CIRRELT

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

Interuniversity Research Centre
on Enterprise Networks, Logistics and Transportation

Implicit Depot Assignments and Rotations in Vehicle Routing Heuristics

Thibaut Vidal
Teodor Gabriel Crainic
Michel Gendreau
Christian Prins

October 2012

CIRRELT-2012-60

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

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

www.cirrelt.ca

Implicit Depot Assignments and Rotations in Vehicle Routing Heuristics

Thibaut Vidal[1,2,*], Teodor Gabriel Crainic[1,3], Michel Gendreau[1,4], Christian Prins[2]

[1] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT)
[2] Laboratoire d'optimisation des systèmes industriels (LOSI), Université de Technologie de Troyes, 12, rue Marie Curie, B.P. 2060, 10010 Troyes, Cedex, France
[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. Vehicle routing variants with multiple depots and mixed fleet present intricate combinatorial aspects related to sequencing choices, vehicle type choices, depot choices, and depots positioning. This paper introduces a dynamic programming methodology for efficiently evaluating compound neighborhoods combining sequence-based moves with an optimal choice of vehicle and depot, and an optimal determination of the first customer to be visited in the route, called rotation. The assignment choices, making the richness of the problem, are thus no more addressed in the solution structure, but implicitly determined during each move evaluation. Two meta-heuristics relying on these concepts, an iterated local search and a hybrid genetic algorithm are presented. Extensive computational experiments demonstrate the remarkable performance of these methods on classic benchmark instances for multi-depot vehicle routing problems with and without fleet mix, as well as the notable contribution of the implicit depot choice and positioning methods to the search performance. The proposed concepts are fairly general, and widely applicable to many other vehicle routing variants.

Keywords: Vehicle routing, dynamic programming, local search, multi-depot, fleet mix.

Acknowledgements. Partial funding for this project has been provided by the Champagne-Ardenne regional council, France, the Natural Sciences and Engineering Council of Canada (NSERC), through its Industrial Research Chair and Discovery Grant programs, and by the Fonds de recherche du Québec - Nature et technologies (FRQNT) through its Team Research Project program. This support is gratefully acknowledged.

---

* Corresponding author: Thibaut.Vidal@cirrelt.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2012

1 Introduction

Vehicle Routing Problems (VRP) with combined assignment choices, such as multi-depot and mixed-fleet settings, appear prominently in many applications related to transportation, production planning, robotics, maintenance, health care or emergency relief. These combinatorial optimization problems require two levels of decisions, related respectively to the sequencing of visits to customers into routes, and the assignment of customers to some global resources, such as depots or vehicle types. Heuristics and meta-heuristics that rely on separate optimization procedures for addressing each aspect, e.g. separate families of local searches, large neighborhoods or crossovers to work on the order of visits, the depot choices or the vehicle types, may overlook a wide range of potential solution refinements involving joint changes in the sequencing and assignment decisions, e.g. swapping two customers and in the meantime changing the vehicle or the depot assigned to the routes. Thus, most advanced meta-heuristics combine these decisions within purposeful optimization procedures to achieve notable performance gains (see Prins 2009b, for example), though the number of combined solution changes tends to become computationally expensive to investigate.

To contribute towards addressing this challenge, this paper proposes a new bidirectional dynamic programming approach to optimally manage the choices of vehicle, depot, and first customer visited in a route, the so-called optimal *rotation*, directly at the level of route evaluations in vehicle routing heuristics. We thus introduce a new Local Search (LS) in which the neighborhoods are solely based on customer-visits relocations and arc exchanges, while dynamic programming-based route evaluation functions produce optimal depot placements, choices and rotations for each alternative route. Since several advanced meta-heuristics for vehicle routing problems requires a Split algorithm to optimally segment a solution represented as a giant tour into several routes, we also derive an advanced *Split* algorithm with compound vehicle assignments, depot choices and rotations. The proposed enhanced procedures work with the same computational complexity as the classic ones from the literature. Thus, the additional capability we introduce does not lead to any additional computational overhead.

As a proof of concept, these methodologies are integrated and tested within two meta-heuristics: a simple multi-start Iterated Local Search (ILS) similar to the one of Prins (2009a) for the capacitated VRP, and a more elaborate Hybrid Genetic Search with Advanced Diversity Control (HGSADC) similar to the one of Vidal et al. (2012a,b, 2013). Two specific problems are investigated, the multi-depot fleet mix problem requiring combined decisions on assignments to vehicles types and depot along with sequencing choices, and the classic multi-depot VRP. Extensive computational experiments, on small- and large-scale benchmark instances with up to 960 customers demonstrate the remarkable performance of the proposed meta-heuristics, as well as the notable contribution of the combined neighborhoods to the search performance.

To facilitate the presentation, we first introduce in Section 2 the problems, notations and variants considered in our experiments. Sections 3 and 4 describe the proposed methodology for optimally managing depot, vehicle choices and rotations within route evaluations, and presents the advanced Split method. The integration of these procedures into a neighborhood-based and a population-based meta-heuristic is discussed in Section 5. The computational experiments are reported in Section 6, and Section 7 concludes.

2 Vehicle routing problems and variants

Vehicle Routing Problems (VRP) aim to design least cost vehicle routes to service geographically dispersed customers (Toth and Vigo 2002, Cordeau et al. 2007, Golden et al. 2008, Laporte 2009, Vidal et al. 2012c). Emphasis is still growing on this family of problems after 50 years of research, mostly because of their major economic impact in many application fields, but also because of the considerable amount of problem variants that must be dealt with to adequately address practical settings. Practical applications, indeed, lead to a variety of problem *attributes* that complement the classic VRP model, and seek to account for customer requirements (e.g., schedules, consistency), network and vehicle characteristics (mixed fleet, multiple depots), and driver's needs (working hour regulations, lunch breaks) among others.

Two attributes especially, *mixed-fleet* and *multi-depot*, are recurrent in many large-scale logistics applications. The problem combining these attributes, known as the Multi-Depot Vehicle Fleet Mix Problem (MDVFMP), can be defined as follows. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $\mathcal{V} = \mathcal{V}^{\text{DEP}} \cup \mathcal{V}^{\text{CST}}$ be a complete undirected graph, in which the d nodes $v_o \in \mathcal{V}^{\text{DEP}}$ represent depots with infinite capacity, and the n nodes $v_i \in \mathcal{V}^{\text{CST}}$ stand for customers. Each customer v_i is characterized by a demand for a non-negative amount of product q_i. The edges $(i,j) \in \mathcal{E}$ represent the possibility of traveling between customers v_i and v_j for a total travel distance of c_{ij}. Finally, w types of vehicles are available in unlimited quantity, any vehicle k being characterized by a base acquisition/depreciation cost e_k, a per-distance-unit cost u_k, and a capacity Q_k. As such, the minimum cost $\Phi(x,q)$ to perform a route with distance x and total demand q is given in Equation (1).

$$\Phi(x,q) = \min_{k \in \{1,\dots,w\}/q \le Q_k} \{e_k + u_k x\} \tag{1}$$

The MDVFMP aims to find a set of routes, as well as their assignment to vehicles and depots, to service each customer once and minimize the total cost. Each route assigned to any vehicle k shall start and end at the same depot location and carry less than q_k of units of products. A mathematical formulation is given in Equations (2-10). In this model, $\mathcal{F}$ represents a fleet containing a large number of vehicles of each type. The binary variable y_{io} represents the depot assignment decision, taking value 1 if and only if customer v_i is assigned to depot v_o. The binary variable x_{ijko} takes value 1 if and only if customer v_j is serviced immediately after v_i by vehicle k from depot o. The objective, presented in Equation (2), includes the base cost of e_k of any used vehicle k, and the distance-based cost $u_k c_{ij}$. Equations (3-5) ensure that exactly one depot is chosen for each customer. Equation (6) ensures the conservation of the flow. Equation (7) enforces the capacity limits for the vehicles, and sub-tours are eliminated by Equation (8).

$$\text{Minimize} \quad \sum_{k \in \mathcal{F}} \sum_{v_o \in \mathcal{V}^{\text{DEP}}} \Big(\sum_{v_i \in \mathcal{V}} e_k x_{oiko} + \sum_{(v_i,v_j) \in \mathcal{E}} u_k c_{ij} x_{ijko} \Big) \tag{2}$$

$$\text{Subject to:} \quad \sum_{v_o \in \mathcal{V}^{\text{DEP}}} y_{io} = 1 \qquad v_i \in \mathcal{V}^{\text{CST}} \tag{3}$$

$$\sum_{v_i \in \mathcal{V}} \sum_{k \in \mathcal{F}} x_{ijko} - y_{io} = 0 \qquad v_j \in \mathcal{V}^{\text{CST}} \; ; \; v_o \in \mathcal{V}^{\text{DEP}} \tag{4}$$

$$\sum_{v_j \in \mathcal{V}} x_{ojko'} = 0 \qquad v_o \in \mathcal{V}^{\text{DEP}} \; ; \; v_{o'} \in \mathcal{V}^{\text{DEP}} \; ; \; v_o \neq v_i \; ; \; k \in \mathcal{F} \tag{5}$$

$$\sum_{v_j \in \mathcal{V}} x_{jiko} - \sum_{v_j \in \mathcal{V}} x_{ijko} = 0 \qquad v_i \in \mathcal{V} \ ; \ v_o \in \mathcal{V}^{\mathrm{DEP}} \ ; \ k \in \mathcal{F} \tag{6}$$

$$\sum_{v_i \in \mathcal{V}} \sum_{v_j \in \mathcal{V}} q_i x_{ijko} \leq Q_k \qquad v_o \in \mathcal{V}^{\mathrm{DEP}} \ ; \ k \in \mathcal{F} \tag{7}$$

$$\sum_{v_i \in S} \sum_{v_j \in S} x_{ijko} \leq |S| - 1 \qquad S \in \mathcal{V}^{\mathrm{CST}} \ ; \ |S| \geq 2 \ ; \ v_o \in \mathcal{V}^{\mathrm{DEP}} \ ; \ k \in \mathcal{F} \tag{8}$$

$$x_{ijko} \in \{0, 1\} \qquad v_i \in \mathcal{V}; \ v_j \in \mathcal{V} \ ; \ v_o \in \mathcal{V}^{\mathrm{DEP}} \ ; \ k \in \mathcal{F} \tag{9}$$

$$y_{iko} \in \{0, 1\} \qquad v_i \in \mathcal{V} \ ; \ v_o \in \mathcal{V}^{\mathrm{DEP}} \ ; \ k \in \mathcal{F} \tag{10}$$

The MDVFMP includes several prominent problems as special cases, such as the Vehicle Fleet Mix Problem (VFMP) when $d = 1$, the Multi-Depot VRP (MDVRP) when $w = 1$, and the Capacitated VRP (CVRP) when $(w, d) = (1, 1)$. The MDVFMP is also NP-hard as a generalization of the CVRP.

Contributions explicitly targeted on the MDVFMP are not frequent in the literature, and we are currently aware of a single meta-heuristic from Salhi and Sari (1997). This neighborhood-based method relies on the same concepts as the Variable Neighborhood Search, and proceeds by changing the neighborhoods by increasing size whenever a local optimum is encountered. Advanced moves that change both the assignment and the sequencing are used once several simpler neighborhoods have been exhausted. The MDVFMP is also a sub-problem of the settings encountered by Irnich (2000), Dondo and Cerdá (2007) and Goel and Gruhn (2008), although a limited fleet has been considered in most cases along with some other problem attributes.

