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

Thibaut Vidal
Teodor Gabriel Crainic
Michel Gendreau
Nadia Lahrichi
Walter Rei

January 2011

CIRRELT-2011-05

# 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 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 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.

**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.

[†]This version updates the CIRRELT-2010-34

---

* Corresponding author: Teodor.Gabriel@cirrelt.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 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., 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; 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. 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 multi-depot VRP, *MDVRP*, the periodic VRP, *PVRP*, and the multi-depot 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), food collection or distribution (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. Thus, for all currently available benchmark instances for the three problem classes, HGSADC identifies either the best known solutions, including the optimal ones, or new best solutions.

To sum up, the main contributions of this article are: 1) A new meta-heuristic that is highly effective for three important vehicle routing problem classes, the MDVRP, the PVRP, and the MDPVRP. The meta-heuristic equals or outperforms the current best methods proposed for each particular class and requires a limited computational effort. Moreover, with very limited adaptation, it also proves extremely competitive for the CVRP. 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 solution cost 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 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 $\mathcal{V} = \mathcal{V}^{\text{DEP}} \cup \mathcal{V}^{\text{CST}}$. The unique vertex $v_0 \in \mathcal{V}^{\text{DEP}}$ represents the depot where the product to be distributed is kept and a fleet of $m$ identical vehicles with capacity $Q$ is based. Vertices $v_i \in \mathcal{V}^{\text{CST}}$ stand for customers $i,\ i = 1, \dots, n$, requiring service and characterized by a non-negative demand $q_i$ and a service duration $\tau_i$. 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 Multi-Depot VRP, $m$ representing the number of vehicles available at each depot. In this case, vertices $v_0, \dots, v_d$ make up the set $\mathcal{V}^{\text{DEP}}$, while the remaining vertices $\mathcal{V}^{\text{CST}}$ stand for customers. 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, 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 a MDPVRP with $d$ depots and $t$ periods into an equivalent PVRP with $d \times t$ periods corresponding to (depot, period) couples. (The Annex 7.1 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 Section 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 meta-heuristic 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 meta-heuristics 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. The PVRP is first transformed into a large VRP containing each customer as many times as given by its frequency and addressed by an ACO method. The problem of distributing the resulting routes among periods is then solved as a graph coloring problem, with occasional changes in customer patterns to progress towards a feasible PVRP solution. In a final step, ACO 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, 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. It has only been outperformed recently by the variable neighborhood search (VNS) of Hemmelmayr et al. (2009), which is built upon various well-known VRP neighborhoods, e.g., string relocate, swap, and 3-opt. Finally, one should notice the VNS algorithm with multilevel refinement strategy of Pirkwieser and Raidl (2010), specifically 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. Thangiah and Salhi (2001) represented solutions as circles in the 2D space, whereas Ombuki-Berman and Hanshar (2009) introduced a mutation operator that targets the depot assignment to "borderline" customers, which are close to several depots. In this case also, however, neighborhood-based methods, such as 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 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. (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; 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 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 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 Genetic Search with Adaptive Diversity Control* (*HGSADC*) meta-heuristic 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 meta-heuristic 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 (Section 4.1), which is often the norm, but also the contribution the individual makes to the diversity of the gene pool. It thus contributes to maintain 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 $< T_{max}$ do
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 $C$ infeasible then insert $C$ into infeasible sub-population; repair with probability $P_{rep}$
7:      if $C$ feasible then insert $C$ into feasible sub-population
8:      if maximum sub-population 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: end while
12: Return best feasible solution
```

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

## 4.1 Search space

The meta-heuristic 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, 2009). 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 $\sigma_0^r \in \mathcal{V}^{\mathrm{DEP}}$, visits a sequence of $n_r$ customers $\sigma_1^r, \ldots, \sigma_{n_r}^r \in \mathcal{V}^{\mathrm{CST}}$, and returns to the same depot $\sigma_{n_r+1}^r = \sigma_0^r$. It is characterized by load $q(r) = \sum\limits_{i=1}^{n_r} q_{\sigma_i^r}$, driving time $c(r) = \sum\limits_{i=0}^{n_r} c_{\sigma_i^r \sigma_{i+1}^r}$, and total duration $\tau(r) = c(r) + \sum\limits_{i=1}^{n_r} \tau_{\sigma_i^r}$.

Let $\omega^{\mathrm{Q}}$ and $\omega^{\mathrm{D}}$ 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.

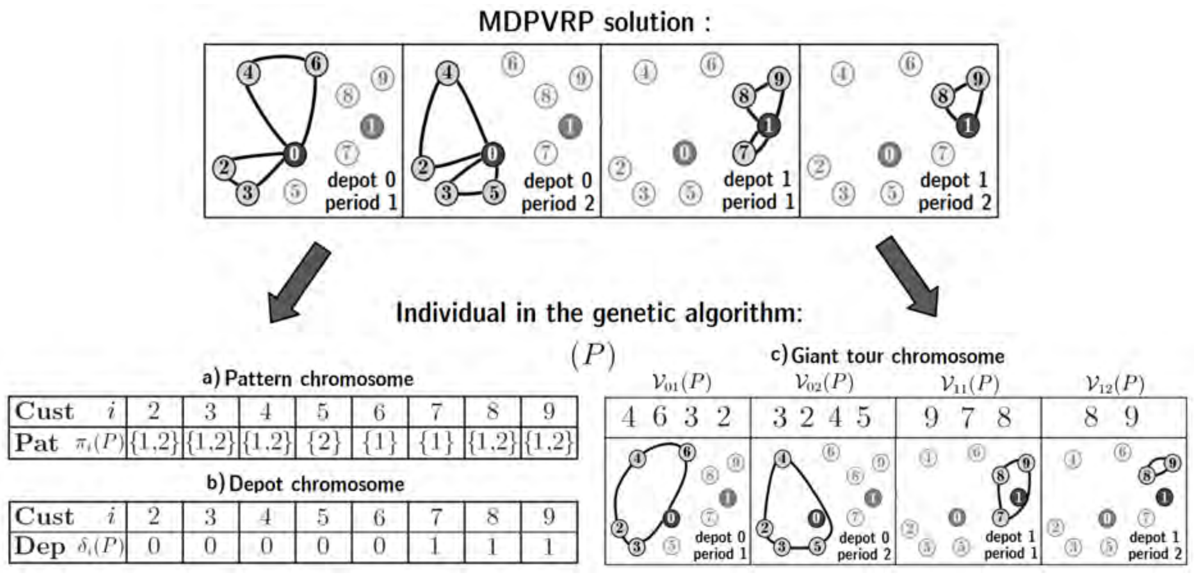
$$\phi(r) = c(r) + \omega^{\mathrm{D}} \max\{0, \tau(r) - T)\} + \omega^{\mathrm{Q}} \max\{0, q(r) - Q\} \tag{1}$$

