The Vehicle Routing Problem with Stochastic Demand and Duration Constraints

Alan L. Erera, Juan C. Morales, and Martin Savelsbergh

The Supply Chain & Logistics Institute
Georgia Institute of Technology
School of Industrial and Systems Engineering
Atlanta, GA 30332-0205

March 4, 2010

Abstract

Time considerations have been largely ignored in the study of vehicle routing problems with stochastic demands, even though they are crucial in practice. We show that tour duration limits can effectively and efficiently be incorporated in solution approaches that build fixed, or a priori, tours for such problems. We do so by assuming that each tour must be duration-feasible for all demand realizations, and determine the maximum duration of a given delivery tour by solving the optimization problem of an adversary. A computational study demonstrates the approach, and shows that enforcing tour duration limits impacts the structure of nearly-best solutions and may create the need for additional tours. However, for the instances considered, the price paid for robustness is small as the increase in total expected tour duration is modest.

1 Introduction

The vehicle routing problem with stochastic demand (VRPSD) has been studied alongside more traditional deterministic routing problems since initial work by Tillman (1969), but has received relatively less attention. Most recent work focuses on finding fixed, or a priori, tours. A key challenge in these problems is modeling tour feasibility. In practice, it is impossible if not extremely costly to ensure feasibility of a set of a priori tours for all possible demand realizations. Thus, most models are more flexible and allow some recourse decisions. Usually, the feasibility of the a priori tour set is determined given a fixed recourse policy that specifies the actions to take whenever an infeasibility, or tour failure, occurs during operations.
The most popular recourse policy studied in the literature is one that we denote detour-to-depot: when a vehicle capacity infeasibility arises, i.e., when the demand of a customer cannot be satisfied given the current vehicle remaining capacity, the vehicle makes an return trip to the depot to restock (or unload) before resuming its tour. Various heuristic and exact optimization approaches for constructing a set of a priori tours minimizing expected costs given this recourse policy have been proposed and analyzed in the literature.

In practice, however, the unrestricted use of detour-to-depot recourse is probably not possible. An out-and-back trip to the depot results in both additional cost and additional travel time, and thus a longer duration of the delivery tour. Since the number of hours that drivers can spend driving or on duty is typically limited by regulation, and since the operating day of most pickup/delivery operations is also constrained by business practices, the duration of an actual tour usually cannot be extended beyond certain hard limits. Thus, the duration of a tour, including any additional travel time incurred during recourse actions, cannot exceed an upper bound. This additional feasibility criterion leads to a problem that we denote the Vehicle Routing Problem with Stochastic Demand and Duration Constraints (VRPSD-DC).

In this paper, we formally define the VRPSD-DC, develop a tabu search heuristic for its solution, and perform a computational study to analyze the impact of duration constraints on the expected delivery costs and the structure of the set of a priori tours. The solution method relies requires solving an “adversarial” optimization problem, which determines a customer demand realization that maximizes the actual execution duration of an a priori tour. We show that this adversarial problem can be solved in pseudo-polynomial time for a large class of recourse policies, but in polynomial time for certain ones, including the typical detour-to-depot recourse policy. The computational study demonstrates that imposing tour duration limits impacts the structure of nearly-best solutions and may create the need for additional vehicle tours. However, for the instances considered, the increase in total expected tour duration observed is relatively modest.

The remainder of the paper is organized as follows. In Section 2, we briefly review related literature. In Section 3, we introduce and define the VRPSD-DC. In Section 4, we discuss the adversarial problem and present a longest path algorithm for its solution. In Section 5, we present a tabu search algorithm for constructing a set of a priori delivery tours with low expected costs. In Section 6, we present the results of an extensive computational study. Finally, in Section 7, we discuss potential future research in this area, and present some final observations.

2 Related Literature

The vehicle routing problem with stochastic demands (VRPSD) is a well-studied problem. For excellent surveys, see Dror et al. (1989) and Gendreau et al. (1996a). Note that we are not aware of any existing research that studies VRPSD problems with tour duration limits.
Modeling and solution approaches for the VRPSD can be divided into three main research streams:

1. Approaches based on chance constrained models;
2. Approaches based on stochastic programming with recourse models; and
3. Approaches based on Markov decision models.

An early chance-constrained model is proposed in Stewart and Golden (1983). The paper presents a model to identify minimum cost tours subject to a threshold constraint on the probability of a tour failure. A similar approach is proposed in Laporte et al. (1989); the model uses fewer variables, but requires a homogeneous fleet of vehicles. Each of these models can be transformed into a deterministic vehicle routing problem under reasonable assumptions. One major deficiency these models is that although the probability of a failure is constrained, the customer locations of failures (and hence their costs) are ignored. Tours with the same \textit{a priori} cost and the same failure probability can have significantly different expected costs, depending on the possible failure locations.

Two-stage stochastic programming models that minimize expected tour costs have also been proposed for VRPSD, and they typically consider recourse actions such that the second stage cost can be determined without optimization; such models are denoted \textit{a priori optimization} models. Dror and Trudeau (1986) provide part of the initial inspiration for this idea. This early paper argues that simpler penalty-type stochastic optimization models can be improved by computing the expected cost of a planned tour taking into consideration the location of a tour failure, assuming that upon failure the recourse action is for the vehicle to travel to the depot to reload before continuing (\textit{i.e.}, detour-to-depot).

