European Journal of Operational Research 285 (2020) 172–182

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Optimal a priori tour and restocking policy for the single-vehicle routing problem with stochastic demands

Alexandre M. Florio[a,*], Richard F. Hartl[a], Stefan Minner[b]

[a] University of Vienna, Oskar-Morgenstern-Platz 1, Vienna 1090, Austria
[b] Technical University of Munich, Arcisstrasse 21, Munich 80333, Germany

ARTICLE INFO

Article history:
Received 26 February 2018
Accepted 26 October 2018
Available online 31 October 2018

Keywords:
Routing
Markov decision processes
Stochastic programming

ABSTRACT

We present a model for the single-vehicle routing problem with stochastic demands (SVRPSD) with optimal restocking. The model is derived from a characterization of the SVRPSD as a Markov decision process (MDP) controlled by a certain class of policies, and is valid for general discrete demand probability distributions. We transform this MDP into an equivalent mixed-integer linear model, which is then used to solve small instances to optimality. By doing so, we are able to quantify the drawbacks associated with the detour-to-depot restocking policy, an assumption of many exact approaches for the (multivehicle) VRPSD. We also examine the tradeoff between the deterministic a priori cost and the stochastic restocking cost for varying route load scenarios. Finally, a wait-and-see model for the SVRPSD is proposed, and is used within a parallel heuristic to solve larger literature instances with up to 150 nodes and Poisson distributed demands. Computational experiments demonstrate the effectiveness of the heuristic approach, and also indicate under which circumstances near-optimal solutions can be obtained by the myopic strategy of a priori route cost minimization.

1. Introduction

1.1. Vehicle routing with stochastic demands

The vehicle routing problem (VRP) is one of the classical optimization problems in logistics. In the VRP, a set of vehicles is available for delivering (or picking-up) goods to (from) customers. Each customer has some demand, and the vehicles have capacities. A valid route begins at some designated depot, visits a sequence of customers, and finishes at the depot. Moreover, in a valid route, the sum of the demands of the visited customers does not exceed the capacity of the vehicle. The objective in the VRP is to design a set of valid routes of minimum total cost visiting every customer once. The cost of a route is usually measured in terms of its total length or duration.

In many practical scenarios some of the problem data may not be known at the time of route planning. For example, traffic conditions may vary causing uncertainty in the travel times. In other cases, customer demands are unknown and are only disclosed upon arrival of the vehicle at their locations. When randomness is present in the input data, the general class of corresponding routing problems is called stochastic VRP (Gendreau, Laporte, & Séguin, 1996). Many subclasses arise depending on the element (or combination of elements) considered stochastic. In this paper, we consider the case of VRP with stochastic demands (VRPSD), which is the most studied in the literature (Gendreau, Jabali, & Rei, 2016).

The notions of optimality and feasibility change when we move into stochastic domain. For example, while an optimal solution to the deterministic VRP minimizes total route cost, in the stochastic version we usually treat the objective as the minimization of the *expected* total cost. On the feasibility side, the translation of constraints is not so direct. When demands are stochastic, the total demand in a route may exceed the capacity of the vehicle. A *failure* is said to occur when a vehicle arrives at a customer with insufficient load to serve that customer's demand. There are different modeling approaches for the VRPSD, depending on how failures are managed.

Many of these approaches are based on two-stage stochastic programming models. In such models, the first stage decisions determine the so-called *a priori tour* (single-vehicle case) or *a priori routes* (multiple-vehicles case). The second-stage decisions determine the *recourse action* with associated *recourse cost*. The objective in these models is the minimization of the total expected cost, composed of the first stage (or a priori) cost and the expected recourse cost. In its simplest form, the recourse can be just a penalty term for unsatisfied demand. However, in many practical scenarios

* Corresponding author.
E-mail addresses: alexandre.de.macedo.florio@univie.ac.at (A.M. Florio), richard.hartl@univie.ac.at (R.F. Hartl), stefan.minner@tum.de (S. Minner).

https://doi.org/10.1016/j.ejor.2018.10.045
0377-2217/

fulfilling all demand is mandatory (e.g., waste collection). In these cases, we must allow the vehicle to perform *replenishment* (or emptying) trips to the depot in order to increase the total capacity available to serve a route. The recourse action now becomes a *restocking policy*, which is a set of rules that govern when the vehicle should perform a replenishment trip to the depot.

Many exact and heuristic methods developed for the VRPSD assume the so-called *detour-to-depot* restocking policy, first stated in Dror, Laporte, and Trudeau (1989). This policy prescribes a replenishment trip if and only if the vehicle does not have sufficient residual capacity to serve the current customer. This is non-optimal, especially when the current customer is located far away from the depot. However, incorporating more involved policies in the (already complicated) two-stage models may quickly lead to intractability. On the other hand, the problem of finding the optimal restocking policy given a fixed a priori tour has long been solved by a simple stochastic dynamic programming algorithm (Yee & Golden, 1980). An optimal restocking policy can perform *preventive* replenishment trips to avoid failures further down the route where a return trip could be costly, and thus is more sophisticated than the detour-to-depot policy.

1.2. Single-vehicle case (SVRPSD)

When replenishment trips are allowed, one can imagine a single-vehicle version of the VRPSD, where all customers are served in the same tour. In this work, we focus on this special case for the following reasons: (1) under optimal restocking, every VRPSD instance admits one optimal solution, in which a single vehicle visits all the customers (Yang, Mathur, & Ballou, 2000); (2) currently, no linear model exists for the VRPSD, so it is natural to start with a model for this case; (3) studying the single-vehicle case gives insights on the effects of different restocking policies on the expected cost of a route. These insights are also valid for the multiple-vehicles case, when each route is considered separately; and (4) in applications where *districting* of customers occurs before the definition of the routes, the underlying problem can be decomposed into several single-vehicle problems.

The SVRPSD[1] has been extensively studied. In Bastian and Rinnooy Kan (1992) penalty, chance-constrained and full service models were proposed, and it was shown that, under detour-to-depot restocking and identically distributed customer demands, the SVRPSD is equivalent to a time-dependent traveling salesman problem. Also under detour-to-depot restocking, Bertsimas (1992) developed closed-form expressions to compute the expected cost of an a priori tour, proposed the *cyclic* heuristic and derived asymptotic results. Computational results of the cyclic heuristic were reported in Bertsimas, Chervi, and Peterson (1995), including comparison with a heuristic based on the savings algorithm (Clarke & Wright, 1964). Hjorring and Holt (1999) proposed an approximated mixed-integer model, and solved it exactly with the integer L-shaped method (Laporte & Louveaux, 1993). Heuristics employing local branching and Monte Carlo sampling were also applied to solve the single-vehicle version (Chepuri & Homem-de Mello, 2005; Rei, Gendreau, & Soriano, 2010). Finally, optimal restocking policies were derived for variants of the problem with multiple products and multiple vehicle compartments (Pandelis, Kyriakidis, & Dimitrakos, 2012; Tatarakis & Minis, 2009; Tsirimpas, Tatarakis, Minis, & Kyriakidis, 2008), and multiple products with customer preferences (Kyriakidis, Dimitrakos, & Karamatsoukis, 2017).

A stream of literature has also been developed for the case where, in addition to restocking decisions, *sequencing* decisions are allowed. By sequencing, we understand the decision of which customer to visit next, after a customer has been served. This alternative approach constitutes the so-called *reoptimization* paradigm. In reoptimization models the routes are not defined at planning time, but are the result of several sequencing decisions made each time a customer is visited and new demand information disclosed. Reoptimization models are clearly more complex, and are usually defined as Markov decision processes (MDPs), such as the single-vehicle model presented in Dror et al. (1989), and solved with approximation techniques (Novoa & Storer, 2009; Secomandi, 2001; Secomandi & Margot, 2009). In these methods, forward recursion algorithms to compute the expected cost of a route under optimal restocking lead to significant speedups (Bertazzi & Secomandi, 2018). Given the augmented decision set, it can be anticipated that the expected cost of a route is smaller under the reoptimization approach. However, in a multi-period environment, solutions obtained under the reoptimization paradigm may completely lack *consistency*, a desirable attribute in many routing applications (Groër, Golden, & Wasil, 2009).

1.3. Multiple-vehicles case

In a *chance-constrained programming* (CCP) approach to the VRPSD, routes are designed so that the probability of a failure is kept low. In Stewart and Golden (1983), the first formulation of this kind was presented. Dror, Laporte, and Louveaux (1993) proposed an exact method for solving a VRPSD with restricted number of failures modeled as a CCP. See also Birge and Louveaux (2011) for an example of CCP applied to a simple routing problem. Exact algorithms for solving the VRPSD with probabilistic constraints have been introduced in Dinh, Fukasawa, and Luedtke (2018) and Noorizadegan and Chen (2018), with the former also handling the case of positively correlated demands. In both algorithms, medium-sized instances (with up to 60 nodes) could be solved to optimality.

Recently, the first approaches addressing the VRPSD under optimal restocking have been proposed (Florio, Hartl, & Minner, 2018; Louveaux & Salazar-González, 2018; Salavati-Khoshghalb, Gendreau, Jabali, & Rei, 2018). These approaches do not explicitly model the stochastic restocking costs, but use different sort of bounds when examining the solution space with sophisticated branch-and-bound algorithms. Optimal restocking policies have been incorporated in heuristics for the VRPSD (Yang et al., 2000), and, recently, also for the generalized VRPSD (Biesinger, Hu, & Raidl, 2018). The multiple-vehicles case under the reoptimization approach has also been studied and solved with approximate dynamic programming methods (Goodson, Ohlmann, & Thomas, 2013; Goodson, Thomas, & Ohlmann, 2016). Apart from these contributions, most other heuristic and exact methods for the VRPSD assumed either the detour-to-depot policy or penalty-based models. We refer to Gendreau et al. (2016) for a review of the related literature.

