University of the Aegean
Dep. of Financial & Management Engineering
Design Operation & Production Systems Lab (DeOPSys Lab)

Replanning in Vehicle Routing with Dynamic Pickups [1]

G. Ninikas, I. Minis

Lab Report, 2012-09

[1] Technical Report, 2012

REPLANNING IN VEHICLE ROUTING WITH DYNAMIC PICKUPS

George Ninikas[2]

Department of Financial and Management Engineering, University of the Aegean, 41 Kountouriotou Str., Chios, 82 100, Greece

g.ninikas@fme.aegean.gr

Abstract

We investigate a dynamic routing problem that seeks to assign, in the most efficient way, dynamic pickup requests that arrive in real-time while a predefined distribution plan is being executed. We refer to this problem as the *Vehicle Routing Problem with Dynamic Pickups (VRPDP)*. In this paper, we address the VRPDP through iterative replanning. In addition to defining the replanning model, we drill-down to significant aspects concerning the replanning process; i.e. i) *how* to replan, ii) *when* to replan, and iii) *what* part of the new plan to communicate to the drivers. On some of these aspects, we establish basic theoretical insights. To solve the replanning problem, we propose a Branch-and-Price (B&P) approach. Furthermore, for cases of high complexity (e.g. without time windows), we propose a novel insertion heuristic based on column generation that provides near optimal solutions in an efficient manner. We use extensive experimentation to test the proposed methods and analyze the related replanning policies. Based on the results obtained we propose replanning guidelines under various operational settings.

Keywords: *dynamic vehicle routing, replanning, dynamic pickup-and-delivery, branch-and-price*

[2] Corresponding author: Tel: +30 (22710) 35450, Fax: +30 (22710) 35484

Problem description and related literature

Increasing competitive pressures and expectations for high-quality service have led urban logistics operators to enhance their offering by responding to requests that arrive in a dynamic fashion. For example, in a typical courier setting, a set of delivery vehicles originating from a local distribution hub (depot), is tasked to deliver (or pickup) orders known prior to the start of operations (offline requests). As the work plan unfolds, however, customer orders are received through a call center, for on-site pickup within the current period of operations. These pickup orders, referred to as dynamic pickups, have to be collected and returned to the hub for further processing. In this work we focus on such situations and seek to allocate in real time dynamically arriving orders to the most appropriate vehicles, either to those en-route or to extra vehicles stationed at the depot. We refer to the related problem as the Vehicle Routing Problem with Dynamic Pickups (VRPDP).

Beyond the courier case, such problems arise naturally in money transfer logistics and repair-maintenance services. Service vehicles are called to serve requests for money or faulty equipment pickups, respectively, which arrive to a dispatch center in a dynamic fashion. Another example may be found in coach transfers. In that case, vehicles that execute planned routes originating from major locations (e.g. airport) and serving predefined drop-off areas (e.g. accommodation sites), are requested to collect passengers from additional locations while en-route.

In these applications, incorporating dynamic requests in the *a-priori* plan may reduce the plan's quality or, even worse, it may lead to infeasibilities. Real-time decision-making appears to be essential for addressing such dynamic situations effectively. That is, an *a-priori* plan may be modified and updated based on the real-time state of the logistics system (once or repeatedly). In this context, we define as *replanning* the problem that seeks to assign these new service requests to the available vehicles, while achieving efficient routing costs and respecting all service constraints. We also investigate key related decisions and actions; that is, i) *how* to replan, ii) *when* to replan, and iii) *what* part of the new plan to communicate to the drivers.

The VRPDP is a dynamic version of the *one-to-many-to-one* pickup and delivery problems (1-M-1-PDPs, Berbeglia *et al.*, 2007; Berbeglia *et al.*, 2010). In the latter, vehicles deliver commodities to customers, while other commodities are picked up from the customers and are transported back to the depot. The VRPDP forms a special case and considers that a) each customer requires only pickup or delivery, and b) pickup and delivery customers may be served in an arbitrary order. Note that the static version of the VRPDP is referred in the literature as the Vehicle Routing Problem with Mixed Linehauls and Backhauls (VRPMB, Parragh *et al.*, 2008), or the Mixed

Vehicle Routing Problem with Backhauls (MVRPB) (Salhi and Nagy, 1999; Ropke and Pisinger, 2006), or Vehicle Routing Problem with Backhauls with Mixed load (VRPBM) (Dethloff, 2002).

Limited work has been conducted on the dynamic counterpart of the PDPs and, to the best of our knowledge, no study has investigated the dynamic version of 1-M-1-PDPs. The majority of the work has focused on dynamic one-to-one PDPs, in which each request has certain origin and destination. Related problems to the one-to-one-PDPs (1-1 PDPs) mostly deal with the transportation of passengers in urban areas, as in the *dial-a-ride problem* (DARP), or in the same-day transportation of letters/parcels, as in the *Dynamic PDP* (DPDP). Since this paper doesn't focus on 1-1 PDPs, we refer the reader to the survey of Berbeglia *et al.* (2010) for solution approaches and related references for this class of problems.

The aforementioned dynamic PDPs can be also seen as a subclass of the general category of dynamic vehicle routing problems (DVRPs – Psaraftis, 1988), which have attracted an increasing body of research over the last two decades. Two major classes of solution approaches may be discerned based on the way to deal with dynamism: i) *local approaches*, which consider no information regarding the future, and ii) *look-ahead approaches*, that incorporate probabilistic information of future events at each replanning event. For the latter ones, the reader may refer to Powel (1996), Bent and Van Hentenryck (2004), Larsen *et al.* (2004) and Ichoua *et al* (2006). We review here work related to local approaches that are relevant to both our problem setting and approach. Results on this class of problems may be found in, among others, Gendreau *et al.* (1999; 2006), Ichoua *et al.* (2000), Larsen *et al.* (2002), Yang *et al.* (2004) and Chen and Xu (2006). Typical solution approaches reported in those papers can be classified in i) local-search or rule-based algorithms, ii) sophisticated optimization-based algorithms, or iii) hybrid approaches of (i) and (ii).

In order to cope with the characteristics of dynamic environments, fast response times are required. For this reason, the majority of the studies have proposed local-search or rule-based algorithms. For example, Shieh and May (1998) studied the DVRP with time windows and propose an insertion-based heuristic, improved by a local search. Gendreau *et al* (1999) and Ichoua *et al* (2000) proposed a tabu search heuristic in order to address a similar problem in which time windows may be violated at some cost. Larsen *et al* (2002) compared various rule-based heuristics with various degrees of dynamism for a dynamic travelling repairman problem in which requests need to be served at a minimum total cost. Their study showed that the route length increases linearly w.r.t. the degree of dynamism. Chen and Xu (2006) proposed a column-generation-based approach for solving a DVRP with hard time windows, in which all requests need to be serviced; the algorithm uses fast heuristics to modify existing columns generated at an earlier stage in order to incorporate the up-to-date information. Those columns are then included and solved within a set-partitioning formulation in an iterative manner. Their approach outperforms an insertion-based heuristic that was used for comparison, but it provides inferior results compared to a similar approach that allows unlimited amount of computational time for solving the underlying static problems.

In addition to the problem definition and the solution approach, a critical problem element is when to reoptimize in such a dynamic environment. Very limited research focused on replanning policies and their impact on the overall solution. The majority of studies (e.g. Gendreau *et al*, 1999; Ichoua *et al*, 2000) reoptimize at every *event*, i.e. upon the arrival of a vehicle at a customer or the introduction (or cancellation) of a customer order. Other studies deal with replanning at certain fixed periods. For example, Larsen (2001) studied the DVRP with time windows introducing the so-called *batching strategies* and analyzed the effect of reoptimization on simple predefined fixed events (e.g. upon the arrival of three customers and every 10 minutes). Chen and Xu (2006) replan at fixed cycles. More recently, Angelelli (2009) applied replanning on predefined time-events (e.g. 1, 2.5 and 5 hours) for a dynamic multi-period vehicle problem.

In this paper, we address the VRPDP through replanning. We define the replanning model, and propose a Branch-and-Price (B&P) approach to solve it. For cases of high complexity (e.g. without time windows), we propose a novel Column Generation-based insertion heuristic that provides near optimal solutions in an efficient manner. Regarding the implementation of replanning, we discuss and analyze related policies and propose guidelines under various operational settings.

The remainder of this paper is structured as follows: Section 2 formalizes the replanning problem within the VRPDP setting. Section 3 presents a Branch-and-Price (B&P) approach to solve the replanning problem, using both an exact algorithm and the proposed heuristic. Section 4 provides insights regarding the question "*when to replan*"; we propose and analyze several replanning strategies. Section 5 presents our testing environment and analyzes and compares the results obtained by various replanning scenarios. Finally, Section 6 summarizes our key findings.

Replanning in VRPDP

Problem overview

Consider a transportation network in a Euclidean plane. A sufficient number of dedicated homogeneous vehicles V with limited capacity $\bar{Q}$ are located at a single depot prior to the start of operations. At time 0, reflecting the beginning of the planning horizon $[0, T_{max}]$, a set of vehicles $K \subset V$ commence the execution of their planned routes to serve a set of offline requests known in advance, while $K_d = V - K$ is the set of vehicles available at the depot. A vehicle, once dispatched, is required to return to the depot until $t = T_{max}$. During the execution of the distribution plan, new customers call-in requesting (pickup) services. These arriving requests will be referred to as *Dynamic Requests* (DRs). Only DRs that arrive during a pre-defined *admissible period* $[0, T_{max} - \tau]$ must be served. Offline requests originally assigned to vehicles K cannot be re-allocated to other vehicles, while DRs may be served by any vehicle $V = K \cup K_d$ as needed.

The problem's scope is to serve all offline requests and allocate all DRs to the vehicles of set V in order to minimize the overall routing costs. Note that we assume sufficient vehicles available in order to serve all requests. In our setting, the allocation of DRs in the available fleet is dealt through iterative *replanning* as described below.

Solution framework

We assume that in the overall planning horizon $[0, T_{max}]$, there will be *L replanning cycles*, each corresponding to an appropriate "static" problem $\Gamma_1, \Gamma_2, \dots, \Gamma_L$, with replanning occurring at time instances $T_\ell, \ell = 1,2, \dots, L$ where $T_0 = 0 < T_1 < \cdots < T_L < T_{max} - \tau$. Replanning cycles ($[T_{\ell-1}, T_\ell]$, $\ell \geq 1$) may not be necessarily of equal duration. The "*static*" problem solved at each replanning time T_ℓ, considers all information known up to this point in time. It is assumed that this problem (Γ_ℓ) is solved instantaneously. The structure of the replanning framework is illustrated in Figure 1.