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 and efficient crossover procedures working on permutations, but requires an algorithm to find the optimal segmentation of the tour into routes



| Cust $i$ | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|
| Pat $\pi_i(P)$ | {1,2} | {1,2} | {1,2} | {2} | {1} | {1} | {1,2} | {1,2} |

| Cust $i$ | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|
| Dep $\delta_i(P)$ | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |

| $\mathcal{V}_{01}(P)$ | $\mathcal{V}_{02}(P)$ | $\mathcal{V}_{11}(P)$ | $\mathcal{V}_{12}(P)$ |
|---|---|---|---|
| 4 6 3 2 | 3 2 4 5 | 9 7 8 | 8 9 |

Figure 1: From a MDPVRP solution to the individual chromosome representation

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 7.2.

## 4.3 Evaluation of individuals

The individual-evaluation function in population-based meta-heuristics 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 meta-heuristics. 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 $\mathcal{N}_{\rfloor\updownarrow\wr\int\rceil}$, computed according to Equation (2). Several distance measures were tested in the experiments leading to the final algorithm. A normalized Hamming distance $\delta^{\mathrm{H}}(P_1, P_2)$, 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 multi-depot, multi-period 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.

$$\Delta(P) = \frac{1}{n_{close}} \sum_{P_2 \in \mathcal{N}_{\rfloor\updownarrow\wr\int\rceil}} \delta^{\mathrm{H}}(P, P_2) \tag{2}$$

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

Let $fit(P)$ and $dc(P)$ 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 $BF(P)$ 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.

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

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 (Section 4.4) and survivor (Section 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 Section 4.6.

## 4.4 Parent Selection and Crossover

The offspring generation scheme of HGADSC 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 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 $\Lambda_1$;
4: Randomly select $n_2 - n_1$ remaining couples to form the set $\Lambda_2$;
5: The remaining $td - n_2$ couples make up the set $\Lambda_{mix}$.

6: STEP 1: INHERIT DATA FROM $P_1$.
7: for each (depot, period) $(o,l)$, belonging to set
8:     $\Lambda_1$: Copy the sequence of customer visits from $\mathcal{V}_{o,l}(P_1)$ to $\mathcal{V}_{o,l}(C)$;
9:     $\Lambda_{mix}$: Randomly (uniform distribution) select two chromosome-cutting points $\alpha_{kl}$ and $\beta_{kl}$;
       copy the $\alpha_{kl}$ to $\beta_{kl}$ 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) $(o,l) \in \Lambda_2 \bigcup \Lambda_{mix}$ 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
        (Section 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 $\Lambda_1$, $\Lambda_2$, and $\Lambda_{mix}$ 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 $\Lambda_1 = \{(\text{d1,p2})\}$, $\Lambda_2 = \{(\text{d1,p1})\}$, and $\Lambda_{mix} = \{(\text{d0,p1}), (\text{d0,p2})\}$.

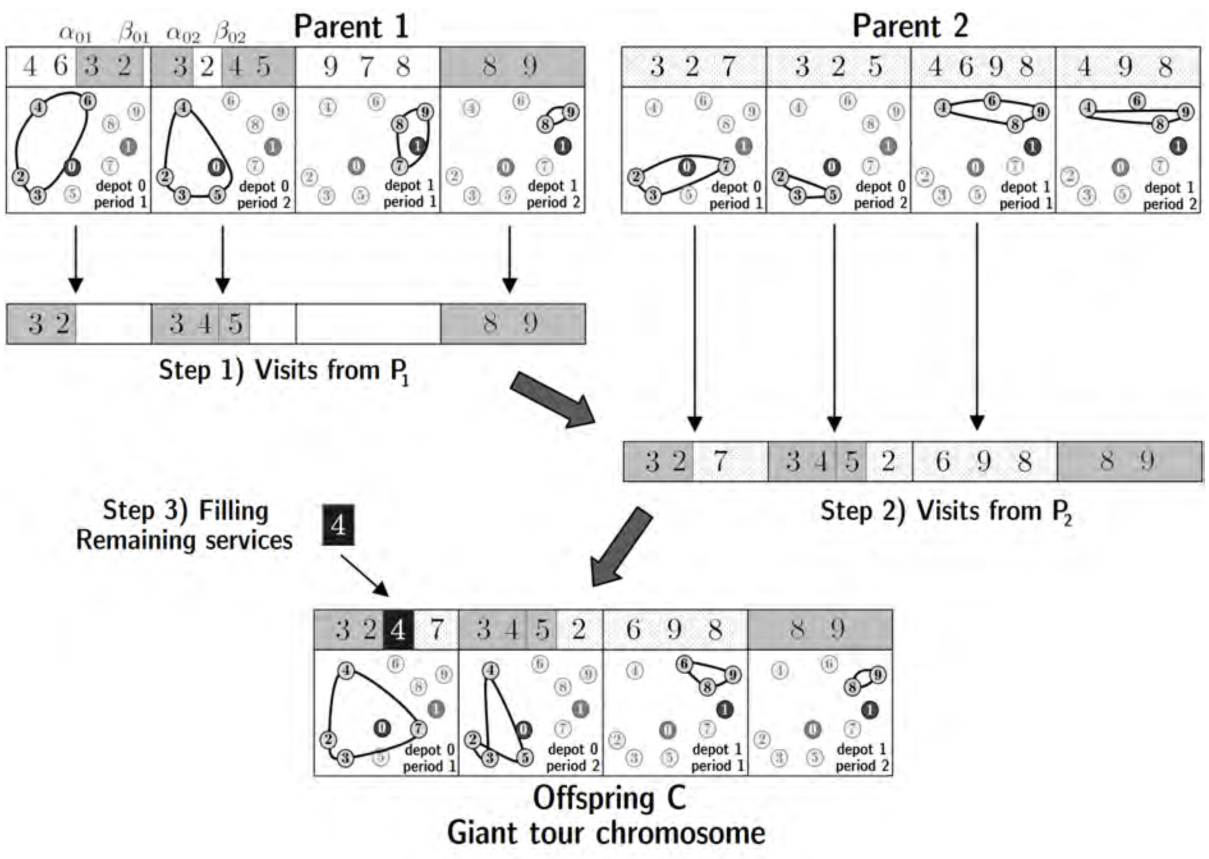


Figure 2: The PIX crossover

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 $\Lambda_1$, or a random subsequence, if it belongs to $\Lambda_{mix}$. Thus, in Figure 2, all customers serviced from depot 1 in period 2 ($\Lambda_1 = \{(d1, p2)\}$) are inherited from $P_1$, while only subsets are inherited for depot 0 at periods 1 and 2 ($\Lambda_{mix} = \{(\text{d0,p1}), (\text{d0,p2})\}$). Step 2 targets the second parent and, thus, the material from (depot, period) couples in $\Lambda_2 \bigcup \Lambda_{mix}$. 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 $\Lambda_2 \bigcup \Lambda_{mix}$ 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] 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 the 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. Yet, to enhance the precision of the insertion, and because the routes must be extracted in all cases, before undertaking the next phase of the meta-heuristic (Section 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, as it includes several local-search procedures based on neighborhoods for the VRP. A *Repair* phase eventually 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 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 $r(u) = r(v)$, 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*, while 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 intra-route move, while moves M8 and M9 are 2-opt* inter-route 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 intra-period 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 re-starting 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 $\mu$ and $\mu + \lambda$ individuals, the former representing the minimum subpopulation size, and the latter 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 $\lambda$ 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, $4\mu$ 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 $\mu$ individuals.

**Penalty parameter adjustment.** The penalty parameters are initially set to $\omega^{\mathrm{D}} = 1$ and $\omega^{\mathrm{Q}} = \bar{c}/\bar{q}$, 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 Section 4.5. Let $\xi^{\mathrm{REF}}$ be a target proportion of naturally-feasible individuals, and $\xi^{\mathrm{Q}}$ and $\xi^{\mathrm{D}}$ 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 $\text{PAR} = \text{Q}, \text{D}$:

- if $\xi^{\mathrm{PAR}} \leq \xi^{\mathrm{REF}} - 0.05$, then $\omega^{\mathrm{PAR}} = \omega^{\mathrm{PAR}} \times 1.2$;

- if $\xi^{\text{PAR}} \geq \xi^{\text{REF}} + 0.05$, 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 $4\mu$ 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 (Section 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 $\mu$ 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 $\lambda$ 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., $\delta^{\text{H}}(P_1, P_2) = 0$, 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 $i = 1 \ldots \lambda$ do
2:     $X \leftarrow$ all individuals having a clone
3:     if $X \neq \emptyset$ 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
6: end for
```

