Reoptimization Approaches for the Vehicle Routing Problem with Stochastic Demands

Nicola Secomandi, Francois Margot

Tepper School of Business, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213-3890, USA, {ns7, fmargot}@andrew.cmu.edu

Tepper Working Paper 2005-E11

Under Review in *Operations Research – Computing and Information Technologies*
Area editor: Robert Fourer
Submitted: December 2005; Revised: April 2007, July 2007

Abstract

We consider the vehicle routing problem with stochastic demands (VRPSD) under reoptimization. We develop and analyze a finite-horizon Markov decision process (MDP) formulation for the single vehicle case, and establish a partial characterization of the optimal policy. We also propose a heuristic solution methodology for our MDP, named partial reoptimization, based on the idea of restricting attention to a subset of all the possible states and computing an optimal policy on this restricted set of states. We discuss two families of computationally efficient partial reoptimization heuristics and illustrate their performance on a set of instances with up to and including 100 customers. Comparisons with an existing heuristic from the literature and a lower bound computed with complete knowledge of customer demands show that our best partial reoptimization heuristics outperform this heuristic and are on average no more than 10-13% away from this lower bound, depending on the type of instances.

1. Introduction

The deterministic vehicle routing problem (VRP) captures a fundamental supply chain management activity: the efficient distribution of products to customers (see, e.g., Nahmias 2005, §6.6). It involves determining a set of minimum cost distribution routes for a fleet of vehicles of limited capacity located at a central depot to serve the known demands of a set of geographically dispersed customers (see, e.g., Golden and Assad 1988, Fisher 1995, and Toth and Vigo 2002). VRP assumes that customer demands are known with certainty and that all the relevant information needed to compute the routes is available before their execution. Psaraftis (1995) points out that in practice customer demands can be uncertain and that most real world VRPs are dynamic, in the sense that routing decisions are updated in response to the information that is revealed to the decision maker during real time operations (Powell, Jaillet and Odoni 1995 introduce a similar characterization). He also challenges the operations research community to take advantage of the possibilities created by advances in information and communication technologies, such as geographical information/position systems (GI/PS) and intelligent vehicle highway systems, by incorporating real-time aspects within novel VRP models and algorithms.

Table 1: Classification of existing solution-approaches

	Replenishment	
Routing	Static	Dynamic
Static	A-priori	Restocking
Dynamic	Not relevant	Reoptimization

Psaraftis (1995, p. 157) presents the single-vehicle VRP with stochastic demands (VRPSD) and reoptimization as a simple prototypical problem for dynamic vehicle routing research that is very difficult to solve exactly. In VRPSD, customer demands are random variables with known probability distributions. Hence, the vehicle capacity may be insufficient to meet a customer's demand during service, in which case a failure occurs and the driver needs to perform a replenishment trip to the depot. Under the reoptimization approach, after serving the demand of a customer, the driver makes a routing and a replenishment decision in a dynamic fashion: given the available vehicle capacity and the set of unserved customers, s/he decides which customer to visit next, whether directly or by performing an en route replenishment at the depot. In the context of this problem, it is easy to envision how one may exploit GI/PS technology. For example, the driver informs a central dispatcher of the amount of a customer realized demand. The dispatcher in real-time communicates to the driver the location of the next customer to visit or any needed replenishment trip to the depot.

Reoptimization is not the only solution approach to VRPSD (see Stewart and Golden 1983, Powell, Jaillet and Odoni 1995, Gendreau, Laporte and Séguin 1996, Bertsimas and Simchi-Levi 1996, and Dror 2002 for reviews). Table 1 categorizes the existing solution approaches according to the static or dynamic nature of their routing and replenishment decisions: static (respectively, dynamic) routes are planned (respectively, unplanned), and static (respectively, dynamic) replenishment deals with them reactively (respectively, proactively). In the *a priori* approach, routes are planned and replenishment is reactive. In the *restocking* approach, routes are planned and replenishment is proactive, whereby a driver serving a route is supplemented with a dynamic replenishment policy that, given the vehicle remaining capacity after the completion of a customer service, may prescribe a replenishment trip to the depot on the way to the next customer. In the *reoptimization* approach, both routing and replenishment decisions depend on the available vehicle capacity and the set of unserved customers.