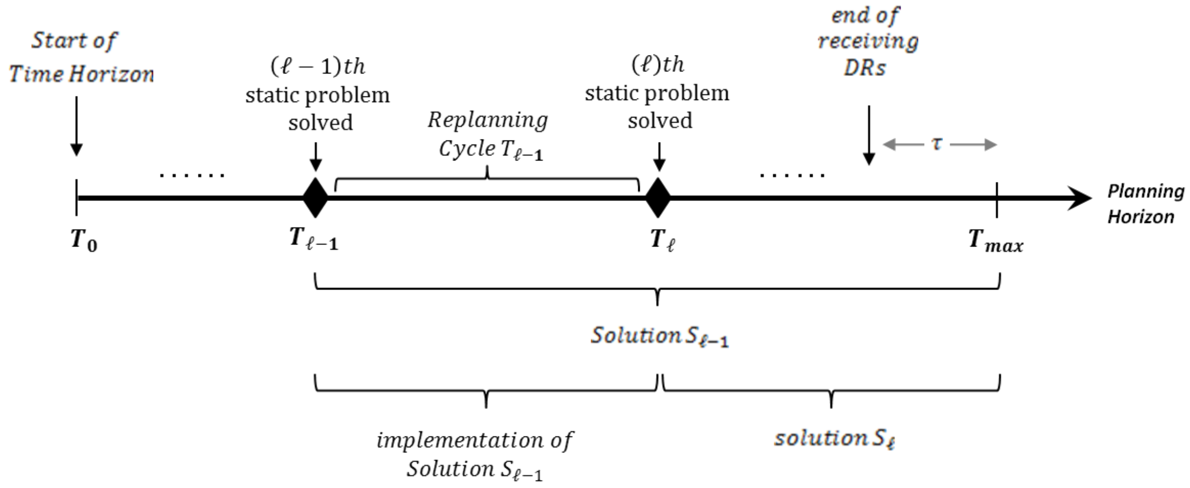

Figure 1. Overview of the replanning framework

A replanning problem $\Gamma_\ell,\ \ell \in \{1, \dots, L\}$ takes into account two sets of orders not yet served: i) the *committed orders*, that include all orders assigned to a vehicle originally or during previous replanning cycle, that have not been served and cannot be re-allocated to other vehicles, and ii) the *flexible orders*, that correspond to newly arrived DRs, or previously arrived DRs not yet served that can be re-assigned to any vehicle $V = K \cup K_d$. Typically, flexible orders correspond to all DRs that have not been served at time T_ℓ. However, there are some practical cases in which this may not be applicable, and DRs assigned to vehicles during a prior replanning cycle and not yet served, may be considered as committed orders. This limitation may be caused by financial transactions, communication with the customer, etc.

For the reason above, depending on the policy, two scenarios are relevant: a) committed orders correspond only to offline requests and flexible orders are all DRs not yet served, and b) committed orders are all orders assigned to vehicles during any previous cycle and not yet served; flexible orders correspond only to newly arrived DRs. Those two cases will be analyzed subsequently in Section 4.

The solution S_ℓ of the static problem of replanning cycle ℓ considers the entire remaining time horizon $[\mathrm{T}_\ell, \mathrm{T}_{\max}]$. Part of this solution is then implemented until the next replanning time $\mathrm{T}_{\ell+1}$. The following assumptions concerning the operational scenario are also considered:

- The current status of the logistics operations is assumed to be known at any time
- Waiting (when needed for a time window opening) is performed at the location of the previously served customer
- The route is updated only at customer locations, i.e. we do not allow diversion (as e.g. in Ichoua *et al.*, 2000).

The replanning problem

In describing the replanning problem we omit index ℓ, since the problem has the same form in any replanning cycle. Let $N = C \cup F$ denote the set of orders which have not been served, where C and F denote the sets of known committed and flexible orders, respectively. Furthermore, $C = \bigcup_{k \in K} C_k$, where C_k represent the set of committed orders assigned to each vehicle K that is en-route. Note that this latter set may include delivery but also pickup orders that are assigned to vehicle $k \in K$ during previous replanning cycles and cannot be re-distributed to other vehicles. Let set $M = \bigcup_{k \in K} \mu_k$, where μ_k represents the current location of vehicle $k \in K$, and node 0 represent the origin and destination depot. We consider a complete directed graph in a Euclidean plane $G = (W, A)$, where $W = C \cup F \cup M \cup \{0\}$ and A the set of arcs connecting all nodes $W (A = \{(i,j): i \in W, j \in W \backslash M\})$. The cost of traversing arc $(i,j), \{i \in W, j \in W \backslash M\}$ is denoted by c_{ij}, while t_{ij} denotes the travel time between these two nodes.

Each order $i \in C \cup F$ is characterized by the following elements:

d_i	is the demand/supply of the order at client i. Delivery orders are associated with a negative value of d_i and pickup orders with a positive one; $d_0 = 0$
s_i	is the service time of order i at the client site; also $s_0 = 0$
h_i	is the arrival time of new order i. $0 < h_i < T_{max} - \tau$ for all DRs.
$[a_i, b_i]$	is the time window of customer i. For orders known prior to time T_0, $0 \leq a_i < b_i \leq T_{max}$ and for DRs, $h_i < a_i < b_i \leq T_{max}$. Additionally, $a_0 = 0$ and $b_0 = T_{max}$.

Three sets of variables are defined: i) x_{ijk} is equal to 1 if arc $(\mathrm{i}, \mathrm{j}) \in \mathrm{A}$ is used by vehicle $\mathrm{k} \in \mathrm{V}$ and zero otherwise, ii) w_{ik} represents the start of service for customer $i \in N$ by vehicle $k \in V$, where for the depot $w_{0k} \geq T$, and iii) Q_{ik} is the load of vehicle $k \in V$ immediately after serving customer $i \in W$. Note that initial load of the vehicle $Q_{\mu_k k}, k \in K$ at each replanning cycle is equal to the total amount to be delivered (and/or picked up) by vehicle k.

The replanning problem for the VRPDP is similar to the formulation proposed by Parragh *et al.* (2008) for the multi-vehicle pickup and delivery problem, which was, in turn, adapted from the model proposed by Cordeau *et al.* (2002) for the VRPTW.

The objective is to minimize the total cumulative routing cost over the planning horizon $[T_{\ell}, T_{max}]$ and is given by:

$$\min(z) = \sum_{k\in V} \sum_{(i,j)\in A} c_{ijk} x_{ijk} \tag{1}$$

Constraints:

$$\sum_{j\in C_k \cup F \cup \{0\}} x_{ijk} = 1 \qquad \forall k \in K, \quad \forall i \in C_k \cup \{\mu_k\} \tag{2}$$

$$\sum_{k\in V} \sum_{j\in W} x_{ijk} = 1 \qquad \forall i \in F \tag{3}$$

$$\sum_{i\in C_k \cup F \cup \{\mu_k\}} x_{i0k} = 1 \qquad \forall k \in K \tag{4}$$

$$\sum_{j\in F} x_{0jk} \le 1 \qquad \forall k \in K_d \tag{5}$$

$$\sum_{j\in F} x_{0jk} = \sum_{j\in F} x_{j0k} \qquad \forall k \in K_d \tag{6}$$

$$\sum_{i\in W} x_{ihk} - \sum_{j\in W} x_{hjk} = 0 \qquad \forall h \in N, \forall k \in V \tag{7}$$

$$Q_{jk} \ge Q_{ik} + d_j - Z(1 - x_{ijk}) \qquad \forall (i,j) \in A, \forall k \in V \tag{8}$$

$$\max\{0, d_i\} \le Q_{ik} \le \min\{\bar{Q}, \bar{Q} + d_i\} \qquad \forall i \in N, \forall k \in V \tag{9}$$

$$w_{jk} \ge w_{ik} + s_i + t_{ij} - Z(1 - x_{ijk}) \qquad \forall (i,j) \in A, \forall k \in V \tag{10}$$

$$\max(a_i, T) \sum_{j\in W} x_{ijk} \le w_{ik} \le b_i \sum_{j\in W} x_{ijk} \qquad \forall k \in V, \forall i \in W \tag{11}$$

$$T \le w_{0k} \le b_0 \qquad \forall k \in K_d \tag{12}$$

$$x_{ijk} \in \{0,1\} \qquad \forall (i,j) \in A, \forall k \in V \tag{13}$$

Constraint (2) specifies that each vehicle k en-route must serve all committed orders originally assigned to it. Constraint (3) ensures that all flexible customers will be served, either by a vehicle *en-route* or by a vehicle available at the depot. Constraints (4) force active vehicles (i.e. those not at the depot) to eventually return to the depot. According to Constraint (5) new vehicles dispatched from the depot in this replanning cycle can only serve flexible customers. Constraint

(6) forces these new vehicles to return to the depot. Constraint (7) ensures flow conservation. Constraints (8) and (9) ensure that the vehicle's capacity limit is respected at all vertices, where Z is a large positive constant. Constraints $(10) - (11)$ ensure that a route is time feasible; constraint (10) updates the start time (of service) along the route, while (11) ensures that the service start time is within the time window of the node. Constraints (12) force the new vehicles K_d to begin at the replanning time and return at the depot within the available planning horizon. Finally, Constraints (13) force the flow variables to assume binary values $\{0, 1\}$.

A Branch-and-Price scheme

To solve the replanning problem of the VRPDP at each replanning cycle, we employ branch-and-price (B&P) (Barnhart *et al.*, 1998; Desaulniers *et al.*, 1998; Desrosiers and Lübbecke, 2005). Our B&P approach has been inspired by related work for a) the VRPTW (Desrochers *et al.*, 1992; Feillet *et al.*, 2004; Feillet *et al.*, 2005; Chabrier, 2006), and b) the PDPTW (Desaulniers *et al.*, 2002; Cordeau *et al.*, 2007; Ropke *et al.*, 2007). Below we describe the enhancements made to the typical B&P approach in order to address the VRPDP. Accordingly we provide an exact and a new heuristic approach.