**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., $fit(J) = nbIndiv$, and thus $BF(J) \geq nbIndiv + 1$. $P$ belongs to the best $nbElit$ solutions in terms of cost, thus $BF(P) \leq nbElit + (1 - \frac{nbElit}{nbIndiv})(nbIndiv) \leq nbIndiv$. Individual $P$ will not be removed as $J$ has a worst biased fitness. □

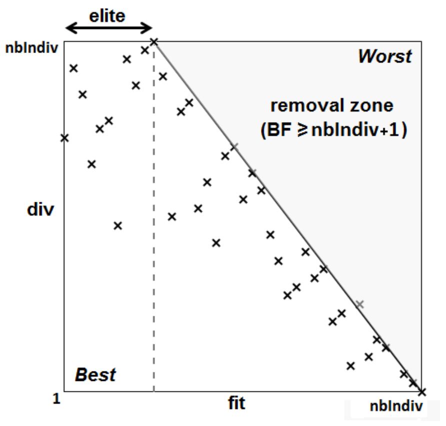


Figure 3: Illustration of the survivor selection property

Figure 3 represents the fitness and diversity measures of a subpopulation taken from our experimentations after the survivor-selection procedure was run, where *nbElit* individuals are considered an elite. The figure also displays the *removal zone*, where individuals with $BF \geq nbIndiv + 1$ can be eliminated according to the biased fitness criteria.

## 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 multi-period, multi-depot settings (Section 5.2), as well as for the capacitated VRP (Section 5.3). The latter is discussed in Section 5.4, while the calibration of the meta-heuristic is discussed in Section 5.1.

HGSADC was implemented in C++. Experiments were run on a 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 (2009) factors (see e-companion 7.3).

## 5.1 Calibration of the HGSADC algorithm

As for most meta-heuristics, 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 *meta-calibration* approach (Mercer and Sampson, 1978), which was shown to perform particularly well for genetic-algorithm calibration (Smit and Eiben, 2009).

Meta-calibration involves solving the problem of parameter optimization by means of meta-heuristics. 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.

Table 1: Calibration Results

| Parameter | | Range | PVRP | MDVRP | MDPVRP | **Final Params** |
|---|---|---|---|---|---|---|
| $\mu$ | Population size | [5,200] | 18 | 24 | 30 | **25** |
| $\lambda$ | 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** |
| $\xi^{\mathrm{REF}}$ | Reference proportion of feasible individuals | [0,1] | 0.10 | 0.30 | 0.20 | **0.2** |

Except for the generation size $\lambda$, 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 $P_m \geq 0.7$. 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 $\lambda$, which is set to 40 for the PVRP, 70 for MDVRP, and 100 for the MDPVRP.

## 5.2 Results on periodic and multi-depot VRPs

HGSADC was tested on the MDVRP and PVRP benchmark instances of Cordeau et al. (1997), 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), and the variable neighborhood search of Hemmelmayr et al. (2009) (HDH), for the PVRP; CGL and the adaptive large neighborhood search of Pisinger and Ropke (2007) (PR) for the MDVRP.

To further study the behavior of HGSADC with respect to the number of iterations, three different stopping conditions were tested for $(It_{NI}, T_{max})$, $(10^4, 10min)$, $(2.10^4, 30min)$, and $(5.10^4, 1h)$. In all cases, the diversification parameter was set to $It_{div} = 0.4 It_{NI}$. 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 with various number of iterations. For each algorithm, we report the averages for the 42 PVRPs and the 33 MDVRPs of the instance computation time (line *Time*) and percentage of deviation from the BKS (*Gap*). Detailed results are provided in the e-companion 7.4.

Table 2: HGSADC performance on PVRP instances

| | **CGL** *(1 run)* | **HDH** *(Avg. 10 runs)* | | | **ALP** | **HGSADC** *(Avg. 10 runs)* | | |
|---|---|---|---|---|---|---|---|---|
| | $15.10^3$ it | $10^7$ it | $10^8$ it | $10^9$ it | — | $10^4$ it | $2.10^4$ it | $5.10^4$ it |
| Time | 4.28 min | 3.34 min | — | — | — | 5.56 min | 13.74 min | 28.21 min |
| Gap | +1.82% | +1.45% | +0.76% | +0.39% | +1.40% | +0.20% | +0.12% | +0.07% |

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

Table 3: HGSADC performance on MDVRP instances

| | **CGL** *(1 run)* | **PR** *(Avg. 10 runs)* | | **HGSADC** *(Avg. 10 runs)* | | |
|---|---|---|---|---|---|---|
| | $15.10^3$ it | $25.10^3$ it | $50.10^3$ it | $10^4$ it | $2.10^4$ it | $5.10^4$ it |
| Time | small | 1.97 min | 3.54 min | 4.24 min | 8.99 min | 19.11 min |
| Gap | +0.96% | +0.52% | +0.34% | -0.01% | -0.04% | -0.06% |

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 our 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 $(5.10^4, 1h)$, previous BKS are retrieved on all runs for 21 instances out of 33, while known optimal solutions from Baldacci and Mingozzi (2009); Baldacci et al. (2010) are retrieved on every run. It is noticeable that HGSADC obtains in a few minutes better PVRP results than HDH, 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 7.4, 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 meta-heuristics (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 multi-depot 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 multi-period settings.

Detailed results of experiments on 34 well-known CVRP instances from the literature are reported in the e-companion 7.5. Very competitive results to state-of-the-art methods were obtained in similar computation times. An overall gap to the BKS of 0.11% 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 meta-heuristic 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. 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.

Table 4: Sensitivity analysis on main HGSADC components

| **Benchmark** | | **No-Edu** | **No-Pop** | **No-Cross** | **No-Inf** | **No-Rep** | **HGSADC** |
|---|---|---|---|---|---|---|---|
| **PVRP** | T | 0.89 min | 4.21 min | 4.42 min | 5.39 min | 5.20 min | 5.56 min |
| | % | +4.24% | +2.19% | +1.94% | +0.80% | +0.19% | +0.20% |
| **MDVRP** | T | 0.83 min | 3.49 min | 4.21 min | 4.45 min | 3.58 min | 4.24 min |
| | % | +7.10% | +9.54% | +7.04% | +0.45% | +0.07% | -0.01% |
| **MDPVRP** | T | 0.89 min | 9.29 min | 11.21 min | 15.47 min | 13.22 min | 15.96 min |
| | % | +25.22% | +16.90% | +8.39% | +1.40% | +0.54% | +0.42 % |

It is noticeable that all these algorithmic components play an important role in the good performance of the proposed meta-heuristic, 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. 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 *HGS0*).