An early exact solution approach for such an \textit{a priori} optimization model is given in Gendreau et al. (1995), where an integer L-shaped method is proposed. Laporte et al. (2002) presents an improved approach capable of handling larger instances. An important element of this improved approach is the use of lower bounds at the root node which helps to speed up solution times. These bounds are calculated under the assumption that the expected value of demand on any tour is less than or equal to the vehicle capacity.

Significant effort has been devoted to the development of heuristics for solving the VRPSD. Stewart and Golden (1983) and Dror and Trudeau (1986) both propose algorithms inspired by the ideas underlying the savings heuristic of Clarke and Wright (1964) for the deterministic VRP. An efficient local search heuristic is presented in Savelsbergh and Goetschalckx (1995), and their computational results indicate that the approach compares favorably to the savings-based algorithms. Gendreau et al. (1996b) extends local search ideas with a tabu search metaheuristic to find solutions to the \textit{a priori} model proposed in Gendreau et al. (1995). The quality of the tabu search is assessed by comparison to results using the exact solution approach for a common set of instances; the tabu search algorithm produced an optimal set of tours in 89.45\% of the cases. Furthermore, an average deviation from optimal
expected cost of only 0.38% was observed. Note that most of the research on recourse models for the VRPSD focuses on simple recourse policies that are separable by vehicle (Bertsimas (1992) and Bertsimas and Simchi-Levi (1994) describe many such policies); an exception is a two-vehicle sharing recourse policy proposed in Ak and Erera (2007).

Another stream of research on the VRPSD focuses on dynamic operating policies that make decisions each time new information is revealed, rather than relying on static recourse policies. Markov decision models are typically proposed; one drawback is that these models require a very large state space, and thus are intractable even for modestly-sized instances. Dror et al. (1989) proposes a single-vehicle model where a decision epoch corresponds to the moment the vehicle arrives at a customer location and its demand is revealed. At that point, two possible decisions can be made prior to serving the customer: (i) not to serve the customer and move to another location, or (ii) serve the customer and then move to another location. No solution approach or computational study is presented.

An interesting approach, based on Markov decision models, is presented in Yang et al. (2000). The authors attempt to specify an optimal restocking (or unloading) policy for the vehicle in conjunction with the routing decisions. Under such a policy, the vehicle might restock (or unload) at the depot before a capacity failure actually occurs. For a given tour, it is shown that the optimal restocking policy has a simple threshold form: after serving a customer if on-board inventory drops below a customer-dependent threshold, then restock, else continue to the next customer in the tour. Although the optimal policy is quite simple, solving a model that further considers routing decisions is difficult, so heuristic approaches are developed. In related work, Secomandi (Secomandi, 2001; 2003) extends this idea to a more general reoptimization framework for the VRPSD, where the sequence of remaining customer visits may be changed each time a vehicle serves a customer (and thus receives new information). A neuro-dynamic programming rollout approach is proposed for problems with a single vehicle, using the cyclic heuristic of Haimovich and Rinnooy Kan (1985).

3 The Vehicle Routing Problem with Stochastic Demand and Duration Constraints

The Vehicle Routing Problem with Stochastic Demands and Duration Constraints (VRPSD-DC) is defined on a directed graph $G = (V_0, A)$ where $V_0 = \{0\} \cup V$ with 0 representing the depot and $V = \{1, \cdots, N\}$ representing the set of customers, and $A = \{(i, j) \mid i \neq j, i, j \in V_0\}$. A cost $l(i, j)$ is associated with each arc $(i, j) \in A$; we assume this cost represents the travel time between customers $i$ and $j$, and that these travel times satisfy the triangle inequality, i.e., $l(i, j) \leq l(i, k) + l(k, j)$ for all $i, j, k \in V_0$. The vehicle capacity is denoted by $Q$ and we assume for exposition that vehicles carry inventory to be delivered to customers.

Customer demands are nonnegative integer-valued random variables with known distributions and are denoted by vector $\tilde{d} \in \mathbb{Z}_+^{|V|}$. Customer demands are assumed to be independently distributed. It is further assumed that support vectors $\underline{d}, \bar{d} \in \mathbb{Z}_+^{|V|}$ are known such
that $d \leq \bar{d} \leq \bar{d}$, and $\bar{d}(i) > 0$ and $\bar{d}(i) < Q$ for all $i \in V$. If we further assume that there is a positive probability that each demand $\bar{d}(i)$ takes every value in the range $[d(i), \bar{d}(i)]$, then the outcome space of demand realizations is

$$U = \{ d \in \mathbb{Z}_+^{|V|} : d \leq \bar{d} \}.$$ 

Finally, for convenience of notation, we further define $\bar{d}(0) = \bar{d}(0) = 0$. Uncertain customer demands are assumed to be revealed with certainty at some time immediately prior to or within each operating day. For example, a common assumption is that customer demand becomes known upon the arrival of a vehicle.

A tour specifies an a priori sequence for a single vehicle in which some subset of customers is visited; all tours start and end at the depot. Each customer is assumed to be served by one and only one tour. Let $T_k = \{ i_1, i_2, \cdots, i_n \}$ denote the sequence of customers to be visited by the tour of vehicle $k$, where $i_j \in V$.

Each day when a vehicle executes its tour, all customer demands must be satisfied, but this may not be possible if the vehicle were to simply serve each customer in its tour in sequence. Under this problem definition, we consider recourse policies under which each vehicle serves customer in its tour according to the a priori ordering, but returns to the depot to restock when necessary to ensure that all customer demands are satisfied. We will denote as recourse actions these restocking detours. It is assumed that recourse policy $P$ determines uniquely when and where recourse actions will occur in each tour for any demand realization.