In contrast, the literature dedicated to its two immediate sub-problems, the MDVRP and the VFMP, is much more furnished. An extensive survey of all methods for these two sub-problems is outside the scope of this section, and we refer to Ombuki-Berman and Hanshar (2009), Vidal et al. (2012c) and Subramanian et al. (2012) to that extent. Several meta-heuristics produce solutions of remarkable quality. For the MDVRP, the current state of the art results are produced by the Hybrid Genetic Search with Advanced Diversity Control (HGSADC) of Vidal et al. (2012a), which relies on efficient crossover and LS-improvement procedures to create new individuals. Of particular interest is the individual evaluation used in HGSADC, which relies on both solution quality and contribution to the population diversity. High-quality solutions have also been generated by the Adaptive Large Neighborhood Search (ALNS) of Pisinger and Ropke (2007), the parallel iterated tabu search heuristic of Cordeau and Maischberger (2012) and the hybrid iterated local search and integer programming approach of Subramanian (2012).

For the VFMP, state-of-the-art heuristics are based either on ILS and integer programming (Subramanian et al. 2012, Subramanian 2012), tabu search (Brandão 2009), variable neighborhood search (Imran et al. 2009), or hybrid genetic algorithms (Liu et al. 2009, Prins 2009b). The previously-mentioned meta-heuristics enable to adequately address large scale problem instances, while smaller-size instances may be manageable with exact methods. The integer programming approach of Baldacci and Mingozzi (2009) especially, based on a set partitioning formulation, has demonstrated its ability to solve most MDVRP and VFMP instances with up to 100 customers.

It should finally be noted that most successful hybrid genetic algorithms for the previous problems (Liu et al. 2009, Prins 2009b, Vidal et al. 2012a,b, 2013) rely on a solution representation as a giant tour without trip delimiters that does not consider the route segmentation. Such a representation reduces the search space and allows for the use of simple crossovers based on permutations, but requires a *Split* algorithm to optimally segment the tour into routes whenever a full solution is needed (Prins 2004). This latter splitting problem is generally solved as a shortest path problem on an acyclic auxiliary directed graph (Beasley 1983). For the CVRP, the VFMP and the MDVRP, advanced Split algorithms with supplementary capabilities have been proposed, in order to combine the segmentation choices with decisions on vehicle type-to-customer assignments (Prins 2009b), depot selections (Kansou and Yassine 2010), or potential route inversions or rotations (Prins et al. 2009). These enhanced procedures were shown to contribute significantly to solution quality, however, although implicit depot choice and positioning has been studied in the context of Split algorithms, no such methodology has been to this date proposed in the context of LS. The next section contributes to fill this gap in the state of the art, by introducing a new dynamic programming-based approach to efficiently explore compound LS neighborhoods with sequence changes, optimal vehicle, depot choices and route rotations.

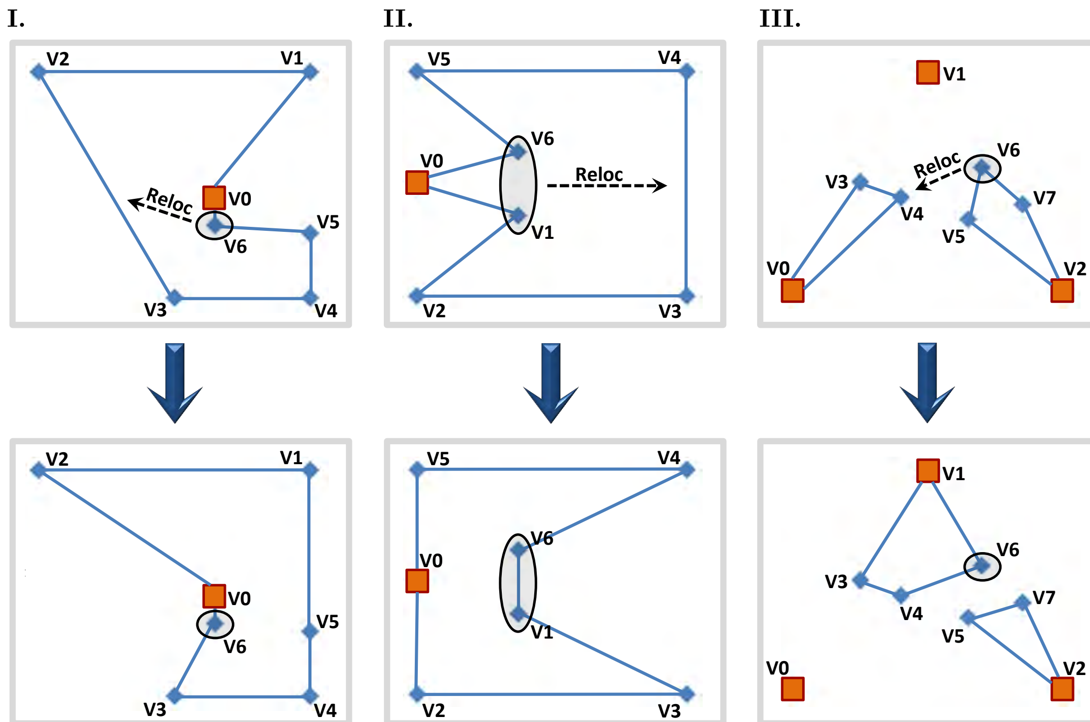
3 Compound neighborhoods with implicit depot positioning and vehicle choices

When dealing with vehicle routing variants with combined assignment choices – for example to depots, days, drivers or vehicle types –, the success of heuristics is often linked to their capacity to tightly integrate decisions together. Any change in the routes, especially, may influence the best way to assign these routes to vehicles, depots, or intervals in drivers schedules, or to rotate the route, such that the exploration of compound neighborhoods is often a key to find new combined choices that could otherwise look unfavorable when examined separately.

Figure 3 displays three euclidean problem instances where these situations arise. Customers are represented with diamonds, while depots are represented by larger squares. For the VRP instance illustrated on the right, the vehicle capacity is $Q = 3$ and any customer v_i for $i \in \{3, \dots, 7\}$ requests one unit of product. All solutions illustrated on the top of this figure are not the global optimum of the problem, but in fact precisely the second best solution (we listed all the solutions when generating the instances). The global optimum is presented below.

It is noteworthy that, even for these small problems with seven nodes, for situations I and II the classical TSP neighborhoods such as 2-opt or Or-opt are insufficient to lead to an optimal solution. The same happens in situation III with the 2-opt*, Relocate, Swap, Cross and I-Cross VRP neighborhoods (see Vidal et al. 2012c for a review of these neighborhoods). Nevertheless, by looking further at the solution changes required to attain the unique global optimum of our instances, we observe that for cases I and II a simple Or-Opt move of customer v_1 or customers (v_1, v_6), combined with an optimal placement of the depot v_0, i.e. rotation of the route, resolves the situation. For the case III, the key is to relocate customer v_6 with (v_3, v_4) and change the depot assigned to the route in a compound way. There is no other manner to attain this solution by local search without accepting deteriorating transitional moves.

Such multi-attribute compound neighborhoods open the way to critical solution refinements, but may be computationally expensive to explore. To give an example, searching any classical 2-opt, 2-opt*, Relocate or Swap neighborhood with combined depot choices, vehicle assignments and rotations would take $O(dwn^3)$ elementary operations when using a straightforward approach that tests all combinations. This complexity is too high to address large problems with meta-heuristics, which usually involve many LS runs.



v_0(13;12)
v_1(20;28) v_2(0;28) v_3(10;0)
v_4(20;0) v_5(20;8) v_6(13;9)

v_0(0;14)
v_1(7.5;10) v_2(0;0) v_3(20;0)
v_4(20;28) v_5(0;28) v_6(7.5;18)

v_0(0;10) v_1(10;30) v_2(20;0)
v_3(5;15) v_4(8;13) v_5(13;10)
v_6(14;17) v_7(17;12)

Figure 1: Solution improvements using compound moves

To address this challenge, we introduce a new efficient search procedure to explore composite VRP neighborhoods based on a bounded number of edge exchanges and node relocations with an optimal choice of vehicle type, depot, and rotation. The proposed approach examines classic VRP neighborhoods of size $O(n^2)$ to produce new alternative sequences of visits, and relies on dynamic programming and incremental route evaluations to optimally determine the other attribute decisions. It exploits the fact that any sequence-based neighborhood involving a bounded number of edge exchanges and vertices relocations, can be assimilated to a recombination of a bounded number of sequences of consecutive visits (Kindervater and Savelsbergh 1997, Vidal et al. 2011). Figure 2 (inspired from Vidal et al. 2011) illustrates this property on a 2-OPT* move, involving two edge exchanges, and on a RELOCATE move involving a vertex relocation.

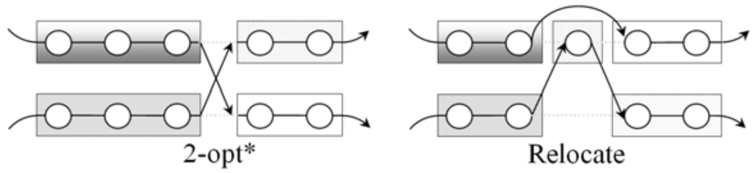


Figure 2: Moves assimilated to recombinations of sequences

The proposed method requires pre-processing dynamic-programming information on the $O(n^2)$ sub-sequences of each incumbent solution, in order to speed up the computations of optimal depot positions for each route produced by the LS. The following values are processed on each subsequence σ of visits to customers : the distance $C(\sigma)$ of the sequence (no visit to the depot is considered), the minimum distance supplement $\hat{C}(\sigma)$ to also visit one depot between the first and the last delivery, and the sum of customers' demands $Q(\sigma)$. By convention, $S(\sigma_0)$ is set to $+\infty$ if the sequence σ_0 is empty or restricted to a single customer. Propositions 1 and 2 enable to exploit this informations to perform efficient move evaluations.

Proposition 1. *Let the* distance supplement $\hat{c}_{ij} = \min_{v_o \in \mathcal{V}^{DEP}}\{c_{io} + c_{oj} - c_{ij}\}$ *be defined as the additional distance required to visit the closest depot between customers v_i and v_j rather than driving directly. The minimum distance to perform the sequence of visits with the best depot choice and route rotation is then given by $Z(\sigma) = \min\{\hat{C}(\sigma) + c_{\sigma_2(|\sigma_2|)\sigma_1(1)}, C(\sigma) + \hat{c}_{\sigma_1(|\sigma_1|)\sigma_2(1)}\}$.*

Proposition 2. *For a sequence $\sigma_0 = (v_i)$ containing a single vertex $\sigma_0 = (v_i)$, $C(\sigma_0) = 0$, $\hat{C}(\sigma_0) = +\infty$, and $Q(\sigma) = q_i$. Furthermore, these values can be derived on larger sequences by induction on the concatenation operation $\oplus$ using Equations (11-13):*

$$C(\sigma_1 \oplus \sigma_2) = C(\sigma_1) + c_{\sigma_1(|\sigma_1|)\sigma_2(1)} + C(\sigma_2) \tag{11}$$

$$\hat{C}(\sigma_1 \oplus \sigma_2) = \min\{\hat{C}(\sigma_1), \hat{c}_{\sigma_1(|\sigma_1|)\sigma_2(1)}, \hat{C}(\sigma_2)\} \tag{12}$$

$$Q(\sigma_1 \oplus \sigma_2) = Q(\sigma_1) + Q(\sigma_2) \tag{13}$$

Equations (11-13) can indeed be used in a pre-processing phase to derive characteristic information on the subsequences of each new incumbent solution, by iteratively appending single customers at the extremities. This pre-processing phase requires $O(n^2)$ calls to the previous operations. Neighbor solutions, assimilated to recombinations of subsequences, are then efficiently evaluated using these equations and the values developed on sequences. Because any local-search move can be assimilated to a bounded number $O(1)$ of concatenations, $O(1)$ calls to these equations lead to the cost for the new routes.

Any use of Equations (11-13) requires $O(d)$ elementary operations in a case with multiple depots, otherwise $O(1)$. The factor d appears within the computation of $\hat{c}_{ij}$ values. Yet, the $\hat{c}_{ij}$ values can be preprocessed at the beginning of the algorithm in $O(dn^2)$, and then be used for all LS runs. Since for the wide majority of neighborhood-based heuristics the number of successive LS is, by far, greater than the number of depots d, the amortized compound move evaluation complexity also drops down from $O(d)$ to $O(1)$ in multi-depot settings.