The *HGS1* variant involves a dispersal rule in the objective space as in Prins (2004). Let $F$ be the fitness function, defined as the cost, and $\Delta_F$ a fitness spacing parameter. Acceptance of an individual $I$ in the population is granted only if $|F(I) - F(C)| \geq \Delta_F$ for all individuals $C$ already in the population. The second variant, named *HGS2*, relies on the population management framework of Sörensen and Sevaux (2006). Let $\Delta_D$ be a spacing parameter and $\delta_H$ the distance measure presented in Section 4.3. To be added to the population, an individual $I$ must obey a dispersal rule, i.e., it must verify $\delta_H(I, C) \geq \Delta_D$ for all $C$ already in the population. In our implementation, the value of $\Delta_D$ changes during run time: strong distance constraints are imposed at the beginning of the search to encourage exploration, whereas the value of $\Delta_D$ 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 Section 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), which is an indication of the

Table 5: 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% |

interest of the hybrid evolution strategy of HGSADC. 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.

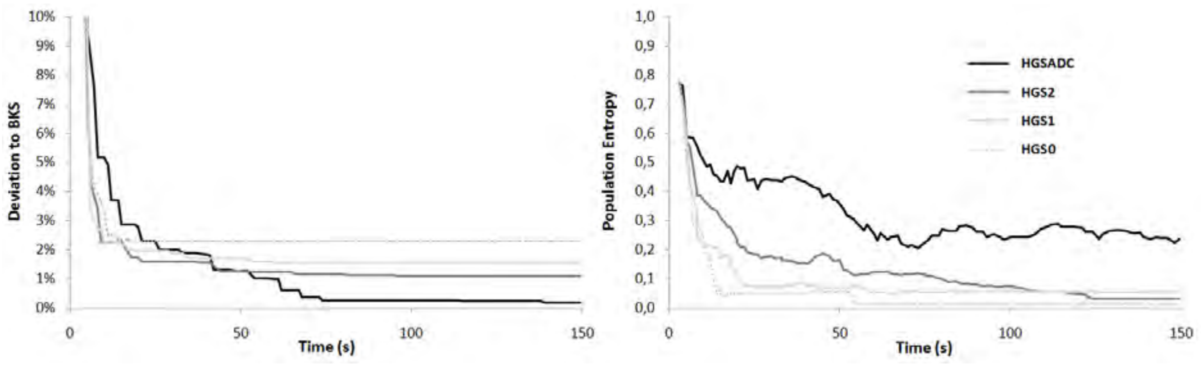


Figure 4: Population entropy and error gap to the BKS for the diversity management strategies on MDPVRP instance pr03

Figure 4 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 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 (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 to avoid premature convergence and to reach high quality solutions.

## 6 Conclusions and Research Perspectives

We proposed 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 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* 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. 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.

## Acknowledgments

While working on this project, T.G. Crainic was the NSERC Industrial Research Chair in 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 our 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

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.

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, 2006.

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 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. Multi-thread cooperative optimization for rich combinatorial problems. In *Proceedings of the 23rd IEEE International Parallel & Distributed Processing Symposium, IPDPS 2009*, 2009.

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.

F. Glover and J.-K. Hao. The case for strategic oscillation. *Annals of Operations Research*, 2009. On-line DOI 10.1007/s10479-009-0597-1.

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, E. A. Wasil, J. P. Kelly, and I. M. Chao. Metaheuristics in vehicle routing. In T. Crainic and G. Laporte, editors, *Fleet management and logistics*, pages 33–56. Kluwer, Boston, 1998.

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.

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. E. Mercer and J. R. 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.

C. Prins. Two memetic algorithms for heterogeneous fleet vehicle routing problems. *Eng. Appl. Artif. Intell.*, 22(6):916–928, 2009. ISSN 0952-1976. doi: http://dx.doi.org/10.1016/j.engappai.2008.10.006.

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. K. Smit and A. E. 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.

E. Zachariadis and C. Kiranoudis. A strategy for reducing the computational complexity of local search-based methods for the vehicle routing problem. *Computers & Operations Research*, 37(12):2089–2105, 2010.

# 7 Annex - Details on formulations, procedures, and computational experiments

Section 7.1 details the multi-depot periodic VRP formulation and the transformation proposition of the MDPVRP into the PVRP. Additional precisions and examples for the Split procedures are given in Section 7.2. The remaining three sections are focused on the experimental studies, and provide respectively details on time comparisons between different CPUs (Section 7.3), detailed results on multi-depot and periodic VRP benchmarks (Section 7.4), and the results for the CVRP benchmarks (Section 7.5).

## 7.1 Problem formulation and transformations

A PVRP formulation was introduced in Cordeau et al. (1997). We now introduce a MDPVRP formulation, as a five-index vehicle flow formulation, and show how it reduces to the PVRP.

Let $c_{ij}$ be the routing cost from vertex $v_i$ to vertex $v_j \in \mathcal{V}$. Let the binary constants $a_{pl}$ be equal to 1 if and only if day $l$ belongs to visit combination (pattern) $p$ and 0 otherwise. Two sets of binary variables are defined. For every $v_i \in \mathcal{V}^{\text{CST}}$, $p \in L_i$, and $v_o \in \mathcal{V}^{\text{DEP}}$, $y_{ipo}$ equals 1 if and only if customer $i$ is assigned to visit combination $p$ and depot $o$. For any $(v_i, v_j) \in \mathcal{V}^2$, $k = 1 \dots m$, $l = 1 \dots t$, and $v_o \in \mathcal{V}^{\text{DEP}}$, $x_{ijklo}$ takes value 1 if and only if vehicle $k$ coming from depot $o$ on day $l$ visits $v_j$ immediately after $v_i$. Using by convention $\tau_o = 0, \forall v_o \in \mathcal{V}^{\text{DEP}}$, the MDPVRP can be stated as follows:

$$\text{Minimize} \quad \sum_{v_i \in \mathcal{V}} \sum_{v_j \in \mathcal{V}} \sum_{k=1}^{m} \sum_{l=1}^{t} \sum_{v_o \in \mathcal{V}^{\text{DEP}}} c_{ij} x_{ijklo} \tag{5}$$

$$\text{Subject to: } \sum_{p \in L_i} \sum_{v_o \in \mathcal{V}^{\text{DEP}}} y_{ipo} = 1 \qquad v_i \in \mathcal{V}^{\text{CST}} \tag{6}$$

$$\sum_{v_j \in \mathcal{V}} \sum_{k=1}^{m} x_{ijklo} - \sum_{p \in L_i} a_{pl} y_{ipo} = 0 \qquad v_i \in \mathcal{V}^{\text{CST}} \;;\; v_o \in \mathcal{V}^{\text{DEP}} \;;\; l = 1 \dots t \tag{7}$$

$$\sum_{v_j \in \mathcal{V}} x_{ojklo} \leq 1 \qquad v_o \in \mathcal{V}^{\text{DEP}} \;;\; k = 1 \dots m \;;\; l = 1 \dots t \tag{8}$$

$$\sum_{v_j \in \mathcal{V}} x_{ijklo} = 0 \quad v_i \in \mathcal{V}^{\text{DEP}} \;;\; v_o \in \mathcal{V}^{\text{DEP}} \;;\; v_o \neq v_i \;;\; k = 1 \dots m \;;\; l = 1 \dots t \tag{9}$$

$$\sum_{v_j \in \mathcal{V}} x_{jiklo} - \sum_{v_j \in \mathcal{V}} x_{ijklo} = 0 \qquad v_i \in \mathcal{V} \;;\; v_o \in \mathcal{V}^{\text{DEP}} \;;\; k = 1 \dots m \;;\; l = 1 \dots t \tag{10}$$

$$\sum_{v_i \in \mathcal{V}} \sum_{v_j \in \mathcal{V}} q_i x_{ijklo} \leq Q \qquad v_o \in \mathcal{V}^{\text{DEP}} \;;\; k = 1 \dots m \;;\; l = 1 \dots t \tag{11}$$