Only reoptimization can take full advantage of the opportunities enabled by the technological advances discussed by Psaraftis (1995). In Dror's words (2002, p. 643), reoptimization "is the most promising [approach] for solving [VRPSD] exactly without narrowly restricting the policy space."

Nevertheless, the existing literature on VRPSD with reoptimization is scant. Dror, Laporte and Trudeau (1989) present a Markov decision process (MDP; Puterman 1994, Bertsekas 2000) formulation of the problem but do not discuss structural properties or attempt to numerically solve their model. Dror (1993) analyzes graph-theoretic properties of a multistage stochastic programming formulation without presenting computational results. Dror, Laporte and Louveaux (1993) consider the reoptimization approach by allowing at most one route failure, but do not discuss the numerical solution of their model. Secomandi (1998, 2000, and 2001) studies various MDP formulations of the problem, both analytically and numerically, presents a rollout policy heuristic for two of them, and shows that its performance dominates that of a different neuro-dynamic programming heuristic (see Bertsekas and Tsitsiklis 1996, Bertsekas, Tsitsiklis and Wu 1997, Sutton and Barto 1998).

Research questions. This brief review brings to light that the extant literature on VRPSD with reoptimization has not yet fully addressed the characterization of an optimal reoptimization policy and the computation of such a policy. Without neglecting the characterization of an optimal reoptimization policy, the main focus of this paper is on the computation of reoptimization policies for VRPSD with a single vehicle. More specifically, our objective is to answer the following research questions. (1) Can the problem be modeled as an MDP in a computationally more advantageous fashion than so far done in the literature? (2) How can this MDP formulation be exploited to design computationally efficient heuristics, which are necessary to solve large size instances?

Contributions. Given our focus, our contributions to the literature are mainly on the computational side, but we also uncover a partial characterization of an optimal reoptimization policy. We present a finite horizon MDP that, while more restrictive than the MDP formulations available in the literature, can be solved to optimality on instances with no more than 15 customers (the relevance of this statement is discussed below). This is a direct consequence of the following modeling restriction: upon a failure, i.e., the vehicle capacity is exhausted while serving a customer demand *and* this demand is not fully served, the driver performs a return trip to the depot to restore the vehicle capacity, and resumes service from the location where the failure occurred. Thus, the service policy is of the split delivery type, but the demand of exactly one customer is partially satisfied when a failure occurs. (If the vehicle capacity is exhausted and a customer demand is fully served a failure does not arise, and hence our MDP allows empty movements.) The partial characterization of the optimal policy is as follows: given a customer location and a set of unvisited customers, the driver's optimal decision between moving to one of these customers directly or with an en route replenishment at the depot is of threshold type in the available capacity, i.e., the en route replenishment is optimal when the remaining capacity falls below a threshold that depends on the customer location and the set of unvisited customers.

With more than 15 customers, unsurprisingly, memory availability becomes a serious obstacle to optimally solve our MDP. Thus, we propose *partial reoptimization* as a computational framework based on the idea of restricting attention to a subset of all the possible MDP states, and solving for an optimal policy on this subset of states. Obviously, an optimal policy on this state subset is not necessarily optimal for the original model. However, the advantage of our approach is that we can reuse the *same* model and solution algorithm developed to compute an optimal reoptimization policy to our MDP for the purposes of computing an optimal partial reoptimization policy. In other words, if one "carefully" selects the subset of states to be considered, one can efficiently compute an optimal policy on the restricted state set by backward dynamic programming. It is not immediate how to carefully select this subset of states. We propose two methods for this purpose, which give rise to two families of heuristics, respectively called partitioning heuristic (PH) and sliding heuristic (SH). The idea behind these methods is to start from a given sequence of customers, perhaps a tour computed under the a priori or restocking approaches, and leverage this customer ordering to generate the sought subset of states. PH and SH differ in their usage of this initial customer sequence: PH works on disjoint blocks of customers, SH on overlapping blocks.

