Chapter 19
Combining Exact Methods to Construct Effective Hybrid Approaches to Vehicle Routing

Rym M'Hallah

Contents

19.1 Introduction ... 435
19.2 Literature Review ... 436
19.3 Problem Description and Formulation ... 438
 19.3.1 Problem Definition ... 438
 19.3.2 Mixed Integer Program ... 438
 19.3.3 Constraint Program ... 441
 19.3.4 Enhancing MIP and CP ... 442
19.4 Solution Approach ... 442
 19.4.1 Stage 1 ... 443
 19.4.2 Stage 2 ... 448
19.5 Results ... 448
19.6 Practical Considerations ... 451
 19.6.1 Single Versus Multiple Depots ... 451
 19.6.2 Distances and Travel Times ... 451
 19.6.3 Time Windows ... 452
 19.6.4 Collection Versus Distribution ... 452
 19.6.5 Packing Constraints ... 452
 19.6.6 Stochastic Travel Times and Time Windows ... 453
19.7 Conclusions ... 454
References ... 454

19.1 Introduction

Globalization has fostered a prosperous climate for industrial growth. However, this growth is coupled with new challenges: a fiercer competition, broadened distribution networks, diversified supply chains, tighter profit margins, and more exigent clients. These latter are demanding lower prices, requiring higher quality levels, and imposing stricter deadlines. Consequently, the manufacturing, service,

R. M'Hallah (✉)
Department of Statistics and Operations Research, College of Science, Kuwait University,
Kuwait City, Kuwait
e-mail: rym.mhallah@ku.edu.kw

A. E. Smith (ed.), *Women in Industrial and Systems Engineering*,
Women in Engineering and Science, https://doi.org/10.1007/978-3-030-11866-2_19

and transportation industries that are affected by this new order have to manage their activities judiciously while dealing with more complex, larger-scale, real-life systems. Managing these activities is equivalent to solving several inter-related industrial engineering problems. Because of the new global environment, industries can no longer address some abstract form of these problems or decompose them into independent subproblems. They have to model the problems as closely as possible to their true real-life context. Such models are important for three reasons. First, they offer a better reflection of the problem. Second, they account for more constraints, take into account the true nature of the variables, and build in intricate interdependencies. Third and last, they enhance the chances of direct real-life implementation of the resulting solutions without any modification and at no supplementary cost.

The optimization of the new complex problems that have emerged is difficult not only because of their compounded nature but most importantly because their components are themselves difficult to solve. This chapter considers one such problem: vehicle routing with multiple time windows (VRPMTW), a problem of substantial impact. Its good management alleviates environmental concerns caused by pollution, reduces costs, and offers better work conditions for drivers and higher satisfaction for clients. In VRPMTW, every client specifies more than one availability period for receiving delivery.

Unlike existing methods, which consider the routing and scheduling aspects of VRPMTW independently, this chapter addresses the problem as a single entity. It approximately solves VRPMTW in reasonable run times using an efficient search technique that takes advantage of the advances of computing technologies. Larger computer memories and faster processors allow the (near-)exact resolution of more realistic problems, an unimaginable phenomenon few years ago.

The proposed approach uses exact techniques as approximate ones. It substitutes difficult constraints with easier ones, and limits the search space to neighborhoods that contain the optimum. Most importantly, it explores the respective strengths of mixed integer programming, constraint programming, and search methods.

Section 19.2 reviews the literature. Section 19.3 defines and models the problem. Section 19.4 details the proposed approach. Section 19.5 presents some results. Section 19.6 discusses some practical considerations. Finally, Sect. 19.7 summarizes this research and presents future extensions.

19.2 Literature Review

Because they occur in most services and industries, vehicle routing problems (VRP) are continuously drawing the attention of researchers (Vidal et al. 2014). This is clearly reflected by the extensive literature and numerous review papers (Adewumi and Adeleke 2018; Braekers et al. 2016; Eksioglu et al. 2009, Gendreau et al. 2008, Koç et al. 2016; Vidal et al. 2013). These surveys classify the literature according to the problem constraints, mathematical models, and solution techniques. Most of them (e.g., Baldacci et al. 2012) single out cases where loads (Alba and Dorronsoro 2008; Gendreau et al. 1994; Minocha and Tripathi 2013; Nagata and Bräysy 2009; Toth and Vigo 2002) and delivery times (Cordeau et al. 2001; Favaretto et al. 2007; Fu et al. 2007; Figliozzi 2010; Moccia et al. 2012; Nagata et al. 2010; Nazif and Lee 2012; Solomon 1987) are constrained. These constraints fathom large parts of the search space and thus eliminate a large number of potential nodes of any tree-based procedure. However, they make finding feasible solutions much harder.

Hashimoto et al. (2006) and Ibaraki et al. (2005), to cite a few, consider VRP with general time windows (TW). They associate to each client more than one delivery window, but penalize the violation of these soft or flexible TWs, with penalties that are not necessarily linear. Beheshti et al. (2015) consider a variant where the multiple soft TWs are not specified by the clients but by the delivery company. Clients rank the TWs in decreasing order of their preference. This variant, which applies in rapid post deliveries, does not guarantee the suitability of the alternatives with the clients' constraints.

Belhaiza et al. (2013) address VRPMTW via a tabu search variable neighborhood search (TSVNS) that minimizes the weighted sum of the trip duration and the penalties associated to the TWs and capacity constraints. Belhaiza and M'Hallah (2016) consider the multiple objective VRPMTW where the criteria are: drivers' utility, total cost, and costumers' utility. They speed their search by fathoming all dominated solutions and investigate the space of solutions that satisfy the Nash equilibrium conditions of a non-cooperative multiple-agent game. Finally, Belhaiza et al. (2017) extend the single objective VRPMTW to the multiple-depot heterogeneous fleet case with the objective of minimizing either the total travel time or the total traveled duration. They address the problem using a multiple-start hybrid genetic variable neighborhood search.

Unlike the techniques surveyed by Archetti and Speranza (2014), the proposed method combines random search with mathematical/constraint programming. This line of research is successfully applied in cutting and packing where the constraints are non-linear (M'Hallah et al. 2013; Al-Mudhahka et al. 2011) and in timetabling where the problem size is large and the planning horizon is long (M'Hallah and Alkhabbaz 2013). Similarly, it is applied in scheduling where a steepest descent heuristic (Laalaoui and M'Hallah 2016) calls iteratively a mixed integer program to insert a job on a machine (M'Hallah 2007; M'Hallah and Al-Khamis 2012) or to enhance an existing solution and assess its optimality gap (M'Hallah 2014) or to choose jobs that will be exchanged among agents (Polyakovskiy and M'Hallah 2014). Recently, it is extended to bin packing where mixed integer/constraint programs apply a look-ahead strategy that reserves areas in the bins for unpacked items (Polyakoskiy and M'Hallah 2018). In all aforementioned applications, the models are augmented with feasibility constraints and with bounds on the objective function values forcing the solvers to only investigate feasible improving directions. They are subject to runtime limits because in most cases, the incumbent is identified at an early stage of the search while most of the computational effort is devoted to proving optimality. Finally, they are fed with partial or complete initial solutions.

19.3 Problem Description and Formulation

Section 19.3.1 defines VRPMTW. Sections 19.3.2 and 19.3.3 model it as a mixed integer and as a constraint program. Finally, Sect. 19.3.4 discusses ways to enhance the performance of both models.