An exact B&P approach

The Restricted Master Problem (RMP)

The Master Problem (MP) for the VRPDP is usually formulated as a set partitioning problem, in which each column corresponds to a feasible route and each constraint corresponds to a customer been served. Hence, we introduce binary coefficient a_{ir} that equals to 1 if order $i \in N$ is included in route $r \in \Omega$ and zero otherwise, as well as a coefficient y_r which equals to 1 if route $r \in \Omega$ is used in the solution and zero otherwise. If c_r denotes the cost of route $r \in \Omega$, then the objective function of the Master Problem is of the following form:

$$(\boldsymbol{SPP}) \quad Minimize \quad \sum_{r \in \Omega} c_r y_r \tag{14}$$

$$subject\ to: \quad \sum_{r \in \Omega} a_{ir} y_r = 1 \qquad \forall i \in N \tag{15}$$

$$y_r = \{0, 1\} \qquad \forall r \in \Omega \tag{16}$$

Consequently, MP involves only constraints imposing single visit at any order (i.e. Constraints (2) and (3) of the original formulation); the remaining constraints of the original problem are handled by the subproblems. Assuming that Ω is the set of all feasible routes (columns), in our formulation this set comprises two subsets, i.e. $\Omega = (\bigcup_{k \in K} \Omega_k) \cup \Omega_p$, where:

- Columns Ω_k correspond to vehicles K already en-route; these routes originate from the current vehicle locations μ_k and end at the depot, and include all committed (C_k) orders and (perhaps) flexible (F) orders.
- Columns Ω_p correspond to vehicles K_d located at the depot. These routes originate and end at the depot, including only F orders.

We denote as Ω' a subset of Ω that contains known and feasible routes. In order to construct this set, we exploit the information from solution $S_{\ell-1}$ obtained during the replanning cycle $(\ell - 1)$, for the interval $[T_{\ell-1}, T_{max}]$. Eliminating all orders that have been served up to T_ℓ, yields a feasible solution $S'_{\ell-1}$ of routes that comprise two types of columns corresponding to vehicles *en-route*:

- Those dispatched at time $T_0 = 0$, which should serve remaining committed orders, and
- Those dispatched from the depot at time $T_{\ell'}$, where $0 < \ell' \leq \ell - 1$, which serve DRs arrived during previous replanning cycles

A note here about committed orders: In addition to the offline requests assigned to the vehicles prior to the start of operations, committed orders may include DRs depending on the policy followed (see Section 2.2). If the policy is to allow re-assignments of all unserved DRs, then the latter are incorporated into the set of flexible orders set along with the newly arrived DRs. Obviously, in this case, columns of the second type above, will become of the form $[\mu_{k'} - depot], \forall k' \in K$.

The feasible set of routes (Ω_k) that cover all committed orders may be used as an initial solution in the set Ω' of the corresponding RMP. For the flexible orders (F set), we generate single-visit trips that originate and finish at the depot, i.e. $[depot - i - depot], \forall i \in F$ to be added to the initial set of columns Ω' (Ω_p columns).

The Pricing Subproblem(s)

Having solved the RMP (by known linear programming techniques), a pricing subproblem (SP) is solved to identify variables (columns) in the set $\Omega \backslash \Omega'$ that have a negative reduced cost w.r.t. the dual solution of the RMP (Desaulniers *et al.*, 2005).

In order to address the requirement that committed orders cannot be re-distributed among other vehicles, we formulate and solve several independent SPs, one for each K vehicle *en-route*. Denote these independent problems by Ψ_k, $\forall k \in K$. The set of orders considered for each Ψ_k consists of the remaining committed orders of vehicle k (C_k set) plus all F orders, i.e. $N_k = C_k \cup F$. This F set is common in each Ψ_k. The solution of each Ψ_k will generate feasible trips (columns) that originate from current vehicle location μ_k and cover all remaining C_k orders and some orders from the F set. The subset of columns generated by each Ψ_k will comprise set $\Omega_k, k \in K$.

In order to consider also the assignment of F orders to vehicles located at the depot, we solve an additional independent subproblem, denoted as Ψ_{K+1}, that includes only the F set, i.e. $N_{K+1} = F$. The solution of this problem generates feasible trips (subject to all constraints) that originate from the depot, serve one or more F orders and return to the depot. The columns generated from Ψ_{K+1} comprise set Ω_p. Figure 2 illustrates the proposed decomposition approach.

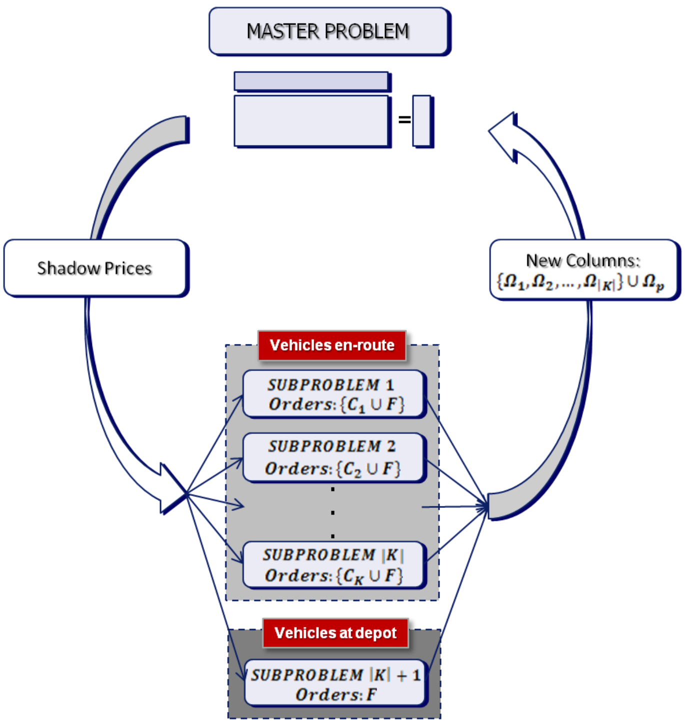

Figure 2. The decomposition approach for the pricing sub-problem

Each one of those SP is modeled as an Elementary Shortest Path Problem with Resource Constraints (ESPPRC), based on the work of Irnich and Desaulniers (2005). In addition to the typical resource constraints that impose restrictions on customer and shift time windows, vehicle capacity and path elementarity, we also employ a new "resource" to ensure that all C_k orders are included in the path (since they cannot be re-allocated to other vehicles). The window of this resource should not exceed the number of C_k orders assigned to each vehicle, i.e. $[0, |C_k|]$. This resource is increased by 1 when using arc (i, j) to reach vertex $j \in C_k$; otherwise it remains unaffected. Based on this, only feasible paths that have consumed $|C_k|$ units of this resource will be kept, as described later.

Solution Procedure for the Pricing Sub-Problem

To solve the pricing subproblems we use a *label correcting algorithm* similar to the one proposed by Feillet *et al.* (2004; 2005). This relies on the creation of multi-dimensional labels by processing nodes in a repetitive manner. Each "*label*" is a vector that corresponds to a partial path δ from source o to vertex $i \in N$, and comprises several components that describe the state of δ, typically the accumulated reduced cost $\tilde{c}_{\delta i}$ as well as the values of the resources at vertex i. Beyond these typical label components, we have introduced $\bar{\bar{c}}_{\delta i}$, the *equilibrium cost*, which represents an upper bound (worst case) of the total modified cost required to serve all *committed* orders not yet included in partial path δ. Note that we drop index k in the remainder of this section, since only committed orders assigned to each vehicle k are considered by each SP. Let $\mathrm{O}(\Lambda_{\delta i}) \subset \mathrm{C}$, denote the set of committed orders included in partial path δ ending at vertex i and $\mathrm{O}'(\Lambda_{\delta i}) \subset \{\mathrm{C} \cup \mathrm{F}\}$ denote the remaining set of all orders $N = C \cup F$ not yet served by partial path δ. Then, the equilibrium cost can be defined as:

$$\bar{\bar{c}}_{\delta i} = \sum_{i \in C \setminus \mathrm{O}(\Lambda')} \left(\max_{h \in O'(\Lambda') \cup \{o\}} (c'_{hi}) + \max_{j \in O'(\Lambda') \cup \{o'\}} (c'_{ij}) \right) \tag{17}$$

where c'_{ij} is the modified cost associated with arc $(i, j) \in A$. This label component is used to capture the requirement for providing service to the committed customers from the very first step of the solution process. Thus, label $\Lambda_{\delta i}$ of partial path δ indicates whether or not the path includes all the required committed orders. This information is used in the dominance criteria described below.

The procedure commences at the source point o with initial label Λ_o and at time $t = t_o^\ell$. For the Ψ_{k+1} subproblems, $t_o^\ell = T_\ell$; for Ψ_k, however, a vehicle may be on its way to the next destination or already serving a customer at replanning time T_ℓ. Therefore, assuming customer $h \in N$ as the source point o, t_o^ℓ is set to $\max(a_h, w_h) + s_h$.

From source point o, each label $\Lambda_{\delta i}$ is extended along all arcs $(i, j) \in A$ to create new labels $\Lambda_{\delta' j}$. A label $\Lambda_{\delta' j}$ is discarded if it is not feasible, i.e. if at least one of its resource components exceeds the resource upper bound. Finally, when a partial path is extended to the ending node 0, then a full feasible path has been generated. This path is a potential solution to the minimization problem. For our case, all labels created for ending node 0 are directly stored if and only if they satisfy the following conditions: i) the criterion of negative reduced cost, i.e, $\tilde{c}_{\delta i} < 0$, and ii) all C orders (for cases where $C \neq \emptyset$) are included in the solution.

In order to avoid enumerating all feasible paths, dominance rules are applied to eliminate (discard) labels that are not Pareto-optimal, while maintaining optimality (Chabrier, 2006). To do so, given two labels $\Lambda_{\delta' i}$ and $\Lambda_{\delta'' i}$ representing two different partial paths δ' and δ'' ending at the same vertex i, we allow $\Lambda_{\delta' i}$ to dominate $\Lambda_{\delta'' i}$ (i.e. the latter is discarded), if $\Lambda_{\delta' i} \leq \Lambda_{\delta'' i}$ (component-wise) and the inequality is strict for at least one component. Note that the resource constraint ensuring that all C orders are included in the path does not participate; instead, component $\bar{\bar{c}}_{\delta i}$ ensures optimality by including all C orders on a path, when needed.

