv})(\mathit{nbIndiv}) \leqslant \mathit{nbIndiv}$. Individual P will not be removed as J has a worst biased fitness. □

5. Computational Experiments

We conducted several sets of experiments to evaluate the performance of HGSADC and to assess the impact on this performance of a number of algorithmic components. The former is performed through comparisons to results of state-of-the-art methods and to best-known solutions (BKS) for the three multiperiod, multidepot settings (§5.2), as well as for the capacitated VRP (§5.3). The latter is discussed in §5.4, while the calibration of the metaheuristic is discussed in §5.1.

HGSADC was implemented in C++. Experiments were run on an AMD Opteron 250 computer with 2.4 GHz clock. To facilitate comparisons with previous work, all CPU times reported in this section and the e-companion were converted into their equivalent Pentium IV 3.0 GHz run times using Dongarra (2011) factors (see e-companion EC.3).

5.1. Calibration of the HGSADC Algorithm

As for most metaheuristics, evolutionary ones in particular, HGSADC relies on a set of correlated parameters and configuration choices for its key operators. In order to identify good parameter values, we adopted the *metacalibration* approach (Mercer and Sampson 1978), which was shown to perform particularly well for genetic algorithm calibration (Smit and Eiben 2009).

Metacalibration involves solving the problem of parameter optimization by means of metaheuristics. In this scope, any evaluation of a set of parameters implies launching automatically the algorithm to be calibrated (HGSADC here) on a restricted set of *training instances* and measuring its effectiveness. We used a meta-evolutionary method, the Evolutionary Strategy with Covariance Matrix Adaptation (CMA-ES) of Hansen and Ostermeier (2001) to perform this optimization, as it necessitates few parameter evaluations to converge towards good solutions.

The calibration was run independently for each problem class, with the dual objective of measuring the dependency of the best parameter set upon the problem class, and identifying an eventual set of parameters suitable for all problem classes considered. Table 1 provides a summary of HGSADC parameters, together with the range of values we estimate to be 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). The calibration results for each class, along with the final choice of parameter values for HGSADC, are also presented.

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 values of Table 1, 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 $P_m \geqslant 0.7$. 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.

Table 1. Calibration results.

Parameter		Range	PVRP	MDVRP	MDPVRP	Final parameters
μ	Population size	[5,200]	18	24	30	**25**
λ	Number of offspring in a generation	[1, 200]	33	87	146	**40/70/100**
el	Proportion of elite individuals, such that $nbElit = el \times \mu$	[0, 1]	0.38	0.45	0.36	**0.4**
nc	Proportion of close individuals considered for distance evaluation, such that $n_{close} = nc \times \mu$	[0, 0.25]	0.24	0.18	0.15	**0.2**
P_m	Education rate	[0, 1]	0.86	0.86	0.70	**1.0**
P_{rep}	Repair rate	[0, 1]	0.57	0.61	0.33	**0.5**
h	Granularity threshold in RI	[0, 1]	0.53	0.36	0.35	**0.4**
ξ^{REF}	Reference proportion of feasible individuals	[0, 1]	0.10	0.30	0.20	**0.2**

5.2. Results on Periodic and Multidepot VRPs

HGSADC was tested on the MDVRP and PVRP benchmark instances of Cordeau et al. (1997), grouped into two sets, S_1 and S_2, containing, respectively, 33 and 42 instances of various sizes, from 50 to 417 customers. It was compared to state-of-the-art methods for these problems: the tabu search of Cordeau et al. (1997) (CGL), the scatter search of Alegre et al. (2007) (ALP), the VNS of Hemmelmayr et al. (2009) (HDH), and the record-to-record ILP approach of Gulczynski et al. (2011) (GGW) for the PVRP; CGL, the fuzzy-logic guided-GA of Lau et al. (2010) (LCTP), and the adaptive large neighborhood search of Pisinger and Ropke (2007) (PR) for the MDVRP.

To study the behavior of HGSADC with respect to the number of iterations, three different stopping conditions were tested for (It_{NI}, T_{max})—(10^4, 10 min), (2.10^4, 30 min), and (5.10^4, 1 h). In all cases, the diversification parameter was set to $It_{\text{div}} = 0.4\, It_{\text{NI}}$. The instance set contains a few very large problems with more than 450 visits to customers over the different periods, for which the population size was reduced by two, and the computation time limit was increased.