This methodology thus enables to compute the route distances resulting from any compound sequence-based move with a best choice of rotation and depot in $O(1)$ amortized time. The implicit rotation optimization is applicable to many VRP variants, and the additional implicit choice of depot has to potential to strongly enhance the capacities of meta-heuristics for multi-depot settings. Finally, in presence of a mixed fleet, the optimal vehicle type can also be chosen in a compound way in $O(w)$ time, and even $O(1)$ once some cost tables of pseudo-polynomial size are preprocessed (Prins 2009b).

4 Split algorithm with vehicle choices, depot assignments and rotations

Before going further with meta-heuristics implementations relying on these concepts, we detail another methodological result which follows from the previous property: an advanced Split algorithm which has the ability to optimally segment the giant tour, choose the vehicles and the depots, and rotate the routes.

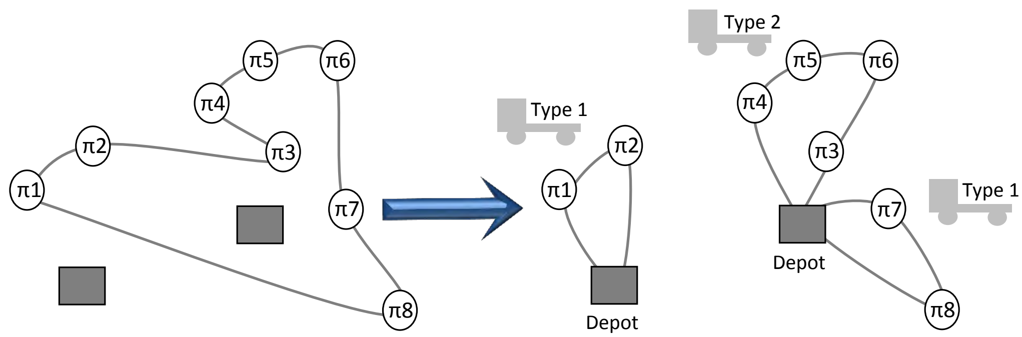


Figure 3: Advanced Split requirements

Consider the example illustrated in Figure 3. Let π_i be the i^{th} customer in the giant tour. The proposed algorithm computes an optimal compound segmentation in three routes corresponding to three sequences of customers (π_1,π_2), (π_3,...,π_6) and (π_7,π_8), with the best depot and vehicle choices. Also, notice that for the sequence (π_3,...,π_6) it is advantageous not to visit the depot between the sequence extremities π_3 and π_6, as it would be in a classic Split procedure, but rather between π_3 and π_4. The compound detection of these rotations is also managed by the method.

As in Beasley (1983) and subsequent works, our advanced Split assimilates the problem of optimally segmenting a giant tour to a shortest path problem on a directed acyclic auxiliary graph $\mathcal{H} = (\mathcal{V}, \mathcal{A})$, where $\mathcal{V}$ contains $n + 1$ nodes indexed from 0 to n. Each arc $(i, j) \in \mathcal{A}$ with $i < j$ represents a potential trip visiting the subsequence of customers π_{i+1} to π_j. The cost h_{ij} of arc (i,j) corresponds to the cost to serve the associated subsequence of customers. The proposed approach accounts for rotations and assignments within this cost definition, and thus h_{ij} stands for the best cost to service the sequence of customers π_{i+1} to π_j, considered as cyclic (π_j is followed by π_{i+1}) while optimally choosing the depot, vehicle type, and rotation. Section 3 gives the methods to compute all these values on subsequences in $O(n^2)$ time.

Once all the costs are determined, applying the Bellman algorithm for undirected acyclic graphs (see Cormen et al. 2001) on $\mathcal{H}$ leads to the best compound segmentation with vehicle types, depot choices and positioning within the routes. The overall splitting algorithm, given in Algorithm 1, works in $O(n^2)$ time. This complexity of $O(n^2)$ is the same as for the basic version of Split (Beasley 1983, Prins 2004) without depot choices and rotations. In this algorithm, Q_{max} denotes the capacity of the largest vehicle type. For the sake of simplicity, we separated the algorithm into a cost computation phase (equivalent to generating the graph $\mathcal{H}$), and a shortest path resolution. These two phases could be done simultaneously to avoid storing the costs, but without impact on the theoretical complexity.

This advanced Split procedure can thus be viewed as a byproduct of advanced subsequence evaluation procedures. The same observation arose in Vidal et al. (2012b), where different operators for sequence evaluations were shown to provide the keys to design splitting algorithms for a wide family of VRP variants. We believe that Split algorithms and local searches with pre-computations on subsequences take their roots on the same dynamic programming concepts, and that as we progress in our understanding of these procedures, these common fundamentals

Algorithm 1 Split with vehicle choice, depot choice and positioning (rotations)

```
//Compute the costs h_ij for 0 ≤ i < j ≤ n in the auxiliary graph H
for i = 1 to n do
   //Single node case
   σ = {π_i}, C(σ) = 0, Ĉ(σ) = +∞, Q(σ) = q_i and h_{i−1,i} = Φ(ĉ_{π_i π_i}, q_i)
   //Append customers to compute costs on larger routes
   for each j = i + 1 to n do
      if Q(σ) + q_{π_j} ≤ Q_max then
         C(σ) = C(σ) + c_{π_{j−1} π_j}
         Ĉ(σ) = min{Ĉ(σ), ĉ_{π_{j−1} π_j}}
         Q(σ) = Q(σ) + q_{π_j}
         h_{i−1,j} = Φ(min{Ĉ(σ) + c_{π_j π_i}, C(σ) + ĉ_{π_j π_i}}, Q(σ))
Solve the shortest path problem on H with costs h_ij, using Bellman algorithm
Return the set of routes associated to the set of arcs of the shortest path
```

become more apparent.

5 Two meta-heuristic applications

The previous move evaluations and Split methodologies have been tested within two types of methods, a multi-start ILS following the guidelines of Prins (2009b), representing perhaps one of the simplest alternative for a neighborhood-based meta-heuristic, and an elaborate population-based meta-heuristic such as HGSADC of Vidal et al. (2012a,b, 2013).

5.1 An iterated local search application

The ILS framework is based on the iterative application of shaking operators and LS-improvement procedures on an incumbent solution to obtain new –hopefully better– solutions. This process is repeated until a maximum number of iterations without improvement $n_{\text{IT-ILS}}$ is attained. In addition, as recommended by Prins (2009a), three enhancements have been added to the method. Firstly, n_{C} solutions are generated at each iteration instead of one, the best solution being kept for the next iteration. This variant of ILS is sometimes called *evolutionary local search* (Wolf and Merz 2007). Secondly, the overall method is started n_{R} times from different initial solutions. Finally, two alternative search spaces are considered, a giant-tour solution representation being used during the shaking phases, while a complete solution representation is used during LS-improvement procedures. We thus rely on the advanced Split procedure of Section 4 to pass from a giant-tour solution representation to a complete solution while efficiently managing assignment choices and rotations. The general structure of the meta-heuristic is presented in Algorithm 2. The algorithm starts from a random solution (random permutation of the giant-tour), and terminates when all n_{R} restarts have been exhausted, or when a maximum time limit T_{MAX} is attained.

In addition, several studies highlighted the benefits of using intermediate penalized infeasible solutions with respect to route constraints (Cordeau et al. 1997, Glover and Hao 2011, Vidal et al. 2012a,b, 2013). Hence, both load and distance constraints are relaxed in our ILS, the cost $\Phi(x, q)$ of a route with distance x and total demand q begin given in Equation (14), where ω^{D}

Algorithm 2 Iterated local search framework

$s_{\text{BEST-EVER}} \leftarrow \emptyset$
for $i_{\text{R}} = 1$ to n_{R} **do**
  $s_{\text{BEST}} \leftarrow \emptyset$; $i_{\text{ILS}} = 0$
  $(s_{\text{FEAS}}, s_{\text{INFEAS}}) \leftarrow$InitializeRandom()
  while $i_{\text{ILS}} < n_{\text{IT-ILS}}$ and Time() $< T_{\text{MAX}}$ **do**
    $S_{\text{CHILDREN}} \leftarrow \emptyset$
    for $s = s_{\text{FEAS}}$ and s_{INFEAS} **do**
      for $i_{\text{C}} = 1$ to n_{C} **do**
        if $i_{\text{ILS}} = k \times n_{\text{R}}$ with $k \in \mathcal{N}^{*+}$ **then** $s_{\text{CURR}} \leftarrow s_{\text{BEST}}$ **else** $s_{\text{CURR}} \leftarrow s$
        $s_{\text{CURR}} \leftarrow$Shaking($s$)
        $s_{\text{CURR}} \leftarrow$AdvancedSplit($s_{\text{CURR}}$)
        $s_{\text{CURR}} \leftarrow$LocalSearch($s_{\text{CURR}}$)
        $S_{\text{CHILDREN}} \leftarrow s_{\text{CURR}}$
        if Infeasible(s_{CURR}) **then** $S_{\text{CHILDREN}} \leftarrow$Repair($s_{\text{CURR}}$)
    $s_{\text{FEAS}} \leftarrow$BestFeasible($S_{\text{CHILDREN}}$)
    $s_{\text{INFEAS}} \leftarrow$BestInFeasible($S_{\text{CHILDREN}}$)
    if Cost(s_{FEAS}) $<$ Cost(s_{BEST}) **then** $s_{\text{BEST}} \leftarrow s_{\text{FEAS}}$; $i_{\text{ILS}} = 0$
    else $i_{\text{ILS}} = i_{\text{ILS}} + 1$
    AdaptPenaltyCoefficients()
if Cost(s_{BEST}) $<$ Cost($s_{\text{BEST-EVER}}$) **then** $s_{\text{BEST-EVER}} \leftarrow s_{\text{BEST}}$
return $s_{\text{BEST-EVER}}$

and ω^{Q} stand for the unit penalties for distance and load excess, respectively. These penalty coefficients are adapted as in Vidal et al. (2012a,b, 2013) relatively to the proportion of feasible solutions generated by the LS.

$$\Phi(x,q) = \min_{k \in \{1,\dots,w\}} \{e_k + u_k x + \omega^{\text{D}} \max\{0, x - D)\} + \omega^{\text{Q}} \max\{0, q - Q_k\}\} \tag{14}$$

The structure of the method has been slightly changed to account for infeasible solutions. First, it is straightforward to extend the Split algorithm to include penalized infeasibility by modifying the subsequence evaluations, and examining subsequences σ with a total demand smaller than two times the maximum capacity ($Q(\sigma) + q_{\pi_j} \leq 2 \times Q_{max}$ – Line 7 of Algorithm 1). In addition, two incumbent solutions are used in the ILS, one feasible, and one infeasible. At each iteration of the ILS, each solution is used to produce n_{C} child solutions. In the event that no feasible solution is currently known, then only the infeasible incumbent solution is used. Furthermore, any infeasible solution undergoes a Repair procedure, where the penalty coefficients are multiplied by a factor of 10, and the LS-improvement procedure is applied to focus the search towards feasible solutions. A last attempt with a factor of 100 is performed if the infeasibility remains. The resulting repaired solution is included in the set of offspring.

The LS-improvement procedure is based on 2-OPT, 2-OPT*, CROSS and I-CROSS neighborhoods restricted of sequences of less than two vertices. The optimal depot choice and position, and vehicle type choice is determined in a compound way using the approach of Section 3. A granularity threshold (Toth and Vigo 2003) is set to limit the neighborhood size, the moves being tested only between a vertex v_i and one of its Γ closest neighbors. Moves are explored in a random order, any improving move being directly applied.

Shaking is done by swapping two couples of random customers within the giant tour. Finally, experimental analyses that adding a probability of *Recall*, that is, jumping back to the current best solution, enhances the search performance by letting the method focus further on elite solution characteristics. We thus included this procedure, and after every n_R successive solution generations without improvement, consider the best solution as starting point for the next shaking and LS.