1.4. Contributions

Our main contribution is from a modeling perspective: we model the SVRPSD, including the expected value of the optimal restocking policy, as a mixed-integer linear model. This model can be used to solve simple instances of the SVRPSD with commercial solvers, without the need for customized implementations. This model may also serve as a starting point for future approaches for the problem based on decomposition techniques.

[1] An alternative and maybe more appropriate name for the single-vehicle version of the problem would be *traveling salesman problem with stochastic demands*, which also emphasizes the fact that a solution consists of a single tour visiting all customers. In order to be consistent with previous literature, however, we kept the convention of calling the problem single-vehicle routing problem with stochastic demands.

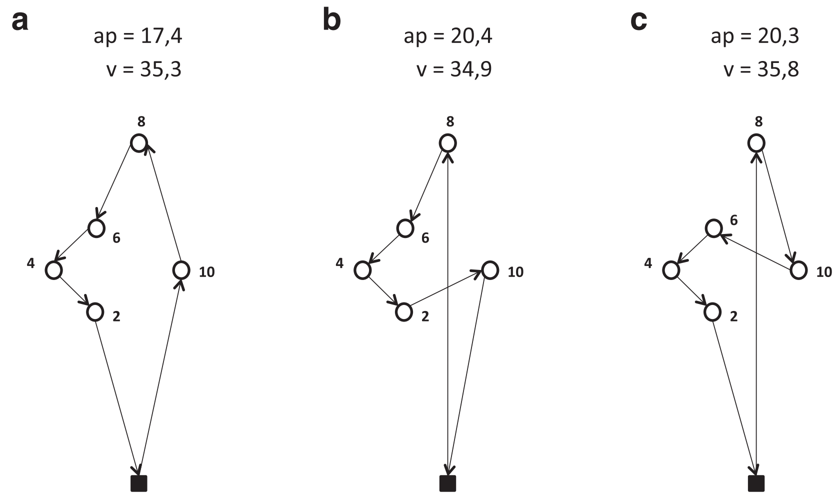

Fig. 1. Solutions to the SVRPSD considering (a) optimal TSP solution coupled with optimal restocking policy; (b) optimal restocking solution; (c) detour-to-depot restocking solution. The values v and ap on top of each solution indicate, respectively, its corresponding expected cost and a priori route cost.

The distinctive characteristic of our model is the simultaneous optimization of the a priori tour and the restocking policy. We do so by first defining a general Markov decision process (MDP) allowing both sequencing and restocking decisions, and then restricting this process by applying *policy constraining*. These constraints define a class of control policies for the general MDP which must comply with some a priori tour. By finding the optimal policy in this class, we also arrive at the optimal restocking policy for some a priori tour. Therefore, we do not rely on the dynamic programming algorithm of Yee and Golden (1980) for route evaluation, but have the optimal restocking policy naturally as an output of the model. In some sense, we unify the a priori problem (finding an optimal tour given a fixed policy) and the restocking problem (finding an optimal policy given a fixed tour) in the same model.

The techniques we employ to build such a model, in particular the policy constraining ideas, provide a general framework for developing models for the SVRPSD with alternative restocking policies. Unfortunately, the price to pay for having a unified model is a mixed-integer linear program with a very large number of decision variables and constraints. Nevertheless, by using policy constraining in different ways, one can derive models which compromise on the optimal restocking policy in favor of simplicity, still doing better than the traditional detour-to-depot recourse models.

Our last contribution is the introduction of a parallel heuristic for solving larger instances of the problem. This heuristic is based on the approximative branch-and-bound method, a framework for heuristically solving mixed-integer linear programs, which we introduce in Section 3.2. The method requires minimal implementation effort, and within reasonable computational time produces very good results, as measured against the known optima of several simplified instances, and against good benchmarks on larger instances.

2. A unified model for the SVRPSD

2.1. Motivating example

Before introducing the model, we motivate the discussion with a simple example. Fig. 1 depicts three solutions for a SVRPSD instance with 5 customers. The customers are located on the plane, and have the following (x, y) coordinates: $(-1, 4)$, $(-2, 5)$, $(-1, 6)$, (0,8) and (1,5). The depot is located at (0,0). The demands are Poisson distributed, with expected values 2, 4, 6, 8 and 10 (as indicated), and the vehicle has a capacity of 15. Therefore, in this instance the route load corresponds to twice the capacity of the vehicle.

Solution (a) is the optimal traveling salesman problem (TSP) solution coupled with the optimal restocking policy, i.e., it is the solution obtained by the myopic strategy of minimizing the a priori route cost alone. Both orientations of the TSP tour have been considered, and the best among them was selected. Solution (b) is the optimal solution to the SVRPSD with optimal restocking. Finally, (c) shows the optimal solution when applying the detour-to-depot restocking policy.

Note that, even though its deterministic a priori cost is 17.2% worse, the optimal solution (b) is 1.1% superior to the myopic strategy. The myopic strategy, however, is better than the optimal detour-to-depot solution. Although this is a relatively small example, it is representative in the sense that the relative order of the solutions, from the best to the worst, is the same as observed in our experiments on larger instances. The percentage difference among the solution values, however, can be significantly higher on larger instances.

While the TSP solution is simply obtained by solving a TSP problem and applying twice (one for each orientation of the optimal TSP tour) the dynamic programming algorithm from Yee and Golden (1980), no mixed-integer linear model currently exists in the literature for computing the optimal restocking solution. We close this gap by deriving a mixed-integer linear program (MILP) for computing optimal a priori tours and restocking policies for the single-vehicle case. We start by devising a general MDP. We then define the class of *Hamiltonian policies*, which are essentially control policies that must conform to some a priori tour. We show that any policy from this class induces a periodic Markov chain in our general MDP. Applying techniques from the literature (Derman, 1962; Derman & Klein, 1965), we model the problem of finding the optimal Hamiltonian policy as a linear program. We show that such a policy must also be optimal to the finite horizon problem of finding an optimal restocking policy. By enforcing Hamiltonian tour constraints, we arrive at a MILP model for simultaneously optimizing a priori tours and restocking policies. We finally reformulate this MILP in a more convenient way.

2.2. Problem description

In what follows, we assume the SVRPSD to be defined on a graph $\mathcal{G}$ with $n+1$ nodes. Node 0 corresponds to the depot, and nodes $1, \dots, n$ to the customers. A transportation cost of c_{ij} is incurred when traveling from i to j. A single vehicle with a capacity of Q is available for servicing the customers. At planning time, the demands of the customers are unknown. We model the discrete demand of customer i by the random variable D_i with probability mass function $p_{D_i}(d)$. Demands are independent and upper-bounded by some large integer d_{max}. It is possible that $d_{max} > Q$. Finally, we only allow partial deliveries to a customer when the realized demand is larger than the remaining load in the vehicle. In this case, the vehicle must deliver the remaining load and travel back and forth to the depot until all demand of the customer has been satisfied. The objective is to find an a priori tour visiting all customers *and* a restocking policy such that all customers have their demands satisfied and the expected total transportation cost is minimized.

2.3. A general Markov decision process

The MDP we present in this section is *not* intended to be a model of the SVRPSD under the reoptimization approach, even though we allow sequencing and restocking decisions at each state. As we will see in Section 2.4, the sequencing decisions will later be enforced to conform to some a priori tour. Only after these constraints are added, the MDP becomes a valid model for the SVRPSD under the a priori optimization approach.

Our general MDP is composed of a set of states, a set of possible decisions for each state, and cost and state transition probabilities associated with each decision. An initial state $\mathcal{S}_0$ corresponds to the vehicle at the depot fully loaded. A final state $\mathcal{S}_F$ corresponds to the vehicle back to the depot after serving some customer. The remaining states are identified by $\mathcal{S}_{i,q}$. The first index i corresponds to the customer just visited. The second index q is the remaining load of the vehicle (if non-negative), or the amount that still needs to be delivered to customer i (if negative). If q is negative, we say that customer i has a *pending* demand. Therefore, the state set is finite and given by $\{\mathcal{S}_0\} \cup \{\mathcal{S}_F\} \cup \{\mathcal{S}_{i,q} \mid i \in \{1, \dots, n\}, q \in \{-d_{max}, \dots, Q\}\}$.

Each state $\mathcal{S}$ is associated with a set of possible decisions $K(\mathcal{S})$. In the initial state $\mathcal{S}_0$ the vehicle can proceed to any customer, so $K(\mathcal{S}_0) = \{k_1, \dots, k_n\}$. At any other state $\mathcal{S} \neq \mathcal{S}_F$, the vehicle can visit another customer directly or indirectly (after a replenishment trip to the depot), or go back to the depot. So we have $K(\mathcal{S}) = \{k_1^d, \dots, k_n^d, k_1^r, \dots, k_n^r, k_0\}$. (The superscript d indicates a direct visit to some customer, and the superscript r indicates a visit to some customer after a replenishment trip to the depot.) At the final state $\mathcal{S}_F$, no further restocking or routing decision is possible. Nevertheless, we define $K(\mathcal{S}_F) = \{k_0\}$, where k_0 is a resetting decision leading the system back to the initial state $\mathcal{S}_0$. This will allow us to perform infinite-horizon analysis later on.