Tables 2 and 3 sum up the comparison of average results from 10 independent runs of HGSADC, with various stopping conditions, to results reported for state-of-the-art algorithms. We first report the averages, over all instances, of the instance computation times (line *Time*) and the percentages of deviation from the BKS (*Gap overall*). We then present average deviations to BKS for the two instance sets, as well as for large problems with more than 150 customers. More detailed results are provided in the e-companion EC.4.

With respect to these experiments, HGSADC seems to perform remarkably well in comparison to other algorithms. During short runs of 10^4 iterations, an average overall gap of +0.20% relative to the previous BKS is achieved for the PVRP, compared to more than +1.40% for the other approaches. Similar performance is observed for MDVRP, with an average gap of −0.01% indicating that the new method is on average better than the previous BKS on all instances. Actually, during these short runs, HGSADC produced new best average results for 41 out of 42 PVRP instances and for all 33 MDVRP instances. It is noteworthy that the average standard deviation per instance obtained by HGSADC is 0.15% for PVRP and 0.05% for MDVRP, meaning that the algorithm is very reliable. New BKS were obtained for 20 instances out of 42 for the PVRP, and 9 instances out of 33 for the MDVRP.

The average computation time is short, barely higher than for other methods, and suitable for many operational decisions. For MDVRP problems especially, only 2.15 min are required, on average, to find the final solution for short MDVRP runs, the rest of the time being spent to reach the time-limit termination criteria. Using the termination criteria (5.10^4, 1 h), previous BKS are retrieved on all runs for 21 instances out of 33, while known optimal solutions from Baldacci and Mingozzi (2009) and Baldacci et al. (2011) are retrieved on every run. It is noticeable that HGSADC obtains, in a few minutes, better PVRP results than HDH,

Table 2. HGSADC performance on PVRP instances.

	CGL		HDH				HGSADC		
	(1 run)		(Avg. 10 runs)			GGW	(Avg. 10 runs)		
	15.10^3 it	ALP	10^7 it	10^8 it	10^9 it	(1 run)	10^4 it	2.10^4 it	5.10^4 it
Time (min)	4.28	3.64	3.34	33.4	334	10.36	5.56	13.74	28.21
Gap overall (%)	+1.82	—	+1.45	+0.76	+0.39	—	+0.20	+0.12	+0.07
Gap S_1 (%)	+1.62	+1.40	+1.43	+0.73	+0.37	+0.94	+0.14	+0.09	+0.04
Gap S_2 (%)	+2.48	—	+1.53	+0.83	+0.44	—	+0.38	+0.23	+0.17
Gap $n \geqslant 150$ (%)	+3.23	—	+2.16	+1.18	+0.62	—	+0.35	+0.20	+0.14

Table 3. HGSADC performance on MDVRP instances.

	CGL	PR			HGSADC		
	(1 run)	(Avg. 10 runs)		LCTP	(Avg. 10 runs)		
	15.10^3 it	25.10^3 it	50.10^3 it	(Avg. 50 runs)	10^4 it	2.10^4 it	5.10^4 it
Time	Small	1.97 min	3.54 min	2.06 min	4.24 min	8.99 min	19.11 min
Gap overall (%)	+0.96	+0.52	+0.34	+0.49	−0.01	−0.04	−0.06
Gap S_1 (%)	+0.58	+0.54	+0.35	+0.39	+0.00	−0.02	−0.03
Gap S_2 (%)	+1.85	+0.47	+0.34	+0.71	−0.04	−0.10	−0.12
Gap $n \geqslant 150$ (%)	+1.40	+0.68	+0.45	+0.70	−0.03	−0.08	−0.10

the previous state-of-the-art method for PVRP, even when HDH runs for 10^9 iterations (100 times the number of iterations for standard HDH runs), corresponding to some 300 minutes of run time.

No benchmark instance set was available for the MDPVRP. We therefore built a set of 10 MDPVRP instances by merging the PVRP and 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 could be found by HGSADC. The full data sets can be obtained from the authors.