Finally, in order to speed up the solution process, we use several acceleration techniques from the literature (see Table 1). The proposed framework provides solutions in a rather efficient time frame (for cases where TWs are present) as compared to a typical VRPTW. This is due to the existence of committed orders that should be served only by the vehicles to which they have been assigned originally, which enables the decomposition of the pricing problem to multiple independent SPs. Furthermore, this requirement strengthens the dominance criteria, which discard a large amount of columns because of the label component $\bar{\bar{c}}_{\delta i}$.

Table 1. Acceleration techniques used

Acceleration Technique	Reference
Unreachable nodes	Feillet *et al.* (2004; 2005), Chabrier (2006)
Limited Discrepancy Search (LDS)	Feillet *et al.* (2005), Athanasopoulos (2011)
Buckets / Storing Processed Labels	Larsen (2001), Chabrier (2006), Athanasopoulos (2011)
Early Termination Criterion	Larsen (2001), Chabrier (2006)
Parallel Implementation of subproblems	

Obtaining integer solutions (B&P)

If no integer solutions are obtained by the CG algorithm, a branch-and-bound (B&B) search scheme is used. The CG algorithm is used to compute lower bounds at each node of the B&B search tree, and we branch on the most fractional binary arc flow variable $\psi_{ij}, (i,j) \in A$; i.e. $\psi_{ij} = \sum_{r \in \Omega'} a_{ijr} y_r$, where a_{ijr} denotes a binary variable equal to 1 if and only if route r traverses arc (i,j) (see Savelsbergh and Sol, 1998; Desrochers *et al.*, 1992 and Desaulniers *et al.*, 1998).

In practice, given that we seek efficient solutions in limited time, one may allow B&B to explore unsolved nodes within an acceptable time limit (e.g. one minute); if an integer solution has not been obtained by that time, the B&B problem may be solved, using only the columns (routes) present in the current RMP. The latter can be calculated using the default integer programming methods of the CPLEX environment.

A heuristic B&P approach

Given the requirements for time efficiency of the solution process, and that the latter should be able to address practical cases with extended solution space (e.g. without time windows), we propose a heuristic procedure to generate negative cost columns to enter the $RMP,$ instead of solving the ESPPRC to optimality.

Generating columns for vehicles en-route

To generate new columns with negative reduced costs, we use a local search procedure to modify certain columns in the initial basis (Ω_k columns). We modify columns with zero reduced cost in order to drive this cost to negative values. The modification is performed using a *cheapest insertion algorithm*, which tries to incorporate in a least-cost fashion each order in F to each column in the initial basis (the latter correspond to vehicles *en-route* assigned to serve C orders). For this insertion we use the flexible order-column combination that results in the minimum reduced cost, as computed from the difference between the reduced costs prior and after the order insertion. Let c_s be the cost (distance) of a column $s(k), k \in K$ prior to the insertion of flexible order f ,and c_{s_f} the post-insertion cost. Also let RC_s and RC_{s_f} the respective reduced costs. Hence, the insertion criterion is provided by the following equation,

$$\begin{aligned} G(s,f) = RC_{s_f} - RC_s = \left(c_{s_f} - \sum\nolimits_{a\in S_f} \pi_a\right) - \left(c_s - \sum\nolimits_{a\in S} \pi_a\right) = \\ = \left(c_{s_f} - \sum\nolimits_{a\in S} \pi_a - \pi_f\right) - \left(c_s - \sum\nolimits_{a\in S} \pi_a\right) = c_{s_f} - c_s - \pi_f \end{aligned} \qquad (18)$$

where $S = \{s(1), s(2), \dots, s(K)\}$ denote the set of Ω_k columns in the optimal basis corresponding only to the columns for vehicles *en-route* at replanning cycle T_ℓ, π_α and π_f correspond to the dual prices of each order $a \in S$ and $f \in F,$ respectively. Using this criterion, each order in F is tested for insertion in all possible positions of each column of the initial basis. Columns with negative reduced cost that are generated during the iterations of this process are maintained as candidates in a pool of columns to be added to the RMP. This operation is terminated when no negative cost columns can be found, or when all orders in F have been tested for insertion.

The pseudocode of the algorithm is given in Figure 3.

Algorithm 1: Heuristic for generating columns for vehicles en-route

```
Ω'' = ∅; // New set of columns generated by the procedure
G = ∅;  // Matrix that stores insertion costs of order f to each column s
RC = ∅  // Matrix that stores the reduced cost of order f to each column s
While F ≠ ∅
   For each column s(k) ∈ S, k ∈ K do
       c_{s(k)} → Cost of column s(k)
       For each order f ∈ F do
          c̄_f^{s(k)} = Inf // Best cost of including order f in column s(k)
          For every feasible arc n in path s for inserting order f ∈ F do
             Apply insertion of order f in path s
             [path(n)] = Apply 2-opt improvement on this temporary path
             s_k^{f_n} = path(n)  // column s with order f on arc n after 2-opt
             c_k^{f_n} = Cost of path(n)
             Compute reduced cost r_k^{f_n} = c_k^{f_n} − Σ_{α∈s(κ)} π_α
             If r_k^{f_n} < 0 then
                Ω'' = Ω'' ∪ {s_f^n}
             End
             If c_k^{f_n} < c̄_f^{s(k)} then
                 G(s,f) = (c_k^{f_n} − c_{s(k)}) − π_f
                 RC(s,f) = r_{s_f^n}
                 c̄_f^{s(k)} = c_k^{f_n}
             End
          End
       End
   End
   If any(RC) ≥ 0
      terminate procedure and return Ω''
   Else
      Find s(k') and p' ∈ N such that g_{s_{p'}} = min(G{s}{n} | ∀ s ∈ S, ∀ n ∈ F)
      Update column s(k') = s(k') ∪ {p'}  // in the best feasible place
      Update matrices G and RC
      Update set of F orders → F = F\{p'}
   End
End
```

Figure 3. Pseudo-code of heuristic approach for generating columns for vehicles en-route

Generating columns for vehicles located at the depot

The solution of the Ψ_{K+1} subproblem for generating columns for vehicles located at depot is possible within the framework described in Section 3.1, since the resources (available remaining time horizon, capacity, etc.) are relatively limited at each replanning cycle. This, of course, holds when the number of orders in the F set is relatively limited. Thus, if the number of F orders is less than or equal to a reasonably small number, e.g. $|F| \leq \Xi$, we use the label correcting algorithm as described in Section 3.1.3. For $|F| > \Xi$, we apply the same algorithm but we exclude path elementarity from the dominance criteria; i.e. a label can be eliminated by another label even if the dominator is not a subtour of the dominated one. This may speed up the solution process, since it eliminates a significant number of columns, but cannot ensure that all feasible Ω_p will be generated, and the optimum will be reached.

Replanning strategies for the VRPDP

We introduce the concept of "*replanning strategy*" that comprises the combination of i) the length of a replanning cycle (i.e. when to replan, referred to as *replanning policy*) and, ii) the part of the plan that is released for implementation (referred to as *implementation tactic*).

Defining the length of replanning cycle(s) is a significant decision; *long* replanning cycles may limit the dispatcher's options (since a larger portion of the route has been completed during previous replanning cycles and fewer options are available for incorporating the newly arrived requests), while *short* replanning cycles may not take advantage of combinations of newly arrived orders. We explore various replanning policies depending on number of DRs that has arrived between two successive replanning instances; that is

- *Single-request replanning (SRR):* Replan upon the arrival of each DR
- *N-request replanning (NRR):* Replan after the arrival of a predefined number (N>1) of DRs

Another type of policy may prescribe replanning at multiples of a fixed period (Fixed-Time Replanning – *FTR policies*). In case DRs arrive uniformly, FTR policies are equivalent to NRR. For cases for which DRs arrive in a non-uniform fashion, our experiments have indicated that FTR policies exhibit inferior behavior compared to their NRR-counterparts. For this reason, we don't deal with such policies in the remainder of this paper.

In addition to the above, we explore two tactics to implement the new plan:

- *Full-Release tactic (FR):* All replanned DRs are released to the fleet immediately for implementation and they cannot be reassigned at later replanning cycles (see the discussion of Section 2.2 on relevant practical cases, in which this tactic is applicable).

- *Partial-Release tactic (PR):* Only the DRs scheduled for implementation prior to the next replanning cycle are released. The remaining DRs are re-considered in the next replanning cycle.

Theoretical insights for replanning strategies

It is reasonable to expect that the PR tactic is superior to FR. Below we examine in which cases this holds assuming the following conditions:

- All orders are served in the final solution
- Vehicles located at the depot are eligible to be dispatched at any $\ell > 0$
- Both release tactics are compared under the same number of replanning cycles
- An optimal method is used for replanning.

We should initially note that in the trivial case of a single-vehicle, both release tactics lead to identical results. This is due to the fact that although FR commits flexible orders for the next replanning cycles, the sequence of customer service within the route of each vehicle (the only one in this trivial case) is not committed; thus, the replanning state is the same for both tactics, which generate identical optimal solutions.

<u>Claim 1</u>: *It is guaranteed that the cost of the overall solution (for $[T_0, T_{max}]$) obtained under the PR tactic is always lower than or equal to the cost of the solution obtained under the FR tactic, for $\ell < 3$.*

Consider a simple example with $L = 2$ replanning cycles, K vehicles scheduled to be dispatched at time T_0 and V available vehicles at the eligible to be dispatched at any $\ell > 0$. Let K_ℓ denote the vehicles en-route considered at every ℓ (which comprise the remaining K vehicles en-route and the ones dispatched during previous replanning cycles, which have not completed their assignments). Let $RP(\omega, \ell)$ denote the replanning problem for each implementation tactic $\omega \in \{FR, PR\}$ with corresponding routing overall cost $O(\omega, \ell)$. Note that $O(\omega, \ell) = O_p(\omega, \ell) + O_f(\omega, \ell)$, where $O_p(\omega, \ell)$ denotes the cost of the already completed portion of the routes up to T_ℓ, and $O_f(\omega, \ell)$ the cost of the solution for $[T_\ell, T_{max}]$.