Again, note that this general MDP is not yet a model of the SVRPSD.

Each decision is associated with some transportation cost. We denote the cost of decision k made in some state $\mathcal{S}_a$ as $C_a(k)$. The decision costs are defined as follows:

$$
\begin{aligned}
C_0(k_j) &= c_{0j} && j \in \{1, \dots, n\}, \\
C_{i,q}(k_j^d) &= (c_{i0} + c_{0i}) \cdot \gamma(q) + c_{ij} && q \in \{-d_{max}, \dots, Q\}, i, j \in \{1, \dots, n\}, \\
C_{i,q}(k_j^r) &= (c_{i0} + c_{0i}) \cdot \gamma(q) + c_{i0} + c_{0j} && q \in \{-d_{max}, \dots, Q\}, i, j \in \{1, \dots, n\}, \\
C_{i,q}(k_0) &= (c_{i0} + c_{0i}) \cdot \gamma(q) + c_{i0} && i \in \{1, \dots, n\}, \\
C_F(k_0) &= 0,
\end{aligned}
$$

where $\gamma(q)$ is the number of roundtrips necessary to fulfill the possible pending demand of customer i, given by:

$$\gamma(q) = \left\lceil -\frac{q}{Q} \right\rceil^+ .$$

Note that the cost to satisfy the pending demand of a customer is incurred at the moment a decision to proceed to another node (customer or depot) is made.

We finally specify the transition probabilities for all pairs of states, for each possible decision. In a given state $\mathcal{S}_a$ when taking decision $k \in K(\mathcal{S}_a)$, the system moves into another state $\mathcal{S}_b$ with probability $\mathcal{P}_{(a)(b)}(k)$. All transition probabilities not explicitly defined have a value of zero. Starting with the initial state $\mathcal{S}_0$ and the final state $\mathcal{S}_F$ we have:

$$\mathcal{P}_{(0)(j,s)}(k_j) = p_{D_j}(Q - s),$$
$$\mathcal{P}_{(F)(0)}(k_0) = 1.$$

For every other state $\mathcal{S}_{i,q}$ we have the following probabilities of moving into state $\mathcal{S}_{j,s}$ or $\mathcal{S}_F$:

$$\mathcal{P}_{(i,q)(j,s)}(k_j^d) = p_{D_j}(Q \cdot \gamma(q) + q - s),$$
$$\mathcal{P}_{(i,q)(j,s)}(k_j^r) = p_{D_j}(Q - s),$$
$$\mathcal{P}_{(i,q)(F)}(k_0) = 1.$$

2.4. Policy constraining

In this section, we enforce the sequencing decisions to comply with some Hamiltonian tour.

A control policy in the previously defined MDP can be stated in terms of non-negative real values $\mathcal{D}_a(k)$, denoting the probability of choosing decision $k \in K(\mathcal{S}_a)$ when in some state $\mathcal{S}_a$. Therefore, for every state $\mathcal{S}_a$ we have $\sum_{k \in K(\mathcal{S}_a)} \mathcal{D}_a(k) = 1$.

Definition 1. Let $\mathcal{D}_a(k)$ be a control policy. Let the set of binary values $\bar{x}_{ij}$, $i, j \in \{0, \dots, n\}$ define a Hamiltonian tour in $\mathcal{G}$. Then, we say $\mathcal{D}_a(k)$ is a Hamiltonian policy if and only if:

$$\mathcal{D}_0(k_j) \le \bar{x}_{0j} \qquad j \in \{1, \dots, n\}, \tag{1}$$

and for all other states $\mathcal{S}_{i,q}$:

$$\mathcal{D}_{i,q}(k_j^d) \le \bar{x}_{ij} \qquad q \in \{-d_{max}, \dots, Q\}, i, j \in \{1, \dots, n\}, \tag{2}$$

$$\mathcal{D}_{i,q}(k_j^r) \le \bar{x}_{ij} \qquad q \in \{-d_{max}, \dots, Q\}, i, j \in \{1, \dots, n\}, \tag{3}$$

$$\mathcal{D}_{i,q}(k_0) \le \bar{x}_{i0} \qquad i \in \{1, \dots, n\}. \tag{4}$$

Proposition 1. *When applied to the general MDP, a Hamiltonian policy leads to an irreducible and periodic Markov chain, with period $n+2$.*

Proof. Consider the Markov chain obtained by applying a Hamiltonian policy to the general MDP, and by discarding every state that is not accessible from $\mathcal{S}_0$. In this Markov chain, from the initial state $\mathcal{S}_0$, the only unconstrained decision is to proceed to node j where $\bar{x}_{0j} = 1$. Therefore, the system must move to some state $\mathcal{S}_{j,s}$ (1 transition). Similarly, from other states $\mathcal{S}_{i,q} \neq \mathcal{S}_F$, the vehicle must proceed to the next node j given by some unique $\bar{x}_{ij} = 1$ (n transitions). Finally, from $\mathcal{S}_F$ the only available decision is to return to $\mathcal{S}_0$ (1 transition). So, after $n+2$ transitions the system must return to $\mathcal{S}_0$. □

Note that a restocking policy can be directly derived from a Hamiltonian policy by simply disregarding the resetting decision in $\mathcal{S}_F$.

In a Hamiltonian policy, no sequencing decisions are possible, yet all restocking decisions are still available. By further enforcing new policy constraints one could propose new policy classes, heuristically meaningful or perhaps tailored to specific applications. For example, inequalities forcing a replenishment when the residual capacity is below a certain threshold level, or before visiting a remote customer, or a customer where partial deliveries are not allowed. Such constraints potentially simplify the MILP introduced in the next section, increasing tractability while still allowing meaningful restocking policies. We conclude this section by showing how policy constraining can be applied to define the class of detour-to-depot restocking policies:

Example 1. In the detour-to-depot restocking policy, no preventive restocking occurs. This is enforced by the following constraints:

$$\mathcal{D}_{i,q}(k_j^r) = 0 \qquad q \in \{-d_{max}, \dots, Q\}, i, j \in \{1, \dots, n\} \tag{5}$$

2.5. A unified MILP

Proposition 1 allows us to apply the technique from Derman and Klein (1965). Assume the general MDP is controlled by some Hamiltonian policy $\mathcal{D}_a(k)$, related to the Hamiltonian tour given by the binary values $\bar{x}_{ij}$, and consider the resulting (irreducible) Markov chain. Then, there exists a unique steady-state probability π_a associated with each state $\mathcal{S}_a$. In the infinite-horizon, π_a corresponds to the fraction of time the Markov chain is in state $\mathcal{S}_a$. Let $y_a(k) \triangleq \pi_a \mathcal{D}_a(k)$, implying $\pi_a = \sum_{k \in K(\mathcal{S}_a)} y_a(k)$. The average cost per state transition of policy $\mathcal{D}_a(k)$ is given by:

$$\sum_{\mathcal{S}_a} \sum_{k \in K(\mathcal{S}_a)} C_a(k)\, y_a(k)\,, \tag{6}$$

where $\sum_{\mathcal{S}_a}$ denotes a summation over all the states. From Markov theory, we know that the probability conservation or balance equations must hold:

$$\sum_{k \in K(\mathcal{S}_a)} y_a(k) = \sum_{\mathcal{S}_b} \sum_{k \in K(\mathcal{S}_b)} y_b(k)\, \mathcal{P}_{ba}(k) \qquad \forall \mathcal{S}_a\,, \tag{7}$$

and since $\sum_{\mathcal{S}_a} \pi_a = 1$:

$$\sum_{\mathcal{S}_a} \sum_{k \in K(\mathcal{S}_a)} y_a(k) = 1\,. \tag{8}$$

By minimizing the linear function (6) on the decision variables $y_a(k) \geq 0$ subject to the linear constraints (7) and (8), we find an optimal Hamiltonian policy related to the Hamiltonian tour $\bar{x}_{ij}$. The corresponding restocking policy must be optimal to the finite-horizon problem of minimizing the expected cost from $\mathcal{S}_0$ to $\mathcal{S}_F$ (by a simple contradiction argument). Moreover, a full loop from $\mathcal{S}_0$ to $\mathcal{S}_0$ has an expected cost of $n+2$ times the objective in (6). Since the resetting decision has a cost of zero, this also corresponds to the expected cost from $\mathcal{S}_0$ to $\mathcal{S}_F$.

So far, we have assumed that an a priori tour was given. Our goal, however, is a model to simultaneously optimize a priori tours and restocking policies. Hence, we include the binary variables x_{ij}, $i, j \in \{0, \dots, n\}$, and the Hamiltonian tour constraints:

$$\sum_{i \in \{0,\dots,n\}} x_{ij} = 1 \qquad j \in \{0, \dots, n\}\,, \tag{9}$$

$$\sum_{j \in \{0,\dots,n\}} x_{ij} = 1 \qquad i \in \{0, \dots, n\}\,, \tag{10}$$

$$\sum_{i,j \in M} x_{ij} \leq |M| - 1 \qquad M \subset \{0, \dots, n\}\,,\ |M| \geq 2\,. \tag{11}$$

Inequalities (1)–(4) become constraints, and are rewritten in terms of $y_a(k)$ and x_{ij} as:

$$y_0(k_j) \leq x_{0j} \qquad j \in \{1, \dots, n\}\,, \tag{12}$$

$$y_{i,q}(k_j^d) \leq x_{ij} \qquad q \in \{-d_{max}, \dots, Q\}, i, j \in \{1, \dots, n\}\,, \tag{13}$$

$$y_{i,q}(k_j^r) \leq x_{ij} \qquad q \in \{-d_{max}, \dots, Q\}, i, j \in \{1, \dots, n\}\,, \tag{14}$$

$$y_{i,q}(k_0) \leq x_{i0} \qquad i \in \{1, \dots, n\}\,. \tag{15}$$

The resulting MILP is a unified model for the SVRPSD under a priori optimization, which considers all possible combinations of a priori tours and restocking policies as the solution space.