$$\sum_{v_i \in \mathcal{V}} \sum_{v_j \in \mathcal{V}} (c_{ij} + \tau_i) x_{ijklo} \leq T \qquad v_o \in \mathcal{V}^{\text{DEP}} \;;\; k = 1 \dots m \;;\; l = 1 \dots t \tag{12}$$

$$\sum_{v_i \in S} \sum_{v_j \in S} x_{ijklo} \leq |S| - 1 \quad S \in \mathcal{V}^{\text{CST}} \;;\; |S| \geq 2 \;;\; v_o \in \mathcal{V}^{\text{DEP}} \;;\; k = 1 \dots m \;;\; l = 1 \dots t \tag{13}$$

$$x_{ijklo} \in \{0, 1\} \qquad v_i \in \mathcal{V};\; v_j \in \mathcal{V} \;;\; v_o \in \mathcal{V}^{\text{DEP}} \;;\; k = 1 \dots m \;;\; l = 1 \dots t \tag{14}$$

$$y_{ipo} \in \{0, 1\} \qquad v_i \in \mathcal{V} \;; p \in L_i \;;\; v_o \in \mathcal{V}^{\text{DEP}} \tag{15}$$

Constraints 6 ensure that exactly one depot and one visit combination are assigned to each customer. Constraints 7 guarantee that customer visits occur only on the periods related to the chosen visit combination on a vehicle coming from the assigned depot, while constraints 8 and 9 enforce respectively the single use of vehicles, and compatibility issues between depot assignments and route starting and ending points. Equations 10 are flow conservation constraints and 11 and 12 enforce limits on the capacity of vehicles and the duration of routes. Subtours are eliminated through 13. This MDPVRP model includes both the MDVRP and the PVRP as a special case when $t = 1$, and $d = 1$ respectively.

Cordeau et al. (1997) showed that an MDVRP could be reduced into a PVRP, by associating a different period to each depot, such that each customer $i$ has a frequency $f_i = 1$ and may be served during any period. A particularity of the resulting PVRP is that routing costs $c_{ijl}$ depend upon the period $l$ considered.

In the same spirit, the MDPVRP also happens to have the same general structure as the PVRP. Indeed, a simple change of indexes associating a period to each (depot, period) pair, transforms the MDPVRP formulation into a PVRP. This transformation is summarized in Proposition 2.

**Proposition 2.** *The MDPVRP reduces to a PVRP with period-dependent routing costs.*

**Proof:** Let $\mathcal{I}$ be an MDPVRP instance with $t$ periods, $d$ depot vertices in $\mathcal{V}^{\text{DEP}}$, and $m$ vehicles per depot. Each customer $i \in \mathcal{V}^{\text{CST}}$ has frequency $f_i$ and pattern list $L_i = \{\{p^i_{11}, \dots, p^i_{1f_i}\}, \dots, \{p^i_{|L_i|1}, \dots, p^i_{|L_i|f_i}\}\}$. We now define an equivalent PVRP instance $\mathcal{J}$, which has $t' = td$ periods, a single depot vertex $v'_0 \in \mathcal{V}'_{dep}$, the same set of customers $\mathcal{V}'_{cus} = \mathcal{V}^{\text{CST}}$, and $m$ vehicles available at each period. Each customer $i$ in the new problem still must be served with frequency $f'_i = f_i$, with a pattern list $L'_i$ containing $d \times |L_i|$ patterns defined by Equation 16. Also, travel costs (durations) $c'_{ijl}$ in the new PVRP are period-dependent to take into account that vehicles operating in period $l$ in the new PVRP were based at depot $v_{\lfloor l/d \rfloor}$ in the MDPVRP. The new distance between $v'_0$ and any customer $v_i$ on period $l$ is thus equal to $c_{v_{\lfloor l/d \rfloor} i}$ in the old problem, while all other distances between customer pairs remain constant among periods.

$$L'_i = \bigcup_{\substack{o \in \{1, \dots, d\} \\ r \in \{1, \dots, |L_i|\}}} \{p^i_{r1} + ot, \dots, p^i_{rf_i} + ot\} \tag{16}$$

Solving the PVRP instance $\mathcal{J}$ leads to a solution of the MDPVRP instance $\mathcal{I}$ in a straightforward manner, as the routes for period $l$ and depot $v_o$ in $\mathcal{I}$ correspond to the routes of period $l + ot$ in $\mathcal{J}$. $\square$

## 7.2 The Split algorithm

The *Split* algorithm introduced in Section 4.2 to extract routes from the giant tour reduces the problem of finding the route delimiters to a shortest path problem.

For a given visiting sequence, let $c_i$ be the customer in position $i$. 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 of nodes $i < j$, arc $(i, j)$ represents the trip $r_{i+1,j}$ starting from the depot, visiting customers $c_{i+1}$ to $c_j$, and coming back to the depot. For each trip, travel time and load are given by Equations (17) and (18). If the total load of a trip including arc $(i, j)$ exceeds $Q_{max} = 2Q$, arc $(i, j)$ is excluded from $\mathcal{A}$. This avoids solutions that are too far from feasibility and reduces the number of arcs. The cost of arcs is noted $\phi(r_{i+1,j})$ and accounts for penalized infeasible solutions:

$$q(r_{i+1,j}) = \sum_{l=i+1,j} q_{S_l} \quad (17)$$

$$t(r_{i+1,j}) = c_{0,S_{i+1}} + \sum_{l=i+1,j-1} (s_{S_l} + c_{S_l,S_{l+1}}) + s_{S_j} + c_{S_j,0} \quad (18)$$

An optimal segmentation of the giant tour into routes 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 5 illustrates the Split algorithm on a sequence of 5 customers $c_1$ to $c_5$. The first graph shows the cost of each arc and the demands at nodes (in bold). In this example, the vehicle capacity is set to $Q = 6$, thus $Q_{max} = 12$, the maximum route duration to $D = 150$, all customers $i$ have an identical service time $d_i = 10$, and the penalty parameters are $\omega^{\mathrm{Q}} = 10$ and $\omega^{\mathrm{D}} = 1$. The corresponding graph $\mathcal{H}$ displays on each arc the route cost including penalties. For example, the route servicing customers $c_3$, $c_4$, and $c_5$ has a cost of $165 + 20 + 15$, the penalties of 20 and 15 corresponding to the load excess of two units and duration, respectively. The optimal solution of the minimum-cost path problem is of cost 260, and made up of the three following routes: route 1 visits $c_1$, route 2 visits $c_2$, $c_3$, and $c_4$, and route 3 visits $c_5$.

Note that in the actual implementation of the algorithm, building the graph $\mathcal{H}$ explicitly is not mandatory. A detailed example of pseudo-code for this procedure can be found in (Chu et al., 2006).

