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

Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation

Solution Integration in Combinatorial Optimization with Applications to Cooperative Search and Rich Vehicle Routing

Nizar El Hachemi
Teodor Gabriel Crainic
Nadia Lahrichi
Walter Rei
Thibaut Vidal

September 2014

CIRRELT-2014-40

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

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

www.cirrelt.ca

Solution Integration in Combinatorial Optimization with Applications to Cooperative Search and Rich Vehicle Routing

Nizar El Hachemi[1,2], Teodor Gabriel Crainic[1,3,*], Nadia Lahrichi[1,4], Walter Rei[1,3], Thibaut Vidal[5]

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

[2] Département de génie industriel, École Mohammadia d'ingénieurs, Université Mohammed V, Avenue Ibnsina, B.P. 765 Agdal, Rabat, Maroc

[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, Polytechnique Montréal, P.O. Box 6079, Station Centre-ville, Montréal, Canada H3C 3A7

[5] Tecgraf, Pontifícia Universidade Católica do Rio de Janeiro, Rua Marquês de São Vicente, 225, Prédio Pe. Laércio Dias de Moura, 22453-900 Rio de Janeiro, RJ, Brazil

Abstract. Structural problem decomposition requires the ability to recombine partial solutions. This recombination task, which we call integration, is a fundamental feature of many methods, both those based on mathematical formulations such as Dantzig-Wolfe or Benders and those based on heuristics that involve sequential solution. Integration may be implicit in mathematical decompositions, but in heuristics this critical task is usually managed by ad-hoc operators, e.g., operators that combine decisions and heuristic adjustments to manage incompatibilities. In this paper, we propose a general framework for integration, which is viewed as a problem in itself with well-defined objectives and constraints. Four different mechanisms are proposed, based on well-known concepts from the literature such as constraining or giving incentives to the critical decisions related to partial solutions. We perform computational experiments on the multi-depot periodic vehicle routing problem to compare the various integration approaches. The strategy that places incentives on selected decisions rather than imposing constraints seems to yield the best results in the context of a cooperative search for this problem.

Keywords: Solution integration, combinatorial optimization, heuristic search, multi-depot periodic vehicle routing.

Acknowledgements. While working on this project, T.G. Crainic was the NSERC Industrial Research Chair in Logistics Management, ESG UQAM, N. El Hachemi and N. Lahrichi were postdoctoral fellows with the Chair, M. Gendreau was the NSERC/Hydro-Québec Industrial Research Chair on the Stochastic Optimization of Electricity Generation, MAGI, Polytechnique Montréal. Partial funding for this project has been provided by the Natural Sciences and Engineering Council of Canada (NSERC), through its Industrial Research Chair, Collaborative Research and Development, 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 de recherche du Québec - Nature et technologies (FRQNT) through its Team Research Project program.

* Corresponding author: Teodor-Gabriel.Crainic@cirrelt.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2014

1 Introduction

Combinatorial optimization problems appear naturally in various application areas; these problems are usually large and complex. Various solution methodologies, both exact and heuristic, have been developed to tackle these problems, and many algorithms apply a form of solution integration. Solution integration is a general process that combines specific solutions of a problem to produce new solutions.

Exact methods that are based on decomposition strategies, such as those proposed by Dantzig and Wolfe [6] and Benders [1], entail a form of solution integration. In these algorithms, at a given iteration, the solution to a master problem is used to instantiate a subproblem that is solved to produce a new column (Dantzig–Wolfe decomposition [6]) or a new cut (Benders decomposition [1]). The new column or cut is then added to the master formulation to strengthen it. Meaningful new information is thus obtained from each solution encountered, and by integrating all the information generated in the master formulation, we can ensure the convergence of the overall solution procedures. Therefore, in the case of both [6] and [1], the solution integration process is clearly defined and formalized by the specific decomposition strategy that is used. However, this is not the case when solution integration is applied in the context of designing heuristic approaches.

Many classes of heuristics and metaheuristics also use solution integration in their search processes. Genetic algorithms [8, 12], scatter search [7], and path relinking [7] are all examples of procedures that employ some form of solution integration (as we highlight in Section 2). In these methods, the solution integration process is usually performed by ad-hoc operators that are tailored to the specific application. However, one usually observes that general underlying solution-integration problems are being addressed by these operators. Unfortunately, in the literature on heuristics and metaheuristics, there is a lack of formalization of these problems and of the methodological questions that they raise. Furthermore, usually no systematic study is conducted to analyze the impact that these problems have on the performance of the associated heuristic procedures.

The present paper aims to compensate for this lack of formalism in the definition and treatment of solution integration applied to heuristic procedures. We make the following contributions:

1. Provide a clear definition of solution integration as a general decision problem;
2. Develop a series of optimization models to formulate the solution-integration problem, based on different strategies (either restrictions or incentives);
3. Apply the models in the context of performing cooperative searches for the multi-depot periodic vehicle routing problem (MDPVRP);

4. Analyze the impact that solution integration has on the performance of this heuristic search for the MDPVRP, and we show the overall efficiency of the integration models proposed.

The rest of the paper is organized as follows. In Section 2, we review and illustrate the problem of solution integration in the context of heuristics and metaheuristics. The problem statement and the different integration models are then described in Section 3. In Section 4, we apply the integration models to cooperative search for the MDPVRP. The computational experiments are reported and analyzed in Section 5. Finally, Section 6 provides concluding remarks.

2 The solution-integration problem

Solution integration is a defining element of many classes of heuristics and thus appears under many guises in the literature. In general terms, given an optimization problem, the solution-integration process first selects a few complete or partial solutions with desirable characteristics, the so-called “good” *input solutions*, and then combines them, in the *integration* step, to generate new solutions that preserve the desirable characteristics and that are “good” with respect to specified criteria, usually quality and diversity.

Population-based metaheuristic search methods are a prime illustration of such processes; here solution integration is essential to the algorithms. Genetic algorithms [8, 12], for example, use a so-called fitness function to assign to each individual solution in a given population a value reflecting its quality and, eventually, its diversity [14] relative to that of the other individuals in the population. This fitness function is then used to select the individuals (the input solutions) that are then combined through certain procedures (the crossover operators) to generate new solutions (the offspring) inheriting the desirable characteristics of the input solutions and enhancing the measures in the fitness function.

Scatter search [7] also evolves a population but applies different selection and combination procedures. It works with a small population of elite solutions (the elite set). It first selects a small number of good solutions from the elite set and then integrates them through a combination procedure, e.g., via linear or convex combinations of sets of variables. The goal is to obtain improved solutions or, at least, solutions that are better in quality, diversity, or both than the less interesting ones in the elite set. Search context information, typically recorded in memories counting the occurrence of particular characteristics (e.g., the persistence of solution elements), is generally used to influence both the selection of the input solutions and the integration step.