Let $L(T_k)$ be the total travel time required by vehicle $k$ to complete its a priori tour if no recourse actions were required, and let $\phi(T_k, P, d)$ be the total additional travel time due to recourse actions given policy $P$ and demand realization $d \in U$. Then, the maximum duration of tour $k$ is

$$L(T_k, P) = L(T_k) + \max_{d \in U} \phi(T_k, P, d).$$

Since the additional travel time incurred due to recourse actions depends on the demand realization, determining the maximum additional travel time due to recourse actions over all possible realizations is an optimization problem. We will refer to it as the adversarial problem, and its purpose is to identify the worst-case demand realization in terms of tour duration. In the remainder of the paper, we use $\Phi(T_k, P)$ to denote $\max_{d \in U} \phi(T_k, P, d)$.

The expected duration of tour $k$ is

$$L_E(T_k, P) = L(T_k) + E[\phi(T_k, P, d)],$$

where $E$ denotes the expectation operator with respect to the customer demand uncertainty space $U$, and thus $E[\phi(T_k, P, d)]$ represents the expected additional travel time incurred by vehicle $k$ due to recourse actions under policy $P$. The VRPSD-DC is then to find a set of tours with minimum total expected duration subject to a hard constraint on individual tour duration. Imposing duration constraints may result in an increased number of a priori tours.
in a solution. Since systems are usually planned such that each tour is operated by a unique vehicle, and since adding additional vehicles results in additional costs, we penalize solutions that use more than the number of vehicles \( m \) used in an optimal solution to the problem without duration constraints:

\[
\text{VRPSD} - \text{DC} \quad \min_{\{T_k\}} \quad \sum_k \mathcal{L}_E(T_k, P) + Fz \\
\text{s.t.} \quad \mathcal{L}(T_k, P) \leq D \quad \forall k, \\
|\{T_k\}| \leq m + z, \\
z \in \mathbb{Z}_{\geq 0},
\]

where \( D \) denotes the maximum travel time duration allowed for a tour, and the customer set \( V \) is partitioned among the tour set \( \{T_k\} \). Decision variable \( z \) captures the excess number of vehicles, and \( F \) is a positive scalar.

The value \( m \) is assumed to be computed under the common assumption that each \textit{a priori} tour requires a unique vehicle. Note that for VRPSD problems that use detour-to-depot recourse policies, it should be clear that in the absence of any additional constraints (for example, on the duration of a tour) all solutions that include each customer on some tour are feasible, including all solutions that include each customer on a single \textit{a priori} tour. However, observe that such single tour solutions need not be optimal. In Figure 1, a simple example with two customers illustrates this idea. Both customer demand random variables \( \tilde{d}(i) \) take value 1 with probability 0.5 and value 2 with probability 0.5, and the vehicle capacity \( Q = 3 \). The single tour solution in the left panel therefore has an expected cost of 4.5, since regardless of which customer is visited first there is a 0.25 likelihood of a recourse action at the second customer. Since the vehicle must pass the depot en route to the second customer, a better solution uses two tours; no recourse is required and the expected cost is 4. Note that this solution could be interpreted differently if we do not assume explicitly that a unique vehicle operates each \textit{a priori} tour. If instead a single vehicle were to operate both tours in the example, then we have introduced a “pre-planned” detour-to-depot trip. Additionally, in the absence of constraints on tour duration, this should be feasible in practice since a pre-planned detour should be easier to accommodate than an unplanned recourse detour.

For simplicity in this paper, we choose not to allow pre-planned detours and assume that each tour in a solution is operated by a unique vehicle.

4 The Adversarial Problem

Consider a single vehicle and its \textit{a priori} tour \( T = \{1, 2, \cdots, n\} \). The adversarial problem seeks to determine the demand realization \( d \in U \) that maximizes \( \phi(T, P, d) \). If the function \( \phi \) is non-decreasing in \( d(i) \) for all \( i \in T \) for \( P \), it is clear that an optimal solution is to set \( d \) equal to \( \overline{d} \). As we will show, this is not the case for all recourse policies; that is, unlike
many robust optimization problems, the worst-case scenario for the adversarial problem is not always found at an extreme point of the box that defines the uncertainty space. Since the size of the uncertainty space for a given tour $T$ is given by

$$\prod_{i \in T} (d(i) - d(i) + 1),$$

simple enumerative approaches for solving the adversarial problem may result in an excessive computational burden that hopefully can be avoided.

In this paper, we focus on solving the adversarial problem for variants of the detour-to-depot recourse policies. In Morales (2006), it is shown how the adversarial problem can be solved for various other recourse policies.

### 4.1 Detour-to-depot recourse

Suppose that the demand of customer $i$ is revealed upon arrival of the vehicle to $i$. We now define two recourse policies, one which allows individual customer demands to be split and one which does not.

**Definition 1 (Splittable detour-to-depot recourse)** $P^S$ is used to denote the following recourse policy given tour $T$: a recourse action is initiated at customer $i \in T$ if and only if the arriving vehicle observes $d(i)$ which is strictly greater than on-board inventory. After satisfying as much of $d(i)$ as possible with on-board inventory, the vehicle restocks at the depot, then returns to $i$ and satisfies any remaining demand before proceeding.