The feasible space of each $RP(\omega, \ell)$, $\ell > 0$ may be formed by considering (a) all feasible combinations of assigning the flexible orders among the vehicles en-route (K_ℓ subproblems) and the vehicles located at depot ($K_\ell + 1$ subproblem), and (b) for each subproblem, all feasible customer sequences of the corresponding set.

The problem solution at each ℓ is affected by the replanning state which is comprised of i) the set of committed orders $C_k(\omega,\ell), k=\{1,2,\dots,K_\ell,K_\ell+1\}$, ii) the set of flexible orders $F(\omega,\ell)$, and iii) the current location of the vehicle(s). Note that there are no committed orders for the $K_\ell+1$ subproblem, i.e. $C_{K_\ell+1}(\omega,\ell)=\emptyset$.

During $\ell=1$ both tactics consider the same replanning state and $O_p(FR,1)=O_p(PR,1)$; therefore, the related re-planning problems (for PR and FR) are identical with identical solutions, and $O(FR,1)=O(PR,1)$.

During $\ell=2$ (at time T_2), it holds that the current locations of the vehicles en-route at T_2 are identical for both tactics and $O_p(FR,2)=O_p(PR,2)$. For each tactic, the related problems consider the sets of orders $N_k(\omega,2)=C_k(\omega,2)\cup F(\omega,2), k=\{1,2,\dots,K_\ell,K_\ell+1\}$; more explicitly:

- *FR-tactic*: $N_k(FR,2)=C_k(FR,2)\cup F(FR,2)=[C_0^k(2)\cup F_k'(2)]\cup F_0(2)$, where $C_0^k(2)$ denotes the set of unserved offline requests and $F_k'(2)$ the subset of DRs arrived during $[T_0,T_1]$ and assigned to vehicle k but not yet served. $F_0(2)$, denotes new orders that arrived during $[T_1,T_2]$. Note that $N_{K_\ell+1}(FR,2)=F_0(2)$.
- *PR-tactic*: $N_k(PR,2)=C_k(PR,2)\cup F(PR,2)=C_0^k(2)\cup[F'(2)\cup F_0(2)]$, where $F'(2)$ denotes all orders arrived during $[T_0,T_1]$ and not yet served. Also, $N_{K_\ell+1}(PR,2)=F'(2)\cup F_0(2)$.

Since $F'(2)=\bigcup_{k\in K}F_k'(2)$, it is clear that $N_k(FR,2)\subseteq N_k(PR,2), \forall k\in\{1,2,\dots,K_\ell,K_\ell+1\}$. Thus, the feasible subspace corresponding to the PR tactic is a superset of that of the FR tactic, and, $O_f(FR,2)\le O_f(PR,2)$; consequently, $O(PR,2)\le O(PR,2)$.

Based on the above, up to $\ell=2$, the PR tactic will always provide superior or equivalent results. For $\ell>2$, however, such a comparison between the two tactics is not possible, since i) the state of the system at each replanning event is not the same and, ii) the cost $O_p(\omega,\ell)$ up to that event is, in general, different for each tactic.

Claim 2: *For any $\ell\ge 3$, and if more than one vehicles are involved (either dispatched at $\ell=0$ or during any $\ell>0$), it is not guaranteed that the overall routing cost under the PR tactic is lower or equal than the one obtained by the FR tactic.*

We will show this claim through a counter-example as illustrated in Figure 4. At $\ell=0$, two vehicles are planned to execute four (4) deliveries (customers 1, 2, 3, 4). During the course of implementing this plan, three (3) DRs arrive and should be incorporated in the plan (customers 5, 6, 7). Replanning happens upon arrival of each DR. Table 2 provides the coordinates of all customers; the depot (denoted by node 0) is located at point (0,0).

Table 2. Customers' coordinates for counter-example of Claim 2

Customer ID	Coordinates (X,Y)
1	$(5,2.5)$
2	$(10,5)$
3	$(10,-5)$
4	$(5,-2.5)$
5	$(10,0)$
6	$(10,10)$
7	$(5,0)$

Figure 4 illustrates two states per implementation tactic for $\ell > 0$; the state prior to replanning ("*Before*") and the state after replanning ("*After*").

At $\ell = 1$, both implementation tactics provide the same routing plans, as expected. At $\ell = 2$, order (6) has arrived. The F set for FR comprises only the new order 6 while for PR the F set includes all DRs not yet served ($\{5,6\}$). PR generates a superior solution, since the flexible order set is a superset of the one considered by FR. However, for $\ell = 3$, the (initial) replanning states of the two implementation tactics are different. Thus, the two generated plans are different, and, in this case, the overall solution of the FR tactic is superior to that of the PR tactic. Table 3 presents the final routing costs after each replanning cycle for both implementation period ($[T_\ell, T_{max}]$) and the entire planning horizon ($[T_0, T_{max}]$).

Table 3. Planned and actual routes under both tactics for three replanning cycles (solid line is the planned route, dotted line the executed route)

	$\boldsymbol{FR - tactic}$		$\boldsymbol{PR - tactic}$	
	$\boldsymbol{Cost\ [T_\ell, T_{max}]}$	$\boldsymbol{Cost\ [T_0, T_{max}]}$	$\boldsymbol{Cost\ [T_\ell, T_{max}]}$	$\boldsymbol{Cost\ [T_0, T_{max}]}$
$\boldsymbol{\ell = 0}$	44.72	44.72	44.72	44.72
$\boldsymbol{\ell = 1}$	31.77	48.54	31.77	48.54
$\boldsymbol{\ell = 2}$	36.18	58.54	35.32	57.68
$\boldsymbol{\ell = 3}$	25.59	58.54	27.23	59.59

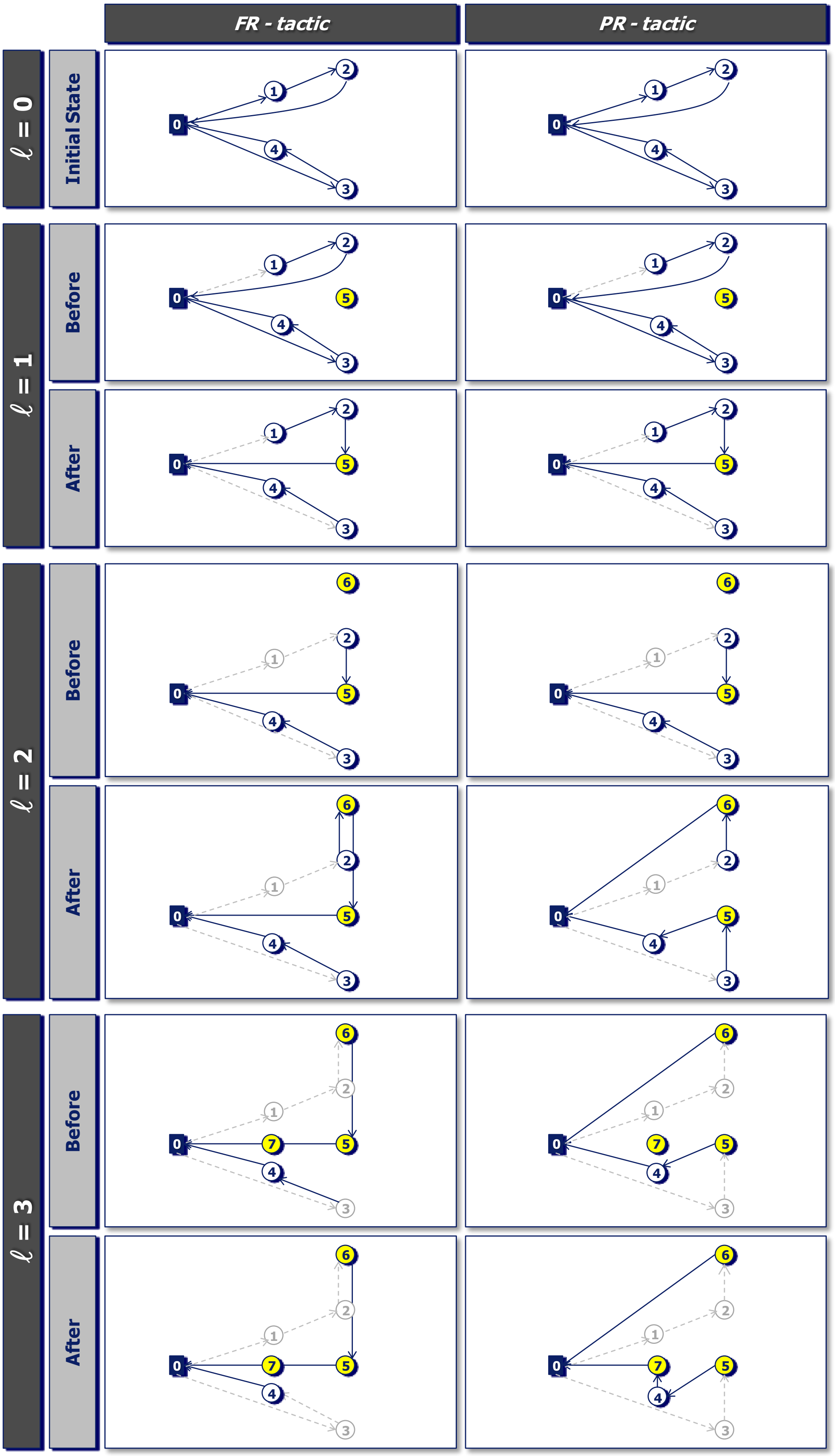

Figure 4. Counter example for Claim 2

Computational experiments

We study below the performance of the proposed replanning strategies (policies and tactics) through extensive experiments.

Experimental setup

We have used the well-known benchmark instances of Solomon (1987). For each instance, we employed the entire 100-customer dataset. Specifically, we have focused our investigation on the impact of the following parameters on the effectiveness of the various operating strategies:

- Customer *geographical distribution*. We used the coordinate patterns of Solomon's datasets R1 and C1 (uniformly distributed customers and clustered customers, respectively).
- Customer *time-windows (TW)*. Tight TW cases may limit the impact of the replanning strategies, since the solution space is significantly limited. For our experiments, the TW range from 5% to 50% of the total planning horizon (T_{max}). Table 4 shows the Solomon instances employed, along with relevant TW information for each instance. Note that the "very wide" TW case was considered only for the uniform distribution instances (R), since the Solomon C1 datasets do not include such a case. In addition, we included datasets vrpnc8 and vrpnc14 of Christofides *et al.* (1979) that have no TW for the uniform and clustered cases, respectively, but use the same customer coordinates as the Solomon datasets. For notation purposes, those datasets will be referred to as R100 and C100, respectively.