5.2 A hybrid genetic search application

To further investigate the contribution of the proposed strategies in the context of a population-based method, we derived an extension of the HGSADC framework of Vidal et al. (2012a,b, 2013) using the implicit depot management during LS and Split. The method relies on the following main elements:

- An hybridization between genetic algorithms and efficient local-improvement procedures. The same LS as in Section 5.1 is used.
- A solution representation as a giant tour without trip delimiters. It should be noted that since the depot choice and management is done implicitly by the LS and Split algorithm of Section 4, there is no need to separate the solution representation among different depots.
- A two sub-population scheme to manage feasible and penalized infeasible solutions w.r.t. route constraints.
- A bi-criteria evaluation of individuals, driven by both solution quality and contribution to the population diversity. This evaluation is used for parent selections and survivors selections when the population size becomes too large.

Algorithm 3 HGSADC framework

$S_{\text{POP}} = (S_{\text{FEAS}}, S_{\text{INF}}) \leftarrow$ InitializePopulation()
$s_{\text{BEST}} \leftarrow \emptyset$; $i_{\text{HGS}} = 0$; $i_{\text{HGS-TOT}} = 0$
while $i_{\text{HGS}} < n_{\text{IT-HGS}}$ and Time() $< T_{\text{MAX}}$ **do**
  $s_{\text{P1}} \leftarrow$ BinTournamentSelection(S_{POP})
  $s_{\text{P2}} \leftarrow$ BinTournamentSelection(S_{POP})
  $s_{\text{OFF}} \leftarrow$ Crossover ($s_{\text{P1}}, s_{\text{P2}}$)
  $s_{\text{OFF}} \leftarrow$ AdvancedSplit(s_{OFF})
  $s_{\text{OFF}} \leftarrow$ LocalSearch(s_{OFF})
  $S_{\text{POP}} \leftarrow s_{\text{OFF}}$
  if Infeasible(s_{OFF}) and Alea() $< P_{\text{REP}}$ **then** $s_{\text{OFF}} \leftarrow$ Repair(s_{OFF}) ; $S_{\text{POP}} \leftarrow s_{\text{OFF}}$
  $i_{\text{HGS-TOT}} = i_{\text{HGS-TOT}} + 1$
  if Cost(s_{OFF}) $<$ Cost(s_{BEST}) **then** $s_{\text{BEST}} \leftarrow s_{\text{OFF}}$; $i_{\text{HGS}} = 0$
  else $i_{\text{HGS}} = i_{\text{HGS}} + 1$
  if TooLarge(S_{FEAS}) **then** SelectSurvivors(S_{FEAS})
  if TooLarge(S_{INF}) **then** SelectSurvivors(S_{INF})
  if $i_{\text{HGS}} = k \times It_{div}$ with $k \in \mathcal{N}^*$ **then** Diversification(S_{POP})
  if $i_{\text{HGS-TOT}} = k \times It_{dec}$ with $k \in \mathcal{N}^*$ **then** DecompositionPhase()
  AdaptPenaltyCoefficients()
return s_{BEST}

The structure of the method is described in Algorithm 3. Initial individuals are first randomly generated and inserted in the appropriate sub-population relatively to their feasibility.

HGSADC then iteratively selects two individuals by binary tournament in the merged sub-populations to serve as input the ordered crossover (OX) operator (see Prins 2004), yielding a single offspring s_{OFF}. This offspring undergoes the advanced Split procedure to optimally delimit and rotate its routes, assign vehicles and depots. Then, s_{OFF} undergoes the LS-improvement procedure of Section 5.1 and is inserted in the population. In case of infeasibility, it is *repaired* with probability P_{REP} and added again to the population.

Each sub-population is monitored separately, a *survivor selection* phase being triggered whenever a maximum size is attained. In addition, *diversification* and *decomposition* procedures (Vidal et al. 2013) are regularly used to enhance exploration and intensify the search around elite solution characteristics. The best found solution is returned once $n_{\text{IT-HGS}}$ successive iterations without improvement have been performed or when a time limit T_{MAX} is attained.

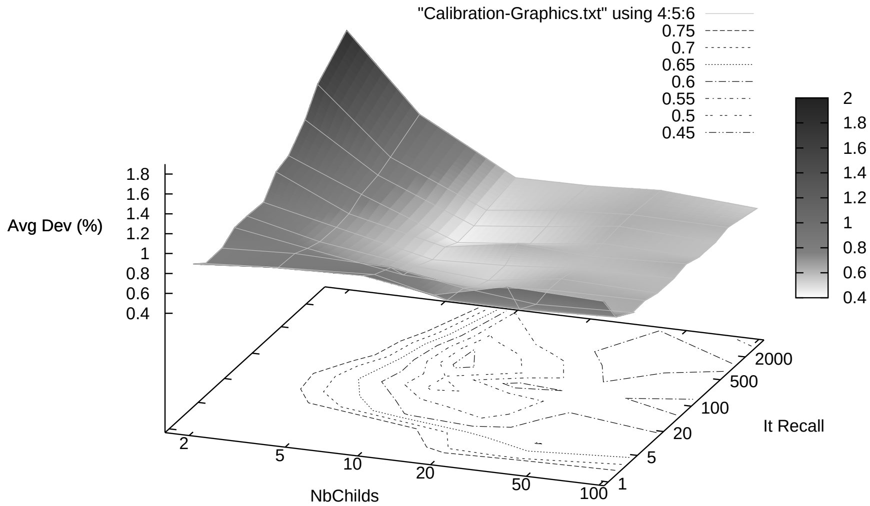
6 Computational experiments

Extensive computational experiments are conducted to analyze the contribution of the compound neighborhoods within the ILS and HGSADC metaheuristics, and compare these methods with state-of-the-art algorithms from the literature. Three main VRP variants are considered: the CVRP and the MDVRP, two seminal problems covered by a huge literature, and the multi-depot vehicle fleet mix problem (MDVFMP), a good example of *rich* problem with compound attributes. The goal, with the ILS experiments, is not to put forward a new champion method, but rather to test the impact of the new compound neighborhoods on a simple LS-based meta-heuristic.

Five sets of benchmark instances are used for the tests. For the CVRP, we rely on the classic benchmark instances of Christofides et al. (1979) (Set A) and Golden et al. (1998) (Set B). Set A includes ten instances with uniformly geographically distributed customers, and four instances with clustered customers. Set B ranges from 200 to 483 customers and displays geometric symmetries.

Three sets of instances (Sets C-E) are used for multi-depot settings. These instances present a mix of uniformly distributed customers and clustered customers. Sets C and D of Cordeau et al. (1997) include 33 MDVRP instances with 50 to 360 customers and two to nine depots. The last Set E, of fourteen large-scale instances, has been introduced in Vidal et al. (2013) for the MDVRP with time windows, and is used for the MDVRP by removing time-window constraints. This set involves 360 to 960 customers and four to twelve depots. MDVFMP instances were also derived, as in Salhi and Sari (1997), by keeping the same customer locations and demands, and generating five types of vehicles v_k such that $Q_k = (0.4 + 0.2k) * \hat{Q}$, $e_k = 70 + 10k$ and $u_k = 0.7 + 0.1k$ for $k \in \{1, \dots, 5\}$, $\hat{Q}$ standing for the vehicle capacity in the original instance. An unlimited fleet is considered in all experiments.

The run time limit for the methods has been set to $T_{max} = 20\ min$ on the medium-scale instance sets (Sets A to D) and $T_{max} = 5\ h$ for the larger instances (Set E). The parameter setting of HGSADC is kept the same as in Vidal et al. (2012a,b, 2013), since an extensive parameter calibration had already been conducted for multi-depot settings. The termination criteria of ILS has been set to $n_{\text{IT-ILS}} = 25000/n_{\mathrm{C}}$, so as to generate 25000 non-improving children before stopping, and the number of restarts is set to $n_{\mathrm{R}} = 5$ to compare with HGSADC and other authors using similar computational effort. The two remaining free parameters, n_{C} and n_{R} were simultaneously calibrated. Several possible values of each parameter were selected, and the method was run 30 times for each configuration on a subset of instances, using different random seeds. For each parameter configuration, the gap to the Best Known Solutions (BKS) in the literature, averaged on all test instances, is reported in Figure 4.



n_C \ n_R	1	2	5	10	20	50	100	200	500	1000
2	0.887	0.778	0.781	0.872	0.874	0.861	1.052	**1.100**	1.309	1.551
4	0.949	0.826	0.815	0.752	0.705	0.698	0.720	**0.801**	0.842	0.908
10	**0.976**	**0.872**	**0.791**	**0.602**	**0.572**	**0.489**	**0.501**	**0.433**	**0.449**	**0.492**
20	0.870	0.702	0.673	0.573	0.521	0.519	0.624	**0.508**	0.528	0.577
40	1.042	0.699	0.598	0.629	0.624	0.578	0.585	**0.599**	0.622	0.624
100	1.011	0.740	0.631	0.639	0.598	0.594	0.626	**0.579**	0.575	0.612
200	0.896	0.632	0.710	0.678	0.568	0.665	0.523	**0.519**	0.747	0.700

Figure 4: ILS performance – Avg Gap to BKS (%) – with different parameter settings

As observed in Figure 4, the combination of parameters $(n_{\mathrm{C}}, n_{\mathrm{R}}) = (10, 200)$ yields the solutions of highest quality. These values were kept for all computational experiments. The algorithm has been coded in C++, compiled with "g++ -O3", and run on an Opteron 250 2.4 GHz CPU for the medium-scale instances (Sets A to D), and an Opteron 275 2.2 GHz CPU for large-scale instances (Set E).

6.1 Impact of compound neighborhoods

The impact of the new compound neighborhoods, within LS and Split, is assessed by means of comparative analyses of several variants of the proposed meta-heuristics, in which the compound optimization of some aspects – rotations, depot choices – has been activated or not. The methods with compound neighborhoods are notated ILS+ and HGSADC+ in the following.

The ILS variant without compound rotation optimization is referred to as *ILS-noR*. Not using both implicit rotations and assignments leads a variant called *ILS-noRD*. In the latter case, customer-to-depots assignments are explicitly managed within the solution representation, using one giant-tour per depot, and applying independently the classic Split procedure on the different giant-tours. The local search of Section 5.1 is used on each separate subset of customers associated to a different depot to perform route improvements (RI), and an additional LS-neighborhood based on customer-to-depot re-assignments is considered during an assignment improvement (AI) phase. These phases are called in the sequence AI-RI-AI-RI.

In the case of HGSADC, deactivating the compound rotations leads to a variant called *HGSADC-noR*. Coming back to an explicit management of the depots means using the original algorithm of Vidal et al. (2012a) on the MDVRP instances with unlimited fleet, notated *HGSADC-noRD*. In this case, the solutions are represented as one giant tour per depot, the PIX crossover of Vidal et al. (2012a) is applied to impact both assignment and sequencing decisions, and the LS-improvement procedure is again separated between route- and assignment-improvement moves.

Tables 1 and 2 report the results of the two meta-heuristics with and without implicit rotations on CVRP instances. The quality of the solutions is compared to those of the best previous methods in the literature: the original HGSADC of Vidal et al. (2012a) (VCGLR12), the HGA with edge-assembly crossover of Nagata and Bräysy (2009) (NB09), and the parallel record-to-record and set covering algorithm of Groër et al. (2011) (GGW11). In these tables, the first three columns display the instances names and characteristics, then the average solution values for the different methods are reported, and finally, the Best Known Solution (BKS) ever found, in previous literature and in our experiments. Best solution values are indicated in boldface for each problem, and new BKS are underlined. The last lines of the tables display the average gap of each method w.r.t. the BKS on each instance set, the average time per run for the methods, and the type of processor used.