19.3.1 Problem Definition

Consider a set $I = \{1, \ \dots \ , n\}$ of n clients that are served, from a single depot d, by a set $K = \{1, \ \dots \ , m\}$ of m vehicles. The depot d, denoted also as client 0, is located in position $\left(o_0^h, o_0^v\right)$ of the Cartesian coordinate system. It is open during the time interval $[\ell_0, u_0]$; i.e., no vehicle leaves d prior to ℓ_0 or returns to d later than u_0. A vehicle may wait at the depot or at a client's site. Let $\delta_{i,j}$ denote the travel times between i and j, $i \in I^*, j \in I^*$, $i \neq j$, where $I^* = I \cup \{0\}$.

Client i, $i \in I$, is characterized by its Cartesian coordinate system's position $\left(o_i^h, o_i^v\right)$, positive demand q_i delivered by a single vehicle, known positive service time s_i, and ordered set $W_i = \left\{w_{i,1}, \dots, w_{i,\overline{p}_i}\right\}$ of $\overline{p}_i$, $\overline{p}_i \in \mathbb{N}^*$, non-overlapping availability TWs. TW p, $p = 1, \dots, \overline{p}_i$, of client i, $i \in I$, is $w_{i,p} = [\ell_{i,p}, u_{i,p}]$, with $0 < \ell_{i,1} < u_{i,1} < \cdots < \ell_{i,p} < u_{i,p} < \cdots < \ell_{i,\overline{p}_i} < u_{i,\overline{p}_i} < \infty$.

Let $I_k \subset I$ be the ordered subset of clients assigned to k. The subsets are mutually exclusive; i.e., $\cup_{k \in K} I_k = I$, $\cup_{k \in K} I_k = \varnothing$. Thus, partial delivery is prohibited. Vehicle k, $k \in K$, has a limited capacity $Q_k > 0$. Its time out of d, including its waiting time, should not exceed $\overline{D}_k > 0$. Its travel duration D_k is the sum of its travel and service time. It excludes any waiting time.

19.3.2 Mixed Integer Program

Herein, VRPMTW assigns the clients to the vehicles and schedules their respective deliveries with the objective of minimizing the total travel time over all vehicles. It can be formulated as a mixed integer program. The model is inspired from the mixed integer program (MIP) of Belhaiza et al. (2013). It uses three types of positive decision variables and three types of binary variables.

The first positive variable $\omega_{i,k}$ denotes the waiting time of vehicle k, $k \in K$, at client i, $i \in I^*$. A vehicle may have to wait prior to starting delivery to a specific client and may choose to wait at d in order to reduce its total time out of d. The second positive variable $t_{i,k}$ indicates the time k, $k \in K$, reaches i, $i \in I^*$. Consequently, k starts delivery to i, $i \in I$, at $t_{i,k} + \omega_{i,k}$. Finally, the third positive variable $c_{i,k}$ is the departure time of k, $k \in K$, from i, $i \in I^*$. It follows that $c_{0,k}$ is the departure time of k from d and $t_{0,k}$ its return time to d.

The first binary variable $z_{i,k} = 1$ if client i, $i \in I$, is assigned to vehicle k, $k \in K$, and 0 otherwise. The second binary variable $x_{i,j,k} = 1$ if i, $i \in I^*$, immediately precedes j, $j \in I^*$, $j \neq i$, and both j and i are served by vehicle k, $k \in K$, and 0 otherwise. Finally, the binary variable $v_{i,p} = 1$ if client i, $i \in I$, is served during $w_{i,p}$, $p = 1, \ldots, \overline{p}_i$, and 0 otherwise.

Then, the VRPMTW using the above six types of decision variables follows.

$$\text{Min} \sum_{k \in K} \sum_{(i,j) \in I^* \times I^*} \delta_{i,j} x_{i,j,k} \tag{19.1}$$

$$s.t. \quad \sum_{k \in K} z_{i,k} = 1, \quad i \in I \tag{19.2}$$

$$\sum_{j \in I^*} x_{i,j,k} - \sum_{j \in I^*} x_{j,i,k} = 0, \quad i \in I^*, k \in K \tag{19.3}$$

$$2x_{i,j,k} - z_{i,k} - z_{j,k} \leq 0, \quad i \in I, j \in I, i \neq j, k \in K \tag{19.4}$$

$$\sum_{k \in K} \sum_{i \in I^*} x_{i,j,k} = 1, \quad j \in I \tag{19.5}$$

$$\sum_{k \in K} \sum_{j \in I^*} x_{i,j,k} = 1, \quad i \in I \tag{19.6}$$

$$\sum_{i \in I} q_i z_{i,k} = Q_k, \quad k \in K \tag{19.7}$$

$$c_{i,k} - t_{i,k} - \omega_{i,k} - s_i + M\left(1 - z_{i,k}\right) \geq 0, \quad i \in I, k \in K \tag{19.8}$$

$$t_{j,k} - c_{i,k} - \delta_{i,j} - M\left(1 - x_{i,j,k}\right) \leq 0, \quad i \in I^*, j \in I^*, i \neq j, k \in K \tag{19.9}$$

$$t_{j,k} - c_{i,k} - \delta_{i,j} + M\left(1 - x_{i,j,k}\right) \geq 0, \quad i \in I^*, j \in I^*, i \neq j, k \in K \tag{19.9$'$}$$

$$t_{i,k} + \omega_{i,k} - \ell_{i,p} + M\left(1 - z_{i,k}\right) + M\left(1 - v_{i,p}\right) \geq 0, \quad i \in I, p \in \left\{1, \ldots, \overline{p}_i\right\}, k \in K \tag{19.10}$$

$$t_{i,k} + \omega_{i,k} - u_{i,p} - M\left(1 - z_{i,k}\right) - M\left(1 - v_{i,p}\right) \leq 0, \quad i \in I, p \in \left\{1, \ldots, \overline{p}_i\right\}, k \in K \tag{19.11}$$

$$\sum_{p=1}^{\overline{p}_i} v_{i,p} = 1, \quad i \in I \tag{19.12}$$

$$c_{0,k} \geq \ell_0, \quad k \in K \tag{19.13}$$

$$t_{0,k} \leq u_0, \quad k \in K \tag{19.14}$$

$$t_{0,k} - c_{0,k} \leq \overline{D}_k, \quad k \in K \tag{19.15}$$

$$\omega_{i,k} \geq 0, z_{i,k} \in \{0, 1\}, \quad i \in I, k \in K \tag{19.16}$$

$$c_{i,k} \geq 0, t_{i,k} \geq 0, \quad i \in I^*, k \in K \tag{19.17}$$

$$v_{i,p} \in \{0, 1\}, \quad i \in I, p \in \left\{1, \ldots, \overline{p}_i\right\} \tag{19.18}$$

$$x_{i,j,k} \in \{0, 1\}, \quad i \in I^*, j \in I^*, i \neq j, k \in K \tag{19.19}$$

where M is a large positive number. Equation (19.1) defines the objective, which minimizes the total travel time. Constraint (19.2) ensures that each client is assigned to a single vehicle. Constraint (19.3) is the classical flow conservation. Constraint (19.4) guarantees that client i may immediately precede client j if and only if both are assigned to the same vehicle k. Constraints (19.5) and (19.6) ensure that each client i is visited once: i has exactly one immediate predecessor and one immediate successor. Constraint (19.7) guarantees that the quantity delivered by a vehicle does not exceed the vehicle's capacity. Constraint (19.8) defines the departure time of vehicle k from client i as the sum of its arrival, service, and waiting time. This equation holds when i is assigned to k, and is redundant otherwise. Constraints (19.9) and (19.9′) compute the arrival time of k to client j as the sum of the departure time of k from the immediately preceding client i and the travel time between i and j. Constraints (19.10) and (19.11) impose that vehicle k starts delivery to client i during TW p of i if and only if i is assigned to k and is served during p. They are redundant otherwise. Constraint (19.12) imposes that delivery to client i occurs during exactly one of the TWs of i. Constraints (19.13) and (19.14) impose that a vehicle k leaves and returns to d during the open time of d, whereas constraint (19.15) limits the time of k out of the depot to the maximal permissible threshold. Finally, constraints (19.16)–(19.19) define the variable types. Solving the above MIP is a challenge.