**Definition 2 (Non-splittable detour-to-depot recourse)** $P^{NS}$ is used to denote the following recourse policy given tour $T$: a recourse action is initiated at customer $i \in T$ if and only if the arriving vehicle observes $d(i)$ which is strictly greater than the on-board inventory. Before satisfying any part of $d(i)$, the vehicle restocks at the depot, returns to $i$, and satisfies the entire demand.
Note that under each policy a vehicle never performs more than one recourse action at a single customer $i$, since $d(i) < Q$; furthermore, the first recourse action will never occur at the first customer in the tour. These two policies share the characteristic that for each $d \in U$, the customers at which recourse actions are initiated are uniquely determined, and that the number of recourse actions is a non-decreasing function of total tour demand $\sum_{i \in T} d(i)$. Additionally, if a recourse action occurs at customer $i$, then the onboard inventory when the vehicle departs after servicing $i$ is strictly greater than zero and strictly less than $Q$.

It has been frequently observed that the detour-to-depot recourse policies as defined have an undesirable property: if the onboard inventory after a delivery at customer $i - 1$ is zero, the vehicle travels to customer $i$ before recognizing the need for a recourse action. This property is easily remedied by extending the policies with an additional recourse action. For example, policy $\mathcal{P}^{NS}$ can be extended as follows:

**Definition 3** $\mathcal{P}^{NS-E}$ is used to denote the following non-splittable recourse policy given tour $T$ with two types of recourse actions:

Type I: a recourse action is initiated at customer $i \in T$ if and only if the arriving vehicle observes $d(i)$ strictly greater than on-board inventory, restocking at depot before serving any part of $d(i)$.

Type II: a recourse action is initiated at customer $i - 1 \in T$ for customer $i$: if after serving $i - 1$ the on-board inventory is zero, then the vehicle travels to the depot to restock before traveling on to customer $i$.

For simplicity, we introduce and discuss the adversarial problem for the basic versions of the detour-to-depot policy, i.e., for $\mathcal{P}^{S}$ and $\mathcal{P}^{NS}$. However, we use policy $\mathcal{P}^{NS-E}$ in our computational study and provide details regarding solving the adversarial problem for the policy in the appendix.

### 4.2 Solving the adversarial problem

Let $\mathcal{P}$ be either the splittable policy $\mathcal{P}^{S}$ or the non-splittable policy $\mathcal{P}^{NS}$. The execution of tour $T$ under policy $\mathcal{P}$ for demand realization $d$ can be described in terms of state variables $(r, i, I_i)$ defined for each customer $i$ in the tour, where $r = k$ if the $k$-th recourse action is initiated at customer $i$ and $r = 0$ if no recourse action is initiated at this customer, and where $I_i$ denotes the on-board inventory of the vehicle when it departs $i$. Note that state variable values are uniquely determined for each $d$.

It is easy to see that the possible set of all states $(r, i, I_i)$ for all demand realizations is compact. For the $i$-th customer in the tour sequence, $0 \leq r \leq \min(i, R)$ where $R \leq n$ denotes the maximum number of recourse actions that can occur, and $1 \leq I_i \leq Q$. Note that $R$ can be determined by executing the tour under $\mathcal{P}$ for demand realization $\bar{d}$.

To solve the adversarial problem, we first characterize the set $S$ of states for which there exists a demand realization $d \in U$ that visits that state. Given $d$ and $\bar{d}$, it is possible to determine the set of customers at which the first recourse action may be initiated, and for
each such customer a corresponding demand realization and the resulting vehicle load at departure.

Let $C^1(i, I_i)$ for $i \in \mathcal{T}$ denote a set of necessary and sufficient conditions that ensures the existence of a demand realization $d \in \mathcal{U}$ for which the first recourse action is initiated at customer $i$ and such that the vehicle load at the departure from customer $i$ is $I_i$. Furthermore, let $C^{r, r+1}(i, I_i, k, I_k)$ for $i, k \in \mathcal{T}$ such that $i < k$ and for $r = 1, \ldots, R - 1$ denote a set of necessary and sufficient conditions that ensures the existence of a demand realization $d \in \mathcal{U}$ for which the $(r + 1)$-th recourse action is initiated at customer $k$ and such that the vehicle load at departure from customer $k$ is $I_k$ given that the $r$-th recourse action is initiated at customer $i$ with vehicle load $I_i$ at departure. Conditions $C^1(i, I_i)$ and $C^{r, r+1}(i, I_i, k, I_k)$ are referred to as recourse conditions. We will show subsequently that such conditions can be developed for policies $\mathcal{P}_{S}$ and $\mathcal{P}_{NS}$.

Given recourse conditions, the adversarial problem can be solved by finding a longest path on an acyclic digraph $G(\mathcal{T}, \mathcal{P}) = (\mathcal{N}, \mathcal{A})$, with node set

$$\mathcal{N} = \{s\} \cup \{t\} \cup \{(r, i, I_i) | r \in \{1, \ldots, R\}, i \in \mathcal{T} \text{ s.t. } i > r, I_i \in \{1, 2, \ldots, Q - 1\}\}, \quad (2)$$

where $s$ and $t$ represent the source and sink node, respectively.

The arc set $\mathcal{A}$ is defined using the recourse conditions and has three groups of arcs:

$$\mathcal{A} = \mathcal{A}^s \cup \mathcal{A}^{r, r+1} \cup \mathcal{A}^t.$$

The first group of arcs $\mathcal{A}^s$ captures the potential first failures and is defined as

$$\mathcal{A}^s = \{(s, (1, i, I_i)) | (1, i, I_i) \in \mathcal{N} \text{ and } i, I_i \text{ satisfy } C^1(i, I_i)\}.$$

The cost of such arcs is the additional travel time of a recourse action at $i$: $l(i, 0) + l(0, i)$.