Table 4. Time-window cases employed

Instance	R101	R105	R109	R112	C101	C105	C109
TW length	10	**30**	Avrg: 59 Min: 37 Max: 83	Avrg: 118 Min: 73 Max: 166	Avrg: 61 Min: 37 Max: 89	Avrg: 122 Min: 75 Max: 177	360
TW length (% of Tmax)	4%	14%	~26%	~51%	5%	10%	29%

- The *degree of dynamism* (dod) (Larsen, 2002), which is defined as the percentage of the number of DRs over the total number of orders in a given dataset. For each instance described previously, we assumed cases of $weak - dod$ (25% DRs), $moderate - dod$ (50% DRs) and $strong - dod$ (75% DRs).

Offline requests were randomly selected from the entire 100-customer dataset and the remaining customers form the set of DRs. Note that only for the instances that involve TW, we skewed the selection of offline orders towards those with early TW opening.

Based on the above, we constructed 27 different cases as shown in Table 5. For each of the 27 cases, we generated 10 different instances (different selection of offline requests), resulting in a total of 270 test instances.

Table 5 summarizes the notation that characterizes each test case; R101-25% refers to an instance with uniform customer geographic distribution, tight time-windows, and 25% dod.

Table 5. Test scenarios

Parameter	Description	Notation	# of levels
Geographical distribution	Uniform	R	2
	Clustered	C	
Time Windows	Tight	101	5
	Medium	105	
	Wide	109	
	Very wide*	112	
	No TW	100	
Degree of Dynamism (dod)	Weak	25%	3
	Moderate	50%	
	Strong	75%	

**used only for uniform geographical distribution (R)*

The initial solutions were obtained as follows:

- For all problems up to wide TWs by the B&P algorithm described in Section 3.1
- For the remaining problems (R112, R100, C100), by a Clark & Wright savings heuristic (Clark and Wright, 1964) followed by a Reactive Tabu Search metaheuristic (Osman and Wassan, 2002) as a post-optimization process.

Table 6 presents the algorithm employed per test case for the solution of the replanning problem.

Table 6. Algorithm employed for solution of each test instance (*B&P = Branch-and-Price, Heur = heuristic*)

DoD \ Instance	R101	R105	R109	R112	R100	C101	C105	C109	C100
25%	B&P	B&P	B&P	HEUR	HEUR	B&P	B&P	HEUR	HEUR
50%	B&P	B&P	B&P	HEUR	HEUR	B&P	B&P	HEUR	HEUR
75%	B&P	B&P	B&P	B&P	HEUR	B&P	B&P	B&P	HEUR

Prior to presenting and discussing the results of the experimental investigation, we discuss the performance of the two algorithms of Table 6. Figure 5 compares the exact and the heuristic methods in terms of solution quality and computational effort. The set-up for this investigation involved $RY-50\%$ datasets, where $Y=(101,105,109,112,100)$. We applied three (3) replanning cycles (i.e. replanned after 18 DRs have been received), and we tested these cases under the PR tactic (because PR considers more flexible orders during each replanning cycle and, thus, it is more time-consuming than FR). Figure 5 presents the average results for 10 replicates per problem w.r.t.: i) the average computational effort per replanning cycle, and ii) the average cost difference of the heuristic method over the exact method.

The B&P algorithm seems to be very efficient in narrow to medium TW-cases compared to HEUR; this fact reverses when TWs are wider or absent. In these latter cases, HEUR seems to be much faster. In terms of solution quality, HEUR has an average deviation of 1.4% w.r.t. exact solution in terms of total routing cost (for all cycles). Note that HEUR may, in some cases, provide better solutions in the overall dynamic problem despite the fact that the solutions at each replanning cycle are slightly inferior (see also Claim 2).

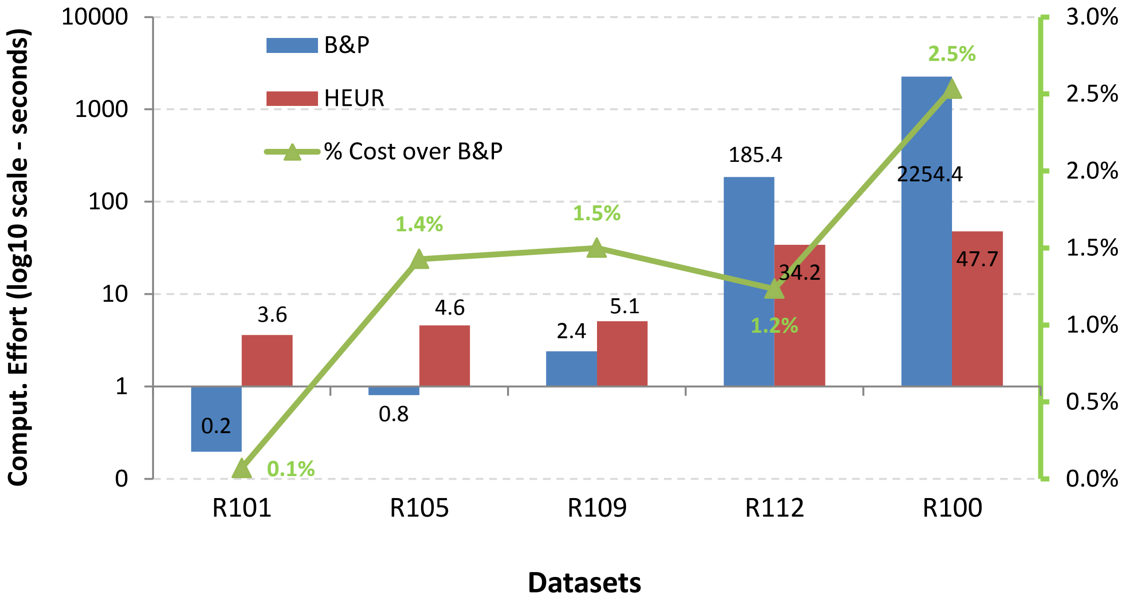

Figure 5. Performance of exact (B&P) and heuristic (HEUR) method

Experimental investigation

We assessed the proposed replanning strategies in two steps: i) Initially we studied their overall performance; ii) subsequently, we drilled down on how key parameters affect the performance of these strategies. All experiments were conducted using a Quad-Core Intel i7 processor of 2.8GHz and 4GB of RAM.

For NRR, we used $N = 1, 0.1\hat{N}, 0.2\hat{N}, 0.33\hat{N}$ (where $\hat{N}$ is the total number of DRs) to yield the NRR-1, NRR-2 and NRR-3 cases. Each policy was tested under the FR and PR release tactics, resulting to a total of eight (8) strategies for each test instance.

Performance of replanning strategies (tactic-policy combination)

Figure 6 presents the average performance of FR and PR under all policies for the various geographical distributions and TW cases. Performance is assessed in terms of excess routing cost per DR, i.e. the final routing cost minus the initial routing cost at time T_0 divided by the number of DRs involved in each dataset. (Note that since the final solution includes all DRs, the comparison of the total excess cost is equivalent to the comparison of the unit cost.) The Figure also shows the cost difference in % $100|(C_{FR} - C_{PR})/C_{PR}|$, where C_{FR} and C_{PR} denote the excess routing costs per DR of tactics FR and PR respectively.

As expected, the PR tactic outperforms FR in all cases, leading to an overall average improvement of about 15%. This improvement increases with the width of TW.

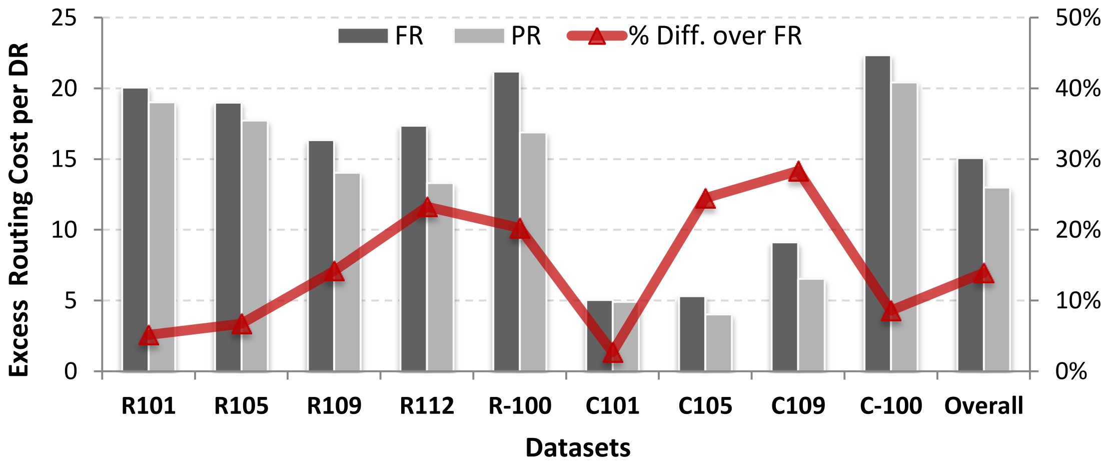

Figure 6. Overall performance of implementation tactics (average of all tactics per data set)

Figure 7 presents the performance of each replanning strategy (policy-tactic combinations) in terms of routing cost per DR. The average is taken over all datasets and the various degrees of dynamism. From this figure, it is clear that the combination SRR-PR has the best overall performance in terms of routing cost per DR. Additionally, the Figure shows that the performance of each tactic is related to the frequency of replanning; the solution for the FR tactic seems to be less efficient when replanning is applied either very frequently (SRR) or infrequently (NRR-3). A possible cause of this may be that frequent replanning (i.e. at every request) does not take advantage of combinations of newly arrived DRs. In the case of infrequently replanning, a larger portion of the route has been completed and fewer options are available for incorporating the newly arrived DRs.

PR seems to be more efficient for shorter replanning cycles; that is because i) short route portions have been completed when replanning is applied, allowing for more options, and, ii) DRs that are not planned to be served until the next replanning timestamp are not committed, providing higher possibilities for DR combinations. The cost difference between the two tactics is decreasing as the number of elapsed DRs per replanning cycle, as expected.