Note that if we consider the set of detour-to-depot policy constraints in (5), implying $y_{i,q}(k_j^r) = 0$ in (14), we arrive at a model for computing optimal a priori routes assuming the detour-to-depot policy.

2.6. Reformulation

In this section, we reformulate the MILP proposed in the previous section. In the new formulation, the number of constraints is considerably reduced, and the objective function assumes a more traditional form, where the a priori tour and the restocking policy are easily recognizable. The new formulation is also more amenable to solution methods, since it has a superior linear relaxation. Our reformulation steps are organized in the following two Lemmas. For the respective proofs, we refer to the appendixes to this paper.

Lemma 1. *In a feasible solution to the unified model presented in Section 2.5, constraints (12) are equivalent to*

$$(n+2)\, y_0(k_j) = x_{0j} \qquad j \in \{1, \dots, n\}\,, \tag{16}$$

constraints (13) and (14) are equivalent to

$$(n+2)\left(\sum_{q=-d_{max}}^{Q} y_{i,q}(k_j^d) + \sum_{q=-d_{max}}^{Q} y_{i,q}(k_j^r) \right) = x_{ij} \quad i, j \in \{1, \dots, n\}, \tag{17}$$

and constraints (15) are equivalent to

$$(n+2) \sum_{q=-d_{max}}^{Q} y_{i,q}(k_0) = x_{i0} \qquad i \in \{1, \dots, n\}\,. \tag{18}$$

The second Lemma concerns the reformulation of the objective function:

Lemma 2. *In a feasible solution to the unified model presented in Section 2.5, the objective function (given by $(n+2)$ times the expression in (6)) is equivalent to*

$$\sum_{i,j \in \{0,\dots,n\}} c_{ij} x_{ij} + (n+2) \sum_{\mathcal{S}_a} \sum_{k \in K(\mathcal{S}_a)} C_a^*(k)\, y_a(k)\,, \tag{19}$$

where $C_a^*(k)$ are defined as

$$
\begin{aligned}
C^*_0(k_j) &= 0 && j \in \{1,\ldots,n\}\,,\\
C^*_{i,q}(k^d_j) &= (c_{i0}+c_{0i})\cdot\gamma(q) && q \in \{-d_{max},\ldots,Q\}, i,j \in \{1,\ldots,n\}\,,\\
C^*_{i,q}(k^r_j) &= (c_{i0}+c_{0i})\cdot\gamma(q)+c_{i0}+c_{0j}-c_{ij} && q \in \{-d_{max},\ldots,Q\}, i,j \in \{1,\ldots,n\}\,,\\
C^*_{i,q}(k_0) &= (c_{i0}+c_{0i})\cdot\gamma(q) && i \in \{1,\ldots,n\}\,,\\
C^*_F(k_0) &= 0\,.
\end{aligned}
$$

In the reformulated unified model, we minimize (19) subject to the balance constraints (7), the Hamiltonian tour constraints (9)–(11), and the Hamiltonian policy constraints (16)–(18). The objective function resembles what usually appears in two-stage models for the VRPSD. The first term corresponds to the deterministic a priori route cost. The second term corresponds to the expected recourse cost for an optimal restocking policy.

The unified model only requires the demands to be discrete and bounded. As long as these assumptions are satisfied, any probability distribution can be used to model the demands, including different ones for different customers. Unfortunately, our model has a number of linear variables and constraints that grows polynomially with the capacity of the vehicle. Therefore, solving it to optimality in practical instances can be computationally prohibitive. For this reason, in the next section we propose an approach for heuristically solving larger instances of the SVRPSD.

3. Wait-and-see model and heuristic solution approach

The heuristic method we introduce later in this section requires a MILP that approximates our target MILP (i.e., the unified model) from below. For the purpose of this heuristic method, the quality of this approximation is not much relevant. A natural way of approximating (from below) the optimal solution of a stochastic minimization problem is computing its wait-and-see solution (Birge & Louveaux, 2011). For this reason, we now introduce a wait-and-see model for the SVRPSD under optimal restocking.

3.1. A wait-and-see model for the SVRPSD

So far, we considered that at route planning time we have a probabilistic description of the demands, which are only disclosed during route execution, sequentially, upon arrival of the vehicle at each customer. We now assume (for the purpose of proposing a wait-and-see model) that all demands are disclosed before route execution, but after route planning, and further that the demands are bounded by the capacity of the vehicle. In this case, we can find the optimal (wait-and-see) a priori tour by minimizing the following objective function:

$$
\sum_{i,j\in\{0,\ldots,n\}} c_{ij}x_{ij} + \mathbb{E}\left[\sum_{i,j\in\{1,\ldots,n\}} (c_{i0}+c_{0j}-c_{ij})\,p^\omega_{ij}\right], \tag{20}
$$

subject to the constraints:

$$
p^\omega_{ij} \le x_{ij} \qquad i,j \in \{1,\ldots,n\}, \omega\in\Omega\,, \tag{21}
$$

$$
\begin{aligned}
l^\omega_i &\le Q && i \in \{0,\ldots,n\}, \omega\in\Omega\,,\\
l^\omega_i &\ge d^\omega_i && i \in \{1,\ldots,n\}, \omega\in\Omega\,,\\
l^\omega_j &\le l^\omega_i - d^\omega_i + Qp^\omega_{ij} + (1-x_{ij})Q && i,j \in \{1,\ldots,n\}, \omega\in\Omega\,,
\end{aligned} \tag{22}
$$

$$
l^\omega_0 \le l^\omega_i - d^\omega_i + (1-x_{i0})Q \qquad i \in \{1,\ldots,n\}, \omega\in\Omega\,, \tag{23}
$$

$$
\begin{aligned}
x_{ij} &\in \{0,1\} && i,j \in \{0,\ldots,n\}\,,\\
p^\omega_{ij} &\in \{0,1\} && i,j \in \{1,\ldots,n\}, \omega\in\Omega\,,\\
l^\omega_0 &\ge 0 && \omega\in\Omega\,,
\end{aligned}
$$

and also subject to the Hamiltonian tour constraints (9)–(11).

This two-stage stochastic program with (integer) recourse considers the a priori tour definition as a first-stage decision, and the points at which restocking is to occur as second-stage decisions. As usual, the binary variables x_{ij} identify the a priori tour. The expected value is taken over all possible joint demand realizations, represented by the set Ω. Each scenario $\omega\in\Omega$ corresponds to a set of demand values $\{d^\omega_1,\ldots,d^\omega_n\}$. The binary variables p^ω_{ij} indicate whether a restocking occurs between customers i and j in scenario ω. Constraint (21) prohibits restocking between non-consecutive nodes. The linear variables l^ω_i, $i\in\{0,\ldots,n\}$, $\omega\in\Omega$, track the load of the vehicle when arriving at node i in scenario ω. Note that l^ω_0 represents the load when arriving back at the depot after all nodes have been visited. Constraints (22) and (23) ensure that the load of the vehicle along a tour is adjusted accordingly. Finally, note that when $|\Omega|=1$ the resulting model is essentially a model for the deterministic capacitated VRP.

3.2. Approximative branch-and-bound heuristic

In this section, we introduce the approximative branch-and-bound heuristic (ABnB), and illustrate how to apply it to solve the SVRPSD under optimal restocking.

We are interested in solving a difficult MILP, called *target problem* (TP), which has the objective function $\mathcal{Z}(\mathbf{x}|\mathbf{y})$, where $\mathbf{x}\in\mathcal{F}$ is a vector containing only integer variables, and $\mathbf{y}$ is a vector containing all the variables of the TP (integer and continuous) except those in $\mathbf{x}$. In the SVRPSD, the TP is the unified model from Section 2.6. We assume an efficient algorithm $\mathcal{A}$ is available for minimizing $\mathcal{Z}(\bar{\mathbf{x}}|\mathbf{y})$ for some fixed $\bar{\mathbf{x}}\in\mathcal{F}$. The result of this minimization is denoted by $\mathcal{A}(\bar{\mathbf{x}})$. This is e.g. the case for the unified model, where given a fixed tour, the optimal restocking solution can be calculated with the stochastic dynamic programming algorithm.