Path relinking [7], sometimes presented as a special case of scatter search or as a mechanism to simultaneously perform diversification and intensification in metaheuristic search, is also based on solution integration. Path relinking also works with an elite set of solutions and proceeds by selecting two solutions, the *initial* and the *guiding*, from this set. The guiding solution (often the current best solution) has the desirable characteristics, and we aim to introduce these characteristics into the initial solution through a series of neighborhood-based moves (solution transformations) that form the relinking path. The idea is that, since good solutions often occur close to other good solutions, we will find improved solutions while building the path (which will be stopped when we reach the guiding solution or earlier if the search is no longer producing good solutions). These improved solutions are inserted into the elite set for the next stage of the method. Search context information, as defined above, is again heavily used to guide the neighborhood moves.

Similar processes are found in cooperative search, where several solution methods explore the solution space simultaneously, exchanging information through data warehousing and processing mechanisms (*central* and *adaptive* memory are encountered most often) [3, 5, 13]. Le Bouthillier and Crainic [10] proposed a central-memory cooperative framework for the vehicle routing problem with time windows, where integration was carried out through the identification and use of globally promising solution characteristics. In this approach, the cooperating solution methods (tabu searches and genetic algorithms) store their good solutions in the central memory. The approach identifies elite sets (the input solutions) and records the persistence of solution elements (arcs) in these sets during the search. It then identifies sets of persistent elements (the desirable characteristics) and passes these structures (called patterns in [10, 11]) to the requesting cooperating methods, along with a solution chosen among the best. The structure is then used to modify the search trajectory of the receiving cooperating method by influencing the selection of the arcs to be included in the solution. A companion paper [11] generalized this approach. The central-memory solutions are separated into three groups according to the solution quality, elite, average, or bad, and the approach monitors the persistence of solution elements within these sets. It then selects patterns of persistent/nonpersistent elements in a particular solution set to guide the cooperating methods (e.g., nonpersistent elements of average solutions allow for a broad exploration, whereas persistent or nonpersistent elements in elite solutions intensify or diversify the search).

Two general conclusions emerge from this brief literature survey. First, there has been no systematic study of solution integration within heuristic search methods for optimization. Second, existing integration procedures are at best specific to a class of solution methods, and they are often ad-hoc, algorithm-specific rules. We can however identify a general solution-integration problem that, given a definition of solution-evaluation measures, desirable characteristics, and desired output solution, is made up of three main components:

1. Selection of input solutions;
2. Identification of desirable characteristics in these solutions;
3. Integration of solutions.

In the following section, we present a formalization of the first two components and introduce several optimization formulations to perform the third.

3 Problem setting and integration methodologies

In this section, we present the essential elements of integration including critical variables, input solutions, integration output, and integration criteria. We also provide complete notation for the four models of integration proposed in this paper.

3.1 Notation

Let model (P) define an optimization problem formulated as follows: $\min_{\mathbf{x}\in X} \quad f(\mathbf{x})$, where $\mathbf{x}$ is an n-dimensional decision vector, $f(\mathbf{x})$ is a real-value function of $\mathbf{x}$, and X defines the domain of $\mathbf{x}$. Given (P), the integration problem considered in this paper is defined as follows: Let $y^i \in X, i \in I$ (where I is a finite index set) be a set of input solutions. The elements of the integrated solution are identified by two measures. First, critical variables define characteristics of the input solutions that should be integrated; we denote them by $\lambda_i \in [0,1]^n$, a vector whose components $j \in N$ (where $N = \{1, ..., n\}$) are defined as follows:

$$\lambda_{i(j)} = \begin{cases} 1 & \text{if variable } j \text{ in input solution } y^i \text{ is critical;} \\ 0 & \text{otherwise.} \end{cases}$$

We note that a variable is defined to be critical if it has been identified as important during the process leading to y^i. Second, the incentive measure is denoted $p_i \in \Re^n_+$, and its components $p_{i(j)}$, $\forall j \in N$, represent the importance of the associated feature.

We denote by $\mathbf{x}^i_c$, $i \in I$, the set of critical variables associated with each input-solution vector y^i. $\mathbf{x}_c$ represents the set of critical variables x for which there is an $i \in I$ such that $x \in \mathbf{x}^i_c$ (i.e., $\mathbf{x}_c = \cup \mathbf{x}^i_c$), and $N(i)$ designates the set of indices of critical variables belonging to y^i (i.e., $N(i) = \{j \in N \mid \lambda_{i(j)} = 1\}$). Finally, we define $\Lambda(j)$ to be the index set of all input solutions that have j as a critical variable (i.e., $\Lambda(j) = \{i \in I \mid \lambda_{i(j)} = 1\}$).

3.2 Problem statement

To define the integration problem, we must specify the input information, y^i, p_i, and λ_i, $i \in I$, and define what will be produced as output and how the output will be evaluated (i.e., the objective).

We distinguish two ways of selecting the input solutions and their critical variables. The first involves choosing input solutions that are disjoint in terms of the critical variables, so that $\forall i \neq j \in I$, we have $\mathbf{x}_c^i \cap \mathbf{x}_c^j = \emptyset$. Thus, the $\mathbf{x}_c^i$ $(i \in I)$ form a partition of $\mathbf{x}_c$. The second approach is simply to choose input solutions that have some critical variables in common, so that $\exists i \neq j \in I$, such that $\mathbf{x}_c^i \cap \mathbf{x}_c^j \neq \emptyset$.

Output

We assume that we wish to produce a feasible solution to the problem or to determine that no such solution exists. The definition of the criteria used to evaluate the output solution is also an important component of the integration problem.

Criteria

We will use either a single function, which may simply be the objective function of the original problem, or a multicriteria fitness function that measures quality, diversity, or closeness to critical variables (e.g., the fitness function defined in Vidal *et al.* [14]. Table 1 provides a summary of the general outline that we propose for the integration problem. The next step is to formulate the problem via an optimization model.

3.3 Integration models

As previously mentioned, the integration problem is formulated via a specific optimization model. We propose four models, each based on a different integration strategy. The choice of model depends on the application and the particular characteristics of the integration strategy.

The two main integration concepts in the literature involve using restrictions to ensure the appearance of desirable features in the result, or encouraging the appearance of given features.

We propose a mathematical formalism for integration based on the concept of critical variables. There are two categories of integrators. The first works with input solutions that are disjoint in terms of the critical variables, and the second allows nondisjoint solutions. We now formulate the different integration models.

INPUT	• Input solutions $y^i \in X$, $i \in I$ (or solution fragments that are partially optimized). • Critical variable vectors λ_i, $i \in I$, that identify the characteristics of the input solutions y^i that should be integrated. • Incentive measures p_i, $i \in I$, that represent the degree of occurrence (of the feature encouraged) in the integrated solution.
OUTPUT	• A feasible solution $\mathbf{x} \in X$ or an empty set if no feasible solution can be obtained.
OBJECTIVE	• Possible criteria: solution quality, solution quality or diversity relative to the input solutions, proximity to critical characteristics identified in the input solutions, combinations of objectives.

Table 1: The integration problem

First category

This category deals with input solutions that are disjoint in terms of the critical variables. Thus, the eligible input solutions y^i ($\forall i \in I$) must satisfy $\forall j \in N$, $|\Lambda(j)| \leq 1$. This category consists of two integrators called I_{PART} and I_{PEN}. We use I_{PART} when we want the critical features of the input solutions to be found in the output of the integration process. To achieve this, I_{PART} sets the critical variables in all the input solutions to their current values as follows:

$$(I_{PART}) \quad \min_{\mathbf{x} \in X} \quad f(\mathbf{x}) \tag{1}$$

$$s.t. \quad x_j = y_j^i \qquad \forall i \in I \,; \forall j \in N : \lambda_{i(j)} = 1 \tag{2}$$

This integration approach gives a straightforward combination of the input solutions. However, constraint (2) may lead to an infeasible problem.