CPLEX fails to identify feasible solutions for instances with more than 20 clients when allocated 3 h of runtime.

19.3.3 Constraint Program

Alternatively, VRPMTW can be modeled as a constraint program. Let $\zeta_{i,k}$ be an optional interval variable indicating the activity of visiting client i using vehicle k. Let the interval variable h_i represent the activity of visiting client i. In addition, let $\boldsymbol{x_k}$ denote the route of k; that is, $\boldsymbol{x_k}$ is the sequence of interval variables $\zeta_{i,k}$. Then, the CP model follows.

$$\text{Min} \sum_{k\in K}\sum_{i\in I^*} \delta_{ij} : \text{next}\left(\boldsymbol{x}_k, i\right) = j \tag{19.20}$$

$$\text{s.t.}\ \ u_{i,1} \geq \text{startOf}\left(h_i\right) \geq \ell_{i,1} \,\|\ldots\, \| u_{i,\overline{p}_i} \geq \text{startOf}\left(h_i\right) \geq \ell_{i,\overline{p}_i},\quad i\in I \tag{19.21}$$

$$\text{startOf}\left(\zeta_{0,k}\right) \geq l_0 \&\& u_0 \geq \text{startOf}\left(\zeta_{n+1,k}\right),\quad k\in K \tag{19.22}$$

$$\text{noOverlap}\left(\boldsymbol{x_k}, \boldsymbol{\delta}\right),\quad k\in K \tag{19.23}$$

$$\text{first}\left(\boldsymbol{x_k}, \zeta_{0,k}\right),\quad k\in K \tag{19.24}$$

$$\text{last}\left(\boldsymbol{x_k}, \zeta_{n+1,k}\right),\quad k\in K \tag{19.25}$$

$$\sum_{i\in I} q_i\ \ \text{presenceOf}\left(b_i^k\right) \leq Q_k,\quad k\in K \tag{19.26}$$

$$\text{endOf}\left(\zeta_{n+1,k}\right) - \text{startOf}\left(\zeta_{0,k}\right) \leq \overline{D}_k\ \ k\in K \tag{19.27}$$

$$\text{alternative}\left(h_i, \zeta_{i,k}\right),\quad k\in K, i\in I \tag{19.28}$$

where || denotes the exclusive "or" operator, && the "and" operator, $\boldsymbol{\delta}$ the travel matrix expressed in time units, and $n+1$ the depot d. Equation (19.20) minimizes the total travel time. Constraint (19.21) ensures that service to client i starts within exactly one of the TWs of i while constraint (19.22) guarantees that a vehicle leaves and returns to d when d is open. Equation (19.23) makes the subsets of clients served by different vehicles mutually exclusive. Equations (19.24) and (19.25) force each vehicle to start and end its tour at d. Equation (19.26) imposes that the load of a vehicle be less than or equal to the vehicle's capacity. Equation (19.27) limits the time out of d of vehicle k. Finally, Eq. (19.28) is the alternative constraint, which assigns every client to a single vehicle. CP obtains near-global optima for some benchmark instances, in particular, when the TWs are wide.

19.3.4 Enhancing MIP and CP

We use the above models as stand-alone heuristics by allocating fixed runtimes to their commercial solvers. To enhance their solvability, we couple them with some strategies.

- First, we reduce the size of the problem by imposing logical assumptions that truck drivers adopt when constructing their solutions. A client j can't immediately follow a client i on the route of vehicle k if the travel time δ_{ij} is too large. Both MIP and CP formulations are augmented by the constraint $x_{ijk} = 0$ if $\left|o_i^h - o_j^h\right| > \frac{\overline{o}^h}{m}$ or $\left|o_i^v - o_j^v\right| > \frac{\overline{o}^v}{m}$ or $\delta_{ij} > \overline{\delta}$, where $\overline{o}^h = \max_{(i,j)\in I^*\times I^*} \left\{o_i^h - o_j^h\right\}$, $\overline{o}^v = \max_{(i,j)\in I^*\times I^*} \left\{o_i^v - o_j^v\right\}$, and $\overline{\delta}$ is a maximal threshold time.
- Second, we start the solvers from several feasible solutions. The multitude of starting points guards against the stagnation in local optima that are far from the global optimum.
- Third, we iteratively fix clients to vehicles, deliveries to TWs, and solve the resulting problems. This avoids the need for disjunctive constraints and the sensitivity of solvers to the choice of M. At each iteration, we augment the model with the tightest known bounds. We obtain the lower bound by removing all TW-related constraints, thus reducing the problem to a series of single machine scheduling problems with sequence-dependent setup times and due date windows. The upper bound is updated every time a VRPMTW feasible solution is obtained.
- Fourth, we iteratively divide the clients and vehicles into two or more disjoint subsets, solve the corresponding subproblems, merge their solutions, and compute the total travel time.
- Fifth, we solve MIP with emphasis on constraint satisfaction rather than on optimization. This yields better quality solutions while tackling larger problems.

19.4 Solution Approach

To apply the above enhancements in a more systematic manner, we approximately solve VRPMTW via a two-stage approach. Stage 1 constructs a pool of initial solutions, where a solution assigns each client to a vehicle and sequences the clients of every vehicle. Stage 1 obtains solutions by one of nine constructive heuristics. These solutions are inspired from the way truck drivers construct their routes. They divide clients into sectors and choose the clients of a single vehicle from adjacent sectors. They construct the routes logically along the grid, a feature that enhances their chance of real-life implementation.

Stage 2 enhances the solutions of Stage 1 by perturbing the incumbent using a steepest descent neighborhood search. The search uses MIP to reassign clients to vehicles and CP to search for a feasible route of a vehicle. It embeds a look-ahead strategy that guards against infeasible/non-improving search directions. Sections 19.4.1 and 19.4.2 detail these two stages.

19.4.1 Stage 1

Because of the different distributions of the clients around the depot and of the varying number of vehicles and clients, it is difficult for a single heuristic to consistently generate good initial solutions. Therefore, we propose a panoply of constructive heuristics CH1, ..., CH9. They all impose a maximal travel time $\overline{\delta}$ between any $(i^{'}, i^{+})$ when appending location i^{+} to the current location $i^{'}$ of a vehicle. In addition to constraining the search space, this conforms to the way drivers plan their routes. Drivers prefer balanced routes; they would rather have equal stretches than a long haul followed/preceded by a series of small hauls. The constructive heuristics differ in the way they assign clients to trucks and how they route them. For example, CH1, CH7, and CH8 start from the closest client to the depot while CH2, ..., CH6 select the first client randomly.