Let $\mathcal{L}(\mathbf{x}|\mathbf{w})$, $\mathbf{x}\in\mathcal{F}$, be the objective function of a simpler MILP, called *approximate problem* (AP). The vector $\mathbf{w}$ contains all the variables of the AP except those in $\mathbf{x}$. Assuming a minimization problem, for some fixed $\bar{\mathbf{x}}\in\mathcal{F}$ it is desirable that the AP bounds the TP from below, or:

$$
\min_{\mathbf{w}} \mathcal{L}(\bar{\mathbf{x}}|\mathbf{w}) \le \mathcal{A}(\bar{\mathbf{x}}) = \min_{\mathbf{y}} \mathcal{Z}(\bar{\mathbf{x}}|\mathbf{y})\,. \tag{24}
$$

We define $\alpha(\bar{\mathbf{x}})$ as the approximation ratio:

$$
\alpha(\bar{\mathbf{x}}) \triangleq \frac{\min_{\mathbf{w}} \mathcal{L}(\bar{\mathbf{x}}|\mathbf{w})}{\mathcal{A}(\bar{\mathbf{x}})}\,. \tag{25}
$$

The ABnB method is *not* a branch-and-bound method in the usual sense. The main idea of the ABnB method is to solve (heuristically) the TP by searching for solutions in the feasible space of the AP. We evaluate these solutions (from a TP perspective) with the algorithm $\mathcal{A}$, and we keep track of the quality of the approximation and the best solution found so far. Instead of coding procedures for performing this search, we leverage on existing branch-and-bound implementations offered by commercial solvers. Consider that the AP has been implemented in some commercial solver. After the three modifications described below, we effectively convert some commercial branch-and-bound implementation into a search procedure to generate a large number of feasible solutions to the TP:

1. At every incumbent node found when applying branch-and-bound on AP: let $[\bar{\mathbf{x}}|\bar{\mathbf{w}}]$ be the incumbent solution vector. We compute $\mathcal{A}(\bar{\mathbf{x}})$, and if $\mathcal{A}(\bar{\mathbf{x}}) < \mathtt{UB}$, we update our best TP upper-bound $\mathtt{UB}$ (initialized to ∞). We insert the approximation ratio $\alpha(\bar{\mathbf{x}})$ in a list $\mathcal{D}$, to keep track of the quality of the approximation. We add a cut to the AP which eliminates all solutions $[\bar{\mathbf{x}}|\mathbf{w}]$, for all $\mathbf{w}$. We finally reject the proposed incumbent.
2. Whenever some *cutoff updating criterion* is satisfied (e.g., after some amount of time has elapsed since the last update, or after some number of solutions have been found), we update the current cutoff value $\mathtt{CO}$ (initialized to ∞) to $f(\mathtt{CO}, \mathtt{UB}, \mathcal{D})$, where f is the *update function*.
3. When branching, we prune every node of the search tree with a lower bound greater or equal to $\min\{\mathtt{UB}, \mathtt{CO}\}$.

Note that, in the first modification, we instruct the solver to discard every incumbent solution to the AP. Therefore, the tree exploration performed by the solver becomes a mechanism for generating a large number of potential solutions to the AP (and, in turn, also to the TP). We keep track of the approximation quality and, from time to time, whenever the cutoff updating criterion is satisfied, we set a new cutoff value, to instruct the solver to discard any solution to the AP that is not promising from the TP perspective. For this inference step, the *quality* of the approximation is not much of an issue, but it is important that it is *consistent*, i.e., the variance of the values in $\mathcal{D}$ should be relatively small.

There are two main advantages in the ABnB method. First, implementation effort is minimal. These three modifications can be implemented using the *callback* architecture offered by major commercial solvers. In addition to that, it is only necessary to implement algorithm $\mathcal{A}$ and the local-search procedure. Second, the method scales very well to a parallel computing environment, since commercial branch-and-bound implementations already support efficient parallelization to explore the nodes of a branch-and-bound tree.

Our implementation of the ABnB heuristic for the SVRPSD uses the stochastic dynamic programming algorithm as the incumbent evaluation algorithm $\mathcal{A}$. The approximate problem is the wait-and-see model presented earlier in this section with continuous p_{ij}^{ω} variables. The recourse cost is approximated by sample averaging over 5 randomly generated scenarios. This small number of scenarios has been determined experimentally, and corresponds to a good tradeoff between consistent approximations and method efficiency. The cutoff value is updated when 6 minutes have been elapsed since the last TP upper-bound was found, and the updating function is defined as $\min(\mathtt{CO}, \max_{\alpha \in \mathcal{L}}(\alpha) \cdot \mathtt{UB})$. We also apply 2-opt local search (Lin, 1965) on the promising incumbent solutions.

4. Computational results

The unified model from Section 2.6 was implemented and solved with CPLEX ® version 12.6.1 branch-and-bound MILP solver. The ABnB heuristic was coded in C++, and also used CPLEX as the MILP solver, on which the necessary adaptations were implemented according to Section 3.2. All experiments were performed on an Intel i7-4790 3.6 gigahertz 4-core processor with available memory of 16 gigabytes.

The subtour elimination constraints were added dynamically to the model, whenever at some node of the search tree they were violated. We also used the unified model to compute optimal a priori tours when using the detour-to-depot policy by considering the additional constraints (5). In both cases, the time limit was set to two hours.

Table 1
Simplified instances, two-point distributions.

Instance	Load	Best	Restock (%)	Gap	TSP-OR (%)		DTD[a]		ABnB
sv1	0.75	317.3	2.8	–	317.3	0.0	317.7	0.1%	317.3
$n = 25$	1.00	325.1	5.2	–	325.1	0.0	326.4	0.4%	325.1
	1.25	333.3	6.9	–	334.2	0.3	336.9	1.1%	333.3
	1.50	341.3	9.1	–	343.7	0.7	348.3	2.0%	341.3
	2.00	358.2	13.4	–	363.4	1.4	371.6	3.7%	358.2
	2.50	376.0	16.9	–	383.5	2.0	395.0	5.1%	376.0
sv2	0.75	290.4	3.2	–	290.4	0.0	293.0	0.9%	290.4
$n = 25$	1.00	297.1	4.6	–	297.8	0.2	304.2	2.4%	297.1
	1.25	303.8	6.9	–	305.8	0.7	316.7	4.2%	303.8
	1.50	310.8	9.0	–	314.3	1.1	329.7	6.1%	310.8
	2.00	325.9	12.9	–	331.1	1.6	354.2	8.7%	325.9
	2.50	343.5	17.4	–	349.3	1.7	–	–	343.5
sv3	0.75	325.5	2.1	–	325.6	0.1	327.6	0.7%	325.5
$n = 25$	1.00	331.0	3.8	–	332.3	0.4	337.6	2.0%	331.0
	1.25	336.3	5.3	–	339.7	1.0	349.3	3.9%	336.3
	1.50	342.8	4.3	–	347.1	1.3	360.7	5.2%	342.8
	2.00	358.6	10.6	–	363.9	1.5	–	–	358.6
	2.50	376.5	13.9	–	383.8	1.9	–	–	376.5
sv4	0.75	442.2	2.0	0.6%	442.5	0.1	–	–	442.2
$n = 50$	1.00	448.8	3.4	2.0%	450.5	0.4	–	–	448.8
	1.25	455.3	4.0	2.7%	459.2	0.9	–	–	455.3
	1.50	465.0	6.7	4.5%	468.4	0.7	–	–	462.8
	2.00	479.9	7.6	5.0%	487.7	1.6	–	–	479.0
	2.50	502.7	11.8	6.8%	508.1	1.1	–	–	495.2
sv5	0.75	402.8	3.0	–	402.8	0.0	403.7	0.2%	402.8
$n = 50$	1.00	412.8	5.4	0.5%	412.8	0.0	–	–	412.8
	1.25	423.2	7.7	3.4%	423.2	0.0	–	–	423.2
	1.50	434.3	10.1	4.4%	434.3	0.0	–	–	434.1
	2.00	454.4	10.3	5.2%	457.2	0.6	–	–	454.4
	2.50	475.3	14.2	6.2%	480.1	1.0	–	–	475.3
sv6	0.75	400.3	2.4	–	400.3	0.0	401.8	0.4%	400.3
$n = 50$	1.00	407.3	4.1	0.6%	407.3	0.0	–	–	407.3
	1.25	415.0	5.9	2.2%	415.0	0.0	–	–	415.0
	1.50	423.3	7.7	3.2%	423.2	0.0	–	–	423.2
	2.00	440.3	10.5	4.8%	440.9	0.1	–	–	440.3
	2.50	458.3	13.1	5.6%	460.5	0.5	–	–	458.3

[a] Blank entries indicate optimal could not be found within the time limit.

4.1. Simplified instances, two-point distributions

Our first goal was to compare the performance of the detour-to-depot (DTD) and optimal restocking policies, for varying load scenarios. For this, we conducted tests on a set of simplified instances, derived from literature instances. In these instances, the demand of some customer i is zero with probability p_0 and u_i with probability $1 - p_0$, where u_i is an integer ranging from 1 to 4. The capacity of the vehicle is set to 10. The simplified instances are fully described in Tables B.4 and B.5, in the last appendix to this paper. The route load is defined as the total expected demand divided by the capacity of the vehicle. Each instance was tested under six different route load scenarios, obtained by varying p_0.

We also compare the results with the solution obtained by applying the optimal restocking policy to the TSP-optimal a priori tour (TSP-OR). The expected cost of a tour may be different depending on the direction of travel (Dror & Trudeau, 1986). For this reason, both orientations of the TSP tour are considered, and the best value is reported. The results are presented in Table 1.

The plain implementation of the unified model solved to optimality all instances with 25 nodes, and two instances with 50 nodes. As observed from the optimality gap of the unsolved instances, the higher the load in a route, the more difficult an instance becomes. Intuitively, on higher load scenarios the SVRPSD loses its TSP nature, and the restocking costs start to play a significant role. The overall effect is a deterioration of the linear relaxation bound, affecting negatively the performance of the branch-and-bound algorithm.