## 7.3 Comparison of run times

All CPU times in this paper were scaled into their equivalent Pentium IV 3.0 Ghz run times. This conversion is based on the assumption that CPU time is approximately linearly proportional to the amount of floating point operations per second (flop/s) performed by the processor. Various types of processors have been tested by solving dense

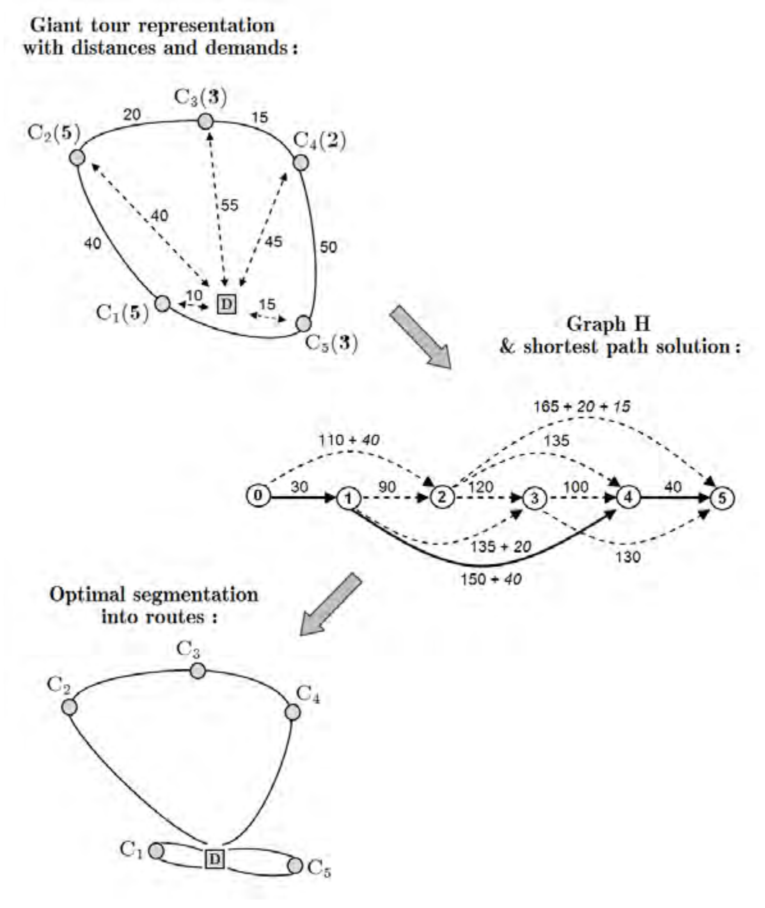

Giant tour representation with distances and demands :
C1(5)
C2(5)
C3(3)
C4(2)
C5(3)
D
Graph H & shortest path solution :
110 + 40
165 + 20 + 15
135
135 + 20
150 + 40
130
Optimal segmentation into routes :

Figure 5: Illustration of a Split graph and shortest-path solution

systems of linear equations, to report their (flop/s) measures in Dongarra (2009). We provide in Table 6 these values for each of the algorithms we compared with, along with the resulting scaling factors used for time conversions. No (flop/s) measure could be found for Pentium IV 2.8 Ghz, Pentium IV 3.2 Ghz, and we made the assumption that the processor speed should be approximately linear with frequency among the processors from the same family.

Table 6: Scaling factors for computation times

| **Authors** | **Processor** | **MFlop/s** | **Factor** |
|---|---|---|---|
| This paper & Nagata and Bräysy (2009) | AMD Opteron 250 2.4 Ghz | 1291 | 0.82 |
| Mester and Braysy (2007) | Pentium IV 2.8 Ghz | — | 0.92 |
| Pisinger and Ropke (2007) | Pentium IV 3.0 Ghz | 1573 | 1.00 |
| Hemmelmayr et al. (2009) | Pentium IV 3.2 Ghz | — | 1.08 |

## 7.4 Detailed results on PVRP, MDVRP and MDPVRP

Tables 7, 8, and 9 present respectively PVRP, MDVRP and MDPVRP results. The first group of columns (1-4) display the instance identifier, number of customers, vehicles, and periods. The next group of columns lists the state-of-the-art methods: the tabu search of Cordeau et al. (1997) (CGL), the scatter search of Alegre et al. (2007) (ALP), and the variable neighborhood search of Hemmelmayr et al. (2009) (HDH) for the PVRP; CGL and the Adaptive Large Neighborhood Search of Pisinger and Ropke (2007) (PR) for the MDVRP (none are available for MDPVRP). This list is followed by the results of the method we propose. We indicate in boldface the best average result among 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 several solutions marked with * by Baldacci and Mingozzi (2009); Baldacci et al. (2010). When upper bounds are improved, the new state-of-the-art solutions are underlined. Finally, the last two lines provide average measures over all instances: the average percentage of error relative to the previous BKS, and computation time for each method.

## 7.5 Experiments on capacitated VRP instances

The VRP appears naturally as a special case of the multi-depot periodic VRPs. The HGSADC method is flexible enough to be used to address this fundamental problem with the following minor changes:

- The distance measure of Section 4.3, originally designed for periodic problems, is replaced by the *broken pairs* distance (Prins, 2009) measuring the proportion of arc similarities between two solutions;
- The PIX crossover is replaced by the simple OX crossover (Prins, 2004) for permutation based solutions. Routes are also concatenated in a cyclic way around the depot to produce the giant tour representation. This should better match related customer visits in the crossover;
- Finally, as the route improvement local search procedure tackles now a large number of customers visits in the same day (up to 483 visits in Golden et al. (1998) instances), the number of neighbor nodes taken into account in moves has been reduced to a small number (20).

All other operators and parameters, including our adaptive diversity management scheme, remain identical.

The algorithm is tested on the traditional benchmarks from the VRP literature. The 14 instances (p01-p14) of Christofides et al. (1979), ranging from 50 to 199 customers, are geographically randomly distributed for the 10 first instances, and otherwise clustered. The 20 large-scale instances (pr01-pr20) of Golden et al. (1998), range from 200 to 483 customers and present geometric symmetries.

Table 10 compares the average results of HGADSC to the actual state-of-the-art algorithms on these benchmarks. The termination criteria $(It_{NI}, T_{max}) = (10^4, \infty)$ is used to perform experiments in a number of iterations and computation time similar to the literature. The first two columns display the instance identifier and the number of customers, while the next group of columns compare the average performance of HGSADC to the performance of the hybrid genetic algorithm of Prins (2004) (P), the guided evolution strategies of Mester and Braysy (2007) (MB), and the edge-assembly crossover based memetic algorithm of Nagata and Bräysy (2009) (NB). On the CVRP, our algorithm differs from (P) only by its population management, infeasible solution use, and the way neighborhoods are restricted in the education operator. These experiments enables thus to state on the benefits of the concepts we introduce. Also, (MB) and (NB) constitute the actual state-of-the-art algorithms for the CVRP. We indicate in boldface the best average result among algorithms for each instance, as well as, in the last two columns, the previous best-known solution (BKS), (found in (MB), (NB) and Zachariadis and Kiranoudis, 2010), and the best solution obtained by HGSADC during all our experiments. When upper bounds are improved, the new state-of-the-art solutions are underlined. The last two lines provide average measures over all instances: the average percentage of error relative to the previous BKS, and computation time for each method.

Table 11 finally reports the results of HGADSC with different termination criteria: $It_{NI} = \{10^4, 2.10^4, 5.10^4\}$. The table format remains the same as previously.

These experiments reveal that HGSADC is competitive with state-of-the-art methods in terms of solution quality and computation time, even though it has no been designed for the CVRP. The results of Table 11 with increased computation times also underline the capability of HGSADC to maintain a very efficient search throughout the run. With a reasonably small increase in the run time (26.37 minutes for $2.10^4$ iterations without without improvement compared to 17.69 minutes for Nagata and Bräysy (2009) state-of-the-art algorithm), HGSADC solution quality already improves upon previously published algorithms.