The second group of arcs $\mathcal{A}^{r, r+1}$ captures potential subsequent failures and is defined as

$$\mathcal{A}^{r, r+1} = \{((r, i, I_i), (r + 1, k, I_k)) | (r, i, I_i) \in \mathcal{N}, (r + 1, k, I_k) \in \mathcal{N},$$

and $i, k, I_i, I_k$ satisfy $C^{r, r+1}(i, I_i, k, I_k)\}.$

Each such arc corresponds to an $(r + 1)$-th failure at $k$ given an $r$-th failure at $i$ with the corresponding inventory levels, with cost equal to the additional travel time required by recourse: $l(k, 0) + l(0, k)$. The final group of arcs $\mathcal{A}^t$ captures tour completion without additional failures and is defined as

$$\mathcal{A}^t = \{((r, i, I_i), t) | (r, i, I_i) \in \mathcal{N} \text{ s.t. } \text{indeg}((r, i, I_i)) > 0 \text{ and } \text{outdeg}((r, i, I_i)) = 0$$

on $(\mathcal{N}, \mathcal{A}^s \cup \mathcal{A}^{r, r+1})\}$

where $\text{indeg}(v)$ and $\text{outdeg}(v)$ denote the indegree and outdegree, respectively, of node $v$.

The cost of such arcs is zero.

Let $L(G(\mathcal{T}, \mathcal{P}))$ be the length of the longest $s-t$ path in the graph. If there does not exist an $s-t$ path, then let $L(G(\mathcal{T}, \mathcal{P})) = 0$. 

9
Lemma 1 For a tour $T$ operated using recourse policy $\mathcal{P}$, $\Phi(T, \mathcal{P}) = L(G(T, \mathcal{P}))$.

Proof. For any demand realization $d$, there is a unique set of customers in $T$ where recourse actions occur. From the construction of network $G(T, \mathcal{P})$, it is clear that every $d \in \mathcal{U}$ is associated with an $s-t$ path in $G(T, \mathcal{P})$, and also, that every $s-t$ path in the network is associated with a demand realization. To evaluate $\Phi(T, \mathcal{P})$, the longest path in the network is identified because this path is associated with a demand realization with maximum additional travel time.

Lemma 2 $L(G(T, \mathcal{P}))$ can be calculated in $O(n^3Q^2)$.

Proof. The number of nodes in the $r$-th layer of $G(T, \mathcal{P})$, represented by $(r, i, I)$ for $i \in T$ and $I \in \{1, \ldots, Q-1\}$, is bounded by $nQ$. $R$ is bounded by $n$, and therefore $|A^{r,r+1}|$ is bounded by $n^3Q^2$. Also, it is clear that $|A^r|$ is bounded by $nQ$ and $|A^r|$ bounded by $n^2Q$. This implies that $O(|A|) = O(n^3Q^2)$, and the result follows since $G(T, \mathcal{P})$ is an acyclic digraph.

4.2.1 Solving the adversarial problem for policy $\mathcal{P}^S$

We now develop recourse conditions for the splittable policy $\mathcal{P}^S$. First, consider a simple and useful lemma presented without proof.

Lemma 3 Consider vectors $\bar{d}, \bar{d} \in \mathbb{Z}_+^n$ such that $\bar{d} \leq \bar{d}$ and $\bar{d} > 0$. If

$$\sum_{\ell=1}^{n-1} d(\ell) \leq M \quad \text{and} \quad \sum_{\ell=1}^{n} \bar{d}(\ell) \geq M + 1,$$

where $M$ is a finite integer, then there exists a vector $d \in \mathbb{Z}_+^n$ such that $\underline{d} \leq d \leq \bar{d}$ that satisfies

$$\sum_{\ell=1}^{n-1} d(\ell) \leq M \quad \text{and} \quad \sum_{\ell=1}^{n} d(\ell) \geq M + 1.$$

Now, recourse conditions can be developed.

Proposition 1 (Recourse condition $C^{r+1}(i, I_i, k, I_k)$ for $\mathcal{P}^S$) Consider $d \in \mathcal{U}$ such that the $r$-th recourse action occurs at customer $i$, and the vehicle departs $i$ with inventory $I_i$. Then, the $(r+1)$-th recourse can occur at customer $k > i$ with corresponding $I_k$ if and only if

$$\sum_{\ell=i+1}^{k-1} d(\ell) \leq I_i \quad \text{and} \quad \sum_{\ell=i+1}^{k} \bar{d}(\ell) \geq I_i + 1;$$
and furthermore, the possible remaining inventory \( I_k \) is bounded by:

\[
Q + I_i - \min \left\{ I_i, \sum_{\ell=i+1}^{k-1} \bar{d}(\ell) \right\} - \bar{d}(k) \leq I_k \leq Q + I_i - \sum_{\ell=i+1}^{k} d(\ell).
\]

**Proof.**

(\( \Rightarrow \)) Let \( d \) be a realization where the \( r \)-th recourse occurs at \( i \) with remaining inventory \( I_i \) and the \((r+1)\)-th occurs at \( k \) with \( I_k \). Such a \( d \) must satisfy

\[
\sum_{\ell=i+1}^{k-1} d(\ell) \leq I_i \quad \text{and} \quad \sum_{\ell=i+1}^{k} d(\ell) \geq I_i + 1;
\]

with corresponding

\[
I_k = I_i - \sum_{\ell=i+1}^{k} d(\ell) + Q.
\]