Some of the constructive heuristics (e.g., CH3 and CH4) use the notion of a sector. When the number of vehicles is large, the grid is divided into a series of radial sectors, where sector k, $k = 1, \ldots, m$, is delimited by the angles $\frac{2(k-1)\pi}{m}$ and $\frac{2k\pi}{m}$, with sectors $m+1$ and $m+2$ coinciding with sectors $k = 1$ and $k = 2$, respectively. Client i, $i \in I$, is part of sector k if its polar coordinate $\theta_i = \tan^{-1}\left(\frac{o_i^v - o_0^v}{o_i^h - o_0^h}\right)$ falls in the sector of k. When the number of vehicles is small, the grid is split into m guillotine rectangular adjacent bands, with sector k corresponding to the band delimited by $o^h = \frac{(k-1)\left(\overline{o}^h - \underline{o}^h\right)}{m}$ and $o^h = \frac{k\left(\overline{o}^h - \underline{o}^h\right)}{m}$, where $\underline{o}^h = \min_{(i,j)\in I^*\times I^*}\left\{o_i^h - o_j^h\right\}$.

Let F denote the set of free clients. Unless differently stated, F is initially set to I. In addition, let $I_k = \{[1], \ldots, [n_k]\}$ denote the ordered set of $n_k{=}|I_k|$ clients of k where $I_k^* = I_k \cup \{d\}$ with $[n_k + 1] = [0] = d$ signaling the end and the beginning of the tour at the depot. Using these definitions, we detail below each of the nine constructive heuristics.

CH1 considers the vehicles sequentially starting with $k = 1$. For the current vehicle k, it initializes the vehicle's current capacity $Q_k^{\text{current}} = 0$ and the index of its current location $i' = d$. Iteratively, it assigns to k a free client $i^+ \in F$ such that

C1. i^+ is the closest client to i',
C2. $Q_k^{\text{current}} + q_{i^+} \leq Q_k$ —i.e., appending the demand of i^+ to the current load of k does not violate the load capacity of k,
C3. delivery to i^+ may start during one of the TWs of i^+, and upon completing delivery to i^+, vehicle k may reach the depot
C4. on-time (prior to u_0),
C5. without exceeding its own maximal time out of the depot $\overline{D}_k$.

Consequently, CH1 updates the vehicle's current load: $Q_k^{\text{current}} = Q_k^{\text{current}} + q_{i^+}$, current location: $i' = i^+$ and the set of free clients: $F = F \setminus \{i^+\}$. When CH1 fails to append any of the free clients to the current route of k, it considers the next vehicle: it sets $k = k + 1$.

Generally, CH1's total travel distance is smaller than those of all other constructive heuristics; however, it may induce large waiting times. It is greedy and myopic by nature.

CH2 assigns all clients of a given sector to a same vehicle. Then, for each vehicle k, it applies CP to search for a feasible route; i.e., it applies Eqs. (19.20)–(19.25) and (19.27) of CP for $K = \{k\}$. When CP fails to identify a feasible route, we relax Eq. (19.25).

CH2 identifies good quality solutions when the sectors concord with the geographical clusters. CH2 offers balanced routes with every vehicle having to eventually serve those far away clients of its sector.

CH3 mimics CH2 except that it considers the clients of a sector sequentially. It assigns each vehicle $k \in K$ to sector k. It then considers the vehicles sequentially starting with $k = 1$. For the current vehicle k, it initializes $Q_k^{\text{current}} = 0$ and the index of its current location $i' = d$. Iteratively, it assigns to vehicle k a free client i^+ from sector k such that $\delta_{i'i^+} < \overline{\delta}$, and constraints (C1)–(C5) hold. Consequently, it updates Q_k^{current}, i', and F.

When it fails to heuristically append any of the free clients of sector k to the current route, CH3 tries to insert a free client using an exact model. Let $f \in F$ denote the free client to be inserted in the route of k. Let the binary decision variable $\chi_{[i]} = 1$, $[i] \in I_k^*$, if client f is inserted immediately before client $[i]$ on the route of k, and 0 otherwise. Let the positive decision variables $t_{[i]}$, $\omega_{[i]}$, and $c_{[i]}$ denote, respectively, the time k reaches $[i]$, waits before delivery, and ends delivery to $[i]$, $[i] \in I_k$, and (c_0, t_0) the time k leaves and returns to d. Finally, let parameter $\Delta_{[i]} = \delta_{[i-1]f} + \delta_{f[i]} - \delta_{[i-1][i]}$ denote the net travel time change caused by inserting client f immediately before client $[i]$. In fact, k no longer travels between $[i-1]$ and $[i]$. Instead, it travels from $[i-1]$ to f to $[i]$. Using the above parameters and decision variables, MIP_f tries to insert iteratively every free client $f, f \in F$, that satisfies C3 to the route of k using the following mixed integer program.

$$\text{Min} \sum_{i \in I_k} \Delta_{[i]} \chi_{[i]} \tag{19.29}$$

$$\text{s.t.} \sum_{i \in I_k} \chi_{[i]} = 1 \tag{19.30}$$

$$c_{[i]} - t_{[i]} - \omega_{[i]} - s_{[i]} = 0, \quad i \in I_k \tag{19.31}$$

$$t_{[i+1]} - c_{[i]} - \delta_{[i][i+1]} \left(1 - \chi_{[i+1]}\right) + M \chi_{[i+1]} \geq 0, \quad i \in I_k \tag{19.32}$$

$$t_{[i+1]} - c_f - \delta_{f[i+1]} \chi_{[i+1]} - M \left(1 - \chi_{[i+1]}\right) \geq 0, \quad i \in I_k \tag{19.32$'$}$$

$$t_{[i]} - c_f - \delta_{f[i]} \chi_{[i]} + M \left(1 - \chi_{[i]}\right) \geq 0, \quad i \in I_k \tag{19.33}$$

$$t_{[i]} + \omega_{[i]} - \ell_{[i]p} + M \left(1 - v_{[i]p}\right) \geq 0, \quad i \in I_k, p \in \left\{1, \ldots, \overline{p}_{[i]}\right\} \tag{19.34}$$

$$t_{[i]} + \omega_{[i]} - u_{[i]p} - M \left(1 - v_{[i]p}\right) \leq 0, \quad i \in I_k, p \in \left\{1, \ldots, \overline{p}_{[i]}\right\} \tag{19.35}$$

$$\sum_{p=1}^{\overline{p}_{[i]}} v_{[i]p} = 1, \quad i \in I_k \tag{19.36}$$

$$c_0 \geq \ell_0 \tag{19.37}$$

$$t_0 \leq u_0 \tag{19.38}$$

$$t_0 - c_0 \leq \overline{D}_k \tag{19.39}$$

$$\omega_{[i]} \geq 0, \quad c_{[i]} \geq 0, \quad t_{[i]} \geq 0, \quad [i] \in I_k \tag{19.40}$$

$$v_{[i]p} \in \{0, 1\} \quad [i] \in I_k, p \in \left\{1, \ldots, \overline{p}_{[i]}\right\} \tag{19.41}$$

$$\chi_{[i]} \in \{0, 1\} \quad i \in I_k^*. \tag{19.42}$$

Equation (19.29) chooses the position of the free client f on the route of vehicle k. It minimizes the additional net travel time. Equation (19.30) imposes that f be inserted in exactly one position. It is possible that f can't be inserted without violating the TWs of the already routed clients. In such a case, MIP$_f$ is infeasible. Equation (19.31) relates the arrival, waiting, service and departure times for client $[i]$. Equations (19.32) and (19.33) compute the arrival time to a client relative to the departure time of the vehicle from the preceding client. Constraints (19.34)–(19.36) ensure that any client's service starts within exactly one of the client's TWs. Equations (19.37)–(19.39) restrict the vehicle's departure from, return to, and time out of d. Finally, constraints (19.40)–(19.42) define the variable types.