We use I_{PEN} when the critical features are preferred but not required. An additional term is added to the objective function to encourage the appearance of the critical characteristics in the solution. The model I_{PEN} is as follows:

$$(I_{PEN}) \quad \min_{\mathbf{x} \in X} \quad f(\mathbf{x}) + \sum_{i \in I} \sum_{j \in N} \lambda_{i(j)} p_{i(j)} (x_j - y_j^i)^2 \tag{3}$$

where $p_{i(j)}$ (the relative importance of the critical feature) represents the weight associated with moving the critical variable x_j toward the value y^i_j. The more we judge that a promising feature should appear in the solution, the higher its weight.

Second category

This category deals with input solutions that have some critical variables in common. It consists of two integrators, I_{AND} and I_{PAND}. We use I_{AND} when only those critical features for which there is consensus may appear in the output of the integration process. To achieve this, I_{AND} sets the critical variables for which there is consensus to their current values. For a given critical variable, consensus occurs if this variable has the same value in all the input solutions in which it appears. Define $\mathcal{C} = \{j \in N \mid \sum_{i \in I} \lambda_{i(j)} \geq 1, y^k_j = y^l_j, \forall k, l \in \Lambda(j), k \neq l\}$ to be the set of critical variables for which there is consensus, and let γ_j, $\forall j \in \mathcal{C}$, be the values of these variables. Then I_{AND} is as follows:

$$(I_{AND}) \quad \min_{\mathbf{x} \in X} \quad f(\mathbf{x}) \tag{4}$$

$$s.t. \quad x_j = \gamma_j \qquad \forall j \in \mathcal{C} \tag{5}$$

In other words, this integrator preserves the common critical features that are present in the input solutions. An important property of this approach is that all the input solutions considered remain feasible in model (4)–(5). Therefore, if $\mathbf{x_{AND}}$ defines an optimal solution to (4)–(5), then $f(\mathbf{x_{AND}}) \leq f(y^i), \forall i \in I$.

I_{PAND} also sets the critical variables for which there is consensus to their current values. The remaining critical variables are encouraged to appear in the solution through the use of the incentive term defined for I_{PEN}. For each such variable just one characteristic can be encouraged; this can be achieved by setting the weights associated with the other critical characteristics to zero. Therefore, I_{PAND} is as follows:

$$(I_{PAND}) \quad \min_{\mathbf{x} \in X} \quad f(\mathbf{x}) + \sum_{i \in I} \sum_{j \in N \setminus \mathcal{C}} \lambda_{i(j)} p_{i(j)} (x_j - y^i_j)^2 \tag{6}$$

$$s.t. \quad x_j = \gamma_j \qquad \forall j \in \mathcal{C} \tag{7}$$

The integration problems I_{PART}, I_{PEN}, I_{AND}, and I_{PAND} retain the same structure (i.e., the constraint set X) as the optimization problem from which they are derived. The models are obtained either by applying a restriction to the original feasible region (I_{PART}, I_{AND}, and I_{PAND}) or by adding an incentive term to the objective function (I_{PEN} and I_{PAND}). The use of a restriction strategy necessarily reduces the feasible space of the integration models compared to the original model. The restriction may even lead to a decomposition of the original problem into a series of smaller subproblems. Therefore, the integration models obtained by applying restrictions are expected to be easier to solve than the original problem. Moreover, the use of the incentive term does not dramatically change the objective function. Thus, any algorithm designed to solve the original problem can be adapted to solve the integration problems.

4 Case study: Integrative cooperative search for the MDPVRP

We illustrate the different integration concepts using the MDPVRP, which is a multiple-decision-set and multiple-attribute routing problem. We use the integrative cooperative search (ICS) framework of Crainic *et al.* [4] to decompose by decision set and to solve the problem, and therefore to generate input solutions for our four integrators. We give criteria for selecting the input information for the integration, and a clear description of the critical characteristics and their associated critical variables. Finally, we present the mathematical formulations of the four integrators described in the previous section.

4.1 The MDPVRP and its decomposition

The MDPVRP designs a set of routes for a fleet of homogeneous vehicles located at different depots. Customers require multiple deliveries with known demands following a defined frequency in the given period. A list of possible visit-combinations is provided for each customer (i.e., lists of periods when visits may occur). Each vehicle performs at most one route per day, the route starts and finishes at the same depot, and each customer must be served from the same depot throughout the time horizon.

We use the notation introduced in Vidal *et al.* [14]. Let $\mathcal{V}^{\text{CST}}$ denote the set of customers and $\mathcal{V}^{\text{DEP}}$ the set of depots, with $\mathcal{V} = \mathcal{V}^{\text{CST}} \cup \mathcal{V}^{\text{DEP}}$. Each customer i requires q_i units, and τ_i is the service time. We assume that $\tau_o = 0$ for each depot v_o. Visits to customer i should occur during one of the possible visit-combinations L_i. Let the binary constants $a_{pl} = 1$ if day l belongs to visit-combination p and 0 otherwise.
A fleet of m homogeneous vehicles with capacity Q is available at each depot. The duration of the routes should not exceed T. For all $v_i, v_j \in \mathcal{V}$, c_{ij} is defined to be the travel time from vertex v_i to v_j. We introduce three sets of binary variables:

$$y'_{io} = \begin{cases} 1 & \text{if customer } i \text{ is assigned to depot } v_o \\ 0 & \text{otherwise;} \end{cases}$$

$$y_{ip} = \begin{cases} 1 & \text{if customer } i \text{ is assigned day combination } p \in L_i \\ 0 & \text{otherwise;} \end{cases}$$

$$x_{ijklo} = \begin{cases} 1 & \text{if customer } i \text{ is followed by } j \text{ in route of vehicle } k \text{ originating from } o \text{ on day } l \\ 0 & \text{otherwise.} \end{cases}$$

Appendix A presents the detailed mathematical formulation of the MDPVRP.

We decompose the problem using the ICS paradigm introduced in [4]. We choose the depot and the period decision sets for the decomposition. We obtain two problems, the periodic vehicle routing problem (PVRP) and the multi-depot vehicle routing problem (MDVRP) for which algorithms are available. Indeed, fixing the customer-to-depot assignments in the MDPVRP yields several PVRPs and fixing the patterns (day-combinations) for all customers yields a subset of MDVRPs. In the latter case, a global linking constraint is added to ensure that the customer-to-depot assignment is the same in each period.

4.2 Integration subproblems for MDPVRP

According to Section 3, to define each integrator subproblem, we need to first identify the input information. Since the MDPVRP decomposes by depot and period decision sets, we use two solutions, one from the solver applied to the MDVRP and the other from the PVRP. Both categories of integrators may be applied.