In a similar fashion, Tables 3 and 4 report the results of the two meta-heuristics with and without implicit rotations and assignment choices on the MDVRP instances. A comparison of solution quality is done with the best current MDVRP algorithms: the ALNS of Pisinger and Ropke (2007) (PR07), the ILS of Subramanian (2012) (S12) and the original HGSADC of Vidal et al. (2012a) (VCGLR12). Some BKS, indicated with a "$*$" are known to be optimal (Baldacci and Mingozzi 2009). Results are reported for these three methods in presence of a fleet size limit. Still, as indicated by Cordeau et al. (1997), in many cases the maximum fleet size is large and does not impact the optimal solution value. Problems for which the fleet size limit appears to have an incidence are indicated in italics.

These results first highlight the notable contribution of implicit depot assignment, which lead to average MDVRP solutions of better quality for both ILS (0.781% for ILS-noR compared to 2.585% for ILS-noRD) and HGSADC (0.100% compared to 0.252%) with similar run time. This contribution is higher on large-scale problems, for which larger solution improvements were still achievable. The impact of the implicit depot management is very large for the ILS approach, which seems to have difficulties to achieve high-quality solutions on large-scale benchmarks without compound sequencing and assignment moves. For these problems, a more thorough exploration of assignment alternatives appear to be necessary, and the compound moves contribute to fulfill this goal.

The use of implicit rotations enhances the solution quality for both ILS and HGSADC, on both CVRP and MDVRP experiments. The largest improvement is again observed on the MDVRP with ILS, with an average gap decrease from 0.781% to 0.515%. No impact on the computational time is observed on MDVRP experiments for both ILS and HGSADC, and thus the effort spent in evaluating compound moves is paid off in terms of convergence speed. For the CVRP, the computational time increases by a factor of two. However, the current HGSADC+ implementation was derived from an existing code, and "skips" depots when evaluating routes using IF instructions. A complete new implementation without any mention of depots may run much faster. Implicit rotation management also lead to LS simplifications, by enabling to

Table 1: Impact of implicit rotations within ILS – CVRP instances

Inst	n	**VCGLR12**	**NB09**	**GGW11**	**ILS-noR**			**ILS+**			**BKS**
		Avg 10	Avg 10	Best 5	Avg 10	Best 10	T(min)	Avg 10	Best 10	T(min)	
A-p01	50	**524.61**	**524.61**	**524.61**	**524.61**	**524.61**	0.89	**524.61**	**524.61**	1.90	524.61
A-p02	75	**835.26**	835.61	**835.26**	836.16	835.32	2.25	835.67	**835.26**	4.56	835.26
A-p03	100	**826.14**	**826.14**	**826.14**	826.78	**826.14**	4.27	826.26	**826.14**	9.30	826.14
A-p04	150	**1028.42**	**1028.42**	**1028.42**	1034.18	1031.96	8.11	1033.39	1031.29	16.87	1028.42
A-p05	199	1294.06	1291.84	1291.50	1315.50	1306.95	11.55	1313.98	1303.45	24.89	1291.29
A-p06	50	**555.43**	**555.43**	**555.43**	**555.43**	**555.43**	1.16	**555.43**	**555.43**	1.91	555.43
A-p07	75	**909.68**	910.41	**909.68**	**909.68**	**909.68**	3.18	**909.68**	**909.68**	3.47	909.68
A-p08	100	**865.94**	**865.94**	**865.94**	**865.94**	**865.94**	3.76	**865.94**	**865.94**	6.29	865.94
A-p09	150	**1162.55**	1162.56	**1162.55**	1166.28	1165.44	9.82	1166.70	1164.11	15.60	1162.55
A-p10	199	1400.23	1398.30	1399.91	1418.05	1416.58	14.60	1416.41	1412.91	24.79	1395.85
A-p11	120	**1042.11**	**1042.11**	**1042.11**	**1042.11**	**1042.11**	5.15	**1042.11**	**1042.11**	11.62	1042.11
A-p12	100	**819.56**	**819.56**	**819.56**	**819.56**	**819.56**	2.19	**819.56**	**819.56**	4.52	819.56
A-p13	120	1543.07	1542.99	1542.36	1549.56	1545.96	7.24	1549.90	1547.81	12.19	1541.14
A-p14	100	**866.37**	**866.37**	**866.37**	**866.37**	**866.37**	2.86	**866.37**	**866.37**	4.84	866.37
B-pr01	240	5627.00	5632.05	5636.96	5647.13	5644.44	22.89	5648.59	5644.42	46.95	5623.47
B-pr02	320	8446.65	8440.25	8447.92	8458.11	8452.72	39.27	8460.78	8451.05	85.09	8404.61
B-pr03	400	**11036.22**	**11036.22**	**11036.22**	11056.98	11041.02	59.85	11056.85	11043.41	120.78	11036.22
B-pr04	480	13624.53	13618.55	13624.52	13646.49	13632.39	85.03	13660.90	13642.33	165.73	13592.88
B-pr05	200	**6460.98**	**6460.98**	**6460.98**	**6460.98**	**6460.98**	13.19	**6460.98**	**6460.98**	28.19	6460.98
B-pr06	280	8412.90	8413.41	8412.90	8413.46	8413.36	24.33	8413.78	8413.36	58.46	8400.33
B-pr07	360	10157.63	10186.93	10195.59	10201.16	10195.59	44.44	10204.74	10195.59	103.81	10102.68
B-pr08	440	11646.58	11691.54	11691.76	11814.54	11734.96	66.48	11847.72	11743.37	142.68	11635.34
B-pr09	255	581.79	581.46	581.92	594.23	593.01	19.61	594.81	592.63	36.12	579.71
B-pr10	323	739.86	739.56	739.82	758.47	757.27	25.85	757.93	756.25	77.97	736.26
B-pr11	399	916.44	916.27	916.14	940.97	936.87	39.64	941.03	936.98	105.76	912.84
B-pr12	483	1106.73	1108.21	1112.73	1141.83	1139.49	62.55	1140.38	1136.28	144.35	1102.69
B-pr13	252	859.64	858.42	858.45	880.25	878.14	16.77	877.55	875.82	32.21	857.19
B-pr14	320	1082.41	1080.84	**1080.55**	1112.22	1110.56	23.42	1106.18	1102.59	55.75	1080.55
B-pr15	396	1343.52	1344.32	1341.41	1380.38	1377.45	40.30	1375.84	1374.52	78.75	1337.92
B-pr16	480	1621.02	1622.26	1619.45	1672.93	1668.66	58.34	1666.15	1663.46	114.03	1612.50
B-pr17	240	708.09	707.78	707.79	713.98	712.70	16.49	713.45	712.24	32.10	707.76
B-pr18	300	998.44	995.91	997.25	1019.51	1016.16	24.17	1017.44	1012.10	46.92	995.13
B-pr19	360	1367.83	1366.70	1366.26	1395.49	1392.94	36.62	1394.29	1390.17	81.92	1365.60
B-pr20	420	1822.02	1821.65	1820.88	1871.64	1863.90	56.16	1872.68	1869.40	106.13	1818.32
Gap Set-A		0.047%	0.033%	0.028%	0.363%	0.258%		0.336%	0.215%		
Gap Set-B		0.267%	0.273%	0.296%	1.873%	1.665%		1.795%	1.552%		
Gap All		0.176%	0.174%	0.186%	1.251%	1.086%		1.194%	1.002%		
T(min)		21.57	21.51	8×3.92			25.07			53.13	
CPU		Opt 2.4G	Opt 2.4G	Xeon 2.3G	Opt 2.4G			Opt 2.4G			

Table 2: Impact of implicit rotations within HGSADC – CVRP instances

Inst	n	**VCGLR12**	**NB09**	**GGW11**	**HGSADC-noR**			**HGSADC+**			**BKS**
		Avg 10	Avg 10	Best 5	Avg 10	Best 10	T(min)	Avg 10	Best 10	T(min)	
A-p01	50	**524.61**	**524.61**	**524.61**	**524.61**	**524.61**	1.13	**524.61**	**524.61**	2.33	524.61
A-p02	75	**835.26**	835.61	**835.26**	**835.26**	**835.26**	1.83	**835.26**	**835.26**	3.32	835.26
A-p03	100	**826.14**	**826.14**	**826.14**	**826.14**	**826.14**	3.23	**826.14**	**826.14**	7.02	826.14
A-p04	150	**1028.42**	**1028.42**	**1028.42**	1028.56	**1028.42**	5.91	**1028.42**	**1028.42**	12.40	1028.42
A-p05	199	1294.06	1291.84	1291.50	1293.88	1291.45	10.38	1292.17	1291.45	22.11	1291.29
A-p06	50	**555.43**	**555.43**	**555.43**	**555.43**	**555.43**	1.29	**555.43**	**555.43**	2.26	555.43
A-p07	75	**909.68**	910.41	**909.68**	**909.68**	**909.68**	2.29	**909.68**	**909.68**	3.41	909.68
A-p08	100	**865.94**	**865.94**	**865.94**	**865.94**	**865.94**	3.75	**865.94**	**865.94**	6.67	865.94
A-p09	150	**1162.55**	1162.56	**1162.55**	**1162.55**	**1162.55**	6.60	**1162.55**	**1162.55**	11.46	1162.55
A-p10	199	1400.23	1398.30	1399.91	1399.62	1396.54	17.17	1398.12	**1395.85**	28.33	1395.85
A-p11	120	**1042.11**	**1042.11**	**1042.11**	**1042.11**	**1042.11**	4.29	**1042.11**	**1042.11**	10.34	1042.11
A-p12	100	**819.56**	**819.56**	**819.56**	**819.56**	**819.56**	2.41	**819.56**	**819.56**	5.20	819.56
A-p13	120	1543.07	1542.99	1542.36	1542.86	1542.86	5.79	1542.52	**1541.14**	10.17	1541.14
A-p14	100	**866.37**	**866.37**	**866.37**	**866.37**	**866.37**	3.41	**866.37**	**866.37**	5.83	866.37
B-pr01	240	5627.00	5632.05	5636.96	5625.75	**5623.47**	25.65	5625.22	**5623.47**	62.00	5623.47
B-pr02	320	8446.65	8440.25	8447.92	8447.92	8447.92	39.04	8444.29	8413.82	103.07	8404.61
B-pr03	400	**11036.22**	**11036.22**	**11036.22**	11051.23	**11036.22**	64.80	**11036.22**	**11036.22**	151.02	11036.22
B-pr04	480	13624.53	13618.55	13624.52	13645.38	13624.53	93.06	13645.38	13624.53	174.19	13592.88
B-pr05	200	**6460.98**	**6460.98**	**6460.98**	**6460.98**	**6460.98**	16.39	**6460.98**	**6460.98**	34.00	6460.98
B-pr06	280	8412.90	8413.41	8412.90	8412.90	8412.90	29.32	8412.90	8412.90	62.69	8400.33
B-pr07	360	10157.63	10186.93	10195.59	10182.45	10115.58	54.07	10168.95	10141.06	130.77	10102.68
B-pr08	440	11646.58	11691.54	11691.76	11649.45	**11635.34**	78.31	11640.99	**11635.34**	161.10	11635.34
B-pr09	255	581.79	581.46	581.92	581.86	**579.71**	28.70	581.93	580.50	91.74	579.71
B-pr10	323	739.86	739.56	739.82	739.01	737.43	81.40	739.78	737.65	151.96	736.26
B-pr11	399	916.44	916.27	916.14	915.41	913.69	111.23	915.87	913.15	230.69	912.84
B-pr12	483	1106.73	1108.21	1112.73	1107.36	1104.96	177.02	1106.99	1105.24	267.33	1102.69
B-pr13	252	859.64	858.42	858.45	859.74	**857.19**	20.11	859.99	858.39	52.81	857.19
B-pr14	320	1082.41	1080.84	**1080.55**	1082.33	**1080.55**	29.25	1081.68	**1080.55**	55.16	1080.55
B-pr15	396	1343.52	1344.32	1341.41	1343.87	1340.62	60.73	1341.68	1339.75	159.65	1337.92
B-pr16	480	1621.02	1622.26	1619.45	1619.53	1616.09	98.69	1618.65	1616.43	226.49	1612.50
B-pr17	240	708.09	707.78	707.79	707.95	707.79	15.98	707.93	707.79	35.04	707.76
B-pr18	300	998.44	995.91	997.25	997.26	**995.13**	35.10	998.32	997.25	66.46	995.13
B-pr19	360	1367.83	1366.70	1366.26	1367.20	1366.48	52.51	1366.95	1366.40	100.58	1365.60
B-pr20	420	1822.02	1821.65	1820.88	1821.09	1819.59	87.84	1821.88	1820.45	173.46	1818.32
Gap Set-A		0.047%	0.033%	0.028%	0.043%	0.012%		0.023%	0.001%		
Gap Set-B		0.267%	0.273%	0.296%	0.272%	0.102%		0.253%	0.119%		
Gap All		0.176%	0.174%	0.186%	0.178%	0.065%		0.158%	0.070%		
T(min)		21.57	21.51	8×3.92			37.31			77.09	
CPU		Opt 2.4G	Opt 2.4G	Xeon 2.3G	Opt 2.4G			Opt 2.4G			

Table 3: Impact of implicit rotations and depot choices within ILS – MDVRP instances.