MIP$_f$ is much easier to solve than MIP. First, the clients' sequence is fixed. Second, the number of alternative positions for inserting f is small. Third and last, for every client $[i]$, $[i] \in I_k$, many TWs can be removed from the model by preprocessing the data. For $p \in \left\{1, \ldots, \overline{p}_{[i]}\right\}, \quad v_{[i]p} = 0$, if

$$u_{[i]p} \leq \sum_{[j]=1}^{[i]} \delta_{[j-1][j]} + \sum_{[j]=1}^{[i-1]} s_{[j]}. \tag{19.43}$$

When the insertion of a free client of the current sector into the route of k deems impossible (i.e., MIP$_f$ is infeasible), CH3 appends it to sector $k+1$.

CH3 is less sensitive to the physical limits imposed by the sectors while it yields good results when the clients are clustered. It maintains low traveled times per vehicle and makes the evolution of the route progress logically.

CH4 proceeds as CH3 does except that it considers clients from larger areas. It assigns to vehicle k a free client i^+ from sectors $k-1$, k, and $k+1$ such that $\delta_{i'i^+} < \overline{\delta}$ and C1–C5 hold. It generally obtains as good routes as CH3 or better, but may require more calls to MIP$_f$ (depending on the distribution of the clients on the grid and of the TWs) as it considers more free clients for each vehicle k, $k \in K$. CH4 is particularly useful in the presence of a mixture of clustered and randomly distributed clients.

CH5 and **CH6** are similar to CH3 and CH4 except that they build a route from both ends: assigning simultaneously two clients to the route: the client that is closest to the previous position of the truck and a client that is closest to the next position in the return path of the truck.

CH7 assigns each client i, $i \in I$, to a vehicle k, $k \in K$, where k is randomly selected from the discrete uniform $[1, m]$. It then applies Eqs. (19.20)–(19.25) of CP to sequence the clients of each vehicle. CH7 is a blind random search. As such, it can't be competitive when the clients are clustered. It can be used as part of a diversification strategy with a population-based meta-heuristic. Alternatively, it can be applied a fixed large number of times, and its best feasible solution is retained for further use. Obviously, it is useful when the clients are uniformly distributed.

CH8 balances the number of clients per truck. It considers the m trucks sequentially, assigning to each truck k, $k \in K$, the client that is closest to the previous location of k. It reiterates through the m trucks until all clients are assigned. When $\frac{n}{m}$ is integer, all trucks have an equal number of clients; otherwise, the numbers of clients for any two trucks do not differ by more than one. It then reduces the violations of the TWs of the clients of each route by applying CP.

CH9 is inspired from CH8 in that it assigns an equal number of clients to each truck except that it only considers clients from the sector of vehicle k, $k \in K$. When it exhausts the clients of its sector, vehicle k chooses clients of sector $k + 1$. Any unassigned clients of a sector k are automatically appended to sector $k + 1$. CH9 uses MIP$_f$ to insert each client to the route of k.

CH3–CH9 apply a look-ahead search strategy that prohibits the search in infeasible directions. They abort constructing a solution if the demand of the free clients can't be assigned to the residual capacities of the vehicles or the residual travel time can't accommodate visiting them. This is guaranteed by solving a binary multiple choice multiple knapsack problem (MCKP) whose decision variables $y_{f,k} = 1$ if the demand of free client$f, f \in F$, can be loaded to vehicle k, $k \in K$, during one of the next iterations. MCKP is also used to choose the subset of free clients that may potentially be appended to the current route of the vehicle k when applied with $K = \{k\}$. Let c_k^{current} be the current total travel time of k and $\underline{\delta}_i = \min_{i' \in I^*\{i\}} \{\delta_{ii'}\}$ be the smallest travel time from i, $i \in I^*$, to any other client i', $i' \in I^*\{i\}$. MCKP follows.

$$\text{Max} \sum_{k \in K} \sum_{f \in F} y_{f,k} \tag{19.44}$$

$$\text{s.t.} \quad \sum_{k \in K} q_f y_{f,k} \leq Q_k - Q_k^{\text{current}}, \quad k \in K \tag{19.45}$$

$$c_k^{\text{current}} + \sum_{f \in F} \left(\underline{\delta}_f + s_f \right) y_{f,k} + \underline{\delta}_0 - t_0 \leq \overline{D}_k, \quad k \in K \tag{19.46}$$

$$\sum_{k \in K} y_{f,k} \leq 1, \quad f \in F \tag{19.47}$$

$$y_{f,k} \text{ Binary}, \quad f \in F, k \in K \tag{19.48}$$

Equation (19.44) provides an upper bound to the number of free clients that can be inserted. Equation (19.45) ensures that the demand of the clients that could be assigned to a vehicle will not exceed the vehicle's residual capacity. Equation (19.46) guarantees that the additional travel time caused by the assignment of free client to a vehicle will not exceed that vehicle's residual travel time. Equation (19.47) reinforces that a free client is served by at most a single vehicle. Finally, Equation (19.48) defines the binary nature of the assignment variable; that is, no partial delivery to a client. When the objective function equals the number of free clients, it may be possible to find a feasible solution to VRPMTW, without any guarantee that such a solution exists. On the other hand, a zero upper bound signals an infeasible direction of the search and halts the construction process. As this problem is solved a large number of times, its integer programming form is only considered when its linear relaxation succeeds in packing all free items.

19.4.2 Stage 2

Stage 2 enhances each solution of Stage 1 by perturbing the incumbent using a steepest descent neighborhood search and using CP to define both feasible and infeasible directions of the search. Three perturbation mechanisms are used:

- Merge I_k and $I_{k'}$ for k and $k^{'}$ serving adjacent sectors and apply CP for $|K| = 2$ and $I = I_k \cup I_{k'}$,
- Remove every client i, $\; i \in I_k$, and insert i into $I_{k'}$, $k \neq k^{'}$, using MIP$_f$
- Remove i_1 and i_2 from I_k and insert them, respectively, into $I_{k'}$ and $I_{k''}$ using MIP$_f$

Inserting client i into the route of vehicle $k^{'}$ is tested if (1) the residual capacity $Q_{k'} - Q_{k'}^{\text{current}}$ of $k^{'}$ exceeds demand q_i, (2) the slack time of $k^{'}$ is less than the minimal additional travel and service times caused by the insertion of i, and (3) the minimal variation Δ of the travel time caused by the swap is negative. That is, CP and MIP$_f$ are only applied when the perturbation may lead to an improving solution. A neighbor is improving if it (1) reduces the fitness of the incumbent or (2) is feasible while the incumbent is not.

19.5 Results

We applied the resulting heuristic on benchmark instances, which have randomly distributed (rm) clients, clustered (cm) clients, randomly distributed clusters of clients (rcm), tight (1) and wide (2) TWs. The tighter the TWs, the larger the number of vehicles is. The heuristic was run on a laptop with a 2.90 GHz Intel Core i7 processor and a 16.0 GB RAM. It was allocated a maximal runtime of 9 min per initial solution per instance. All calls to CP and MIP were allocated 3 and 10 s of runtime with all travel times rounded to their nearest integer.