The critical characteristics of the solutions are the customer-to-depot assignment and the customer-to-pattern assignment. Formally, let S_1 and S_2 be two partial solutions selected as input information where S_1 is an MDVRP solution and S_2 is a PVRP solution; d_{ji} and p_{ji} are respectively the depot and pattern of customer i for solution S_j. Therefore, d_{1i}, p_{1i} are the critical characteristics of S_1 and d_{2i}, p_{2i} are the critical characteristics of S_2.

Integrators I_{PART} and I_{PEN} deal with two disjoint families of critical characteristics. I_{PART} **sets** the customer-to-depot assignments (y'_{io}) of the first family and the customer-to-pattern assignments (y_{ip}) of the second family.

$$(I_{PART}) \text{ Minimize } \sum_{v_i \in \mathcal{V}} \sum_{v_j \in \mathcal{V}} \sum_{k=1}^{m} \sum_{l=1}^{t} \sum_{v_o \in \mathcal{V}^{\text{DEP}}} c_{ij} x_{ijklo} \tag{8}$$

$$\text{subject to: } y'_{id_{1i}} = 1 \qquad v_i \in \mathcal{V}^{\text{CST}} \tag{9}$$

$$y_{ip_{2i}} = 1 \qquad v_i \in \mathcal{V}^{\text{CST}} \tag{10}$$

$$\text{All constraints of MDPVRP model} \tag{11}$$

The objective is to minimize the routing cost where the constraints fix the customers' assignment to depot 9 and pattern 10. Note that the MDPVRP in which this assignment is complete separates into (number of depots)*(number of periods) smaller VRPs. State-of-the-art algorithms such as Vidal *et al.* [14] may be used.

Integrator I_{PEN} **encourages** the critical characteristics to appear; an incentive term is introduced in the objective function. Let ρ_o and ρ_p be n-dimensional positive vectors to encourage the critical characteristics in solutions S_1 and S_2 respectively. The incentive term is $\sum_{v_i \in \mathcal{V}^{\text{CST}}} \rho_o^i (1 - y'_{id_{1i}}) + \sum_{v_i \in \mathcal{V}^{\text{CST}}} \rho_p^i (1 - y_{ip_{2i}})$.

$$(I_{PEN}) \quad \text{Minimize} \quad \sum_{v_i \in \mathcal{V}} \sum_{v_j \in \mathcal{V}} \sum_{k=1}^{m} \sum_{l=1}^{t} \sum_{v_o \in \mathcal{V}^{\text{DEP}}} c_{ij} x_{ijklo} + \sum_{v_i \in \mathcal{V}^{\text{CST}}} \rho_o^i (1 - y'_{id_{1i}}) + \sum_{v_i \in \mathcal{V}^{\text{CST}}} \rho_p^i (1 - y_{ip_{2i}}) \quad (12)$$

$$\text{subject to: All constraints of MDPVRP model} \quad (13)$$

Integrators I_{AND} and I_{PAND} deal with a subset of the two selected solutions. I_{AND} fixes the customer-to-depot and customer-to-pattern assignments common to S_1 and S_2. Let $\mathcal{V}^{\text{CST}'}$ be the set of customers i satisfying $d_{1i} = d_{2i}$, and define $\mathcal{V}^{\text{CST}''}$ to be the set of customers i such that $p_{1i} = p_{2i}$. I_{AND} may be formulated as follows:

$$(I_{AND}) \quad \text{Minimize} \quad \sum_{v_i \in \mathcal{V}} \sum_{v_j \in \mathcal{V}} \sum_{k=1}^{m} \sum_{l=1}^{t} \sum_{v_o \in \mathcal{V}^{\text{DEP}}} c_{ij} x_{ijklo} \quad (14)$$

$$\text{subject to: } y'_{id_{1i}} = 1 \qquad v_i \in \mathcal{V}^{\text{CST}'} \quad (15)$$

$$y_{ip_{2i}} = 1 \qquad v_i \in \mathcal{V}^{\text{CST}''} \quad (16)$$

$$\text{All constraints of MDPVRP model} \quad (17)$$

The resulting integration problem is smaller than the original MDPVRP. The objective function is unchanged, and constraints (15) and (16) are added for the set pairs (customer, depot) or (customer, day combination) that are common to both parent solutions S_1 and S_2. The other constraints of the MDPVRP remain the same. There are few MDVRP algorithms in the literature; to the best of our knowledge, Vidal *et al.* [14] is the only approach for solving this problem directly. A generalized version of the unified tabu search proposed by Cordeau *et al.* [2], namely GUTS, is also available [4].

Integrator I_{PAND} uses the set $\overline{\mathcal{V}^{\text{CST}'}}$ that is the complement of $\mathcal{V}^{\text{CST}'}$, i.e., $\overline{\mathcal{V}^{\text{CST}'}} \cup \mathcal{V}^{\text{CST}'} = \mathcal{V}^{\text{CST}}$, and similarly for $\overline{\mathcal{V}^{\text{CST}''}}$. Let ρ_o and ρ_p represent respectively a $|\overline{\mathcal{V}^{\text{CST}'}}|$-

dimensional and a $|\overline{\mathcal{V}^{\text{CST}''}}|$-dimensional positive vector to encourage any critical characteristics that are not common to solutions S_1 and S_2. In I_{PAND}, we add to the objective function of I_{AND} two positive terms to encourage critical characteristics (see (18)).

$$(I_{PAND})\ \text{Minimize} \sum_{v_i\in\mathcal{V}}\sum_{v_j\in\mathcal{V}}\sum_{k=1}^{m}\sum_{l=1}^{t}\sum_{v_o\in\mathcal{V}^{\text{DEP}}} c_{ij}x_{ijklo} + \sum_{v_i\in\overline{\mathcal{V}^{\text{CST}'}}}\rho_o^i(1-y'_{id_{1i}}) + \sum_{v_i\in\overline{\mathcal{V}^{\text{CST}''}}}\rho_p^i(1-y_{ip_{2i}}) \quad (18)$$

$$\text{subject to: } y'_{id_{1i}} = 1 \qquad v_i \in \mathcal{V}^{\text{CST}'} \quad (19)$$

$$y_{ip_{2i}} = 1 \qquad v_i \in \mathcal{V}^{\text{CST}''} \quad (20)$$

$$\text{All constraints of MDPVRP model} \quad (21)$$

I_{PAND} fixes the same customer-to-depot and customer-to-pattern assignments as in I_{AND}. Customer-to-depot assignments (y'_{io}) for the solution selected from the MDVRP solver and customer-to-pattern assignments (y_{ip}) for the solution selected from the PVRP solver (which we have not determined yet) are encouraged via the objective function.

There is a direct relationship between the four models: the critical assignments are either fixed or encouraged, and either all assignments or only a subset are fixed. The models use constraints and incentive terms in the objective function to implement the chosen integration strategy.

5 Computational Experiments

To test the efficiency of our integrators we used the data sets from Vidal *et al.* [14]: 10 MDPVRP problems of various sizes, ranging from instances with 48 customers and 96 services to instances with 288 customers and 864 services, on a horizon of four or six days. The customers are randomly located, with some of them being grouped into clusters to imitate many practical problem settings.

We coded the integrators and the ICS application in C++ and used a cluster of Itanium II 1.5 Ghz processors, with 64-bit values for all the data (including distances) and calculations.