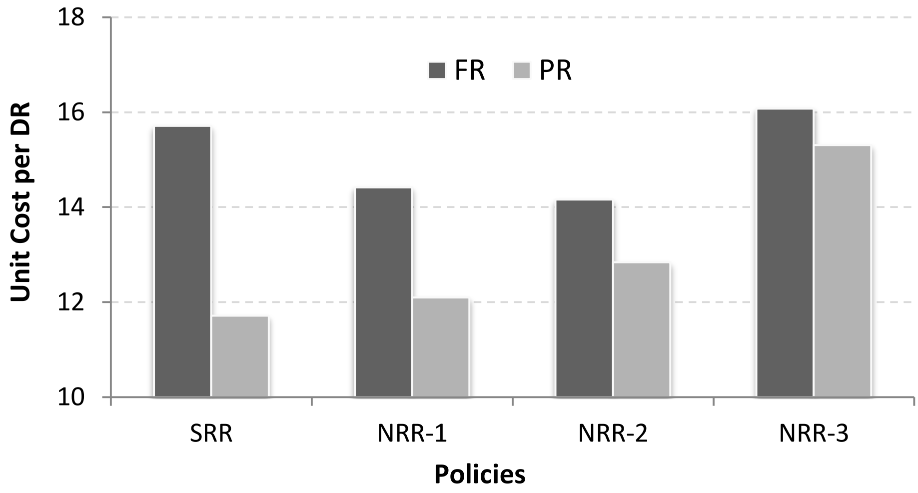

Figure 7. Average Performance of all strategies over all test instances

Parametric study of strategies

We investigated the effects of the following factors (parameters) on performance: i) the time-window pattern, and ii) the degree of dynamism.

Factor Analysis I: Performance of replanning strategies w.r.t. time-windows

Figures 8 and 9 show the overall performance of all policies under various TW patterns for the FR and PR tactic, respectively. The y-axis provides the percentage difference over the best reported value of a policy-tactic combination (i.e. over all strategies). For the FR case, SRR and NRR-3 have a strong correlation with the length of the TW: the performance of SRR deteriorates when TW lengths increase, while the performance of NRR-3 improves significantly. That is because wide TWs allow for more combinations when batch replanning takes place. However, in non-TW cases, very infrequent replanning deteriorates solution quality. Medium-interval replanning cycles (NRR-1 and NRR-2) display almost similar behavior for all TW patterns.

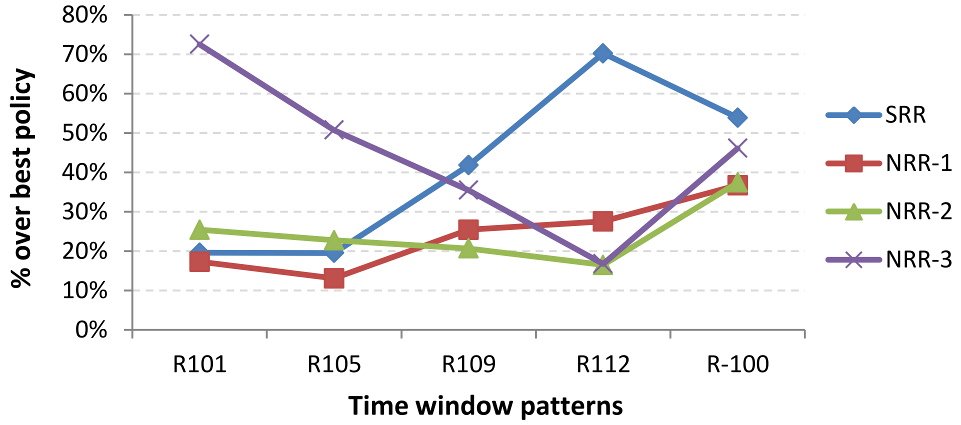

Figure 8. Performance of all policies under the **FR** tactic for various TW patterns

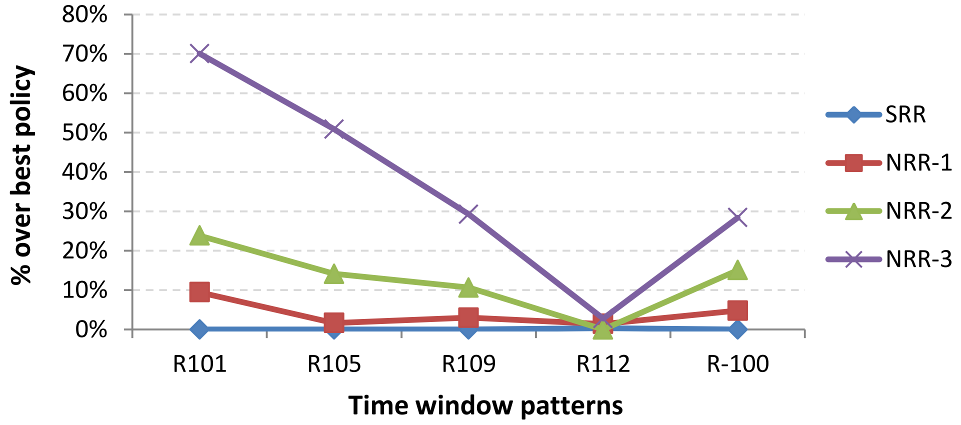

Figure 9. Performance of all policies under the **PR** tactic for various TW patterns

For the PR tactic, all policies follow similar behavior; that is, more frequent replanning (SRR and NRR-1) always yield better results. This is possibly due to a similar argument to Claim 1; that is, frequent replanning allows re-allocation of orders in a more flexible manner, and, this enhances the feasible space. This of course is true for each replanning cycle, but may not hold for the entire problem, if this involves several replanning cycles.

Factor Analysis II: Performance of replanning strategies w.r.t. the degree of dynamism

Figures 10 and 11 present the performance of each policy w.r.t. the degree of dynamism (dod) for the FR and PR tactics, respectively. Note that this is the average performance over all parameter values (except, of course, dod) and is provided as the percentage difference over the best reported value of policy-tactic combination. For FR, medium-interval replanning cycles favor solution quality for all dod; very frequent or infrequent replanning display inferior performance, which is pronounced in cases of strong-dod.

For the PR tactic (Figure 11), SRR and NRR-1 outperform all other policies for all cases of dod. Overall, the performance of policies with wider replanning intervals seems to deteriorate with the increase of degree of dynamism. This may be attributed to the fact that in strong dynamic environments (limited number of offline customers) many vehicles are dispatched from the depot during replanning due to the high number of new DRs, causing additional cost because of the empty mileage (travel from and to depot). Infrequent replanning in such cases causes vehicles en-route to return to depot at an early stage (because of the limited number of committed orders) and new vehicles to be frequently dispatched in order to cover the high demand.

For the reason mentioned above, in environments with strong dynamics, SRR under PR might perform worse than a moderately frequent replanning frequency. This is indeed validated by Figure 12, in which we analyze the performance of policies under the PR tactic for the R112 case (very wide TW). Figure 12 shows that moderately frequent polices (NRR-1, NRR-2) can perform slightly better (on average) than SRR for environments with moderate to strong dynamics (note that SRR was found to be the best policy only in 4 out of 10 replicates on this dataset).

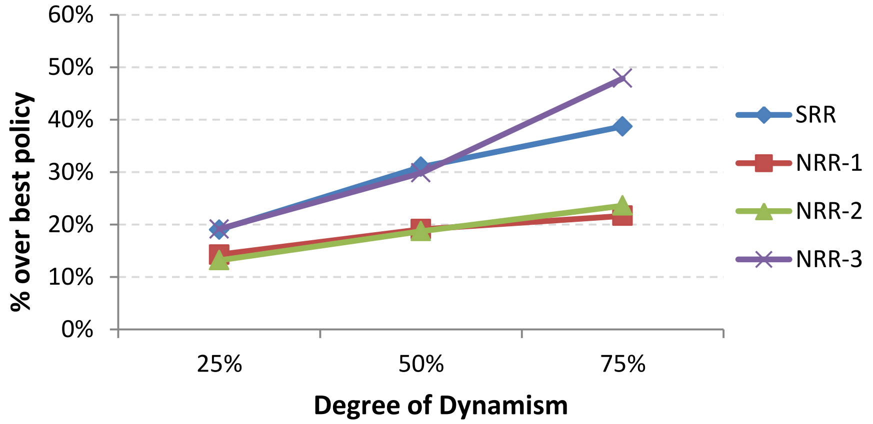

Figure 10. Performance of policies under **FR** tactic for various degrees of dynamism

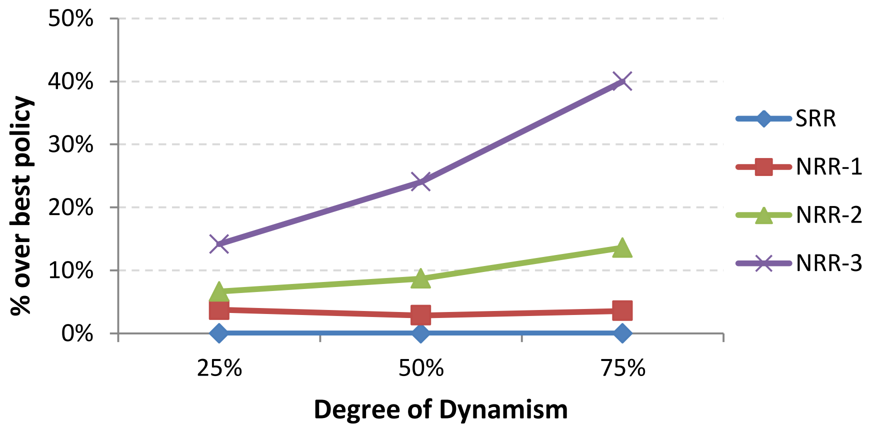

Figure 11. Performance of policies under **PR** tactic for various degrees of dynamism

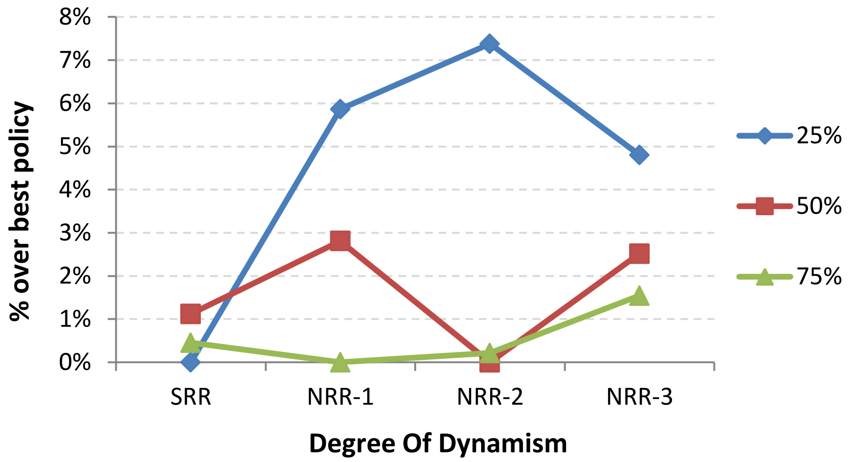

Figure 12. Performance of policies under PR tactic for R112 and various degrees of dynamism

Conclusions

We have described a dynamic *one-to-many-to-one* pickup and delivery problem, referred to as the Vehicle Problem with Dynamic Pickups (VRPDP) that seeks to assign in the most efficient way dynamic pickup requests that arrive in real-time fashion while a predefined distribution plan is being executed. We addressed the VRPDP through iterative replanning. In addition to defining the replanning model, we drilled-down to significant aspects concerning the replanning process, i.e. i) *how* to replan, ii) *when* to replan, and iii) *what* part of the new plan to communicate to the drivers.