Table 2
CMT instances, Poisson distributions.

Instance	Q	Load	TSP-OR	ABnB (%)		A priori cost (%)	
CMT-1	1036	0.75	428.9	428.9	0.0	428.9	0.0
$n = 50$	777	1.00	434.9	434.9	0.0	428.9	0.0
TSP*= 428.9[a]	622	1.25	446.3	442.3	−0.9	432.0	0.7
	518	1.50	463.4	442.3	−4.5	432.0	0.7
	389	2.00	480.0	456.5	−4.9	429.0	0.0
	311	2.50	494.3	464.7	−6.0	435.5	1.5
	222	3.50	535.5	503.2	−6.0	451.5	5.3
	155	5.00	600.0	565.2	−5.8	450.5	5.0
CMT-2	1819	0.75	544.4	544.4	0.0	544.4	0.0
$n = 75$	1364	1.00	548.1	548.1	0.0	544.4	0.0
TSP = 544.4	1091	1.25	571.5	555.7	−2.8	549.9	1.0
	909	1.50	571.6	557.4	−2.5	552.7	1.5
	682	2.00	595.7	569.7	−4.4	553.8	1.7
	546	2.50	605.9	581.2	−4.1	562.5	3.3
	390	3.50	660.9	618.4	−6.4	576.9	6.0
	273	5.00	708.1	693.6	−2.1	593.3	9.0
CMT-3	1944	0.75	640.2	640.2	0.0	640.2	0.0
$n = 100$	1458	1.00	646.1	646.1	0.0	640.2	0.0
TSP = 640.2	1166	1.25	652.5	649.6	−0.4	648.3	1.3
	972	1.50	652.5	652.5	0.0	640.2	0.0
	729	2.00	666.6	661.8	−0.7	656.8	2.6
	583	2.50	697.1	676.3	−3.0	671.3	4.9
	417	3.50	704.6	695.4	−1.3	654.6	2.3
	292	5.00	821.4	783.2	−4.7	692.2	8.1
CMT-4	2980	0.75	707.9	707.9	0.0	707.9	0.0
$n = 150$	2235	1.00	712.2	712.2	0.0	707.9	0.0
TSP = 707.9	1788	1.25	719.3	719.0	0.0	716.9	1.3
	1490	1.50	732.4	726.8	−0.8	718.6	1.5
	1118	2.00	749.1	733.4	−2.1	723.1	2.2
	894	2.50	756.7	730.7	−3.4	711.0	0.4
	639	3.50	796.5	771.9	−3.1	742.7	4.9
	447	5.00	861.3	805.6	−6.5	764.5	8.0

[a] Value of the optimal TSP tour on the instance.

When the load in a route is less than or equal to the capacity of the vehicle, the myopic solution obtained by ignoring the restocking cost and minimizing only the a priori cost is of high quality. In fact, in all instances with a load of 0.75 and 1.00 solved to optimality, the TSP solution coupled with the optimal restocking policy was always within 0.4% of the optimal solution. Note that this solution is much simpler from an algorithmic perspective, as it only involves applying twice (one for each orientation) a dynamic programming algorithm on the optimal TSP solution. The ABnB heuristic performed well on the set of simplified instances, matching all optimal solutions found by the unified model, even with significantly less available computational time (only up to 20 minutes were allowed in these runs). In addition, new upper-bounds were found for 5 instances.

4.2. Larger instances, Poisson distributions

In order to further assess the performance of the heuristic, we executed a second round of tests on larger, more practical instances. These new instances were obtained by combining 8 different load scenarios with 4 instances from Christofides, Mingozzi, and Toth (1979) adapted to the stochastic case. The adaptation consisted basically in considering the demands as Poisson distributed, with the rates as given by the demand values stated in the instances. The load scenarios ranged from 0.75 to 5.00. The two highest loads experimented (3.50 and 5.00) requires the vehicle to perform several replenishment trips, and in practice may mean that the route must be executed across multiple periods (especially when the number of customers is high). The capacity of the vehicle in each instance was determined by dividing the total expected demand by the load value, and rounding the result to the nearest integer. The time limit for these runs was set to 1 hour. The results are presented in Table 2.

Table 3
Louveaux and Salazar-González (2018) instances, triangular distributions.

Instance	Q	Load	Distr.	LS-18	ABnB (%)	
E031-09h	84	1.79	3	332.753	332.475	−0.1
$n = 30$	79	1.90	3	335.296	335.218	0.0
	84	1.79	9	337.674	337.198	−0.1
	79	1.90	9	344.525	343.205	−0.4
E051−05e	139	1.80	3	441.000	441.000	0.0
$n = 50$	132	1.89	3	441.311	441.154	0.0
	139	1.80	9	443.006	443.006	0.0
	132	1.89	9	448.083	446.616	−0.3
E076−07s	209	1.79	3	549.005	552.103	0.6
$n = 75$	198	1.89	3	550.164	550.314	0.0
	209	1.79	9	550.815	550.352	−0.1
	198	1.89	9	554.800	560.799	1.1
E101−08e	278	1.80	3	640.001	641.017	0.2
$n = 100$	264	1.89	3	641.732	642.802	0.2
	278	1.80	9	641.304	642.839	0.2
	264	1.89	9	646.119	646.376	0.0
A034−02f	92	1.79	3	1404.64	1406.24	0.1
$n = 33$	87	1.90	3	1418.72	1417.81	−0.1
	92	1.79	9	1436.02	1435.71	0.0
	87	1.90	9	1484.13	1484.72	0.0
A048-03f	131	1.79	3	1812.02	1812.03	0.0
$n = 47$	124	1.90	3	1818.16	1819.35	0.1
	131	1.79	9	1827.96	1831.37	0.2
	124	1.90	9	1864.33	1870.68	0.3
A071−03f	195	1.79	3	1979.43	1979.52	0.0
$n = 70$	185	1.89	3	1985.93	1986.15	0.0
	195	1.79	9	1993.74	1995.71	0.1
	185	1.89	9	2027.83	2034.64	0.3

We observe that for load values of 0.75 and 1.00, the ABnB heuristic is unable to improve upon the solution given by the TSP tour coupled with the optimal restocking policy. These results are in line with the previous findings, which attested the quality of the TSP-based solution when the route loads are low. When the load increases, however, the TSP solution quickly deteriorates. Already at a load of 1.25, the TSP solution can be at least 2.8% suboptimal, and this gap grows to 6.4% when the load is 3.50, and to 6.5% when the load is 5.00.

Finally, we compare the a priori length of the best solution found by the ABnB heuristic with the optimal TSP length for each instance. For load values up to 1.50, the a priori length of the best solutions found are no more than 1.5% above the TSP length. On the other hand, at higher load values, the best solutions found have an a priori length up to 9.0% higher than the TSP length. Therefore, the overall solution value can be improved even when using moderately suboptimal TSP a priori tours. This further confirms how the SVRPSD, under the a priori approach and optimal policy assumption, loses its similarity with the TSP in the presence of high loads.

4.3. Louveaux and Salazar-González (2018) instances, triangular distributions

Our last set of tests were conducted on instances from Louveaux and Salazar-González (2018). We selected all 28 instances with 2 vehicles, which were solved either to optimality (24 instances) or to a very small gap (4 instances). The solution values of these instances (mostly optimal) provide a good benchmark to the ABnB heuristic, since using a single tour is likely not to offer much advantage when compared to using only 2 routes. As in the original work, the demands were assumed identically distributed, and followed triangular distributions with either 3 or 9 possible demand values (for details, see Louveaux & Salazar-González, 2018). We also allowed a maximum computational time of 1 hour for these runs. The results are presented in Table 3.

The ABnB method was able to find solutions with values very close to the reference values (average deviation of only 0.08%). In some cases, solutions with lower costs could be found, due to additional gains obtained by combining all customers into a single tour. These gains, however, were only marginal when using one tour instead of 2 routes.

5. Conclusions

In this paper, we considered the single-vehicle routing problem with stochastic demands (SVRPSD) under optimal restocking. We developed a unified model for the problem, which simultaneously computes an optimal a priori tour and an optimal restocking policy. The model can be easily adapted to other restocking policies by considering additional policy constraints. In addition, the model is valid for general (discrete) demand probability distributions and route loads. Our results show that the optimal restocking solution is up to 8.7% superior to the detour-to-depot solution. The difference, however, is only significant when the expected demand in a route is equal to or exceeds the capacity of the vehicle, a scenario that has not been traditionally considered in the literature. Moreover, the solution obtained by simply combining the optimal TSP tour with the optimal restocking policy outperforms the optimal detour-to-depot solution, for every load scenario considered, and at a much lower algorithmic sophistication.

Furthermore, we introduced the approximative branch-and-bound (ABnB) framework to heuristically solve difficult mixed-integer linear programs, and illustrated how to apply it to find good solutions to the SVRPSD. With a modest implementation effort and computing time, the ABnB heuristic was able to find the optimal solution of all simplified instances we could solve to optimality with the unified model. The method is also appropriate for solving larger instances of the SVRPSD, as shown by experiments performed on literature instances with up to 150 nodes. In these experiments, the solutions obtained with the ABnB heuristic outperformed by up to 6.5% the good upper-bound given by combining the TSP tour with the optimal restocking policy.