We developed three families of tests. The first compares the individual performance of each integrator in terms of the computational time, the quality of the resulting solution, and the distance between the input and output solutions computed as follows:

$$dist(S_1, S_2, C) = \frac{\sum_{i \in \mathcal{V}^{\text{CST}}} (\delta_{d_i^C}^{d_{1i}} + \delta_{p_i^C}^{p_{2i}})}{2card(\mathcal{V}^{\text{CST}})}$$

where S_1 is a solution associated with the MDVRP partial solver and S_2 is a solution associated with the PVRP partial solver. We tested two strategies for the selection of S_1 and S_2. First, we randomly selected them from the best 25% of the solutions. Second, we performed five uniform random selections for each parent (S_1 and S_2) among the best 25% of the solutions, generating 25 combinations (pairs) of parent solutions. We retained the pair where the two solutions were closest in terms of distance (i.e., correlated parents). C is the complete solution resulting from integrating S_1 and S_2, and d_{1i} and d_i^C represent the depot for customer i in solution S_1 and in the complete solution C. Similarly, p_{2i} and p_i^C denote the patterns (allowable day-combinations of visits and days) for customer i associated with S_2 and with C. Finally, δ is a discrete function defined as follows:

$$\delta_i^i = 1, \ \forall i, \ and \ \delta_j^i = 0, \ \forall i \neq j$$

The second family of tests studies the impact of the quality of S_1 and S_2 on the complete solution C obtained by the integration process. For both families, we used the HGA developed by Vidal *et al.* [14] to solve the integration problem. The third family of tests measures the impact of different combinations of integrators in the ICS context. It uses a generalization of the UTS [2] method for partial solvers and the HGA of Vidal *et al.* [14] for the integration.

Tables 2 and 3 report the average percentage gap in relation to the best known solution (BKS) obtained from Lahrichi *et al.* [9]. The gap is computed using the formula $\frac{z^* - BKS}{BKS}$ where z^* is the best solution found. The tables also report the average time in seconds to perform the integration, and the average distance between the complete solution resulting from the integration and the associated input solutions. These results show that I_{AND} outperforms the other approaches in terms of the solution quality (average deviation from BKS) for both parent-selection strategies. To strengthen our conclusions, we conducted a Friedman test to compare the different integrators, considering the solution quality (% gaps) obtained for 10 instances by running each of them 10 times. We performed the test for correlated parents, and Figure 2 gives the results as boxplots of paired differences in solution quality. The grey boxes indicate statistically significant pairwise effects as highlighted by a Friedman test with post-hoc. This statistical test confirms that I_{AND} is better than I_{PAND}, I_{PART}, I_{PEN}, and I_{RAND} with high confidence: $p < 0.00045$, $p < 10^{-5}$, $p < 10^{-5}$, and $p < 10^{-5}$ respectively (see the first four boxes of Figure 2). However, in terms of computing time and the average distance between the complete solution and the input solutions, I_{PART} is the best approach. It provides solutions with the maximum possible common attributes with the input solutions, since

Criterion	Integrator	Instance									
		01	02	03	04	05	06	07	08	09	10
Parents	Gap_{BKS} (%)	0.55	0.35	0.98	1.62	1.29	1.38	0.90	0.74	0.41	2.60
Gap_{BKS} (%)	I_{PART}	0.87	0.15	1.98	1.47	1.71	1.47	1.78	0.26	0.52	3.29
	I_{AND}	0.07	0.03	0.53	0.86	0.57	0.83	0.34	0.20	0.14	2.07
	I_{PAND}	0.18	0.05	0.53	0.86	0.59	0.78	0.37	0.22	0.22	2.2
	I_{PEN}	0.15	0.37	1.00	1.43	3.26	2.17	0.98	0.98	2.87	6.68
	I_{RAN}	0.10	0.91	1.11	2.46	5.96	2.79	1.16	3.05	4.22	8.27
$Time$ (s)	I_{PART}	0.31	0.80	2.50	6.15	20.36	22.01	0.78	2.41	7.97	40.20
	I_{AND}	0.45	2.02	10.07	22.86	42.44	52.57	3.02	9.82	27.56	64.27
	I_{PAND}	0.43	1.53	6.73	17.69	35.30	40.49	2.28	6.95	19.48	61.25
	I_{PEN}	0.90	3.43	9.00	29.49	62.55	71.24	4.24	18.14	66.19	104.80
	I_{RAN}	1.32	6.97	17.06	36.93	69.25	89.6	5.73	25.64	72.92	121.12
$Dist.$	I_{PART}	0	0	0	0	0	0	0	0	0	0
	I_{AND}	0.16	0.14	0.15	0.14	0.09	0.09	0.22	0.16	0.08	0.10
	I_{PAND}	0.09	0.04	0.05	0.04	0.02	0.02	0.12	0.07	0.03	0.03
	I_{PEN}	0.10	0.05	0.09	0.06	0.06	0.05	0.18	0.13	0.14	0.16
	I_{RAN}	0.24	0.31	0.32	0.30	0.35	0.32	0.42	0.42	0.38	0.39

Table 2: Integrator results with random choice of parents

it greatly reduces the search space by fixing the optimized parts (attributes) from each input solution. It is however the worst approach in terms of solution quality. When we consider the three criteria simultaneously, I_{PAND} seems to be the most efficient approach since it offers a good compromise.

Criterion	Integrator	Instance									
		01	02	03	04	05	06	07	08	09	10
Parents	Gap_{BKS} (%)	0.87	1.15	1.59	2.74	2.57	4.71	2.01	3.77	2.72	4.27
Gap_{BKS} (%)	I_{PART}	0.74	0.08	2.00	2.43	4.30	5.74	1.44	0.32	1.25	4.14
	I_{AND}	0.06	0.02	0.25	0.90	0.79	1.60	0.28	0.33	0.72	2.09
	I_{PAND}	0.33	0.15	0.58	1.16	1.23	2.24	0.31	0.35	0.97	2.61
	I_{PEN}	0.24	0.17	0.73	1.39	2.13	2.41	0.88	0.54	2.06	5.56
	I_{RAN}	0.05	0.66	1.08	2.11	3.49	3.32	1.13	2.70	4.00	6.93
$Time$ (s)	I_{PART}	0.31	0.70	2.49	5.36	18.10	20.71	0.80	2.10	8.37	57.11
	I_{AND}	0.39	1.43	8.27	20.04	37.5	55.82	2.36	8.03	30.10	74.53
	I_{PAND}	0.39	1.20	5.18	11.59	31.33	36.03	1.91	5.53	19.85	65.43
	I_{PEN}	0.80	2.59	8.54	22.22	48.93	64.37	3.68	12.51	47.93	116.38
	I_{RAN}	1.30	7.24	16.42	32.62	58.63	87.95	5.94	22.88	63.64	140.95
$Dist.$	I_{PART}	0	0	0	0	0	0	0	0	0	0
	I_{AND}	0.09	0.11	0.11	0.11	0.10	0.13	0.15	0.14	0.11	0.08
	I_{PAND}	0.02	0.03	0.03	0.03	0.03	0.03	0.08	0.05	0.02	0.03
	I_{PEN}	0.02	0.02	0.03	0.03	0.02	0.04	0.11	0.06	0.06	0.08
	I_{RAN}	0.23	0.29	0.32	0.3	0.35	0.32	0.43	0.43	0.4	0.39