The maximum run time was increased to 30 minutes for these experiments (It_{NI} remains set to 10^4) to account for the higher difficulty of the MDPVRP. The average results of 10 runs of HGSADC on these new instances are reported in the e-companion EC.4, and compared to the best solutions ever found during all our experiments. An average error gap of +0.42% was observed, which is reasonable given the increased problem difficulty. The average standard deviation per instance is now 0.26%, illustrating the increased irregularity of the search space. Keeping the best solution of the 10 runs leads to significantly better solutions with an average error gap of +0.13%, but requires more computational resources. This approach corresponds to the well-known independent-search strategy for parallel metaheuristics (Crainic and Toulouse 2010). More sophisticated parallel-search strategies, based on cooperation, in particular, could be used to improve the exploration of the search space and reach better results.

5.3. Results on the Capacitated VRP

The CVRP is a special case of multidepot periodic problems, when $d = 1$ and $t = 1$. HGSADC can thus be used to address the CVRP with very minor changes in the distance measure and the parameters, even though its operators were designed for multiperiod settings.

Detailed results of experiments on 34 well-known CVRP instances from the literature are reported in the e-companion EC.5. Very competitive results to state-of-the-art methods were obtained in similar computation times. An overall gap to the BKS of 0.10% was thus observed, which is equal to the performance of the best, highly specialized algorithm in the literature (Nagata and Bräysy 2009). We also retrieved 12 new best-known solutions for Golden et al. (1998) instances. The diversity management method we propose seems to compensate for the lack of problem-tailored operators and opens several promising avenues of research.

5.4. Sensitivity Analysis of Algorithmic Components

A second set of experiments targeted the analysis of the impact on the performance of the proposed metaheuristic of various algorithmic components. Sensitivity analysis was thus performed on "traditional" hybrid genetic components by "removing" each of them in turn.

The "No-Education" version was obtained by setting the probability of offspring education P_m to 0, which also meant that no repair was performed. Local-search improvement methods are thus exclusively used to produce the initial population. In the "no-population" version, $\lambda = 1$, $\mu = 0$, and *nbElite* = 1. Thus, only one individual appears in each subpopulation, the population management mechanism then behaving as a steady-state population management where the offspring replaces the parent only if it improves. The crossover either combines an individual with itself, or both feasible and infeasible individuals. Only one parent was selected by binary tournament for the "no-crossover" version, underwent education, and was inserted into the population. The parent selection was performed only in the feasible subpopulation for the "no-infeasible" algorithm, a constant high penalty value being enforced through the search. Finally, setting the repair probability P_{rep} to 0 yielded the "no-repair" version. The results are reported in Table 4, each column corresponding to the average time and gap to BKS of HGSADC without the respective component.

It is noticeable that all these algorithmic components play an important role in the good performance of the proposed metaheuristic, the most crucial being education followed by population, crossover, infeasible solutions, and repair to a lesser extent.

The second part of the sensitivity analysis was dedicated to the adaptive population diversity control mechanism, which is a cornerstone of the proposed methodology.

Table 4. Sensitivity analysis on main HGSADC components.

Benchmark	No-edu	No-pop	No-cross	No-inf	No-rep	HGSADC
PVRP						
T (min)	0.89	4.21	4.42	5.39	5.20	5.56
%	+4.24	+2.19	+1.94	+0.80	+0.19	+0.20
MDVRP						
T (min)	0.83	3.49	4.21	4.45	3.58	4.24
%	+7.10	+9.54	+7.04	+0.45	+0.07	−0.01
MDPVRP						
T (min)	0.89	9.29	11.21	15.47	13.22	15.96
%	+25.22	+16.90	+8.39	+1.40	+0.54	+0.42

We therefore compared its performance to those of two mechanisms from the literature, mechanisms that proved their worth in their respective contexts. Two new algorithms were thus derived from HGSADC to conform to each of these two rules, as well as a variant without diversity control (identified as *HGS*0).

The *HGS*1 variant involves a dispersal rule in the objective space as in Prins (2004). Let F be the fitness function, defined as the cost, and Δ_F a fitness spacing parameter. Acceptance of an individual I in the population is granted only if $|F(I) - F(C)| \geqslant \Delta_F$ for all individuals C already in the population. The second variant, named *HGS*2, relies on the population management framework of Sörensen and Sevaux (2006). Let Δ_D be a spacing parameter and δ_H the distance measure presented in §4.3. To be added to the population, an individual I must obey a dispersal rule, i.e., it must verify $\delta_H(I, C) \geqslant \Delta_D$ 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 the value of Δ_D decreases progressively toward zero as the method approaches the termination criteria to encourage the exploitation of good solutions. For both methods, we use an incremental population management and only individuals with a fitness below the median of the population can be discarded.