Table 19.1 illustrates the variation of the solution values obtained by the constructive heuristics for 24 benchmark instances. Columns 1–5 display the label of the instance, the minimal and maximal number of TWs $\underline{p}$ and $\overline{p}$, the minimal and maximal time gap $\underline{\tau}$ and $\overline{\tau}$ between two consecutive TWs, the minimal and maximal ranges $\underline{w}$ and $\overline{w}$ of the TWs, and the number of vehicles m. Columns 6 and 7 display

Table 19.1 Variation of the solution values of Stage 1

	$(\underline{p},\overline{p})$	$(\underline{\tau},\overline{\tau})$	$(\underline{w},\overline{w})$	m	z^*	σ	Δ (%)	CH
rm101	(5,9)	(10,30)	(10,30)	10	1013.81	39.18	4.49	6
rm102	(5,7)	(10,30)	(10,30)	9	969.87	25.93	4.81	4
rm103	(4,7)	(10,50)	(10,30)	9	918.82	36.88	4.23	5
rm104	(3,6)	(10,70)	(10,30)	9	907.67	28.41	1.54	4
rm105	(2,6)	(10,100)	(10,30)	9	890.13	29.57	0.33	9
rm106	(2,3)	(50,100)	(30,50)	9	912.34	35.19	1.27	1
rm107	(1,3)	(50,150)	(30,50)	9	900.26	33.57	1.72	1
rm108	(1,2)	(100,200)	(50,100)	9	960.05	26.23	4.80	2
Average							2.90	
rcm101	(5,10)	(10,30)	(10,30)	10	1097.26	49.27	3.17	4
rcm102	(5,7)	(10,30)	(10,50)	10	1186.83	48.02	5.24	3
rcm103	(3,7)	(10,50)	(10,50)	10	1174.08	59.79	3.73	5
rcm104	(3,5)	(10,50)	(10,50)	10	1208.73	37.17	8.72	3
rcm105	(2,5)	(10,70)	(10,70)	10	1253.75	38.26	5.72	8
rcm106	(2,4)	(30,70)	(30,70)	10	1281.41	48.09	7.38	7
rcm107	(1,3)	(30,100)	(30,70)	11	1370.62	72.74	6.44	1
rcm108	(1,3)	(30,100)	(30,100)	11	1406.69	74.68	5.56	9
Average							5.75	
cm201	(5,10)	(100,150)	(50,100)	5	937.26	29.27	3.79	3
cm202	(5,7)	(100,200)	(50,100)	6	816.80	36.56	3.29	9
cm203	(3,7)	(100,300)	(50,100)	5	962.22	33.03	1.64	1
cm204	(3,5)	(100,500)	(50,100)	5	852.61	40.65	2.45	3
cm205	(2,5)	(200,500)	(100,200)	4	1058.42	32.04	3.86	1
cm206	(2,4)	(200,700)	(100,200)	4	941.89	38.84	3.58	7
cm207	(1,3)	(200,1000)	(100,300)	4	1173.93	27.34	3.60	5
cm208	(1,3)	(500,1000)	(100,500)	4	964.94	33.85	3.70	3
Average							3.24	

$\overline{z}$ and s, the average and standard deviation of CH1 to CH9 solution values. Column 7 reports $\Delta = \frac{(\overline{z}-z^u)}{z^u}100\%$, the percent deviation of $\overline{z}$ from the upper bound z^u obtained by TSVNS while column 8 indicates the index of the CH yielding the best solution value.

Table 19.1 suggests that none of the constructive heuristics obtains the best initial solution value consistently. Each of them is adapted to a particular class of TWs and tailored to a different distribution of clients on the grid. Table 19.1 further indicates that the initial solutions are reasonably good with a 3.96% average Δ from z^u.

Table 19.2 assesses the heuristic's performance. Columns 1–5 and 6–10 display the label of the instance, the number of vehicles m, the heuristic's best solution value z^*, the known bound z^u, and $\Delta^* = \frac{(z^*-z^u)}{z^u}100\%$the percent deviation of z^* from z^u.

None of the constructive heuristics makes stage 2 systematically obtain the best traveled time. This suggests that the nine local optima identified by stage 2 are not

Table 19.2 Heuristic's performance on benchmark instances

Instance	m	z^u	z^*	Δ^* (%)	Instance	m	z^u	z^*	Δ^* (%)
rm101	10	970.23	951	−2.02	rm201	3	686.42	688	0.23
rm102	9	925.34	921	−0.47	rm202	2	684.35	670	−2.14
rm103	9	881.49	883	0.17	rm203	2	674.01	657	−2.59
rm104	9	893.88	879	−1.69	rm204	2	664.88	657	−1.20
rm105	9	887.16	881	−0.70	rm205	2	651.30	648	−0.51
rm106	9	900.90	899	−0.21	rm206	2	672.80	659	−2.09
rm107	9	885.07	896	1.22	rm207	2	657.27	657	−0.04
rm108	9	916.04	912	−0.44	rm208	2	663.59	653	−1.62
Average				−0.52					−1.25
cm101	10	1101.16	1147	4.00	cm201	5	903.08	886	−1.93
cm102	11	1139.73	1190	4.22	cm202	6	790.81	825	4.14
cm103	12	1120.47	1152	2.74	cm203	5	946.67	979	3.30
cm104	14	1248.04	1251	0.24	cm204	5	832.21	834	0.21
cm105	10	1010.23	1096	7.83	cm205	4	1019.10	1069	4.67
cm106	10	982.22	999	1.68	cm206	4	909.37	930	2.22
cm107	11	1056.50	1059	0.24	cm207	4	1133.13	1179	3.89
cm108	10	967.33	968	0.07	cm208	4	930.54	980	5.05
Average				2.63					2.69
rcm101	10	1063.52	1069	0.51	rcm201	2	778.68	738	−5.51
rcm102	10	1127.68	1160	2.79	rcm202	2	815.90	757	−7.78
rcm103	10	1131.84	1124	−0.70	rcm203	2	721.97	695	−3.88
rcm104	10	1111.81	1128	1.44	rcm204	2	698.41	701	0.37
rcm105	10	1185.89	1184	−0.16	rcm205	2	754.51	711	−6.12
rcm106	10	1193.37	1196	0.22	rcm206	2	769.62	764	−0.74
rcm107	11	1287.67	1310	1.70	rcm207	3	749.78	756	0.82
rcm108	11	1332.57	1340	0.55	rcm208	2	742.70	681	−9.06
Average				0.79					−3.99

in the same vicinity. Indeed, their delivery plans differ in terms of routes and clients' sequencing. Thus the importance of the multiple restart of the heuristic.

The nine constructive heuristics are fundamental to building good initial starting points. Stage 2 obtains consistently lower traveled distances when initiated from these nine solutions than when initiated randomly. The comparison of the solution values obtained in Tables 19.1 and 19.2 highlights the importance of stage 2 in identifying a near-global optimum.

The proposed heuristic lowers the travel time for 22 out of 48 benchmark instances with the reduction reaching 9.06% for rcm208. All the time reductions were recorded for rm and rcm classes. The analysis of the results further showed that the best known solutions z^u correspond to solutions that are on the edge of feasibility. For instance, any slight increase of the travel time between pairs of clients may cause the violation of the TW constraints. This questions the validity of the best known solutions in a real-life context, as explained below.

19.6 Practical Considerations

Rincon-Garcia et al. (2018) discuss barriers to the implementation of VRP-related solutions in real-life contexts. Herein, we enumerate few such barriers and explain how the proposed model can handle them.