Table 3: Integrator results with correlated choice of parents

From Tables 2 and 3 and specifically the integrators I_{AND} and I_{PAND}, we deduce

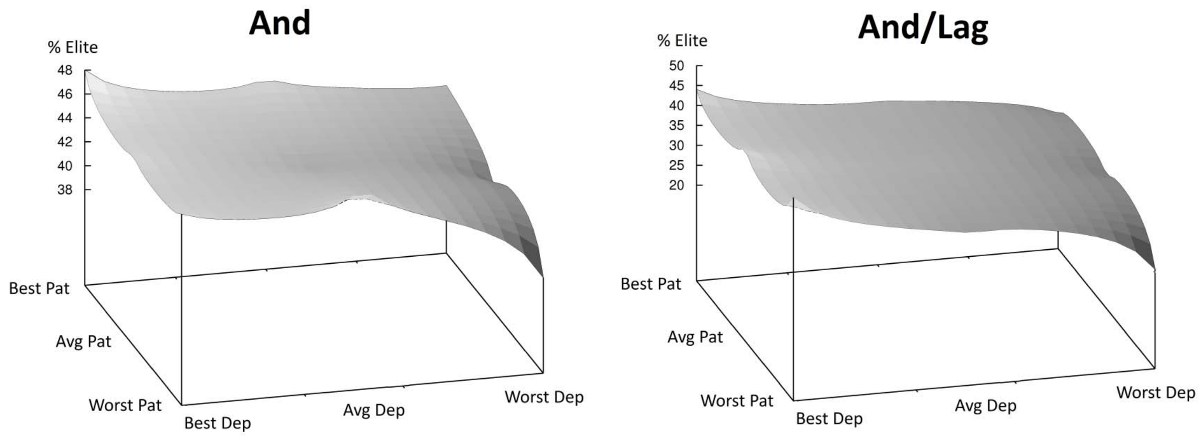

Figure 1: Selection of parents

that neither of the two parent-selection strategies is dominant in terms of the average gap. However, we observe that selecting parents that are close to each other leads to solutions that are close to the parents in a relatively short integration time. This finding is consistent, since starting with correlated parents greatly reduces the search space of I_{AND} and I_{PAND}. Hence, their computing times are short, and the solutions will have more attributes in common with the parents and therefore they will be closer to the parents.

We also studied the impact of parent quality on the complete solution obtained by integration (see Figure 1). We classified the solutions according to the following criteria: a high-quality solution is among the best 25% of the solutions; an average solution is among the next 25%; and the remaining solutions are low quality. Figure 1 shows a surface giving the percentage of elite complete solutions resulting from integration in terms of the quality of the parent solutions (horizontal axes) for MDVRP and PVRP. For both I_{AND} and I_{PAND}, the complete solution is strongly related to the quality of the parent solution associated with the MDVRP, since the percentage of elite solutions decreases rapidly as the MDVRP parent quality decreases, while the slope of the curve along the axis that represents the PVRP solution quality is almost zero. This is because choosing a bad pattern for a client may lead to a poor complete solution, but this can be partially corrected by making a good choice of the remaining attributes. However, a poor choice of the depot generally leads to poor solutions with long routes.

To demonstrate the benefits of cooperation in the ICS context, we used four processors for the integration. We tested several combinations of integrators (with each integrator running on one processor). We ran each scenario (combination) for 30 minutes and 10 times for each MDPVRP instance. We compared the solutions in terms of the average deviation from the BKS. By analyzing Table 4, we found that two combinations of integrators ($4I_{PEN}$ and $2I_{AND} + 2I_{PEN}$) give good results. I_{AND} and I_{PAND} outperform

Time (min)	Integrator combination	Instance										Avg. Dev.
		01	02	03	04	05	06	07	08	09	10	
30	$4I_{PART}$	0	1.67	1.07	2.17	8.21	4.78	1.00	4.77	4.82	10.37	3.89
	$4I_{AND}$	0	2.67	1.16	3.34	4.05	5.56	0.97	1.49	0.45	3.36	2.30
	$4I_{PAND}$	0	1.97	1.27	1.81	3.12	4.40	0.55	5.20	1.47	4.73	2.45
	$4I_{PEN}$	0	0	0.05	0.32	0.72	0.71	0	0	0.44	2.49	0.47
	$2I_{AND} + 2I_{PEN}$	0	0	0	0.34	0.90	0.57	0	0	0.47	1.79	0.40
	$I_{PART} + I_{AND} + I_{PAND} + I_{PEN}$	0	0	0.02	0.41	1.04	0.55	0	0	0.43	2.39	0.48
15	$4I_{PART}$	0	2.32	1.34	3.36	9.00	10.41	1.00	6.75	5.45	11.07	5.07
	$4I_{AND}$	0	2.67	1.55	4.25	5.02	8.84	0.97	2.95	0.75	4.23	3.12
	$4I_{PAND}$	0	2.48	1.36	3.09	3.92	8.06	0.55	5.57	1.83	5.75	3.26
	$4I_{PEN}$	0	0	0.08	0.43	1.10	0.93	0	0	0.63	3.30	0.65
	$2I_{AND} + 2I_{PEN}$	0	0	0.02	0.41	1.14	0.87	0	0	0.79	2.68	0.59
	$I_{PART} + I_{AND} + I_{PAND} + I_{PEN}$	0	0	0.05	0.54	1.17	1.11	0	0	0.67	3.18	0.72
10	$4I_{PART}$	0	3.13	1.57	4.75	11.54	9.52	1.00	7.62	6.31	11.87	5.73
	$4I_{AND}$	0	2.67	1.73	5.49	5.45	11.00	0.97	4.33	1.06	4.75	3.74
	$4I_{PAND}$	0	2.48	1.41	3.67	4.42	10.15	0.55	5.59	2.19	6.19	3.66
	$4I_{PEN}$	0	0.02	0.12	0.57	1.47	1.11	0	0	0.72	3.88	0.79
	$2I_{AND} + 2I_{PEN}$	0	0.29	0.05	0.44	1.39	1.54	0	0	0.99	3.81	0.82
	$I_{PART} + I_{AND} + I_{PAND} + I_{PEN}$	0	0.66	0.10	0.62	2.03	1.60	0	0	0.85	3.89	0.91
5	$4I_{PART}$	0	4.11	3.19	6.95	14.60	false	1.00	8.81	8.28	12.72	6.63
	$4I_{AND}$	0	3.35	1.86	7.51	7.33	13.32	1.10	6.87	2.61	6.17	5.01
	$4I_{PAND}$	0	2.51	1.62	4.71	6.13	11.52	0.61	7.73	4.27	7.09	4.62
	$4I_{PEN}$	0	0	0.20	0.77	2.06	1.79	0	0.03	1.55	4.36	1.08
	$2I_{AND} + 2I_{PEN}$	0	0	0.28	0.79	2.86	2.60	0	0.11	1.73	5.18	1.35
	$I_{PART} + I_{AND} + I_{PAND} + I_{PEN}$	0	0	0.60	1.05	3.37	2.40	0	0.38	1.43	5.79	1.50

Table 4: Deviation from BKS for different integrator combinations

I_{PART} and I_{PEN} in terms of individual performance, but this behavior is not maintained in a cooperative search context. In fact, I_{AND} and I_{PAND} could not be distinguished in the cooperative search context (see Table 4), especially when we used combinations of four identical integrators. When we compared the individual performance of the integrators, good quality parents were selected. Thus, selecting common attributes is generally a good strategy since these attributes are associated with very good solutions. In the context of cooperative search, we notice that pools of partial elite solutions contain a high percentage of identical solutions. Thus, most of the integrators that we have developed deal with almost identical solutions. In this situation the search space is quite limited. In the case of I_{PEN}, which encourages the appearance of parent attributes, the search space is not limited by the fixing strategy, so a larger neighborhood is explored; this explains I_{PEN}'s performance.