We perform an extensive numerical study to assess the performance of different PH and SH versions on instances with up to and including 100 customers, which are *large* VRPSD instances. Here we compare their performance against that of the rollout policy of Secomandi (2001) and, for instances with no more than 90 customers, a wait and see (a posteriori) unsplit delivery lower bound, which we compute by solving to optimality deterministic VRP instances with perfect knowledge of customer demand realizations. With one exception, the considered PH and SH versions outperform the rollout policy on all the instances considered, and the average deviation of our best performing heuristic from the a posteriori unsplit delivery lower bound is no more than 10-13%, depending on the type of instances. Notice that even if our MDP uses a split delivery service policy, we employ a lower bound under an unsplit delivery service policy because it is easier to compute. However, we have verified that the expected costs of the PH and SH partial reoptimization policies under an unsplit delivery service policy are only slightly higher than those obtained in the split delivery case: for the instances considered in this paper, the difference between the values of the split and unsplit policies is around 0.02% on average. Thus, our usage of an unsplit delivery lower bound to evaluate the effectiveness of split delivery heuristics seems reasonable.

Relevance. The partial characterization of the optimal policy to our MDP is important because it generalizes a similar property that holds in the restocking case (Proposition 19, page 77, in Secomandi 1998, and Theorem 1, page 102, in Yang, Mathur and Ballou 2000). However, the basic relevance of this paper lies in providing a computational framework for the design of computationally efficient reoptimization heuristics for VRPSD, which remain the only practical methods to (heuristically) solve the problem at hand despite recent advances in the solution of MDPs, in particular in the context of routing problems (Powell 2003, §4.4.2, Adelman 2003, 2004, Kleywegt, Nori and Savelsbergh 2004). We are able to compute reoptimization policies for an MDP formulation of the problem on large instances that outperform the best heuristic available in the literature, while also faring well against the a posteriori unsplit delivery lower bound. This should be contrasted with the following statements regarding MDP approaches to VRPSD made in the literature: Dror, Laporte and Trudeau (1989, p. 174) write that "at present, the value of these [...] solution frameworks is mainly descriptive," and Dror (1993, p. 440), quoting L'Ecuyer, states that "an optimistic statement would be that we could solve three-node problems with a lot of computational effort." While our results certainly benefit from the availability of more powerful computers since the late-eighties/early-nineties, they are enabled in a more important way by our methodological choice of restricting attention to a carefully selected subset of states. This is a fundamentally different approach to (heuristically) solving VRPSD relative to what has so far been done in the literature on VRPSD with reoptimization. In other words, our MDP formulation is important not because we can solve exactly 15, as opposed to 3, customer instances. Rather it is important because it provides a clear starting point for the design of computationally efficient reoptimization heuristics, of which we present two in this paper.

As stated by Psaraftis (1988, pp. 156-157), VRPSD with reoptimization is a foundational model in stochastic and dynamic vehicle routing research. In this respect, our proposed solution methodology is relevant because it provides a framework that other researchers can use and extend for the development of heuristics for this canonical model. In terms of practical relevance, while we do not deal with an application of our model to a real life situation, it is important to point out that availability of computationally efficient and effective heuristics is crucial for potential implementation of the model studied here, or, most likely, modifications thereof, in real life settings. As summarized by Bertsimas and Simchi-Levi (1996) and Dror (2002), these include cash collection/distribution from/to automatic telling machines, delivery of packages from a post office, sludge disposal, and delivery of home heating oil, among others. Chan, Carter and Burnes (2001) and Singer, Donoso and Jara (2002) describe applications of a related, but significantly richer, model in the context of distribution of manufacturing materials and liquefied petroleum gas, respectively, in response to stochastic demands.