19.6.1 Single Versus Multiple Depots

Even though MIP and CP are given for a single depot, extending them to a multiple-depot case is straightforward. Two cases are possible: Subsets of vehicles are already allocated to each of the depots or the vehicles are allocated to the depots once an optimal solution to the problem is obtained. The former case is the more prevalent because vehicles are generally tagged to depots and goods are already stocked in specific depots. Such a case reduces to solving several single-depot problems. The latter case can be handled by defining I^* as the union of the set of depots and the set of clients. Then, Equations (19.9) and (19.9′) will apply to all $i \in I^*$ while (19.12)–(19.14) will be defined for each of the depots $d \in I^* \backslash I$. The same applies to Equations (19.22)–(19.24) of CP.

19.6.2 Distances and Travel Times

In MIP and CP, the time matrix $\boldsymbol{\delta}$ may be the result of a transformation into time units of the Euclidean distances that separate any pair of locations $(i,j) \in I^* \times I$. These distances are an accurate estimation of the true distances in the absence of natural or human made barriers or when locations are very far away, i.e., when the time needed to cover the travel distance is much larger than delays that may be caused by local traffic or the routing within the neighborhood of either i or j. In this case, transforming the distances into time units can be based on the average speed of the vehicle (on the highway). On the other hand, when the depot and clients are within the same city, a more relevant measure may be the actual distance the vehicles have to travel. For example, in newer cities, these distances could be estimated by the sum of the distances along the two axes because of the grid structure of the streets. In older cities, these distances should correspond to the true route that the vehicle will use. For instance, they can be approximated using Google Maps or similar software. Their transformation into time units should account for local traffic. The actual travel times are stochastic in nature because of traffic, weather conditions, road construction, unforeseen accidents, driver's state of mind, time of day, etc. Consequently, the time estimates corresponding to the traveled distances should be worst case estimates. This will avoid generating solutions that are on the edge of feasibility and that will cause tardy deliveries.

In the proposed models, travel times are not necessarily symmetric. Even when distances are equal, travel times may differ. In this sense, the above models are general. Travel time can always be expressed as a polynomial relationship of trip time.

19.6.3 Time Windows

The TWs are herein defined as hard; i.e., they must be satisfied for any solution to be feasible. This may be the case for vehicles in cities with travel bans or for deliveries in downtown areas or for private households. To ensure that their deliveries be on-time, constructors tend to account for possible traffic delays and reduce the range of their TWs. That is, their declared TWs are much tighter than their true TWs. For businesses, TWs are either soft or slightly flexible. Businesses tend to wait few extra minutes hoping to receive a delivery rather than to have it postponed. Similarly, households give conservative/tighter TWs than their true availability. They may resort to the help of a neighbor or a family member to receive the delivery. All these factors can be easily accommodated by altering Equations (19.9)–(19.14) and (19.21)–(19.22), e.g., by adding a flexibility factor φ and studying the sensitivity of the solutions to the size of φ.

19.6.4 Collection Versus Distribution

Both models apply whether the goods are being delivered or collected. However, for a distribution problem, items requested by a customer could be in different depots. In this case, the problem can be divided into two steps: Collecting all goods into a single depot (or moving from a national depot to a local one), then distributing the cumulated goods (i.e., delivery from a local depot to customers). Both steps can be perceived as VRPMTW with the clustering solved as in the generalized traveling salesman problem.

19.6.5 Packing Constraints

Equations (19.7) and (19.22) are loading constraints, expressed in terms of the capacity of the vehicle. They assume that the loads are shapeless volumes. However, unless the goods are liquid (e.g., petroleum for gas stations, water), these constraints are a relaxation of three-dimensional non-overlap and envelopment constraints. In some instances, the packing has to also account for the accessibility of the items as a function of their order of delivery. Adding such constraints eliminates a large number of alternative solutions and reduces the search space.

19.6.6 Stochastic Travel Times and Time Windows

In MIP and CP, the travel times are deterministic. Yet, delays always occur. Travel times may vary according to time of day, traffic, weather conditions, etc. Even though delays are generally undesirable, they may be used in some circumstances to the advantage of the vehicle. When a vehicle is delayed on a route that doesn't include enough slack time to compensate for it, deliveries may have to be postponed to a later TW for a particular client but may allow an earlier delivery to another client. In such circumstances, it is judicious to apply a dynamic rerouting of the non-served clients with the objective of minimizing the travel time and the violation of TW penalties. This is of course only possible in small cities where deliveries are within the same vicinity.

Because travel times are non-deterministic, the above models assume that travel times and TWs correspond to worst case scenarios rather than average or modal values. In real life, these bounds are fuzzy. Vehicles reaching clients early may be able to start service immediately. Similarly, those reaching few minutes late may still get a chance to serve their clients during the intended TWs. Again, clients are generally willing to wait few extra minutes for the arrival of a delivery and may be available prior to the TW; i.e., the TWs' bounds are very conservative (i.e., tightest bounds).

When the average in lieu of the worst case performance is sought, a stochastic VRPMTW is at hand. It can be addressed via a stochastic program, which determines the routes that minimize the expected total travel time. The stochastic model can be tackled using a sample average approximation (Al-Khamis and M'Hallah, 2011) that iteratively determines the optimal routes of the vehicles for given samples of uncertain travel times and TWs. This method converges to the optimal travel time in the expected sense as the number of sampled scenarios increases. Each sample corresponds to a deterministic VRPMTW.

An alternative is to apply robust optimization (Mulvey et al. 1995). This method identifies solution robust (near-optimal) model robust (almost feasible) delivery plans. These plans are less sensitive to the uncertainty of the problem. They bound the infeasibility of the plans corresponding to the different scenarios and their distances from the optima by transforming the uncertain problem into a goal program. The weighted goal reflects the tradeoff between feasibility and optimality under all possible scenarios with the weights reflecting the likelihood of each scenario. Despite its attractiveness, robust optimization may become intractable for VRPMTW because of the very large number of possible scenarios, a number that increases with the number of clients, of TWs, and vehicles.

19.7 Conclusions

This chapter proposed a general framework for solving a complex compounded problem. This framework explores the complementary strengths of constraint and mixed integer programming. It builds and enhances solutions using constraint/mixed integer programming-based neighborhood search. The search is guided by feasibility constraints and look-ahead strategies that fathom large parts of the infeasible search space and focuses the search towards improving solutions. This framework was illustrated on the vehicle routing problem with multiple time windows. It can be extended to other compounded problems such as vehicle routing with packing constraints, bin packing with due dates, scheduling with cutting constraints or assembly limitations or resource unavailability.

This framework has several advantages. It requires no tuning and guarantees the reproducibility of the results. These two features can't be achieved by meta-heuristics that rely on seeds of random numbers. Such meta-heuristics include genetic algorithms, simulated annealing, ant colonies, swarm optimization, and bee colonies. Such techniques are generally assessed via their "average performance," a practice that is not justifiable in the service industry.