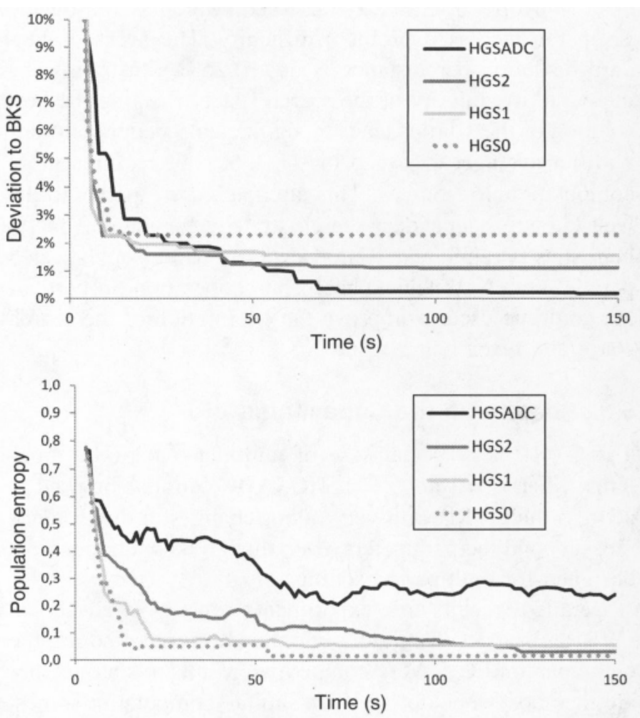
Table 5 reports the average gaps to BKS and average run times for each method on the instances presented in §5.2. One observes that the results verify that applying the dispersal rule with respect to the solution space (HGS2) is more effective than using the dispersal rule with respect to the objective space (HGS1). One also observes that proceeding without diversity management yields rather poor results compared to all other strategies. The best results are definitely obtained with the proposed adaptive diversity management method, which yields the best average gap for an equivalent computational effort.

Table 5. Comparison of population-diversity management mechanisms.

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

Figure 3 illustrates the behavior of the four population-diversity management strategies during one of the runs (150 seconds) on MDPVRP instance pr03, as measured by the population entropy and the gap to the BKS. The population entropy is computed as the average distance from one individual to another. All algorithms close the gap to less than 2.50% within 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 metaheuristic is still regularly improving its best

Figure 3. Population entropy and error gap to the BKS for the diversity management strategies on MDPVRP instance pr03.



found solution as the time limit approaches, despite being already very close to the best-known solution (a gap of 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, on the other hand, maintains a healthy diversity in the population, as illustrated by a 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 us to avoid premature convergence and to reach high-quality solutions.

6. Conclusions and Research Perspectives

We proposed a new hybrid genetic search metaheuristic to efficiently address several classes of multidepot 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 paper introduces several methodological contributions. In particular, in the crossover and education operators, the management of infeasible solutions, the individual evaluation procedure driven both by solution cost and the contribution to population diversity and, more generally, the adaptive population management mechanism that enhances diversity, allows a broader access to reproduction, and preserves 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* metaheuristic 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 metaheuristics for each particular problem class. Moreover, with minimal adjustments, it obtains comparable results to the best methods for the CVRP.

Among the many interesting avenues of research, we mention 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.

Electronic Companion

An electronic companion to this paper is available as part of the online version at http://dx.doi.org/10.1287/opre.1120.1048.

Acknowledgments

The authors thank the referees for their helpful comments that significantly contributed to improving the quality of the paper. While working on this project, T. G. Crainic was the NSERC Industrial Research Chair in Logistics Management, ESG, University of Quebec in Montreal, N. Lahrichi was postdoctoral fellow with the Chair, and M. Gendreau was the NSERC/Hydro-Québec Industrial Research Chair on the Stochastic Optimization of Electricity Generation, MAGI, École Polytechnique. 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 Grant programs, by the authors' partners CN, Rona, Alimentation Couche-Tard, la Fédération des producteurs de lait du Québec, 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

Alegre J, Laguna M, Pacheco J (2007) Optimizing the periodic pick-up of raw materials for a manufacturer of auto parts. *Eur. J. Oper. Res.* 179(3):736–746.