Regarding "*how to replan*", we propose a Branch-and-Price (B&P) approach. For cases of high complexity (e.g. without time-windows), we propose a novel insertion heuristic that is employed within a column generation framework. The latter provides efficient solutions with a limited deviation from the optimal solution (1.4% on the average).

Regarding the "when" and "how" to replan, we presented and analyzed typical replanning policies found in practice, i.e.: i) replanning upon the arrival of each DR, ii) replanning after a certain number of DRs have been received. In addition, we investigated the effect of two implementation tactics: i) immediate release of all DRs for implementation (FR) and, ii) release of only those DRs that are scheduled for implementation prior to the next replanning cycle (PR). We provided theoretical insights regarding the expected behavior of those tactics and we showed through extensive experimentation that replanning upon the arrival of each DR under the PR tactic provides superior results on the average. However, this policy seems to be the least favorite, when the FR tactic is employed.

Furthermore, our experimentation under various operating scenarios has shown the following: i) When the business case allows it, one should always replan under PR tactic in as short replanning intervals as possible. ii) When FR tactic is unavoidable due to the characteristics of the practical environment, one should prefer replanning over short to medium intervals for cases with tight to medium TW, and medium to large intervals for wider TW cases. iii) When replanning in environments with strong dynamism, medium interval policies (regardless of tactic) seem to provide the safest option; iv) finally, one should always prefer size-driven policies (instead of time-driven ones), in order to avoid the negative effect of those policies in environments where DRs may arrive in a non-uniform fashion.

References

Angelelli, E., Bianchessi, N., Mansini, R., Speranza, M.G. (2009). Short Term Strategies for a Dynamic Multi-Period Routing Problem, Transportation Research Part C 17, pp. 106-119

Athanasopoulos, T. (2011). The multi-period vehicle routing problem and its applications, PhD Dissertation, University of the Aegean, Department of Financial and Management Engineering, Chios, Greece

Barnhart, C., Johnson, E.L., Nemhauser, G.L., Savelsbergh, M.W.P., Vance, P.H. (1998). Branch-and-price: Column generation for solving huge integer programs. Operations Research 46(3), pp. 316–329.

Bent, R. W., Van Hentenryck P. (2004). Scenario-based planning for partially dynamic vehicle routing with stochastic customers, Operations Research 52, pp. 977–987.

Berbeglia, G., Cordeau, J.-F., Gribkovskaia, I., Laporte, G. (2007). Static pickup and delivery problems: A classification scheme and survey, TOP 15, pp. 1–31.

Berbeglia, G., Cordeau, J.-F., Laporte, G. (2010). Dynamic pickup and delivery problems, European Journal of Operational Research 202, pp. 8–15.

Chabrier, A. (2006). Vehicle routing problem with elementary shortest path based column generation, Computers & Operations Research 33, pp. 2972–2990.

Chen, Z.L., Xu, H. (2006). Dynamic column generation for dynamic vehicle routing with time windows, Transportation Science 40, pp. 74–88.

Christofides, N., Mingozzi, A., Toth, P. (1979). The vehicle routing problem. In: Christofides, N., Mingozzi, A., Toth, P., Sandi C. (eds) Combinatorial Optimization. John Wiley, Chichester, pp. 315-339

Cordeau, J.F., Desaulniers, G., Desrosiers, J., Solomon, M.M., Soumis, F. (2002). VRP with time windows. In: Toth, P., Vigo, D. (eds) The Vehicle Routing Problem. SIAM Monographs on Discrete Mathematics and Applications vol 9. SIAM, Philadelphia, PA, pp. 175–193.

Cordeau, J.-F., Laporte, G., Potvin, J.-Y., Savelsbergh, M.W.P., (2007). Transportation on demand. In: Barnhart, C., Laporte, G. (Eds.), Handbooks in Operations Research and Management Science: Transportation, vol. 14. North-Holland, Amsterdam, pp. 429–466.

Desaulniers, G., Desrosiers, J., Ioachim, I., Solomon, M.M., Soumis, F., Villeneuve, D. (1998). A unified framework for deterministic time constrained vehicle routing and crew scheduling problems. In: Crainic T, Laporte, G. (eds) Fleet Management and Logistics. Kluwer Academic Publisher, Boston, Dordrecht, London, pp. 57–93.

Desaulniers, G., Desrosiers, J., Erdmann, A., Solomon, M.M., Soumis, F. (2002). Vehicle routing problem with pickup and delivery. In: Toth, P., Vigo, D. (eds) The Vehicle Routing Problem. SIAM Monographs on Discrete Mathematics and Applications vol 9. SIAM, Philadelphia, PA, pp. 225–242

Desaulniers, G., Desrosiers, J., Solomon, M.M. (eds) (2005) Column Generation, No. 5. In: GERAD 25th Anniversary, Springer

Desrochers, M., Desrosiers, J., Solomon, M. (1992). A new optimization algorithm for the vehicle routing problem with time windows. Operations Research 40, pp. 342–354.

Desrosiers, J., Lübbecke, M. (2005). A primer in column generation. In Desaulniers, G., Desrosiers, J., Solomon, M.M. (Eds.), Column Generation, pp. 1–32, Springer, New York.

Dethloff, J. (2002). Relation between vehicle routing problems: An insertion heuristic for the vehicle routing problem with simultaneous delivery and pick-up applied to the vehicle routing problem with backhauls, Journal of the Operational Research Society 53, pp. 115–118.

Feillet, D., Dejax, P., Gendreau, M., Gueguen, C. (2004). An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems. Networks 44 (3), pp. 216-229.

Feillet, D., Gendreau, M., Rousseau, L.M. (2005). New refinements for the solution of vehicle routing problems with branch and price, Technical Report C7PQMR PO2005-08-X, Center for Reasearch on Transportation, Montreal.

Gendreau, M., Guertin F., Potvin J. Y., Taillard É. (1999). Parallel tabu search for real-time vehicle routing and dispatching, Transportation Science 33, pp. 381–390.

Gendreau, M., Guertin, F., Potvin, J.-Y., Séguin, R. (2006). Neighborhood search heuristics for a dynamic vehicle dispatching problem with pick-ups and deliveries, Transportation Research Part C 14, pp. 157–174.

Ichoua, S., Gendreau M., Potvin J. Y. (2000). Diversion issues in realtime vehicle dispatching, Transportation Science 34, pp. 426–438.

Ichoua, S., Gendreau, M., Potvin, J.-Y. (2006). Exploiting knowledge about future demands for real-time vehicle dispatching. Transportation Science 40, pp. 211–225.

Irnich, S., Desaulniers, G. (2005). Shortest path problems with resource constraints. In: Desaulniers, G., Desrosiers, J., Solomon, M.M. (Eds.), Column Generation, Springer, New York, pp. 33–65.

Larsen, A. (2001). The dynamic vehicle routing problem, Ph.D. Thesis, Institute of Mathematical Modelling, Technical University of Denmark.

Larsen, A., Madsen O. B. G., Solomon M. M. (2002). Partially dynamic vehicle routing—Models and algorithms, Journal of the Operational Research Society 53, pp. 637–646.

Larsen, A., Madsen O. B. G., Solomon M. M. (2004). The a-priori dynamic travelling salesman problem with time windows, Transportation Science 38, pp. 459–572.

Larsen, J. (2001). Parallelization of the Vehicle Routing Problem with Time Windows, PhD thesis, Informatics and Mathematical Modelling, Technical University of Denmark.

Osman, I. H., Wassan, N. A. (2002). A reactive tabu search meta-heuristic for the vehicle routing problem with back-hauls, Journal of Scheduling 5, 263-285

Parragh, N. S., Doerner, K. F., Hartl, R.F. (2008). A survey on pickup and delivery problems. Part I: Transportation between customers and depot,

Powell, W. B. (1996). A stochastic formulation of the dynamic assignment problem, with an application to truckload motor carriers, Transportation Science 30, pp. 195–219.

Psaraftis, H.N. (1988). Dynamic vehicle routing problems. In: Golden, B.L., Assad, A.A. (Eds.), Vehicle Routing: Methods and Studies. North-Holland, Amsterdam, pp. 223–248.

Ropke, S., Cordeau, J.-F., Laporte, G. (2007). Models and Branch-and-Cut Algorithms for Pickup and Delivery Problems with Time Windows, Networks 49 (4), pp. 258-272.

Ropke, S., Pisinger, D. (2006). A unified heuristic for a large class of vehicle routing problems with backhauls. European Journal of Operations Research 171, pp. 750–775.

Salhi, S., Nagy, G. (1999). A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling, Journal of Operational Research Society 50, pp. 1034–1042.

Savelsbergh, M.W.P., Sol, M. (1998). DRIVE: Dynamic routing of independent vehicles, Operations Research 46, pp. 474–490.

Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints, Operational Research 35, pp. 254–265.

Yang, J., Jaillet, P., Mahmassani, H. (2004). Real-time multi-vehicle truckload pick-up and delivery problems, Transportation Science 38, pp. 135–148.