Since \( d \in \mathcal{U} \), this implies that

\[
\sum_{\ell=i+1}^{k-1} d(\ell) \leq \sum_{\ell=i+1}^{k-1} d(\ell) \leq I_i \quad \text{and} \quad \sum_{\ell=i+1}^{k} \bar{d}(\ell) \geq \sum_{\ell=i+1}^{k} d(\ell) \geq I_i + 1;
\]

and that

\[
Q + I_i - \sum_{\ell=i+1}^{k} d(\ell) \leq I_k \leq Q + I_i - \sum_{\ell=i+1}^{k} d(\ell).
\]

Since no recourse occurs between \( i \) and \( k \), \( \sum_{\ell=i+1}^{k-1} d(\ell) \leq I_i \) and thus

\[
Q + I_i - \min \left\{ I_i, \sum_{\ell=i+1}^{k-1} \bar{d}(\ell) \right\} - \bar{d}(k) \leq I_k \leq Q + I_i - \sum_{\ell=i+1}^{k} d(\ell).
\]

(\( \Leftarrow \)) The existence of a demand realization in \( \mathcal{U} \) such that the \((r+1)\)-th recourse takes place for \( k \) given that the \( r \)-th takes place for \( i \) with on-board inventory \( I_i \) follows directly from Lemma 3; \( I_k \) just needs to be bounded accordingly.

When the vehicle departs from the depot for the first time, the system conditions are equivalent to those that would occur had a recourse occurred at “customer” 0 (i.e., the depot) with \( I_0 = Q \); the following result then follows directly from Proposition 1.

**Proposition 2 (Recourse condition \( C^1(i, I_i) \) for \( \mathcal{P}^S \))** Recourse conditions \( C^1(i, I_i) \) are equivalent to \( C^{0,1}(0, Q, i, I_i) \).

Propositions 1 and 2 clearly enable the construction of \( \mathcal{G} \) recursively in pseudopolynomial time, beginning with customers at which the first recourse may occur.
4.2.2 Solving the adversarial problem for policy $\mathcal{P}^{NS}$

It is not difficult to extend the ideas from the previous section to develop a pseudopolynomial approach to solve the adversarial problem for tour $T$ operated under recourse policy $\mathcal{P}^{NS}$. However, we can take advantage of an additional feature of this policy to develop a polynomial approach. Consider the following useful observation.

**Observation 1** Consider a tour $T$ operated using policy $\mathcal{P}^{NS}$, and a demand realization $d$ such that a recourse action occurs at customer $i \in T$. Then, after $i$ is served the on-board inventory is always $Q - d(i)$.

This observation suggests that we can derive simpler recourse conditions in this case since, given a recourse action at a customer, onboard inventory can be characterized with only the corresponding demand range and vehicle capacity. The following lemma provides additional insight.

**Lemma 4** Consider a tour $T$ operated using policy $\mathcal{P}^{NS}$ and a demand realization $d$ where the $(r - 1)$-th recourse action occurs at $j$. If the $r$-th recourse action occurs at $i > j$, then

$$d(i) \geq d(i/j) \equiv \max \left\{ 1, Q + 1 - \sum_{\ell = j}^{i-1} d(\ell) \right\}.$$ 

Proof. Under $\mathcal{P}^{NS}$, the vehicle satisfies the demand of customer $j$ before proceeding. A demand realization $d$ that next initiates recourse at $i$ must satisfy

$$\sum_{\ell = j}^{i-1} d(\ell) \leq Q, \text{ and} \quad (3)$$

$$\sum_{\ell = j}^{i} d(\ell) \geq Q + 1. \quad (4)$$

Since $d \in \mathcal{U}$, we have $d(\ell) \leq \bar{d}(\ell)$ and summing over $\ell$ gives

$$\sum_{\ell = j}^{i-1} d(\ell) \leq \sum_{\ell = j}^{i-1} \bar{d}(\ell).$$

The above expression together with (3) implies

$$\sum_{\ell = j}^{i-1} d(\ell) \leq \min \left\{ Q, \sum_{\ell = j}^{i-1} \bar{d}(\ell) \right\}.$$
From (4), we have

\[ d(i) \geq Q + 1 - \sum_{\ell=j}^{i-1} d(\ell) \geq Q + 1 - \min \{ Q, \sum_{\ell=j}^{i-1} \bar{d}(\ell) \} \geq \max \{ 1, Q + 1 - \sum_{\ell=j}^{i-1} \bar{d}(\ell) \}. \]

Observe that \( d(i/j) \) is a lower bound on the demand of customer \( i \) if a recourse action is to occur at customer \( i \) given that the preceding recourse action occurred at customer \( j \). Thus, in order to determine at which customer the \( (r+1) \)-th recourse action may occur, we only need to know where the \( (r-1) \)-th and \( r \)-th recourse actions occurred; we need not explicitly track vehicle inventory levels.