Inst	n	d	**PR07**	**VCGLR12**	**S12**	**ILS-noRD**			**ILS-noR**			**ILS+**			**BKS**
			Avg 10	Avg 10	Avg 10	Avg 10	Best 10	T(min)	Avg 10	Best 10	T(min)	Avg 10	Best 10	T(min)	
C-p01	50	4	**576.87**	**576.87**	**576.87**	**576.87**	**576.87**	0.53	**576.87**	**576.87**	0.50	**576.87**	**576.87**	0.49	576.87*
C-p02	50	4	**473.53**	**473.53**	**473.53**	**473.53**	**473.53**	0.81	473.67	**473.53**	1.21	473.63	**473.53**	1.41	473.53*
C-p03	75	2	*641.19*	*641.19*	*641.19*	641.19	641.19	0.96	641.08	**640.65**	0.99	641.08	**640.65**	1.00	640.65*
C-p04	100	2	*1006.09*	*1001.24*	*1001.04*	1001.84	**999.21**	2.57	1002.12	**999.21**	1.88	1000.01	**999.21**	1.81	999.21*
C-p05	100	2	752.34	**750.03**	750.21	**750.03**	**750.03**	2.09	750.92	**750.03**	2.84	750.72	**750.03**	2.81	750.03
C-p06	100	3	883.01	**876.50**	**876.50**	878.65	**876.50**	2.06	879.82	**876.50**	1.63	877.43	**876.50**	1.60	876.50*
C-p07	100	4	889.36	882.24	**881.97**	886.85	884.66	2.29	886.72	**881.97**	1.72	885.71	**881.97**	1.71	881.97*
C-p08	249	2	4421.03	4397.71	4393.70	4426.63	4401.79	19.65	4424.72	4401.17	15.54	4392.96	4382.91	14.85	4372.78
C-p09	249	3	3892.50	3863.45	3864.22	3903.12	3891.91	19.30	3899.90	3882.66	14.69	3889.82	3878.25	12.75	3858.66
C-p10	249	4	3666.85	3634.03	3634.72	3679.20	3660.30	19.92	3657.40	3635.52	14.43	3651.73	3635.71	12.45	3631.11
C-p11	249	5	3573.23	3546.95	3546.15	3588.35	3560.27	19.52	3586.44	3565.95	14.73	3574.43	3557.57	12.94	3546.06
C-p12	80	2	1319.13	**1318.95**	**1318.95**	**1318.95**	**1318.95**	1.09	**1318.95**	**1318.95**	1.35	**1318.95**	**1318.95**	1.51	1318.95*
C-p13	80	2	**1318.95**	**1318.95**	**1318.95**	**1318.95**	**1318.95**	0.98	**1318.95**	**1318.95**	1.24	**1318.95**	**1318.95**	1.23	1318.95
C-p14	80	2	**1360.12**	**1360.12**	**1360.12**	**1360.12**	**1360.12**	0.98	**1360.12**	**1360.12**	1.34	**1360.12**	**1360.12**	1.17	1360.12
C-p15	160	4	2519.64	**2505.42**	**2505.42**	**2505.42**	**2505.42**	4.18	2507.46	**2505.42**	3.82	**2505.42**	**2505.42**	3.25	2505.42
C-p16	160	4	2573.95	**2572.23**	**2572.23**	**2572.23**	**2572.23**	3.32	**2572.23**	**2572.23**	3.15	**2572.23**	**2572.23**	3.17	2572.23
C-p17	160	4	**2709.09**	**2709.09**	2710.21	2716.89	**2709.09**	3.43	2711.32	**2709.09**	3.48	2710.21	**2709.09**	3.03	2709.09
C-p18	240	6	3736.53	**3702.85**	**3702.85**	3713.02	**3702.85**	13.98	3731.55	3714.56	9.82	3711.32	**3702.85**	10.32	3702.85
C-p19	240	6	3838.76	**3827.06**	3827.55	3828.29	**3827.06**	7.95	3834.44	**3827.06**	7.06	3831.98	**3827.06**	7.32	3827.06
C-p20	240	6	4064.76	**4058.07**	**4058.07**	4064.76	**4058.07**	9.26	4088.92	**4058.07**	8.70	4069.80	**4058.07**	8.02	4058.07
C-p21	360	9	5501.58	**5474.84**	**5474.84**	5525.93	5490.11	20.00	5540.82	5506.26	32.59	5530.58	5496.40	20.00	5474.84
C-p22	360	9	5722.19	**5702.16**	5705.84	5719.73	**5702.16**	19.66	5724.26	5714.46	19.99	5717.64	**5702.16**	18.63	5702.16
C-p23	360	9	6092.66	**6078.75**	**6078.75**	6118.29	**6078.75**	20.00	6129.08	6112.46	30.04	6124.99	6112.46	20.00	6078.75
D-pr01	48	4	**861.32**	**861.32**	**861.32**	**861.32**	**861.32**	1.02	**861.32**	**861.32**	1.18	**861.32**	**861.32**	1.12	861.32
D-pr02	96	4	*1308.17*	*1307.34*	*1308.53*	1299.08	1297.44	3.96	**1296.25**	**1296.25**	3.03	**1296.25**	**1296.25**	2.82	1296.25
D-pr03	144	4	1810.66	1803.80	1804.09	1804.55	**1803.80**	6.61	1803.81	**1803.80**	5.79	1803.81	**1803.80**	5.66	1803.80
D-pr04	192	4	*2073.16*	*2058.31*	*2060.93*	2054.15	**2042.45**	11.41	2047.99	**2042.45**	8.13	2044.26	**2042.45**	8.09	2042.45
D-pr05	240	4	*2350.31*	*2335.81*	*2338.12*	2377.66	2342.77	20.00	2333.43	2326.50	12.77	2330.19	2326.35	11.41	2324.12
D-pr06	288	4	*2695.74*	*2680.95*	*2685.23*	2684.11	2675.71	20.00	2679.96	2673.00	18.72	2670.77	2668.76	18.06	2663.56
D-pr07	72	6	*1089.56*	*1089.56*	*1089.56*	1075.53	**1075.12**	1.85	**1075.12**	**1075.12**	1.64	**1075.12**	**1075.12**	1.55	1075.12
D-pr08	144	6	*1675.74*	*1664.99*	*1665.08*	1667.48	1660.16	6.44	1660.21	1658.71	4.32	1658.68	**1658.23**	4.30	1658.23
D-pr09	216	6	*2144.84*	*2133.52*	*2135.37*	2159.27	2147.68	18.88	2144.57	2136.67	8.88	2139.08	**2131.70**	8.85	2131.70
D-pr10	288	6	*2905.43*	*2885.39*	*2882.41*	2861.08	2826.91	20.00	2819.33	2812.39	20.38	2815.00	2810.25	17.96	2805.53
E-pr11	360	4	–	–	–	5144.80	5096.71	164.31	5071.03	5029.61	70.36	5026.89	4999.04	72.89	4994.67
E-pr12	480	4	–	–	–	6593.86	6522.94	281.29	6446.22	6424.10	149.53	6418.54	6391.48	168.58	6367.67
E-pr13	600	4	–	–	–	8100.34	8054.60	300.02	7791.39	7746.68	261.83	7743.51	7689.19	293.95	7645.29
E-pr14	720	4	–	–	–	9681.79	9588.94	300.03	9279.51	9211.41	300.02	9239.96	9184.68	300.06	9101.67
E-pr15	840	4	–	–	–	11390.61	11235.84	300.03	10817.77	10782.85	300.02	10762.94	10726.17	300.03	10598.70
E-pr16	960	4	–	–	–	13025.41	12720.01	300.04	12196.66	12143.63	300.03	12128.21	12062.69	300.02	11919.71
E-pr17	360	6	–	–	–	4901.09	4880.87	172.32	4817.03	4797.91	68.73	4792.53	4772.62	80.47	4761.70
E-pr18	520	6	–	–	–	6869.04	6770.66	300.02	6585.97	6536.84	186.31	6562.69	6531.79	176.90	6504.36
E-pr19	700	6	–	–	–	9362.54	9170.60	300.03	8804.79	8770.25	300.02	8775.67	8736.22	300.01	8639.44
E-pr20	880	6	–	–	–	10730.24	10553.01	300.03	10040.07	10000.46	300.02	9980.76	9946.18	300.03	9825.50
E-pr21	420	12	–	–	–	4839.75	4712.22	295.36	4632.10	4613.14	99.47	4605.51	4596.09	109.73	4582.62
E-pr22	600	12	–	–	–	6651.39	6466.61	300.03	6230.97	6193.52	283.34	6207.59	6181.67	264.14	6141.63
E-pr23	780	12	–	–	–	9041.86	8778.31	300.04	8176.64	8119.01	300.01	8128.24	8078.99	300.02	8014.10
E-pr24	960	12	–	–	–	11547.32	11020.10	300.06	10134.47	10067.97	300.03	10054.52	10040.63	300.04	9909.49
Gap/T Set-C			0.413%	0.049%	0.049%	0.378%	0.148%	8.46	0.429%	0.157%	8.38	0.268%	0.093%	7.02	
Gap/T Set-D			1.171%	0.747%	0.795%	0.778%	0.298%	11.02	0.250%	0.096%	8.48	0.133%	0.046%	7.98	
Gap/T Set-E			–	–	–	7.501%	5.495%	279.54	1.740%	1.193%	229.98	1.194%	0.745%	233.35	
Gap/T All			–	–	–	2.585%	1.772%	89.75	0.781%	0.453%	74.41	0.515%	0.277%	74.64	
T(min) Set-CD			3.95	5.18	10.45	–	–	9.23	–	–	8.41	–	–	7.31	
CPU			P-IV 3G	Opt 2.4G	I7 2.9G	Opt 2.2 & Opt 2.4G			Opt 2.2 & Opt 2.4G			Opt 2.2 & Opt 2.4G			

Table 4: Impact of implicit rotations and depot choices within HGSADC – MDVRP instances.