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

Baldacci R, Bartolini E, Mingozzi A, Valletta A (2011) An exact algorithm for the period routing problem. *Oper. Res.* 59(1):228–241.

Beltrami EJ, Bodin LD (1974) Networks and vehicle routing for municipal waste collection. *Networks* 4(1):65–94.

Blakeley F, Bozkaya B, Cao B, Hall W, Knolmajer J (2003) Optimizing periodic maintenance operations for Schindler Elevator Corporation. *Interfaces* 33(1):67–79.

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

Chao IM, Golden BL, Wasil E (1995) An improved heuristic for the period vehicle routing problem. *Networks* 26(1):25–44.

Chu F, Labadi N, Prins C (2006) A scatter search for the periodic capacitated arc routing problem. *Eur. J. Oper. Res.* 169(2):586–605.

Coene S, Arnout A, Spieksma FCR (2010) On a periodic vehicle routing problem. *J. Oper. Res. Soc.* 61(12):1719–1728.

Cordeau J-F, Gendreau M, Laporte G (1997) A tabu search heuristic for periodic and multidepot vehicle routing problems. *Networks* 30(2):105–119.

Cordeau J-F, Laporte G, Mercier A (2001) A unified tabu search heuristic for vehicle routing problems with time windows. *J. Oper. Res. Soc.* 52(8):928–936.

Cordeau J-F, Laporte G, Savelsbergh MWF, Vigo D (2007) Vehicle routing. Barnhart C, Laporte G, eds. *Transportation*. Handbooks in Operations Research and Management Science (North-Holland, Amsterdam), 367–428.

Crainic TG, Toulouse M (2010) Parallel meta-heuristics. Gendreau M, Potvin J-Y, eds. *Handbook of Metaheuristics* (Springer, New York), 497–541.

Crainic TG, Crisan GC, Gendreau M, Lahrichi N, Rei W (2009) Multi-thread cooperative optimization for rich combinatorial problems. *23rd IEEE Internat. Parallel and Distributed Processing Sympos., IPDPS 2009*.

Dantzig GB, Ramser JH (1959) The truck dispatching problem. *Management Sci.* 6(1):80–91.

Dongarra J (2011) Performance of various computers using standard linear equations software. Technical report, University of Tennessee, Knoxville, TN.

Drummond LMA, Ochi LS, Vianna DS (2001) An asynchronous parallel metaheuristic for the period vehicle routing problem. *Future Generation Comput. Systems* 17(4):379–386.

Francis PM, Smilowitz KR, Tzur M (2008) The period vehicle routing problem and its extensions. Golden BL, Raghavan S, Wasil EA, eds. *The Vehicle Routing Problem: Latest Advances and New Challenges* (Springer, New York), 73–102.

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

Glover F, Hao J-K (2011) The case for strategic oscillation. *Ann. Oper. Res.* 183(1):163–173.

Golden BL, Wasil EA (1987) Computerized vehicle routing in the soft drink industry. *Oper. Res.* 35(1):6–17.

Golden BL, Raghavan S, Wasil EA, eds. (2008) *The Vehicle Routing Problem: Latest Advances and New Challenges* (Springer, New York).

Golden BL, Wasil EA, Kelly JP, Chao IM (1998) Metaheuristics in vehicle routing. Crainic TG, Laporte G, eds. *Fleet Management and Logistics* (Kluwer, Boston), 33–56.

Gulczynski D, Golden B, Wasil E (2011) The period vehicle routing problem: New heuristics and real-world variants. *Transportation Res. Part E: Logist. Transportation Rev.* 47(5):648–668.

Hadjiconstantinou E, Baldacci R (1998) A multidepot period vehicle routing problem arising in the utilities sector. *J. Oper. Res. Soc.* 49(12):1239–1248.

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

Hemmelmayr VC, Doerner KF, Hartl RF (2009) A variable neighborhood search heuristic for periodic routing problems. *Eur. J. Oper. Res.* 195(3):791–802.

Hoff A, Andersson H, Christiansen M, Hasle G, Løkketangen A (2010) Industrial aspects and literature survey: Fleet composition and routing. *Comput. Oper. Res.* 37(12):2041–2061.