References

Adewumi AO, Adeleke OJ (2018) A survey of recent advances in vehicle routing problems. Int J Syst Assur Eng Manag 9(1):155–172

Alba E, Dorronsoro B (2008) A hybrid cellular genetic algorithm for the capacitated vehicle routing problem. In: Abraham A, Grosan C, Pedrycz W (eds) Engineering evolutionary intelligent systems. Springer, Heidelberg, pp 379–422

Al-Khamis T, M'Hallah R (2011) A two-stage stochastic programming model for the parallel machine scheduling problem with machine capacity. Comput Oper Res 38(12):1747–1759

Al-Mudahka I, Hifi M, M'Hallah R (2011) Packing circles in the smallest circle: An adaptive hybrid algorithm. J Oper Res Soc 62(11):1917–1930

Archetti C, Speranza MG (2014) A survey on matheuristics for routing problems. Eur J Comput Optim 2(4):223–246

Baldacci R, Mingozzi A, Roberti R (2012) Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints. Eur J Oper Res 218(1):1–6

Beheshti AK, Hejazi SR, Alinaghian M (2015) The vehicle routing problem with multiple prioritized time windows: a case study. Comput Ind Eng 90:402–413

Belhaiza S, M'Hallah R (2016) A pareto non-dominated solution approach for the vehicle routing problem with multiple time windows. In: IEEE Congress on Evolutionary Computation Proceedings, Vancouver, Canada, pp 3515–3524

Belhaiza S, Hansen P, Laporte G (2013) A hybrid variable neighborhood tabu search heuristic for the vehicle routing problem with multiple time windows. Comput Oper Res 52:269–281

Belhaiza S, M'Hallah R, Ben Brahim G (2017) A new hybrid genetic variable neighborhood search heuristic for the vehicle routing problem with multiple time windows. In: IEEE Congress on Evolutionary Computation, San Sebastian, Spain, pp 1319–1326

Braekers K, Ramaekers K, van Nieuwenhuyse I (2016) The vehicle routing problem: state of the art classification and review. Comput Ind Eng 99:300–313

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

Eksioglu B, Vural AV, Reisman A (2009) The vehicle routing problem: a taxonomic review. Comput Ind Eng 57(4):1472–1483

Favaretto D, Moretti E, Pellegrini P (2007) Ant colony system for a VRP with multiple time windows and multiple visits. J Interdisciplinary Math 10(2):263–284

Figliozzi MA (2010) An iterative route construction and improvement algorithm for the vehicle routing problem with soft time windows. Transp Res 18(5):668–679

Fu Z, Eglese R, Li LYO (2007) A unified tabu search algorithm for vehicle routing problems with soft time windows. J Oper Res Soc 59(5):663–673

Gendreau M, Hertz A, Laporte G (1994) A tabu search heuristic for the vehicle routing problem. Manag Sci 4(10):1276–1290

Gendreau M, Potvin JY, Bräysy O, Hasle G, Lokketangen A (2008) Metaheuristics for the vehicle routing problem and its extensions: a categorized bibliography. In: Golden B, Raghavan S, Wasil E (eds) The vehicle routing problem: latest advances and new challenges. Springer, New York, pp 143–169

Hashimoto H, Ibaraki T, Imahori S, Yagiura M (2006) The vehicle routing problem with flexible time windows and traveling times. Discret Appl Math 154(16):2271–2290

Ibaraki T, Imahori S, Kubo M, Masuda T, Uno T, Yagiura M (2005) Effective local search algorithms for routing and scheduling problems with general time-window constraints. Transp Sci 39(2):206–232

Koç Ç, Bektaş T, Jabali O, Laporte G (2016) Thirty years of heterogeneous vehicle routing. Eur J Oper Res 249(1):1–21

Laalaoui Y, M'Hallah R (2016) A binary multiple knapsack model for single machine scheduling with machine unavailability. Comput Oper Res 72(8):71–82

M'Hallah R (2007) Minimizing total earliness and tardiness on a single machine using a hybrid heuristic. Comput Oper Res 34(10):3126–3142

M'Hallah R (2014) Minimizing total earliness and tardiness on a permutation flow shop using VNS and MIP. Comput Ind Eng 75:142–156

M'Hallah R, Alkhabbaz A (2013) Scheduling of nurses: a case study of a Kuwaiti health care unit. Oper Res Healthcare 2(1–2):1–19

M'Hallah R, Al-Khamis T (2012) Minimizing total weighted earliness and tardiness on parallel machines using a hybrid heuristic. Int J Prod Res 50(10):2639–2664

M'Hallah R, Alkandari A, Mladenovic N (2013) Packing unit spheres into the smallest sphere using VNS and NLP. Comput Oper Res 40(2):603–615

Minocha B, Tripathi S (2013) Two phase algorithm for solving VRPTW problem. Int J Artif Intel Expert Syst 4:1–16

Moccia L, Cordeau JF, Laporte G (2012) An incremental tabu search heuristic for the generalized vehicle routing problem with time windows. J Oper Res Soc 63(2):232–244

Mulvey JM, Vanderbei RJ, Zenios SA (1995) Robust optimization of large-scale systems. Oper Res 43(2):264–281

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

Nagata Y, Bräysy O, Dullaert W (2010) A penalty-based edge assembly memetic algorithm for the vehicle routing problem with time windows. Comput Oper Res 37(4):724–737

Nazif H, Lee L (2012) Optimized crossover genetic algorithm for capacitated vehicle routing problem. Appl Math Models 36(5):2110–2117

Polyakoskiy S, M'Hallah R (2018) A hybrid feasibility constraints-guided search to the two-dimensional bin packing problem with due dates. Eur J Oper Res 266:819–839

Polyakovskiy S, M'Hallah R (2014) A multi-agent system for the weighted earliness tardiness parallel machine problem. Comput Oper Res 44(4):15–136

Rincon-Garcia N, Waterson BJ, Cherrett TJ (2018) Requirements from vehicle routing software: perspectives from literature, developers and the freight industry. Transp Rev 38(1):117–138

Solomon M (1987) Algorithms for the vehicle routing and scheduling problems with time window constraints. Oper Res 35(2):254–265

Toth T, Vigo D (2002) Models, relaxations and exact approaches for the capacitated vehicle routing problem. Discret Appl Math 123:427–512

Vidal T, Crainic TG, Gendreau M, Prins C (2013) Heuristics for multi-attribute vehicle routing problems: a survey and synthesis. Eur J Oper Res 231(1):1–21

Vidal T, Crainic TG, Gendreau M, Prins C (2014) A unified solution framework for multi-attribute vehicle routing problems. Eur J Oper Res 234(3):658–673

Rym M'Hallah is currently a professor at the Department of Statistics and Operations Research at Kuwait University, and a professor of Quantitative Methods and Information System at the University of Sousse, Tunisia. She joined academia after industrial experience in Tunis, Tunisia. Dr. M'Hallah has earned degrees in industrial engineering and operations research, all from Pennsylvania State University.

Dr. M'Hallah's research focuses on modeling and optimizing large-scale systems using operations research techniques: mathematical programming, meta-heuristics, simulation, scheduling, and quality control. Applications include health care, manufacturing and transportation, scheduling, and cutting and packing. She has an extended international research collaboration network. Her current interest is enhancing health care systems and service industries using optimization techniques.

Dr. M'Hallah's choice of career was influenced at a very early stage by her grandmother and parents who persuaded her of the limitless benefits of a scientific career: notoriety along with a life style that involved discovering new worlds and making a difference in people's lives. Dr. M'Hallah was naturally gifted in Mathematics. Opting for engineering was a matter of luck: She got a scholarship to study in the USA. She was a natural optimizer, which made improving processes and enhancing designs a perfect match. She has been blessed with unconditionally supportive parents, inspiring women role models, enlightening scientific mentors, an encouraging husband and two loving wise kids. Her resilience and determination to succeed have garnered her more than 45 peer-reviewed journal papers. Today, she remains in the field of Science and Engineering as an educator, researcher, and mentor. She likes the challenges that real-life problems bring: the continuous/lifetime/independent learning, the personal growth, and the international collaborations with the friendships they build and the new experiences they offer. Last but not least, she appreciates the freedom of time management and of investigating her research interests.