Inst	n	d	**PR07**	**VCGLR12**	**S12**	**HGSADC-noRD**			**HGSADC-noR**			**HGSADC+**			**BKS**
			Avg 10	Avg 10	Avg 10	Avg 10	Best 10	T(min)	Avg 10	Best 10	T(min)	Avg 10	Best 10	T(min)	
C-p01	50	4	**576.87**	**576.87**	**576.87**	**576.87**	**576.87**	0.65	**576.87**	**576.87**	1.10	**576.87**	**576.87**	1.09	576.87*
C-p02	50	4	**473.53**	**473.53**	**473.53**	**473.53**	**473.53**	0.90	**473.53**	**473.53**	1.77	**473.53**	**473.53**	1.69	473.53*
C-p03	75	2	*641.19*	*641.19*	*641.19*	640.92	**640.65**	1.47	**640.65**	**640.65**	2.56	**640.65**	**640.65**	2.53	640.65*
C-p04	100	2	*1006.09*	*1001.24*	*1001.04*	1000.18	**999.21**	3.48	1001.04	1001.04	3.59	1000.66	**999.21**	3.43	999.21*
C-p05	100	2	752.34	**750.03**	750.21	**750.03**	**750.03**	2.84	**750.03**	**750.03**	5.12	**750.03**	**750.03**	4.89	750.03
C-p06	100	3	883.01	**876.50**	**876.50**	**876.50**	**876.50**	1.99	**876.50**	**876.50**	3.31	**876.50**	**876.50**	3.02	876.50*
C-p07	100	4	889.36	882.24	**881.97**	883.83	**881.97**	3.02	**881.97**	**881.97**	3.75	**881.97**	**881.97**	3.75	881.97*
C-p08	249	2	4421.03	4397.71	4393.70	4386.59	4384.15	17.83	4382.51	4375.49	26.36	4383.63	4375.49	19.93	4372.78
C-p09	249	3	3892.50	3863.45	3864.22	3864.85	3859.17	16.48	3865.11	3859.76	27.73	3860.77	3859.17	19.51	3858.66
C-p10	249	4	3666.85	3634.03	3634.72	3639.61	**3631.11**	18.43	3631.71	**3631.11**	23.54	3631.71	**3631.11**	17.71	3631.11
C-p11	249	5	3573.23	3546.95	3546.15	3553.28	**3546.06**	13.13	3547.47	**3546.06**	17.31	3547.37	**3546.06**	17.14	3546.06
C-p12	80	2	1319.13	**1318.95**	**1318.95**	**1318.95**	**1318.95**	1.57	**1318.95**	**1318.95**	2.77	**1318.95**	**1318.95**	2.85	1318.95*
C-p13	80	2	**1318.95**	**1318.95**	**1318.95**	**1318.95**	**1318.95**	1.91	**1318.95**	**1318.95**	2.94	**1318.95**	**1318.95**	2.85	1318.95
C-p14	80	2	**1360.12**	**1360.12**	**1360.12**	**1360.12**	**1360.12**	1.88	**1360.12**	**1360.12**	2.38	**1360.12**	**1360.12**	2.53	1360.12
C-p15	160	4	2519.64	**2505.42**	**2505.42**	**2505.42**	**2505.42**	4.42	**2505.42**	**2505.42**	7.66	**2505.42**	**2505.42**	7.79	2505.42
C-p16	160	4	2573.95	**2572.23**	**2572.23**	**2572.23**	**2572.23**	5.28	**2572.23**	**2572.23**	7.71	**2572.23**	**2572.23**	7.75	2572.23
C-p17	160	4	**2709.09**	**2709.09**	2710.21	**2709.09**	**2709.09**	5.27	**2709.09**	**2709.09**	8.29	**2709.09**	**2709.09**	8.31	2709.09
C-p18	240	6	3736.53	**3702.85**	**3702.85**	**3702.85**	**3702.85**	8.81	**3702.85**	**3702.85**	13.48	**3702.85**	**3702.85**	14.01	3702.85
C-p19	240	6	3838.76	**3827.06**	3827.55	**3827.06**	**3827.06**	9.90	**3827.06**	**3827.06**	15.34	**3827.06**	**3827.06**	15.11	3827.06
C-p20	240	6	4064.76	**4058.07**	**4058.07**	**4058.07**	**4058.07**	9.55	**4058.07**	**4058.07**	16.55	**4058.07**	**4058.07**	16.20	4058.07
C-p21	360	9	5501.58	**5474.84**	**5474.84**	5476.36	**5474.84**	19.83	**5474.84**	**5474.84**	29.18	**5474.84**	**5474.84**	20.07	5474.84
C-p22	360	9	5722.19	**5702.16**	5705.84	**5702.16**	**5702.16**	19.94	**5702.16**	**5702.16**	33.19	**5702.16**	**5702.16**	20.00	5702.16
C-p23	360	9	6092.66	**6078.75**	**6078.75**	**6078.75**	**6078.75**	19.97	**6078.75**	**6078.75**	38.69	6080.43	**6078.75**	20.00	6078.75
D-pr01	48	4	**861.32**	**861.32**	**861.32**	**861.32**	**861.32**	1.12	**861.32**	**861.32**	2.10	**861.32**	**861.32**	2.05	861.32
D-pr02	96	4	*1308.17*	*1307.34*	*1308.53*	**1296.25**	**1296.25**	2.70	**1296.25**	**1296.25**	4.74	**1296.25**	**1296.25**	4.67	1296.25
D-pr03	144	4	1810.66	1803.80	1804.09	**1803.80**	**1803.80**	4.93	**1803.80**	**1803.80**	8.54	**1803.80**	**1803.80**	8.84	1803.80
D-pr04	192	4	*2073.16*	*2058.31*	*2060.93*	2043.09	**2042.45**	10.99	**2042.45**	**2042.45**	13.24	**2042.45**	**2042.45**	12.84	2042.45
D-pr05	240	4	*2350.31*	*2335.81*	*2338.12*	2325.91	**2324.12**	16.40	2326.30	**2324.12**	25.88	2325.09	**2324.12**	19.51	2324.12
D-pr06	288	4	*2695.74*	*2680.95*	*2685.23*	2664.85	**2663.56**	19.26	2663.88	**2663.56**	34.66	2664.57	**2663.56**	20.00	2663.56
D-pr07	72	6	*1089.56*	*1089.56*	*1089.56*	**1075.12**	**1075.12**	1.78	**1075.12**	**1075.12**	3.01	**1075.12**	**1075.12**	3.04	1075.12
D-pr08	144	6	*1675.74*	*1664.99*	*1665.08*	1658.52	**1658.23**	6.36	**1658.23**	**1658.23**	7.73	**1658.23**	**1658.23**	7.85	1658.23
D-pr09	216	6	*2144.84*	*2133.52*	*2135.37*	2132.70	**2131.70**	10.39	2132.96	**2131.70**	17.23	2132.95	**2131.70**	15.65	2131.70
D-pr10	288	6	*2905.43*	*2885.39*	*2882.41*	2809.72	2807.17	18.60	2808.19	**2805.53**	38.64	2808.63	2807.11	20.00	2805.53
E-pr11	360	4	–	–	–	5018.62	**4994.67**	73.02	5001.94	**4994.67**	88.13	5004.67	**4994.67**	81.90	4994.67
E-pr12	480	4	–	–	–	6396.26	6371.96	127.92	6392.27	**6367.67**	109.81	6389.08	6375.87	139.18	6367.67
E-pr13	600	4	–	–	–	7683.73	7651.43	250.91	7668.48	**7645.29**	215.08	7668.70	7648.71	175.03	7645.29
E-pr14	720	4	–	–	–	9185.56	9163.63	293.97	9136.79	9106.37	294.81	9129.58	**9101.67**	296.51	9101.67
E-pr15	840	4	–	–	–	10682.96	10614.31	300.44	10627.80	10607.65	300.11	10638.93	**10598.70**	296.53	10598.70
E-pr16	960	4	–	–	–	12071.44	12016.08	301.22	11948.52	**11919.71**	300.38	11948.39	11921.00	300.32	11919.71
E-pr17	360	6	–	–	–	4772.85	**4761.70**	77.00	4765.72	4762.19	73.01	4766.29	**4761.70**	73.10	4761.70
E-pr18	520	6	–	–	–	6539.55	6504.81	158.53	6518.19	6504.72	192.22	6517.60	6504.62	162.11	6504.36
E-pr19	700	6	–	–	–	8707.25	8667.11	290.42	8669.38	8641.05	296.36	8669.61	**8639.44**	296.34	8639.44
E-pr20	880	6	–	–	–	9883.17	9834.39	300.74	9845.57	9826.77	300.28	9845.52	**9825.50**	300.26	9825.50
E-pr21	420	12	–	–	–	4603.04	4583.67	75.69	4595.10	**4582.62**	88.00	4596.54	4586.41	88.90	4582.62
E-pr22	600	12	–	–	–	6183.22	6154.31	198.82	6162.68	6147.01	207.01	6157.81	**6141.63**	233.83	6141.63
E-pr23	780	12	–	–	–	8101.79	8050.99	289.35	8037.81	8021.97	283.35	8032.95	**8014.10**	294.29	8014.10
E-pr24	960	12	–	–	–	10049.78	10004.94	300.01	9938.18	**9909.49**	300.48	9926.27	9910.02	300.57	9909.49
Gap/T Set-C			0.413%	0.049%	0.049%	0.056%	0.012%	8.20	0.027%	0.012%	12.80	0.023%	0.003%	10.09	
Gap/T Set-D			1.171%	0.747%	0.795%	0.037%	0.006%	9.25	0.026%	0.000%	15.58	0.025%	0.006%	11.45	
Gap/T Set-E			–	–	–	0.729%	0.275%	217.00	0.270%	0.026%	217.79	0.257%	0.020%	217.06	
Gap/T All			–	–	–	0.252%	0.089%	70.62	0.100%	0.014%	74.45	0.093%	0.009%	72.03	
T(min) Set-CD			3.95	5.18	10.45	–	–	8.52	–	–	13.64	–	–	10.50	
CPU			P-IV 3G	Opt 2.4G	I7 2.9G	Opt 2.2 & Opt 2.4G			Opt 2.2 & Opt 2.4G			Opt 2.2 & Opt 2.4G			

Table 5: Performance of HGSADC – MDVFMP instances

Inst n d	**SS97**	**HGSADC+**			**BKS**
	–	Avg 10	Best 10	T(min)	
C-p01 50 4	1526.7	**1477.73**	**1477.73**	1.87	1477.73
C-p02 50 4	992.8	**957.73**	**957.73**	2.44	957.73
C-p03 75 2	1611.1	**1569.67**	**1569.67**	3.37	1569.67
C-p04 100 2	2361.9	**2292.64**	**2292.64**	4.64	2292.64
C-p05 100 2	1498.4	**1453.64**	**1453.64**	7.42	1453.64
C-p06 100 3	2277.5	**2208.66**	**2208.66**	5.34	2208.66
C-p07 100 4	2297.1	**2198.91**	**2198.91**	5.19	2198.91
C-p08 249 2	6718.6	6448.26	**6441.36**	20.00	6441.36
C-p09 249 3	6211.4	6021.41	**5998.70**	20.00	5998.70
C-p10 249 4	6018.7	5817.81	**5807.53**	20.00	5807.53
C-p11 249 5	6030.8	5773.28	**5770.42**	19.74	5770.42
C-p12 80 2	2108.2	**2072.18**	**2072.18**	3.60	2072.18
C-p13 80 2	2126.8	**2096.39**	**2096.39**	3.56	2096.39
C-p14 80 2	**2160.1**	**2160.12**	**2160.12**	4.18	2160.12
C-p15 160 4	4116.2	**3973.47**	**3973.47**	9.61	3973.47
C-p16 160 4	4178.9	**4119.76**	**4119.76**	9.93	4119.76
C-p17 160 4	4344.1	4323.09	**4309.09**	13.95	4309.09
C-p18 240 6	6217.0	**5887.43**	**5887.43**	19.80	5887.43
C-p19 240 6	6233.6	**6130.36**	**6130.36**	19.47	6130.36
C-p20 240 6	6493.1	6481.23	**6469.21**	20.00	6469.21
C-p21 360 9	9184.6	8710.75	**8709.26**	20.00	8709.26
C-p22 360 9	9332.0	9164.65	**9151.64**	20.00	9151.64
C-p23 360 9	9706.6	9728.87	**9714.41**	20.00	9714.41
D-pr01 48 4	–	**1181.47**	**1181.47**	2.09	1181.47
D-pr02 96 4	–	**1901.39**	**1901.39**	6.81	1901.39
D-pr03 144 4	–	**2712.71**	**2712.71**	9.97	2712.71
D-pr04 192 4	–	3371.35	**3370.85**	24.15	3370.85
D-pr05 240 4	–	4068.98	**4066.52**	28.20	4066.52
D-pr06 288 4	–	4677.35	**4669.16**	62.08	4669.16
D-pr07 72 6	–	**1550.87**	**1550.87**	3.56	1550.87
D-pr08 144 6	–	**2705.46**	**2705.46**	10.73	2705.46
D-pr09 216 6	–	3642.57	**3637.39**	28.84	3637.39
D-pr10 288 6	–	4980.33	**4973.74**	48.63	4973.74
E-pr11 360 4	–	7342.40	**7323.10**	150.65	7323.10
E-pr12 480 4	–	9472.47	**9436.13**	251.09	9436.13
E-pr13 600 4	–	11567.96	**11526.27**	291.36	11526.27
E-pr14 720 4	–	13824.41	**13778.18**	300.01	13778.18
E-pr15 840 4	–	16393.71	**16352.74**	301.06	16352.74
E-pr16 960 4	–	18509.70	**18471.52**	300.46	18471.52
E-pr17 360 6	–	7048.26	**7025.01**	168.08	7025.01
E-pr18 520 6	–	9810.34	**9775.42**	245.35	9775.42
E-pr19 700 6	–	13312.76	**13280.04**	301.52	13280.04
E-pr20 880 6	–	15683.36	**15631.64**	301.52	15631.64
E-pr21 420 12	–	7087.02	**7072.79**	170.80	7072.79
E-pr22 600 12	–	9804.69	**9776.65**	288.99	9776.65
E-pr23 780 12	–	12878.26	**12840.99**	301.05	12840.99
E-pr24 960 12	–	16360.55	**16343.33**	300.02	16343.33
Gap/T Set-C	2.769%	0.067%	0.000%	11.92	
Gap/T Set-D	–	0.053%	0.000%	22.51	
Gap/T Set-E	–	0.282%	0.000%	262.28	
Gap/T All	–	0.128%	0.000%	88.75	
T(min) Set-C	4.10	–	–	11.92	
CPU	VAX 4000	Opt 2.2 & Opt 2.4G			