Holland JH (1975) *Adaptation in Natural and Artificial Systems. An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence* (The University of Michigan Press, Ann Arbor).

Kang KH, Lee YH, Lee BK (2005) An exact algorithm for multi depot and multi period vehicle scheduling problem. *Lecture Notes in Computer Science*, Vol. 3483 (Springer, Berlin), 350–359.

Lacomme P, Prins C, Ramdane-Cherif W (2005) Evolutionary algorithms for periodic arc routing problems. *Eur. J. Oper. Res.* 165(2):535–553.

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

Lau HCW, Chan TM, Tsui WT, Pang WK (2010) Application of genetic algorithms to solve the multidepot vehicle routing problem. *IEEE Trans. Automation Sci. Engrg.* 7(2):383–392.

Matos AC, Oliveira RC (2004) An experimental study of the ant colony system for the period vehicle routing problem. *Lecture Notes in Computer Science*, Vol. 3172 (Springer, Berlin), 1–29.

Mercer RE, Sampson JR (1978) Adaptive search using a reproductive metaplan. *Kybernetes* 7(3):215–228.

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

Ombuki-Berman B, Hanshar T (2009) Using genetic algorithms for multidepot vehicle routing. Pereira FB, Tavares J, eds. *Bio-Inspired Algorithms for the Vehicle Routing Problem* (Springer, Berlin), 77–99.

Parthanadee P, Logendran R (2006) Periodic product distribution from multidepots under limited supplies. *IIE Trans.* 38(11):1009–1026.

Pirkwieser S, Raidl GR (2010) Multilevel variable neighborhood search for periodic routing problems. *Lecture Notes in Computer Science*, Vol. 6022 (Springer, Berlin), 226–238.

Pisinger D, Ropke S (2007) A general heuristic for vehicle routing problems. *Comput. Oper. Res.* 34(8):2403–2435.

Prins C (2004) A simple and effective evolutionary algorithm for the vehicle routing problem. *Comput. Oper. Res.* 31(12):1985–2002.

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

Smit SK, Eiben AE (2009) Comparing parameter tuning methods for evolutionary algorithms. *Proc. 11th Conf. Evol. Comput.* (IEEE Press, Piscataway, NJ), 399–406.

Sörensen K, Sevaux M (2006) MAPM: Memetic algorithms with population management. *Comput. Oper. Res.* 33(5):1214–1225.

Teixeira J, Antunes AP, de Sousa JP (2004) Recyclable waste collection planning—A case study. *Eur. J. Oper. Res.* 158(3):543–554.

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

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

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

Yang WT, Chu LC (2000) A heuristic algorithm for the multidepot periodic vehicle routing problem. *J. Inform. Optim. Sci.* 22:359–367.

Yu B, Yang ZZ, Xie JX (2011) A parallel improved ant colony optimization for multidepot vehicle routing problem. *J. Oper. Res. Soc.* 62(1):183–188.

Thibaut Vidal is a member of the Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT). His main research interests are in vehicle routing, scheduling, neighborhood search design, and metaheuristics.

Teodor Gabriel Crainic is professor of operations research, transportation, and logistics and is the Logistics Management Chair in the School of Management at the University of Quebec in Montréal, Canada. He is senior scientist at CIRRELT, the Interuniversity Research Center for Enterprise Networks, Logistics and Transportation, and director of its Intelligent Transportation Systems Laboratory. His main research interests are in network, integer, and combinatorial optimization, metaheuristics, and parallel computation applied to the planning and management of complex systems, mainly in transportation, logistics, and telecommunication.

Michel Gendreau is professor of operations research at the Department of Applied Mathematics and Industrial Engineering of École Polytechnique de Montréal, Canada, where he holds the NSERC/Hydro-Québec Chair on the Stochastic Optimization of Electricity Generation. His main research area is the application of operations research methods to energy, transportation, and telecommunication planning. He is the editor-in-chief of *Transportation Science*.

Nadia Lahrichi is professor of operations research at the Department of Mathematics and Industrial Engineering, École Polytechnique de Montréal, Canada. Her main research focuses on transportation and healthcare management, more specifically on the organization of healthcare services, patient flow, and resource optimization.

Walter Rei is professor of operations research and logistics at the School of Management, University of Quebec in Montréal, Canada. He is also a member of the Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT). His main research interests include integer stochastic programming and combinatorial optimization.