Novelty. The idea of heuristically solving an MDP by selectively considering only a subset of states is not new. For example, aggregation techniques are discussed by Whitt (1978) and White and White (1989). In contrast, we do not employ an aggregation technique in this paper since partial reoptimization works on a subset of the original MDP states. Whitt (1979) discusses an approximation approach that is more closely related to ours, in that it also entails solving the MDP on a subset of states of the given MDP without state aggregation. However, it is not clear how one could employ the work of Whitt (1979) in our specific setting. Thus, our work is novel in this respect since it provides two specific methods to construct a restricted set of states for our MDP formulation of VRPSD.

Since we do not employ value function approximations, our approach is also distinct from the emerging research area on approximate dynamic programming (Bertsekas and Tsitsiklis 1996, Sutton and Barto 1998, Bertsekas 2000, Chapter 6, Adelman 2006, Powell 2007). Applications of these ideas in the context of stochastic routing and scheduling include Adelman (2004) and Kleywegt, Nori and Savelsbergh (2004) for the stochastic inventory routing problem, a problem related to VRPSD and also studied by Dror and Trudeau (1986) and Trudeau and Dror (1992), Bertsekas and Castanon (1999) for stochastic scheduling problems, Secomandi (1998, 2000, 2001) for VRPSD, and Secomandi (2003) for a traveling salesman problem with stochastic travel times.

In particular, Adelman (2003, 2004) uses math programming methods, what he calls the price directed approach, to compute lower bounds and control policies for inventory routing problems. While we also estimate a lower bound using math programming techniques, our choice of problem and our approach to lower bound and control policy computation are different from his. However, it might be possible to extend and the price directed approach to tackle VRPSD with reoptimization.

Gendreau, Laporte and Séguin (1995), Hjorring and Holt (1999), and Laporte, Louveaux and Van Hamme (2002) optimally solve two stage stochastic programming formulations of versions of VRPSD under the a priori approach. Since we focus on VRPSD with reoptimization, application of stochastic programming to our problem would require a multistage formulation, such as the one in Dror (1993). Solving this formulation to optimality for the size of problems that we consider is currently impractical.

One of our families of heuristics, SH, is related to a dynamic programming based heuristic developed by Balas (1999), and studied by Balas and Simonetti (2001), in the context of a deterministic traveling salesman problem. We study a different problem since the traveling salesman problem analyzed by these authors is not capacitated and does not feature demand uncertainty.

Organization. The remainder of this paper is organized as follows. Section 2 presents our

MDP formulation and establishes a partial characterization of the optimal policy, together with some related properties. Section 3 provides a generic description of the partial reoptimization approach. Section 4 introduces two specific partial reoptimization families of heuristics, PH and SH. Section 5 presents the computation of two lower bounds. Section 6 discussed our extensive numerical results. We conclude in §7 by summarizing our main results and discussing further research avenues.

2. Reoptimization

We restrict our attention to a business situation that features distribution, rather than collection, of items. Let the set of nodes of a given complete network be $\{0, 1, \ldots, N\}$, with N a positive integer. Node 0 denotes the depot and $\mathcal{N} := \{1, 2, \ldots, N\}$ is the set of customers. Distances $d(i, j)$ between any two nodes i and j are known, symmetric, and satisfy the triangle inequality: $d(i, j) \leq d(i, \ell) + d(\ell, j)$, with ℓ an additional node. There is a single vehicle initially located at the depot with capacity Q, a positive integer. Let $\tilde{\xi}_i$, $i = 1, 2, \ldots, N$, be the discrete random variable that describes customer i demand. Its probability mass function is known and denoted $p_i(k) := \Pr\{\tilde{\xi}_i = k\}$, $k = 0, 1, \ldots, K \leq Q$, with K a nonnegative integer. Random variables $\tilde{\xi}_i$'s are stochastically independent of each other and of the vehicle routing/replenishment policy. The realization of $\tilde{\xi}_i$ becomes known upon arrival of the vehicle at customer location i. The total depot capacity is at least $N \cdot K$, so that all customers can be fully served.