We conclude with a few remarks of practical relevance. On instances with route loads equal to or less than the capacity of the vehicle, the a priori cost clearly dominates the expected restocking cost. In these cases, combining the TSP solution with the optimal restocking policy leads to a near-optimal solution to the SVRPSD. The higher the load, the more important it becomes to explicitly consider the restocking cost. Already at a load of 1.25, the TSP-based solution can be almost 3% suboptimal. More elaborate solution approaches are particularly recommended in high load scenarios. In such cases, the myopic strategy of a priori cost minimization is clearly suboptimal, suggesting the existence of a tradeoff between restocking and a priori costs.

Acknowledgment

We thank two anonymous referees for their comments, which led to significant improvements in this paper.

Appendix A. Proof of Lemmas 1 and 2

Lemma 1. *In a feasible solution to the unified model presented in Section 2.5, constraints (12) are equivalent to*

$$(n+2)\, y_0(k_j) = x_{0j} \qquad j \in \{1,\dots,n\}, \tag{16}$$

constraints (13) and (14) are equivalent to

$$(n+2)\left(\sum_{q=-d_{max}}^{Q} y_{i,q}(k_j^d) + \sum_{q=-d_{max}}^{Q} y_{i,q}(k_j^r)\right) = x_{ij} \quad i,j \in \{1,\dots,n\}, \tag{17}$$

and constraints (15) are equivalent to

$$(n+2)\sum_{q=-d_{max}}^{Q} y_{i,q}(k_0) = x_{i0} \qquad i \in \{1,\dots,n\}. \tag{18}$$

Proof. We start with constraints (12). In a feasible solution, x_{ij} are integer and define a Hamiltonian tour, and so the resulting Markov chain is periodic, with a period of $n+2$ (see Proposition 1). $\mathcal{S}_0$ is the single state in its period class, so $\pi_0 = \sum_{i=1}^{n} y_0(k_i) = (n+2)^{-1}$. Let l be the first node in the tour, so $x_{0l} = 1$. Considering nonnegativity of $y_0(k_i)$ and constraints (12), it must hold that $y_0(k_i) = (n+2)^{-1}$ if $k = l$, and 0 otherwise. Therefore, (16) holds. The implication in the opposite direction is trivial.

We now analyze constraints (13) and (14). With the same argument based on periodicity, all states corresponding to the same node are in the same period class. Hence, the following must hold:

$$\sum_{q=-d_{max}}^{Q} \pi_{i,q} = \sum_{q=-d_{max}}^{Q} \sum_{k\in K(\mathcal{S}_{i,q})} y_{i,q}(k) = (n+2)^{-1} \qquad i \in \{1,\dots,n\} \tag{A.1}$$

Let l be the node following i in the tour, so $x_{il} = 1$. If $l = 0$, then constraints (13) and (14) are not applicable. So, whenever constraints (13) and (14) are applicable, we have $x_{i0} = 0$ and, by (15), $\sum_{q=-d_{max}}^{Q} y_{i,q}(k_0) = 0$. Considering nonnegativity of $y_{i,\,q}(k)$ and constraints (13) and (14), it must be that:

$$\sum_{q=-d_{max}}^{Q} y_{i,q}(k_j^d) + \sum_{q=-d_{max}}^{Q} y_{i,q}(k_j^r) = \begin{cases} (n+2)^{-1}, & \text{if } j = l, \\ 0, & \text{otherwise}. \end{cases}$$

Therefore, (17) holds. The implication in the opposite direction is trivial.

We finally consider (15). Let l be the node immediately preceding 0 in the tour, so $x_{l0} = 1$. All states corresponding to node l must be in the same period class. Considering nonnegativity of $y_{i,\,q}(k)$ and constraints (13)–(15), it must be that:

$$\sum_{q=-d_{max}}^{Q} y_{i,q}(k_0) = \begin{cases} (n+2)^{-1}, & \text{if } i = l, \\ 0, & \text{otherwise}. \end{cases}$$

Therefore, (18) holds. The implication in the opposite direction is trivial. □

Lemma 2. *In a feasible solution to the unified model presented in Section 2.5, the objective function (given by $(n+2)$ times the expression in (6)) is equivalent to*

$$\sum_{i,j\in\{0,\dots,n\}} c_{ij}x_{ij} + (n+2)\sum_{\mathcal{S}_a}\sum_{k\in K(\mathcal{S}_a)} C_a^*(k)\, y_a(k), \tag{19}$$

where $C_a^*(k)$ are defined as

$$\begin{aligned}
C_0^*(k_j) &= 0 & j &\in \{1,\dots,n\}, \\
C_{i,q}^*(k_j^d) &= (c_{i0}+c_{0i})\cdot\gamma(q) & q &\in \{-d_{max},\dots,Q\}, i,j \in \{1,\dots,n\}, \\
C_{i,q}^*(k_j^r) &= (c_{i0}+c_{0i})\cdot\gamma(q) + c_{i0} + c_{0j} - c_{ij} & q &\in \{-d_{max},\dots,Q\}, i,j \in \{1,\dots,n\}, \\
C_{i,q}^*(k_0) &= (c_{i0}+c_{0i})\cdot\gamma(q) & i &\in \{1,\dots,n\}, \\
C_F^*(k_0) &= 0.
\end{aligned}$$

Table B.4
Description of the simplified instances sv1 to sv6 – nodes 0–25.

n	sv1			sv2			sv3			sv4			sv5			sv6		
	x	y	D	x	y	D	x	y	D	x	y	D	x	y	D	x	y	D
0	40	40	–	40	40	–	40	40	–	35	35	–	35	35	–	35	35	–
1	22	22	3	36	26	3	21	45	4	41	49	3	35	17	4	55	45	2
2	45	35	3	55	20	2	33	34	4	55	20	4	15	30	3	25	30	4
3	50	50	4	55	45	1	26	59	2	20	50	2	10	43	2	55	60	1
4	40	66	3	55	65	2	35	51	1	30	60	1	20	65	1	50	35	4
5	62	35	1	62	57	4	62	24	1	30	25	4	15	10	1	30	5	1
6	21	36	4	33	44	1	9	56	2	10	20	4	5	30	3	20	40	1
7	62	48	4	66	14	3	44	13	1	15	60	2	45	65	2	45	20	4
8	26	13	1	11	28	3	7	43	4	45	10	3	55	5	2	65	35	4
9	17	64	3	41	46	3	55	34	2	65	20	3	45	30	2	35	40	1
10	35	16	2	52	26	2	43	26	3	41	37	1	64	42	2	40	60	2
11	31	76	2	22	53	1	26	29	4	31	52	4	35	69	4	53	52	4
12	50	40	4	55	50	3	54	10	1	65	55	3	63	65	1	2	60	2
13	60	15	3	47	66	1	30	60	1	20	20	1	5	5	1	60	12	4
14	30	50	2	12	17	4	15	14	4	40	25	2	42	7	2	24	12	2
15	16	19	3	21	48	2	50	30	2	23	3	4	11	14	3	6	38	1
16	51	42	4	50	15	4	48	21	1	2	48	2	8	56	4	13	52	1
17	12	38	2	15	56	3	29	39	1	6	68	3	47	47	2	49	58	3
18	54	38	4	55	57	3	67	41	1	27	43	2	37	31	3	57	29	3
19	10	70	4	6	25	3	65	27	3	63	23	3	53	12	3	32	12	4
20	40	60	2	70	64	1	64	4	2	36	26	3	21	24	1	17	34	4
21	36	6	4	30	20	3	20	30	4	12	24	2	24	58	4	27	69	3
22	15	5	1	50	70	2	57	72	2	15	77	2	62	77	1	49	73	2
23	45	42	3	38	33	3	50	4	1	67	5	2	56	39	1	37	47	3
24	66	8	4	59	5	4	35	60	2	37	56	2	57	68	4	47	16	2
25	27	24	3	40	20	3	40	37	1	44	17	2	46	13	1	49	11	3

Table B.5
Description of the simplified instances sv4 to sv6 – nodes 26–50.

n	sv4			sv5			sv6		
	x	y	D	x	y	D	x	y	D
26	49	42	2	53	43	3	61	52	4
27	57	48	4	56	37	3	55	54	3
28	15	47	1	14	37	4	11	31	4
29	16	22	2	4	18	4	28	18	3
30	26	52	2	26	35	4	31	67	4
31	15	19	2	22	22	3	18	24	3
32	26	27	4	25	24	1	22	27	4
33	25	21	1	19	21	3	20	26	2
34	18	18	2	37	52	4	49	49	3
35	52	64	1	20	26	2	40	30	2
36	21	47	4	17	63	4	31	62	4
37	52	33	4	51	21	2	42	41	4
38	31	32	2	5	25	4	12	42	2
39	36	16	3	52	41	4	27	23	4
40	17	33	2	13	13	2	57	58	1
41	62	42	1	42	57	1	16	57	1
42	8	52	3	7	38	1	27	68	4
43	30	48	4	43	67	3	58	48	3
44	58	27	4	37	69	4	38	46	1
45	46	10	4	61	33	3	62	63	2
46	63	69	3	32	22	2	45	35	4
47	59	15	3	5	6	4	10	17	4
48	21	10	2	5	64	4	30	15	1
49	39	10	3	32	39	2	25	32	2
50	25	55	2	48	28	3	56	37	3

Proof. The objective function in the original formulation can be rewritten as $(n+2)$ multiplied by

$$\sum_{j\in\{1,\dots,n\}} C_0(k_j)\,y_0(k_j) + \sum_{\mathcal{S}_{i,q}} \sum_{k\in K(\mathcal{S}_{i,q})} C_{i,q}(k)\,y_{i,q}(k)\,, \tag{A.2}$$

where $\sum_{\mathcal{S}_{i,q}}$ indicates a summation over all states except states $\mathcal{S}_0$ and $\mathcal{S}_F$. The first term in (A.2) corresponds to $\mathcal{S}_0$, and the second term corresponds to all other states $\mathcal{S}_{i,q} \neq \mathcal{S}_F$. We omit from (A.2) the term corresponding to $\mathcal{S}_F$ which is zero.