We can now develop a somewhat simpler set of recourse conditions. To begin, we show that the first recourse action \( (r = 1) \) occurs at node \( i \) via conditions identical to those for the splittable policy. We present the following result without a proof because of its similarity to the proof of Proposition 1:

**Proposition 3 (Recourse condition \( C^1(i) \) for \( P^{NS} \))** There exists a demand realization \( d \in U \) such that the first recourse occurs at \( i \) if and only if

\[ \sum_{\ell=1}^{i-1} d(\ell) \leq Q \quad \text{and} \quad \sum_{\ell=1}^{i} \bar{d}(\ell) \geq Q + 1. \]

Next, we develop conditions that characterize where the \( (r+1) \)-th recourse action may occur, given the locations of the \( (r-1) \)-th and \( r \)-th recourse actions:

**Proposition 4 (Recourse condition \( C^{r-1,r+1}(j,i,k) \) for \( P^{NS} \))** Assume there exists a demand realization \( d \in U \) such that the \( (r-1) \)-th recourse occurs at \( j \) and the \( r \)-th recourse occurs at \( i > j \) for \( r > 0 \). The \( (r+1) \)-th recourse can occur at \( k > i \) if and only if

\[ \max \{ d(i/j), d(i) \} + \sum_{\ell=i+1}^{k-1} d(\ell) \leq Q \quad \text{and} \quad \sum_{\ell=i}^{k} \bar{d}(\ell) \geq Q + 1. \]

**Proof.**

\( \Rightarrow \) Since a recourse action occurs at \( i \), by Observation 1 the on-board inventory after \( i \) is served is \( Q - d(i) \). Hence, such a demand realization \( d \) must satisfy

\[ \sum_{\ell=i}^{k-1} d(\ell) \leq Q \quad \text{and} \quad \sum_{\ell=i}^{k} \bar{d}(\ell) \geq Q + 1. \]
Since \( d \in \mathcal{U} \), we have \( Q + 1 \leq \sum_{\ell=1}^{k} d(\ell) \leq \sum_{\ell=1}^{k} \overline{d}(\ell) \), which corresponds to the second inequality in the proposition. Also, since \( \overline{d}(\ell) \leq d(\ell) \) and, by Lemma 4, \( d(\ell) \leq d(i) \), we have \( \max \{d(i), d(j)\} \geq \sum_{\ell} \overline{d}(\ell) \geq \sum_{\ell} d(\ell) \leq Q \).

(\( \Rightarrow \)) Given that the \( r \)-th recourse occurs at \( i \) then the existence of the \((r+1)\)-th recourse at \( k \) depends only on the values of \( d(i), \cdots, d(k) \); we now identify the corresponding bounds of a feasible demand realization. Consider vector \( d' \in \mathbb{Z}_{+}^{[k-i+1]} \) and let \( d'(\ell) = d(i + \ell - 1) \) for \( \ell = 1, \cdots, k-i+1 \). Consider vectors \( d', \overline{d} \in \mathbb{Z}_{+}^{[k-i+1]} \) and let \( d'(1) = \max \{d(i), d(i)\} \), \( \overline{d}(\ell) = d(i + \ell - 1) \) for \( \ell = 2, \cdots, k-i+1 \) and \( \overline{d}(\ell) = \overline{d}(i + \ell - 1) \) for \( \ell = 1, \cdots, k-i+1 \). The result then follows from Lemma 3 and Lemma 4; for the former, let \( M = Q \), \( n = k-i+1 \), \( \overline{d} = d' \) and \( \overline{d} = \overline{d}'. \)

Observe that Proposition 4 can be used with \( r = 1 \) to determine the customers where the second recourse may occur given that the first recourse occurred at \( i \) by setting \( j = 0 \), since we assume that \( \overline{d}(0) = d(0) = 0 \).

We now show that Propositions 3 and 4 allow us to define a new network for solving the adversarial problem for policy \( \mathcal{P}^{NS} \). The adversarial problem for \( \mathcal{P}^{NS} \) can be solved using a digraph where node \((r,i/j)\) represents that recourse action \( r \) occurs at customer \( i \) preceded by recourse \((r-1)\) at customer \( j \). An arc between two nodes represents the existence of a demand realization that results in consecutive recourse actions at the corresponding customers. Let \( G_{r}(T, \mathcal{P}^{NS}) = (N, A) \) denote this network. The set of nodes is

\[
N = \{s\} \cup \{t\} \cup \{(1,i/0) \mid i \in T \setminus \{1\}\} \cup N',
\]

where \((1,i/0)\) indicates that the first recourse action occurs at customer \( i \), and

\[
N' = \{(r,i/j) \mid r \in \{2,\cdots,R\}, i \in \{r+1,\cdots,n\}, j \in \{r,\cdots,i-1\}\},
\]

where \((r,i/j)\) is as defined earlier. Again, the arc set \( A \) is defined in terms of recourse conditions and consists of three groups of arcs:

\[
A = A^t \cup A^{r+1} \cup A^t.
\]

The first group of arcs \( A^t \) captures the potential first failures and is defined as

\[
A^t = \{(s, (1,i/0)) \mid (1,i/0) \in N \text{ and } i \text{ satisfies } C^1(i)\}.
\]

The cost of such arcs is the additional travel time of a recourse action at \( i \): \( l(i,0) + l(0,i) \). The second group of arcs \( A^{r+1} \) captures potential subsequent failures and is defined as

\[
A^{r+1} = \{((r,i/j), (r+1,k/i)) \mid (r,i/j) \in N, (r+1,k/i) \in N, \text{ and } i,j,k \text{ satisfy } C^{r-1,r+1}(j,i,k)\}.
\]

More precisely, an arc corresponds to an \((r+1)\)-th failure at \( k \) given an \( r \)-th failure at \( i \) and an \((r-1)\)-th failure at \( j \). The cost of such arcs is the additional travel time of the
The final group of arcs $A^t$ captures tour completion without additional failures and is defined as

$$A^t = \{((r, i/j), t) \mid (r, i/j) \in \mathcal{N} \text{ s.t } \text{indeg}((r, i/j)) > 0 \text{ and outdeg}((r, i/j)) = 0 \text{ on } (\mathcal{N}, \mathcal{A}^s \cup \mathcal{A}^{r+1})\}.$$ 

The cost of such arcs is zero.