Our MDP formulation assumes the following split delivery service policy: If the vehicle becomes empty while serving a customer, the driver immediately executes a return trip to the depot to restore capacity to Q and then finishes supplying this customer (one trip suffices since $K \leq Q$). As explained below, this assumption allows us to formulate the problem as a finite-horizon MDP with $N + 1$ stages in set $\mathcal{S} := \{0, 1, \ldots, N\}$, with stage $s \in \mathcal{S}$ corresponding to the number of unvisited customers. If this assumption were not imposed, the problem could not be formulated as a finite-horizon MDP (see, e.g., the MDP formulations of Secomandi 2000, 2001). The initial stage is N. From stage $s \in \mathcal{S} \setminus \{0\}$ the vehicle moves to stage $s - 1$. Stage 0 is the terminal stage. In stage N the unique state is

$$(0, Q, \mathcal{N}),$$

which corresponds to the vehicle being at the depot with full capacity and all customers remaining to be served. Let $\mathcal{R}_s(\ell) \subset \mathcal{N}$ be a set of customers yet to be visited when the vehicle is at location ℓ in stage $s \in \mathcal{S} \setminus \{N\}$ and has finished serving customer ℓ, i.e., $|\mathcal{R}_s(\ell)| = s$ and $\ell \notin \mathcal{R}_s(\ell)$. We refer to $\mathcal{R}_s(\ell)$ as a set of remaining customers in stage s when the vehicle is at location ℓ. Note that $\mathcal{R}_0(\cdot) \equiv \emptyset$. Denote $q \in \mathcal{Q} := \{0, \ldots, Q\}$ the amount of available vehicle capacity *after* satisfying the demand of a customer. A state in stage $s \in \mathcal{S} \setminus \{N\}$ is

$$(\ell, q, \mathcal{R}_s(\ell)),$$

which corresponds to the vehicle being at customer location ℓ with q available capacity and set of remaining (unvisited) customers $\mathcal{R}_s(\ell)$. The state set in stage $s \in \mathcal{S} \setminus \{N\}$ is $\mathcal{X}_s := \{(\ell, q, \mathcal{R}_s(\ell)),\ \ell \in \mathcal{N}, q \in \mathcal{Q}, \mathcal{R}_s(\ell) \subset \mathcal{N}, |\mathcal{R}_s(\ell)| = s, \ell \notin \mathcal{R}_s(\ell)\}$, and the state set is $\mathcal{X} := \{(0, Q, \mathcal{N})\} \cup (\bigcup_{s \in \mathcal{S} \setminus \{N\}} \mathcal{X}_s)$. The cardinality of the state space is $1 + N(Q+1)2^{N-1}$. This can be easily verified using the Binomial Theorem, by noting that the number of states per stage is 1 in stage N and $N(Q+1)(N-1)_s$ in stage $s \in \mathcal{S} \setminus \{N\}$, where

$$(N-1)_s := \begin{pmatrix} N-1 \\ s \end{pmatrix}. \tag{1}$$

In any state in the final stage, 0, the only possible action is to move to the depot. In the initial state in stage N, the possible actions correspond to visiting one of the customers. In any other state $(\ell, q, \mathcal{R}_s(\ell))$ in any other stage $s \in \mathcal{S} \setminus \{0, N\}$, the possible actions are which customer $j \in \mathcal{R}_s(\ell)$ to visit next, whether directly (a case labeled D) or by fully replenishing at the depot (a case labeled R). Both routing and replenishment decisions are dynamic, in the sense of Psaraftis (1995), because they are state dependent.