To conclude, we believe that the success of I_{AND} and I_{PEN} in the context of ICS is mainly due to their complementarity, since I_{AND} allows search intensification in a region of good solutions while I_{PEN} allows search diversification (critical characteristics are encouraged but not fixed). As we did for the individual integrators with correlated parents, we performed a Friedman test to compare the different combinations of integrators in ICS, measuring the solution quality (% gaps) obtained for 10 instances. Figure 3 confirms our analysis based on Table 4: the combinations $4I_{PEN}$, $2I_{AND} + 2I_{PEN}$, and $I_{PART} + I_{AND} + I_{PAND} + I_{PEN}$ are significantly better than $4I_{PART}$, $4I_{AND}$, and $4I_{PAND}$ with high confidence ($p < 10^{-5}$). However, it is impossible to conclude from the statistical test which combination dominates.

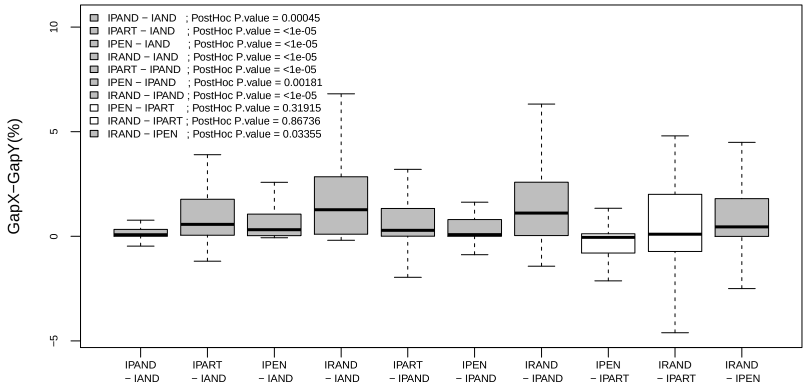

Figure 2: Boxplots of paired differences in solution quality for different integrators when parents are correlated

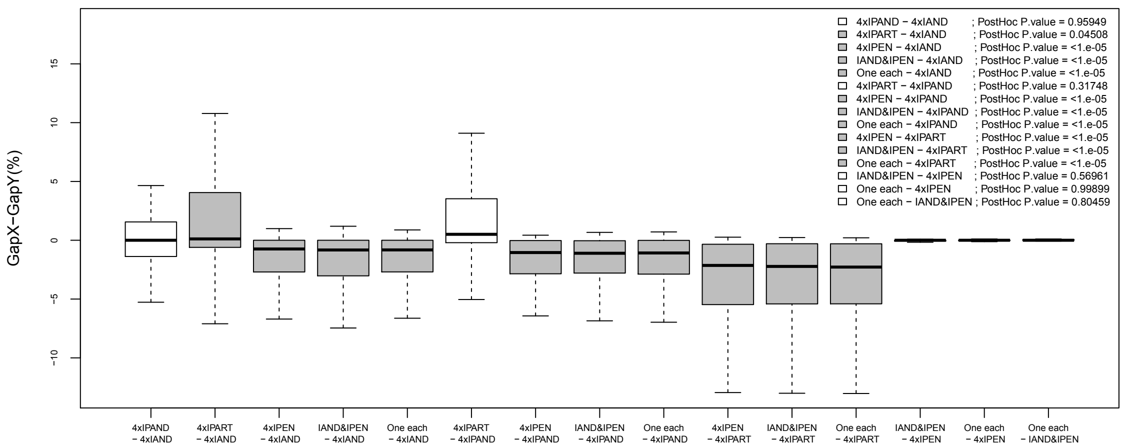

Figure 3: Boxplots of paired differences in solution quality for different combinations of integrators in the context of ICS

6 Conclusion

We have presented a general approach to deal with integration. Four different strategies have been used to build integrators, and two of them have given good results. The first fixes only the critical variables that have a consensus among the parent solutions (input information) in the integration subproblem. The second in addition encourages the remaining critical characteristics to appear in the integrated solution by following a nearly-Lagrangian relaxation procedure. In the ICS context, we have investigated the use of several combinations of integrators. The results indicate that two combinations ($4I_{PEN}$ and $2I_{AND} + 2I_{PEN}$) give better results in terms of the average deviation from the BKS of ICS.

Future research directions will involve developing exact methods for integration. We believe that exact integrators may improve the quality of the complete solutions or guarantee the solution quality. It might be possible to use exact integration methods to provide robust guidance in the context of ICS.

Acknowledgments

While working on this project, T.G. Crainic was the NSERC Industrial Research Chair in Logistics Management, ESG UQAM, N. El hachemi and N. Lahrichi were postdoctoral fellows with the Chair, and M. Gendreau was the NSERC/Hydro-Québec Industrial Research Chair on the Stochastic Optimization of Electricity Generation, MAGI, École Polytechnique. Partial funding for this project has been provided by the Natural Sciences and Engineering Council of Canada (NSERC), through its Industrial Research Chair, Collaborative Research and Development, 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 through its Team Research Project program.

References

[1] Benders, J.F. Partitioning procedures for solving mixed-variables programming problems. *Numer. Math.* 4, 238–252, 1962.

[2] Cordeau, J.F., G. Laporte, 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.

[3] Crainic, T.G. Parallel computation, co-operation, tabu search. In C. Rego, and B. Alidaee (Eds.). *Metaheuristic Optimization Via Memory and Evolution: Tabu Search and Scatter Search*. Kluwer Academic Publishers, Norwell, MA, 283–302, 2005.

[4] Crainic, T.G., G. Crisan, M. Gendreau, N. Lahrichi, W. Rei. Multi-thread integrative cooperative optimization for rich combinatorial problems. In *IPDPS '09: Proceedings of the 2009 IEEE International Symposium on Parallel & Distributed Processing*, pp. 1–8, 2009.

[5] Crainic, T.G., M. Toulouse. Parallel meta-heuristics. In M. Gendreau and J.-Y. Potvin (Eds.). *Handbook of Metaheuristics (2nd Edition)*. Springer, 497–541, 2010.

[6] Dantzig, G.B., P. Wolfe. Decomposition principle for linear programs. *Operations Research*, 8, 101–111, 1960.