Considering (16), the first term in (A.2) multiplied by $(n+2)$ is equal to

$$\sum_{j\in\{0,\dots,n\}} c_{0j}x_{0j}\,. \tag{A.3}$$

The second term in (A.2) can be further rewritten as

$$\sum_{\mathcal{S}_{i,q}} \sum_{j\in\{1,\dots,n\}} C_{i,q}(k_j^d)\,y_{i,q}(k_j^d) + \sum_{\mathcal{S}_{i,q}} \sum_{j\in\{1,\dots,n\}} C_{i,q}(k_j^r)\,y_{i,q}(k_j^r) + \sum_{\mathcal{S}_{i,q}} C_{i,q}(k_0)\,y_{i,q}(k_0)\,. \tag{A.4}$$

Considering (17), the sum of the two first terms in (A.4) multiplied by $(n+2)$ is equal to

$$\sum_{i,j\in\{1,\dots,n\}} c_{ij}x_{ij} + (n+2)\sum_{\mathcal{S}_a} \sum_{k\in K(\mathcal{S}_a)} C_a^*(k)\,y_a(k)\,. \tag{A.5}$$

Finally, considering (18) the last term in (A.4) multiplied by $(n+2)$ is equal to

$$\sum_{i\in\{0,\dots,n\}} c_{i0}x_{i0}\,. \tag{A.6}$$

The reformulated objective function in (19) follows by combining (A.3), (A.5) and (A.6). □

Appendix B. Description of the simplified instances

The simplified instances sv1 to sv6 used to test the unified model are based on instances CMT2 (75 nodes) and CMT4 (150 nodes) from Christofides et al. (1979). Instance sv1 is obtained from instance CMT2 by considering only every third node from CMT2, starting with the first node. Instances sv2 and sv3 are obtained in a similar way, but starting with the second and third nodes respectively. The depot coordinates in sv1, sv2 and sv3 correspond to the depot coordinates in CMT2. The same rule is applied to generate instances sv4, sv5 and sv6 from instance CMT4.

The demand parameter of each node in instances sv1 to sv6 is obtained by taking 1 plus the remainder of the division by 4 of the demand of the original node in instance CMT2 or CMT4, thus yielding an integer in the interval [1,4].

For avoidance of doubt, in the following tables we provide all the coordinates and demand parameters of instances sv1 to sv6.

References

Bastian, C., & Rinnooy Kan, A. H. G. (1992). The stochastic vehicle routing problem revisited. *European Journal of Operational Research, 56*, 407–412.

Bertazzi, L., & Secomandi, N. (2018). Faster rollout search for the vehicle routing problem with stochastic demands and restocking. *European Journal of Operational Research, 270*, 487–497.

Bertsimas, D. J. (1992). A vehicle routing problem with stochastic demand. *Operations Research, 40*, 574–585.

Bertsimas, D. J., Chervi, P., & Peterson, M. (1995). Computational approaches to stochastic vehicle routing problems. *Transportation Science, 29*, 342–352.

Biesinger, B., Hu, B., & Raidl, G. (2018). A genetic algorithm in combination with a solution archive for solving the generalized vehicle routing problem with stochastic demands. *Transportation Science, 52*, 673–690.

Birge, J. R., & Louveaux, F. V. (2011). *Introduction to stochastic programming*. Springer Science & Business Media.

Chepuri, K., & Homem-de Mello, T. (2005). Solving the vehicle routing problem with stochastic demands using the cross-entropy method. *Annals of Operations Research, 134*, 153–181.

Christofides, N., Mingozzi, A., & Toth, P. (1979). The vehicle routing problem. *Combinatorial optimization* (pp. 315–338). Chichester: Wiley.

Clarke, G., & Wright, J. W. (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research, 12*, 568–581.

Derman, C. (1962). On sequential decisions and Markov chains. *Management Science, 9*, 16–24.

Derman, C., & Klein, M. (1965). Some remarks on finite horizon Markovian decision models. *Operations Research, 13*, 272–278.

Dinh, T., Fukasawa, R., & Luedtke, J. (2018). Exact algorithms for the chance-constrained vehicle routing problem. *Mathematical Programming, 172*(1–2), 105–138.

Dror, M., Laporte, G., & Louveaux, F. V. (1993). Vehicle routing with stochastic demands and restricted failures. *ZOR Zeitschrift für Operations Research Methods and Models of Operations Research, 37*, 273–283.

Dror, M., Laporte, G., & Trudeau, P. (1989). Vehicle routing with stochastic demands: Properties and solution frameworks. *Transportation Science, 23*, 166–176.

Dror, M., & Trudeau, P. (1986). Stochastic vehicle routing with modified savings algorithm. *European Journal of Operational Research, 23*, 228–235.

Florio, A. M., Hartl, R. F., & Minner, S. (2018). New exact algorithm and solution properties for the vehicle routing problem with stochastic demands. *Technical Report*. arXiv: 1806.08549.

Gendreau, M., Jabali, O., & Rei, W. (2016). 50th anniversary invited article—future research directions in stochastic vehicle routing. *Transportation Science, 50*, 1163–1173.

Gendreau, M., Laporte, G., & Séguin, R. (1996). Stochastic vehicle routing. *European Journal of Operational Research, 88*, 3–12.

Goodson, J. C., Ohlmann, J. W., & Thomas, B. W. (2013). Rollout policies for dynamic solutions to the multivehicle routing problem with stochastic demand and duration limits. *Operations Research, 61*, 138–154.

Goodson, J. C., Thomas, B. W., & Ohlmann, J. W. (2016). Restocking-based rollout policies for the vehicle routing problem with stochastic demand and duration limits. *Transportation Science, 50*, 591–607.

Groër, C., Golden, B. L., & Wasil, E. (2009). The consistent vehicle routing problem. *Manufacturing & Service Operations Management, 11*, 630–643.

Hjorring, C., & Holt, J. (1999). New optimality cuts for a single vehicle stochastic routing problem. *Annals of Operations Research, 86*, 569–584.

Kyriakidis, E. G., Dimitrakos, T. D., & Karamatsoukis, C. C. (2017). Optimal delivery of two similar products to N ordered customers with product preferences. *International Journal of Production Economics*. doi:10.1016/j.ijpe.2017.12.017.

Laporte, G., & Louveaux, F. V. (1993). The integer L-shaped method for stochastic integer programs with complete recourse. *Operations Research Letters, 13*, 133–142.

Lin, S. (1965). Computer solutions of the traveling salesman problem. *Bell System Technical Journal, 44*, 2245–2269.

Louveaux, F. V., & Salazar-González, J. J. (2018). Exact approach for the vehicle routing problem with stochastic demands and preventive returns. *Transportation Science*. doi:10.1287/trsc.2017.0780.

Noorizadegan, M., & Chen, B. (2018). Vehicle routing with probabilistic capacity constraints. *European Journal of Operational Research, 270*, 544–555.

Novoa, C., & Storer, R. (2009). An approximate dynamic programming approach for the vehicle routing problem with stochastic demands. *European Journal of Operational Research, 196*, 509–515.

Pandelis, D. G., Kyriakidis, E. G., & Dimitrakos, T. D. (2012). Single vehicle routing problems with a predefined customer sequence, compartmentalized load and stochastic demands. *European Journal of Operational Research, 217*, 324–332.

Rei, W., Gendreau, M., & Soriano, P. (2010). A hybrid Monte Carlo local branching algorithm for the single vehicle routing problem with stochastic demands. *Transportation Science, 44*, 136–146.

Salavati-Khoshghalb, M., Gendreau, M., Jabali, O., & Rei, W. (2018). An exact algorithm to solve the vehicle routing problem with stochastic demands under an optimal restocking policy. *European Journal of Operational Research, 273*, 175–189.

Secomandi, N. (2001). A rollout policy for the vehicle routing problem with stochastic demands. *Operations Research, 49*, 796–802.

Secomandi, N., & Margot, F. (2009). Reoptimization approaches for the vehicle-routing problem with stochastic demands. *Operations Research, 57*, 214–230.

Stewart, W. R., & Golden, B. L. (1983). Stochastic vehicle routing: A comprehensive approach. *European Journal of Operational Research, 14*, 371–385.

Tatarakis, A., & Minis, I. (2009). Stochastic single vehicle routing with a predefined customer sequence and multiple depot returns. *European Journal of Operational Research, 197*, 557–571.

Tsirimpas, P., Tatarakis, A., Minis, I., & Kyriakidis, E. G. (2008). Single vehicle routing with a predefined customer sequence and multiple depot returns. *European Journal of Operational Research, 187*, 483–495.

Yang, W.-H., Mathur, K., & Ballou, R. H. (2000). Stochastic vehicle routing problem with restocking. *Transportation Science, 34*, 99–112.

Yee, J. R., & Golden, B. L. (1980). A note on determining operating strategies for probabilistic vehicle routing. *Naval Research Logistics Quarterly, 27*, 159–163.