An optimal policy satisfies the following set of Bellman equations that, at least in principle, can be solved by dynamic-programming backward recursion. Denote $V_s(\cdot)$ the optimal cost-to-go or value function vector in stage s. For stage 0, define

$$V_0(\ell, q, \mathcal{R}_0(\ell)) := d(\ell, 0), \forall (\ell, q, \mathcal{R}_0(\ell)) \in \mathcal{X}_0.$$

For each stage $s \in \mathcal{S} \setminus \{0, N\}$, by defining $\mathcal{R}_{s-1}(j; \ell) := \mathcal{R}_s(\ell) \setminus \{j\}$, the optimal value function in state $(\ell, q, \mathcal{R}_s(\ell))$ is

$$V_s(\ell, q, \mathcal{R}_s(\ell)) \quad = \quad \min \left\{ V_s^D(\ell, q, \mathcal{R}_s(\ell)), V_s^R(\ell, q, \mathcal{R}_s(\ell)) \right\}, \ \forall (\ell, q, \mathcal{R}_s(\ell)) \in \mathcal{X}_s,$$

where $V_s^D(\ell, q, \mathcal{R}_s(\ell))$ and $V_s^R(\ell, q, \mathcal{R}_s(\ell))$ are, respectively, the cost-to-go values in this stage and state corresponding to visiting the best next-customer directly and by first replenishing at the depot (hence the superscripts D and R):

$$V_s^D(\ell, q, \mathcal{R}_s(\ell)) \quad := \quad \min_{j \in \mathcal{R}_s(\ell)} \left\{ d(\ell, j) + \sum_{k=0}^{q} p_j(k) V_{s-1}(j, q-k, \mathcal{R}_{s-1}(j; \ell)) \right.$$

$$+ \sum_{k=q+1}^{K} p_j(k) \left[V_{s-1}(j, q+Q-k, \mathcal{R}_{s-1}(j;\ell)) + 2d(j,0)\right] \Bigg\}$$

$$V_s^R(\ell, q, \mathcal{R}_s(\ell)) \quad := \quad \min_{j \in \mathcal{R}_s(\ell)} \left\{ d(\ell,0) + d(0,j) + \sum_{k=0}^{K} p_j(k) V_{s-1}(j, Q-k, \mathcal{R}_{s-1}(j;\ell)) \right\}.$$

For stage N, the optimal value function in the unique state $(0, Q, \mathcal{N})$ is

$$V_N(0, Q, \mathcal{N}) = \min_{j \in \mathcal{N}} \left\{ d(0,j) + \sum_{k=0}^{K} p_j(k) V_{N-1}(j, Q-k, \mathcal{N} \setminus \{j\}) \right\}.$$

The optimal action in stage 0 is obvious. In state $(\ell, q, \mathcal{R}_s(\ell))$ in stage $s \in \mathcal{S} \setminus \{0, N\}$ define

$$j_s^D(\ell, q, \mathcal{R}_s(\ell)) \quad := \quad \arg\min_{j \in \mathcal{R}_s(\ell)} \Bigg\{ d(\ell,j) + \sum_{k=0}^{q} p_j(k) V_{s-1}(j, q-k, \mathcal{R}_{s-1}(j;\ell))$$

$$+ \sum_{k=q+1}^{K} p_j(k) \left[V_{s-1}(j, q+Q-k, \mathcal{R}_{s-1}(j;\ell)) + 2d(j,0)\right] \Bigg\}$$

$$j_s^R(\ell, q, \mathcal{R}_s(\ell)) \quad := \quad \arg\min_{j \in \mathcal{R}_s(\ell)} \left\{ d(\ell,0) + d(0,j) + \sum_{k=0}^{K} p_j(k) V_{s-1}(j, Q-k, \mathcal{R}_{s-1}(j;\ell)) \right\}.$$

Note that $j_s^D(\ell, q, \mathcal{R}_s(\ell))$ and $j_s^R(\ell, q, \mathcal{R}_s(\ell))$ denote the best next-customer locations associated with cases D and R, respectively, in stage s and state $(\ell, q, \mathcal{R}_s(\ell))$. We define an action as a pair that includes the next customer to visit as its first entry, and either the label D or R as its second entry to indicate whether this customer should be reached directly from the current location or by first visiting, and hence replenishing at, the depot. Then, an optimal action in stage $s \in \mathcal{S} \setminus \{0, N\}$ and state $(\ell, q, \mathcal{R}_s(\ell))$, denoted $a_s(\ell, q, \mathcal{R}_s(\ell))$, is

$$a_s(\ell, q, \mathcal{R}_s(\ell)) \quad := \quad \begin{cases} (j_s^D(\ell, q, \mathcal{R}_s(\ell)), D) & \text{if } V_s^D(\ell, q, \mathcal{R}_s(\ell)) \leq V_s^R(\ell, q, \mathcal{R}_s(\ell)) \\ (j_s^R(\ell, q, \mathcal{R}_s(\ell)), R) & \text{if } V_s^D(\ell, q, \mathcal{R}_s(\ell)) > V_s^R(\ell, q, \mathcal{R}_s(\ell)) \end{cases}. \qquad (2)$$