In addition, 12 new best known solutions are reported on Golden et al. (1998) instances. All these solutions present equal numbers of vehicles and shorter distances than previous BKS from the literature. In the special case of problem pr07, our algorithm also succeeds in finding a solution with one less route (8 routes instead of 9 for previous BKS), and also shorter distance. This solution is reported in details:

**pr07 – Total Distance: 10102.7 – 8 vehicles**

| Route | Distance | Load | Customer Visits |
|---|---|---|---|
| 1 | 1298.05 | 900 | 0 22 58 94 130 166 167 168 204 240 276 312 348 347 311 275 239 203 202 238 274 310 346 345 309 273 237 236 272 308 344 343 307 271 235 199 163 127 91 55 20 0 |
| 2 | 1263.58 | 900 | 0 23 59 95 131 132 133 169 205 241 277 313 349 350 314 278 242 206 207 243 279 315 351 352 316 280 244 208 172 171 170 134 135 136 137 138 102 101 100 99 98 97 96 60 61 25 24 0 |
| 3 | 1179.91 | 900 | 0 26 62 63 64 65 66 67 103 139 175 174 173 209 245 281 317 353 354 318 282 246 210 211 247 283 319 355 356 320 284 248 212 176 177 141 140 104 105 69 68 32 31 30 29 28 27 0 |
| 4 | 1298.33 | 900 | 0 1 36 35 72 71 107 143 179 215 216 252 217 218 254 290 326 325 289 253 288 324 360 359 323 287 251 250 286 322 358 357 321 285 249 213 214 178 142 106 70 34 33 0 |
| 5 | 1267.10 | 900 | 0 6 42 41 40 39 38 74 73 109 110 146 147 148 149 150 186 222 258 294 330 329 293 257 221 185 184 220 256 292 328 327 291 255 219 183 182 181 145 180 144 108 37 2 3 4 5 0 |
| 6 | 1263.58 | 900 | 0 8 7 43 44 80 79 78 77 76 75 111 112 113 114 115 151 187 223 259 295 331 332 296 260 224 188 189 225 261 297 333 334 298 262 226 190 154 153 152 116 117 118 82 81 45 9 0 |
| 7 | 1242.66 | 900 | 0 11 10 46 47 83 119 120 156 155 191 227 263 299 335 336 300 264 228 192 193 229 265 301 337 338 302 266 230 194 158 157 121 122 123 124 125 89 88 87 86 85 84 48 49 13 12 0 |
| 8 | 1289.48 | 900 | 0 21 57 56 92 93 129 128 164 165 201 200 198 234 270 306 342 341 305 269 233 197 196 232 268 304 340 339 303 267 231 195 159 160 161 162 126 90 54 53 52 51 50 14 15 16 17 18 19 0 |

Table 7: 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 | — | 3.34 min | 5.56 min | | | |

** Optimality was proven within 0.02% precision, such that some "optimal" solutions can still be slightly improved

Table 8: 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 | 4372.78 |
| p09 | 249 | 12 | 3 | 3920.85 | 3892.50 | **3868.59** | 9.50 | 3873.64 | 3858.66 |
| p10 | 249 | 8 | 4 | 3714.65 | 3666.85 | **3636.08** | 9.82 | 3650.04 | 3631.11 |
| 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 | 1803.80 |
| pr04 | 192 | 8 | 4 | 2094.24 | 2073.16 | **2059.36** | 5.22 | 2060.93 | 2058.31 |
| pr05 | 240 | 10 | 4 | 2408.10 | 2350.31 | **2340.29** | 9.56 | 2337.84 | 2331.20 |
| pr06 | 288 | 12 | 4 | 2768.13 | 2695.74 | **2681.93** | 10.00 | 2685.35 | 2676.30 |
| 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 | 2133.20 |
| pr10 | 288 | 12 | 6 | 3089.62 | 2905.43 | **2886.59** | 10.00 | 2889.49 | 2868.26 |
| Avg Gap to BKS | | | | +0.96% | +0.34% | -0.01% | | | |
| Avg Time | | | | small | 3.54 min | 4.24 min | | | |

Table 9: 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 10: Results for Christofides et al. (1979) and Golden et al. (1998) CVRP instances

| | | Average | | | | | BKS | |
|---|---|---|---|---|---|---|---|---|
| **Inst** | **n** | **P** | **MB** | **NB** | **HGSADC** | **T(min)** | **prev BKS** | **HGSADC** |
| | | (1 run) | (1 run) | (10 runs) | (10 runs) | | — | (all exp.) |
| p01 | 50 | **524.61** | **524.61** | **524.61** | **524.61** | 0.43 | 524.61 | 524.61 |
| p02 | 75 | **835.26** | **835.26** | 835.61 | **835.26** | 0.96 | 835.26 | 835.26 |
| p03 | 100 | **826.14** | **826.14** | **826.14** | **826.14** | 1.27 | 826.14 | 826.14 |
| p04 | 150 | 1031.63 | **1028.42** | **1028.42** | **1028.42** | 2.87 | 1028.42 | 1028.42 |
| p05 | 199 | 1300.23 | **1291.29** | 1291.84 | 1294.06 | 5.94 | 1291.29 | 1291.45 |
| p06 | 50 | **555.43** | **555.43** | **555.43** | **555.43** | 0.48 | 555.43 | 555.43 |
| p07 | 75 | 912.3 | **909.68** | 910.41 | **909.68** | 1.09 | 909.68 | 909.68 |
| p08 | 100 | **865.94** | **865.94** | **865.94** | **865.94** | 1.14 | 865.94 | 865.94 |
| p09 | 150 | 1164.25 | **1162.55** | 1162.56 | **1162.55** | 2.53 | 1162.55 | 1162.55 |
| p10 | 199 | 1420.2 | 1401.12 | **1398.3** | 1400.23 | 8.22 | 1395.85 | 1395.85 |
| p11 | 120 | **1042.11** | **1042.11** | **1042.11** | **1042.11** | 1.15 | 1042.11 | 1042.11 |
| p12 | 100 | **819.56** | **819.56** | **819.56** | **819.56** | 0.84 | 819.56 | 819.56 |
| p13 | 120 | 1542.97 | **1541.14** | 1542.99 | 1543.07 | 2.83 | 1541.14 | 1541.14 |
| p14 | 100 | **866.37** | **866.37** | **866.37** | **866.37** | 1.19 | 866.37 | 866.37 |
| pr01 | 240 | 5648.04 | 5627.54 | 5632.05 | **5627.00** | 11.68 | 5626.81 | 5623.47 |
| pr02 | 320 | 8459.73 | 8447.92 | **8440.25** | 8446.65 | 20.75 | 8431.66 | 8404.61 |
| pr03 | 400 | **11036.22** | **11036.22** | **11036.22** | **11036.22** | 27.99 | 11036.22 | 11036.22 |
| pr04 | 480 | 13728.80 | 13624.52 | **13618.55** | 13624.52 | 43.67 | 13592.88 | 13624.53 |
| pr05 | 200 | **6460.98** | **6460.98** | **6460.98** | **6460.98** | 2.56 | 6460.98 | 6460.98 |
| pr06 | 280 | 8412.90 | **8412.88** | 8413.41 | 8412.90 | 8.38 | 8404.26 | 8412.90 |
| pr07 | 360 | 10267.50 | 10195.56 | 10186.93 | **10157.63** | 22.94 | 10156.58 | 10102.7 |
| pr08 | 440 | 11865.40 | 11663.55 | 11691.54 | **11646.58** | 40.67 | 11663.55 | 11635.3 |
| pr09 | 255 | 596.89 | 583.39 | **581.46** | 581.79 | 16.22 | 580.02 | 579.71 |
| pr10 | 323 | 751.41 | 741.56 | **739.56** | 739.86 | 25.86 | 738.44 | 736.26 |
| pr11 | 399 | 939.74 | 918.45 | **916.27** | 916.44 | 45.61 | 914.03 | 912.84 |
| pr12 | 483 | 1152.88 | 1107.19 | 1108.21 | **1106.73** | 95.67 | 1104.84 | 1102.69 |
| pr13 | 252 | 877.71 | 859.11 | **858.42** | 859.64 | 9.36 | 857.19 | 857.19 |
| pr14 | 320 | 1089.93 | 1081.31 | **1080.84** | 1082.41 | 14.12 | 1080.55 | 1080.55 |
| pr15 | 396 | 1371.61 | 1345.23 | 1344.32 | **1343.52** | 39.15 | 1340.24 | 1337.92 |
| pr16 | 480 | 1650.94 | 1622.69 | 1622.26 | **1621.02** | 58.27 | 1616.33 | 1612.50 |
| pr17 | 240 | 717.09 | 707.79 | **707.78** | 708.09 | 7.06 | 707.76 | 707.76 |
| pr18 | 300 | 1018.74 | 998.73 | **995.91** | 998.44 | 14.40 | 995.13 | 995.13 |
| pr19 | 360 | 1385.60 | 1366.86 | **1366.70** | 1367.83 | 27.91 | 1365.97 | 1365.60 |
| pr20 | 420 | 1846.55 | **1820.09** | 1821.65 | 1822.02 | 38.23 | 1819.99 | 1818.32 |
| Avg Gap | | +1.00% | +0.13% | +0.10% | +0.11% | | | |
| Avg Time | | — | 14.20 min | 17.64 min | 17.69 min | | | |