[7] Glover, F., M. Laguna, M. Marti. Fundamentals of scatter search and path relinking. *Control and Cybernetics*, 39(3), 653–684, 2000.

%newblock*Journal of the Operational Research Society*, 49(12), 1239–1248, 1998.

%newblock*European Journal of Operational Research*, 175(3), 2006.

[8] Holland, J.H. *Adaptation in Natural and Artificial Systems*. The University of Michigan Press, Ann Arbor, MI, 1975.

[9] Lahrichi, N., T.G. Crainic, M. Gendreau, W. Rei, G.C. Crisan, T. Vidal. An integrative cooperative search framework for multi-decision-attribute combinatorial optimization. *Publication CIRRELT-2012-42, Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport, Université de Montréal, Montreal, QC, Canada, 2012.*

[10] Le Bouthillier, A., T.G. Crainic. A cooperative parallel meta-heuristic for the vehicle routing problem with time windows. *Computers & Operations Research* 32(7), 1685–1708, 2005.

[11] Le Bouthillier, A., T.G. Crainic, P. Kropf. A guided cooperative search for the vehicle routing problem with time windows. *IEEE Intelligent Systems* 20(4), 36–42, 2005.

[12] Reves, C.R. Genetic algorithms. In M. Gendreau and J.-Y. Potvin (Eds.). *Handbook of Metaheuristics* (2nd Edition). Springer, 109–140, 2010.

[13] Taillard, E.D., L.M. Gambardella, M. Gendreau, J.Y. Potvin. Adaptative memory programming: A unified view of metaheuristics. *European Journal of Operational Research*, 135, 1–16, 2001.

[14] Vidal, T., T.G. Crainic, M. Gendreau, N. Lahrichi, and W. Rei. A hybrid genetic algorithm for multi-depot and periodic vehicle routing problems. *Operations Research* 60(3), 611–624, 2012.

Appendix A: MDPVRP Formulation

$$\text{Minimize} \quad \sum_{v_i \in \mathcal{V}} \sum_{v_j \in \mathcal{V}} \sum_{k=1}^{m} \sum_{l=1}^{t} \sum_{v_o \in \mathcal{V}^{\text{DEP}}} c_{ij} x_{ijklo} \tag{22}$$

$$\text{subject to:} \quad \sum_{p \in L_i} y_{ip} = 1 \qquad v_i \in \mathcal{V}^{\text{CST}} \tag{23}$$

$$\sum_{v_o \in \mathcal{V}^{\text{DEP}}} y'_{io} = 1 \qquad v_i \in \mathcal{V}^{\text{CST}} \tag{24}$$

$$\sum_{v_j \in \mathcal{V}} \sum_{k=1}^{m} x_{ijklo} - \sum_{p \in L_i} a_{pl} y_{ip} = 0 \qquad v_i \in \mathcal{V}^{\text{CST}} \; ; \; v_o \in \mathcal{V}^{\text{DEP}} \; ; \; l = 1 \ldots t \tag{25}$$

$$\sum_{v_j \in \mathcal{V}} \sum_{k=1}^{m} x_{ijklo} \leq y'_{io} \qquad v_i \in \mathcal{V}^{\text{CST}} \; ; \; v_o \in \mathcal{V}^{\text{DEP}} \; ; \; l = 1 \ldots t \tag{26}$$

$$\sum_{v_i \in \mathcal{V}} \sum_{k=1}^{m} x_{ijklo} \leq y'_{jo} \qquad v_j \in \mathcal{V}^{\text{CST}} \; ; \; v_o \in \mathcal{V}^{\text{DEP}} \; ; \; l = 1 \ldots t \tag{27}$$

$$\sum_{v_j \in \mathcal{V}} x_{ojklo} \leq 1 \qquad v_o \in \mathcal{V}^{\text{DEP}} \; ; \; k = 1 \ldots m \; ; \; l = 1 \ldots t \tag{28}$$

$$\sum_{v_j \in \mathcal{V}} x_{ijklo} = 0 \qquad v_i \in \mathcal{V}^{\text{DEP}} \; ; \; v_o \in \mathcal{V}^{\text{DEP}} \; ; \; v_o \neq v_i \; ; \; k = 1 \ldots m \; ; \; l = 1 \ldots t \tag{29}$$

$$\sum_{v_i \in \mathcal{V}} x_{ijklo} = 0 \qquad v_j \in \mathcal{V}^{\text{DEP}} \; ; \; v_o \in \mathcal{V}^{\text{DEP}} \; ; \; v_o \neq v_j \; ; \; k = 1 \ldots m \; ; \; l = 1 \ldots t \tag{30}$$

$$\sum_{v_j \in \mathcal{V}} x_{jiklo} - \sum_{v_j \in \mathcal{V}} x_{ijklo} = 0 \qquad v_i \in \mathcal{V} \; ; \; v_o \in \mathcal{V}^{\text{DEP}} \; ; \; k = 1 \ldots m \; ; \; l = 1 \ldots t \tag{31}$$

$$\sum_{v_i \in \mathcal{V}} \sum_{v_j \in \mathcal{V}} q_i x_{ijklo} \leq Q \qquad v_o \in \mathcal{V}^{\text{DEP}} \; ; \; k = 1 \ldots m \; ; \; l = 1 \ldots t \tag{32}$$

$$\sum_{v_i \in \mathcal{V}} \sum_{v_j \in \mathcal{V}} (c_{ij} + \tau_i) x_{ijklo} \leq T \qquad v_o \in \mathcal{V}^{\text{DEP}} \; ; \; k = 1 \ldots m \; ; \; l = 1 \ldots t \tag{33}$$

$$\sum_{v_i \in S} \sum_{v_j \in S} x_{ijklo} \leq |S| - 1 \quad S \in \mathcal{V}^{\text{CST}} \; ; \; |S| \geq 2 \; ; \; v_o \in \mathcal{V}^{\text{DEP}} \; ; \; k = 1 \ldots m \; ; \; l = 1 \ldots t \tag{34}$$

$$x_{ijklo} \in \{0,1\} \quad v_i \in \mathcal{V}; \; v_j \in \mathcal{V} \; ; \; v_o \in \mathcal{V}^{\text{DEP}} \; ; \; k = 1 \ldots m \; ; \; l = 1 \ldots t \tag{35}$$

$$y'_{io} \in \{0,1\} \quad v_i \in \mathcal{V} \; ; v_o \in \mathcal{V}^{\text{DEP}} \tag{36}$$

$$y_{ip} \in \{0,1\} \quad v_i \in \mathcal{V} \; ; p \in L_i \tag{37}$$

Constraints (23), (24), and (25) ensure that customer i is assigned to one depot and has only one day-combination. Constraints (26) and (27) force customers i and j to be served by a route associated with depot v_o if and only if they are assigned to depot v_o. Constraints (28) force the solution to contain at most m routes on each day. Constraints (29) and (30) ensure that each route can be associated with only one depot. Constraints (31) are flow conservation constraints for each customer i and each day l. Constraints (32) ensure that the capacity of each vehicle k for each day l is satisfied. Constraints (33) ensure that the maximum duration of each route associated with vehicle k on day l is limited to T. Finally, Constraints (34) eliminate subtours.