Here, a direct movement is arbitrarily chosen over a movement with replenishment if their optimal cost-to-go values are equal. In stage N, the optimal action from the unique initial state $(0, Q, \mathcal{N})$ is

$$j_N(0, Q, \mathcal{N}) := \arg\min_{j \in \mathcal{N}} \left\{ d(0,j) + \sum_{k=0}^{K} p_j(k) V_{N-1}(j, Q-k, \mathcal{N} \setminus \{j\}) \right\}.$$

We now establish a partial characterization of the optimal policy and some additional structural results. Proposition 1, based on Lemma 1, establishes that in each stage the optimal value function is weakly decreasing (nonincreasing) in the amount of available capacity given a customer location and a set of remaining customers. Based on this property, Proposition 2 shows that in each stage, obviously excluding the first and the last stages, the optimal choice between a direct movement and a movement with a replenishment at the depot is of threshold type in the amount of available capacity, given a customer location and a set of remaining customers.

Lemma 1 (Restricted bound on the value of additional capacity)**.** *In stage* $s \in \mathcal{S} \setminus \{0, N\}$*, for given customer* ℓ *and set* $\mathcal{R}_s(\ell)$*, if* $V_s(\ell, q, \mathcal{R}_s(\ell))$ *is weakly decreasing in* q*, then* $V_s(\ell, q^1, \mathcal{R}_s(\ell)) - V_s(\ell, q^2, \mathcal{R}_s(\ell)) \leq 2d(0, \ell)$*,* $\forall q^1, q^2$ *such that* $0 \leq q^1 < q^2 \leq Q$*.*

Proposition 1 (More capacity is weakly better)**.** *In every stage* $s \in \mathcal{S} \setminus \{N\}$*, for given customer* ℓ *and set* $\mathcal{R}_s(\ell)$*,* $V_s(\ell, q, \mathcal{R}_s(\ell))$ *is weakly decreasing in* $q \in \mathcal{Q}$*.*

Proposition 2 (Threshold-type replenishment)**.** *In every stage* $s \in \mathcal{S} \setminus \{0, N\}$*, for given customer* ℓ *and set* $\mathcal{R}_s(\ell)$*, the optimal action between replenishing and moving directly to the next customer is of threshold type in available capacity* $q \in \mathcal{Q}$*:*

$$a_s(\ell, q, \mathcal{R}_s(\ell)) = \begin{cases} (j_s^R(\ell, q, \mathcal{R}_s(\ell)), R) & \text{if } q < \bar{q}_s(\ell, \mathcal{R}_s(\ell)) \\ (j_s^D(\ell, q, \mathcal{R}_s(\ell)), D) & \text{if } q \geq \bar{q}_s(\ell, \mathcal{R}_s(\ell)) \end{cases}.$$

The threshold $\bar{q}_s(\ell, \mathcal{R}_s(\ell))$ *is* 0 *if* $V_s^D(\ell, q, \mathcal{R}_s(\ell)) \leq V_s^R(\ell, q, \mathcal{R}_s(\ell))$ $\forall q \in \mathcal{Q}$*, i.e., never replenish, and* $\min\left\{q \in \mathcal{Q} : V_s^D(\ell, q, \mathcal{R}_s(\ell)) = V_s^R(\ell, q, \mathcal{R}_s(\ell))\right\}$ *otherwise.*