Table 11: Behaviour of HGSADC on CVRP instances, when run times increase

| **Inst** | **n** | **$10^4$it** | | **$2.10^4$it** | | **$5.10^4$it** | | | *HGSADC* |
|---|---|---|---|---|---|---|---|---|---|
| | | *Avg sol.* | *T(min)* | *Avg sol.* | *T(min)* | *Avg sol.* | *T(min)* | *prev BKS* | *new BKS* |
| p01 | 50 | **524.61** | 0.43 | **524.61** | 0.96 | **524.61** | 2.96 | 524.61 | 524.61 |
| p02 | 75 | **835.26** | 0.96 | **835.26** | 1.84 | **835.26** | 4.98 | 835.26 | 835.26 |
| p03 | 100 | **826.14** | 1.27 | **826.14** | 2.30 | **826.14** | 6.33 | 826.14 | 826.14 |
| p04 | 150 | **1028.42** | 2.87 | 1028.56 | 4.46 | **1028.42** | 10.65 | 1028.42 | 1028.42 |
| p05 | 199 | 1294.06 | 5.94 | 1294.41 | 7.83 | **1291.74** | 19.52 | 1291.29 | 1291.45 |
| p06 | 50 | **555.43** | 0.48 | **555.43** | 1.07 | **555.43** | 3.28 | 555.43 | 555.43 |
| p07 | 75 | **909.68** | 1.09 | **909.68** | 2.10 | **909.68** | 5.89 | 909.68 | 909.68 |
| p08 | 100 | **865.94** | 1.14 | **865.94** | 2.38 | **865.94** | 6.68 | 865.94 | 865.94 |
| p09 | 150 | **1162.55** | 2.53 | **1162.55** | 4.78 | **1162.55** | 12.00 | 1162.55 | 1162.55 |
| p10 | 199 | 1400.23 | 8.22 | 1398.58 | 18.37 | **1397.70** | 33.09 | 1395.85 | 1395.85 |
| p11 | 120 | **1042.11** | 1.15 | **1042.11** | 2.35 | **1042.11** | 6.65 | 1042.11 | 1042.11 |
| p12 | 100 | **819.56** | 0.84 | **819.56** | 1.73 | **819.56** | 5.06 | 819.56 | 819.56 |
| p13 | 120 | 1543.07 | 2.83 | 1543.04 | 5.37 | **1542.86** | 12.65 | 1541.14 | 1541.14 |
| p14 | 100 | **866.37** | 1.19 | **866.37** | 2.35 | **866.37** | 6.83 | 866.37 | 866.37 |
| pr01 | 240 | 5627.00 | 11.68 | 5625.44 | 24.18 | **5625.10** | 66.90 | 5626.81 | 5623.47 |
| pr02 | 320 | 8446.65 | 20.75 | 8438.79 | 39.52 | **8419.25** | 103.91 | 8431.66 | 8404.61 |
| pr03 | 400 | **11036.22** | 27.99 | **11036.22** | 53.52 | **11036.22** | 116.43 | 11036.22 | 11036.22 |
| pr04 | 480 | **13624.52** | 43.67 | **13624.52** | 50.67 | **13624.52** | 175.22 | 13592.88 | 13624.52 |
| pr05 | 200 | **6460.98** | 2.56 | **6460.98** | 4.96 | **6460.98** | 13.25 | 6460.98 | 6460.98 |
| pr06 | 280 | **8412.90** | 8.38 | **8412.90** | 15.94 | **8412.90** | 41.95 | 8404.26 | 8412.90 |
| pr07 | 360 | 10157.63 | 22.94 | **10132.61** | 43.56 | 10134.90 | 108.49 | 10156.58 | 10102.7 |
| pr08 | 440 | 11646.58 | 40.67 | **11635.30** | 63.65 | **11635.30** | 152.42 | 11663.55 | 11635.3 |
| pr09 | 255 | 581.79 | 16.22 | 581.65 | 30.44 | **581.08** | 51.69 | 580.02 | 579.71 |
| pr10 | 323 | 739.86 | 25.86 | 739.94 | 48.89 | **738.92** | 104.02 | 738.44 | 736.26 |
| pr11 | 399 | 916.44 | 45.61 | 915.54 | 78.51 | **914.37** | 131.23 | 914.03 | 912.84 |
| pr12 | 483 | 1106.73 | 95.67 | 1106.93 | 112.03 | **1105.97** | 196.52 | 1104.84 | 1102.69 |
| pr13 | 252 | 859.64 | 9.36 | 859.24 | 19.24 | **859.08** | 25.56 | 857.19 | 857.19 |
| pr14 | 320 | 1082.41 | 14.12 | 1082.21 | 18.13 | **1081.99** | 43.35 | 1080.55 | 1080.55 |
| pr15 | 396 | 1343.52 | 39.15 | 1343.29 | 43.97 | **1341.95** | 109.14 | 1340.24 | 1337.92 |
| pr16 | 480 | 1621.02 | 58.27 | 1619.49 | 79.24 | **1616.92** | 174.70 | 1616.33 | 1612.50 |
| pr17 | 240 | 708.09 | 7.06 | 707.85 | 10.27 | **707.84** | 28.98 | 707.76 | 707.76 |
| pr18 | 300 | 998.44 | 14.40 | 998.18 | 20.91 | **996.95** | 40.83 | 995.13 | 995.13 |
| pr19 | 360 | 1367.83 | 27.91 | 1366.92 | 35.94 | **1366.39** | 81.49 | 1365.97 | 1365.60 |
| pr20 | 420 | 1822.02 | 38.23 | 1822.09 | 45.03 | **1819.75** | 88.31 | 1819.99 | 1818.32 |
| Avg Gap | | +0.11% | | +0.08% | | +0.03% | | | |
| Avg Time | | 17.69 min | | 26.37 min | | 58.56 min | | | |