define moves only between sequences of visits to customers, thus avoiding to deal with special cases related to depots.

These experiments finally illustrate the good performance of the proposed HGSADC+ meta-heuristic, which matches or outperforms other current state-of-the-art CVRP and MDVRP with an average gap to BKS of 0.158% for the CVRP, and 0.093% for the MDVRP. In all cases, HGSADC variants produce solutions of better quality than ILS. As a counterpart, ILS is simpler and in general faster. All known optimal solutions have been retrieved. Run times remains moderate on medium-scale instances, of a magnitude comparable to those of other methods in the literature. The standard deviation of solution costs from HGSADC+ is small (0.069%, 0.071% for the CVRP and MDVRP), thus showing that high-quality solutions are produced in a consistent manner.

6.2 Addressing a rich problem – the MDVFMP

A second set of experiments has been conducted to investigate the performance of the proposed HGSADC+ on a rich vehicle routing variant, the MDVFMP. The literature is scarce on this particular problem. We compare the proposed method to the variable neighborhood heuristic of Salhi and Sari (1997) (SS97) and to the best known solution founds during multiple runs. Table 5 reports the results of the methods on the modified MDVRP instances, using the same format as previously.

As highlighted in Table 5, major solution improvements were still achievable on these instances. The average gap obtained with HGSADC+ is 0.067%, compared to 2.769% for SS97. Still, it should be noted that SS97 was executed on an old system and processor, and that extended runs on modern computers may lead to decreased gaps. Nevertheless, HGSADC+ demonstrate its ability to find the best known solutions in a consistent manner on this difficult problem: for 14/23 instances of set C with 50 to 240 customers the best known solution has been reached on all 10 runs, and the overall standard deviation of solution costs remains very small (0.089%).

7 Conclusions

In this paper, an efficient dynamic programming methodology was introduced for managing compound customer-to-depots assignments, rotations and vehicles choices within neighborhood searches for vehicle routing. Two meta-heuristics based on these concepts, an ILS and a HGSADC, have been proposed. These approaches produce solutions of remarkable quality on classic CVRP, MDVRP and MDVFMP benchmark instances with unlimited fleet. Extensive experiments demonstrate the notable contribution of the proposed implicit depot management to the search performance. The implicit rotations have a smaller but noticeable impact, and may simplify several aspects of LS implementations. The proposed methodology is general, and broadly applicable to many VRP variants.

Promising avenues of research involves generalizing the approach to other multi-attribute VRPs, possibly with limited fleet. Finally, this research is part of a general effort aiming to identify efficiently manageable subproblems to reduce the size of solution spaces, and similar subproblems and management methods may be investigated on other VRP variants and combinatorial optimization settings.

Acknowledgments

Partial funding for this project has been provided by the Champagne-Ardenne regional council, France, the Natural Sciences and Engineering Council of Canada (NSERC) through its Industrial Research Chair and Discovery Grant programs, and by the Fonds québécois de la recherche sur la nature et les technologies (FQRNT) through its Team Research Project program. This support is gratefully acknowledged.

References

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

Beasley, J.E. 1983. Route first – cluster second methods for vehicle routing. *Omega* **11**(4) 403–408.

Brandão, J. 2009. A deterministic tabu search algorithm for the fleet size and mix vehicle routing problem. *European Journal of Operational Research* **195**(3) 716–728.

Christofides, N., A. Mingozzi, P. Toth. 1979. The vehicle routing problem. N. Christofides, A. Mingozzi, P. Toth, C. Sandi, eds., *Combined Optimization*. Wiley, Chichester, UK, 315–338.

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

Cordeau, J.-F., G. Laporte, M.W.P. Savelsbergh, D. Vigo. 2007. Vehicle Routing. C. Barnhart, G. Laporte, eds., *Transportation*. No. 3 in Handbooks in Operations Research and Management Science, Elsevier, North-Holland, Amsterdam, 367–428.

Cordeau, J.-F., M. Maischberger. 2012. A Parallel Iterated Tabu Search Heuristic for Vehicle Routing Problems. *Computers & Operations Research* **39**(9) 2033–2050.

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

Dondo, R., J. Cerdá. 2007. A cluster-based optimization approach for the multi-depot heterogeneous fleet vehicle routing problem with time windows. *European Journal of Operational Research* **176**(3) 1478–1507.

Glover, F., J.-K. Hao. 2011. The case for strategic oscillation. *Annals of Operations Research* **183**(1) 163–173.

Goel, A., V. Gruhn. 2008. A general vehicle routing problem. *European Journal of Operational Research* **191**(3) 650–660.

Golden, B., S. Raghavan, E. Wasil, eds. 2008. *The Vehicle Routing Problem: Latest Advances and New Challenges*. Springer, New York.

Golden, B.L., E.A. Wasil, J.P. Kelly, I.M. Chao. 1998. The impact of metaheuristics on solving the vehicle routing problem: algorithms, problem sets, and computational results. T.G. Crainic, G. Laporte, eds., *Fleet management and logistics*. Kluwer Academic Publishers, Boston, MA, 33–56.

Groër, C., B. Golden, E. Wasil. 2011. A Parallel Algorithm for the Vehicle Routing Problem. *INFORMS Journal on Computing* **23**(2) 315–330.

Imran, A., S. Salhi, N.A. Wassan. 2009. A variable neighborhood-based heuristic for the heterogeneous fleet vehicle routing problem. *European Journal of Operational Research* **197**(2) 509–518.

Irnich, S. 2000. A multi-depot pickup and delivery problem with a single hub and heterogeneous vehicles. *European Journal of Operational Research* **122**(2) 310–328.

Kansou, A., A. Yassine. 2010. New upper bounds for the multi-depot capacitated arc routing problem. *International Journal of Metaheuristics* **1**(1) 81–95.

Kindervater, G.A.P., M.W.P. Savelsbergh. 1997. Vehicle routing: Handling edge exchanges. E.H.L. Aarts, J.K. Lenstra, eds., *Local Search in Combinatorial Optimization*. Princeton Univ Pr, 337–360.

Laporte, G. 2009. Fifty Years of Vehicle Routing. *Transportation Science* **43**(4) 408–416.

Liu, S., W. Huang, H. Ma. 2009. An effective genetic algorithm for the fleet size and mix vehicle routing problems. *Transportation Research Part E: Logistics and Transportation Review* **45**(3) 434–445.

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

Nagata, Y., O. Bräysy. 2009. Edge Assembly-Based Memetic Algorithm for the Capacitated Vehicle Routing Problem. *Networks* **54**(4) 205–215.

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

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

Prins, C. 2004. A Simple and Effective Evolutionary Algorithm for the Vehicle Routing Problem. *Computers & Operations Research* **31**(12) 1985–2002.

Prins, C. 2009a. A GRASP - Evolutionary Local Search Hybrid for the Vehicle Routing Problem. F.B. Pereira, J. Tavares, eds., *Bio-Inspired Algorithms for the Vehicle Routing Problem*. Springer-Verlag Berlin Heidelberg, 35–53.

Prins, C. 2009b. Two memetic algorithms for heterogeneous fleet vehicle routing problems. *Engineering Applications of Artificial Intelligence* **22**(6) 916–928.

Prins, C., N. Labadi, M. Reghioui. 2009. Tour splitting algorithms for vehicle routing problems. *International Journal of Production Research* **47**(2) 507–535.

Salhi, S., M. Sari. 1997. A multi-level composite heuristic for the multi-depot vehicle fleet mix problem. *European Journal of Operational Research* **103**(1) 95–112.

Subramanian, A. 2012. Heuristic, Exact and Hybrid Approaches for Vehicle Routing Problems. Ph.D. thesis, Universidade Federal Fluminense, Niterói, Brazil.

Subramanian, A., P.H.V. Penna, E. Uchoa, L.S. Ochi. 2012. A Hybrid Algorithm for the Heterogeneous Fleet Vehicle Routing Problem. *European Journal of Operational Research* **221**(2) 285–295.

Toth, P., D. Vigo, eds. 2002. *The vehicle routing problem*. SIAM Monographs on Discrete Mathematics and Applications, Society for Industrial Mathematics, Philadelphia, PA, USA.

Toth, P., D. Vigo. 2003. The Granular Tabu Search and Its Application to the Vehicle-Routing Problem. *INFORMS Journal on Computing* **15**(4) 333–346.

Vidal, T., T.G. Crainic, M. Gendreau, N. Lahrichi, W. Rei. 2012a. A Hybrid Genetic Algorithm for Multi-Depot and Periodic Vehicle Routing Problems. *Operations Research* **60**(3) 611–624.

Vidal, T., T.G. Crainic, M. Gendreau, C. Prins. 2011. A Unifying View on Timing Problems and Algorithms. Tech. rep., CIRRELT.

Vidal, T., T.G. Crainic, M. Gendreau, C. Prins. 2012b. A Unified Solution Framework for Multi-Attribute Vehicle Routing Problems. Tech. rep., CIRRELT.

Vidal, T., T.G. Crainic, M. Gendreau, C. Prins. 2012c. Heuristics for Multi-Attribute Vehicle Routing Problems : A Survey and Synthesis. Tech. rep., CIRRELT.

Vidal, T., T.G. Crainic, M. Gendreau, C. Prins. 2013. A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows. *Computers & Operations Research* **40**(1) 475–489.

Wolf, S., P. Merz. 2007. Evolutionary local search for the super-peer selection problem and the p-hub median problem. T. Bartz-Beielstein, M.J. Belsa Aguilera, C. Blum, B. Naujoks, A. Roli, G. Rudolph, M. Sampels, eds., *Hybrid Metaheuristics*, *LNCS*, vol. 4771. Springer Berlin Heidelberg, 1–15.