These propositions generalize properties of the optimal restocking policy (Proposition 19, p. 77, in Secomandi 1998 and Theorem 1, p. 102, in Yang, Mathur and Ballou 2000). Moreover, the threshold property can be exploited to reduce the burden of numerically computing the optimal policy since in a given stage and for each given customer location and set of remaining customers one can stop computing the "direct" and "replenishment" actions for capacity value below this capacity threshold. In other words, if, again in a given stage and for each given customer location and set of remaining customers, the replenishment action is optimal at some capacity level, then it remains so for all lower capacity levels. However, in our numerical experiments with the partial reoptimization approach, we found these computational savings to be rather small even with large capacity values. For example, on instances with 100 customers and values of parameter Q equal to 895 and 1063 we obtained computational savings of 1.5% on average, with a maximum of 2.89%, on about 35 Cpu seconds (note that these values of Q are somewhat extreme and the instances studied in §6 use smaller value for this parameter).

Corollary 1, which easily follows from Lemma 1 and Proposition 1, generalizes Lemma 1 by establishing that in stage $s \in \mathcal{S} \setminus \{0, N\}$, for given customer ℓ and set $\mathcal{R}_s(\ell)$, the optimal cost-to-go as a function of available capacity lies in a band whose width is no more than twice the distance from ℓ to the depot.

Corollary 1 (Value bands)**.** *In every stage $s \in \mathcal{S} \setminus \{0, N\}$, for given customer ℓ and set $\mathcal{R}_s(\ell)$, it holds that $V_s(\ell, 0, \mathcal{R}_s(\ell)) - V_s(\ell, Q, \mathcal{R}_s(\ell)) \leq 2d(0, \ell)$.*

This property is interesting because it could be exploited when solving the problem using a neuro-dynamic programming methodology (as in Secomandi 2000). In particular, it could be used to place bounds on the values of the optimal value function approximation during its (statistical) estimation phase.

Notice that in our MDP formulation we impose the restriction that the vehicle must replenish when a failure occurs, i.e., when the vehicle is empty but a customer demand has not yet been fully served. However, if the vehicle is empty after *completely* serving the demand of a given customer, we do not force the vehicle to replenish. In this case, it may be optimal for the vehicle to move empty to a customer location. Corollary 2 establishes a simple necessary condition for this action to be strictly optimal, namely that there is positive probability that the demand of this customer is zero.

Corollary 2 (Meaningful customer visits)**.** *If in stage $s \in \mathcal{S} \setminus \{0, N\}$ and state $(\ell, 0, \mathcal{R}_s(\ell))$ it holds that action $(j_s^D(\ell, 0, \mathcal{R}_s(\ell)), D)$ is strictly optimal, i.e., $V_s^D(\ell, 0, \mathcal{R}_s(\ell)) < V_s^R(\ell, 0, \mathcal{R}_s(\ell))$, then $p_{j_s^D(\ell,0,\mathcal{R}_s(\ell))}(0) > 0$.*

According to this property, if what might be otherwise considered a "risky" trip to a customer is strictly optimal, then it must be that there is positive probability that no demand needs to be satisfied at this customer location. Thus, if customer demands are positive with probability 1, there is no value for the vehicle to move empty to a customer location.

As previously discussed, the cardinality of the state space is $O(NQ2^N)$. Therefore, the time requirement of computing an optimal reoptimization policy by dynamic-programming backward-recursion is $O(N^2KQ2^N)$. These space and time requirements grow exponentially in N. However, with 15 (or less) customers it is practical to solve the proposed MDP to optimality in a few seconds on typical instances. With more than 15 customers, memory, rather than Cpu times, becomes a serious issue. Hence, in §3 we propose partial reoptimization as an efficient approach to compute heuristic policies to this MDP on such instances.

To conclude this section, one may observe that a more realistic MDP would remove the requirement that the vehicle immediately replenish upon a failure and that it resume service from the location where the failure occurred. Removing this restriction, in the sense that the vehicle be allowed to visit a different customer after replenishing upon a failure, or more generally after a failure, would require formulating the problem as a stochastic shortest path problem with a state redefined