Let $L(G_1(T, P^{NS}))$ be the length of the longest $s-t$ path in the graph. If there does not exist an $s-t$ path, then by definition $L(G_1(T, P^{NS})) = 0$.

**Theorem 1** For a tour $T$ operated using policy $P^{NS}$, $\Phi(T, P^{NS}) = L(G_1(T, P^{NS}))$. Furthermore, $L(G_1(T, P^{NS}))$ can be computed in $O(n^4)$ time by solving a longest $s-t$ path problem on $G_1(T, P^{NS})$.

**Proof.** By construction $\Phi(T, P^{NS})$ is equal to the length of the longest $s-t$ path in $G_1(T, P^{NS})$. Observe that the number of nodes in the $r$-th layer of the graph for $r > 1$ is bounded by $\sum_{i=1}^{n-r} i = \frac{(n-r)(n-r+1)}{2}$; each node $(r, i/j)$ is connected to at most $n - i$ nodes in layer $r + 1$. Therefore, the number of arcs between layer $r$ and layer $r + 1$ is $O(n^3)$. The number of layers $R$ is bounded by $n$, so the total number of arcs in $A^{r+1}$ is $O(n^4)$. The number of arcs in $A^s$ is bounded by $n$. The number of nodes in the graph is bounded by $n^3$, so the number of arcs in $A^t$ is also bounded by $n^3$. Therefore, the total number of arcs in the graph is $O(n^4)$ and the result follows because the digraph is acyclic.

**Example 1:** Consider the instance in Figure 2 where a vehicle with capacity $Q = 4$ operates tour $T = \{1, 2, \cdots, 7\}$. Table 1 shows the values of the state variables (as defined in Section 4.2) for demand realization $\overrightarrow{d}$ for $T$ operated using recourse policy $P^S$ and $P^{NS}$, respectively.
Table 1: Values of the state variables for demand realization $\bar{d}$ when the tour is operated using recourse policy $\mathcal{P}^S$ and $\mathcal{P}^{NS}$, respectively.

<table>
<thead>
<tr>
<th>$i$</th>
<th>$d(i)$</th>
<th>$r$</th>
<th>$I_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Observe that $\phi(T, \mathcal{P}^S, \bar{d}) = 12$, $\phi(T, \mathcal{P}^{NS}, \bar{d}) = 10$ and that $R = 3$ in both cases. For policy $\mathcal{P}^S$, Figure 3 shows the adversarial network $G(T, \mathcal{P}^S)$. Additionally, the thicker arcs in the figure identify a set of alternative longest $s-t$ paths in the network. Each demand realization that maximizes $\Phi(T, \mathcal{P}^S)$ triggers return trips to the depot at customer 3, 5, and 7 with corresponding cost 20. For policy $\mathcal{P}^{NS}$, Figure 4 shows $G_1(T, \mathcal{P}^{NS})$ as well as a longest path. Again, recourse actions in a worst-case realization are triggered at customers 3, 5 and 7, with an associated cost of 20.

Before discussing our tabu search algorithm for solving instances of VRPSD-DC, we present another example to illustrate some interesting characteristics of the VRPSD-DC.

**Example 2:** Consider the instance shown in Figure 5 where a vehicle with capacity $Q = 6$ operates tour $T = \{1, 2, \cdots, 6\}$. For recourse policy $\mathcal{P}^{NS}$, the maximum additional travel time due to recourse actions, (i.e., the value of the solution to the adversarial problem) is 16 with recourse actions at customer 3 and 5, which results in a total duration of 31 (shown on the left in Figure 6). Interestingly, if we consider the tour without customer 3, i.e., $T = \{1, 2, 4, 5, 6\}$, then the maximum additional travel time due to recourse actions is 18 with recourse actions at customer 2 and 5, for a total duration of 33 (shown on the right in Figure 6). Thus, removing a customer from a tour does not necessarily lead to a reduction in maximum total duration; in fact, it may lead to an increase. In the example, the adversary has better options once customer 3 was removed from the tour; tour failures at customer 2 and customer 5 are now possible. Similar results can be obtained under $\mathcal{P}^S$ by changing the demand interval for customer 4 from $[2,3]$ to $[4,5]$. This example of course also implies that adding a customer to a tour may decrease its maximum total duration.

It is also interesting to observe that the maximum total duration is sensitive to the order in which a tour is traversed. Under $\mathcal{P}^{NS}$, for example, when the tour in the example is traversed in reverse order, i.e., $T = \{6, 5, 4, 3, 2, 1\}$, then the maximum additional travel time due to recourse actions is 8 (down from 18) with recourse actions at customers 2 and 5.

The above properties are not due specifically to the duration constraints, but are the result of the stochastic demands. The locations of any recourse actions determine the expected cost of a tour and these locations are affected by the set of customers in the tour as well as the orientation of the tour.
5 A Tabu Search Heuristic

We have designed and implemented a tabu search heuristic for solving instances of the VRPSD-DC under the detour-to-depot recourse policy $\mathcal{P}^{NS-E}$. We focus on this policy, since non-splittable policies are the most commonly studied in the VRPSD literature. Tabu search is a powerful metaheuristic technique successfully used in many contexts to solve hard optimization problems. Briefly, at each iteration such methods explore a neighborhood of the current solution, always moving to the best option available even if this implies a degradation on the value of the objective function. Cycling is avoided by maintaining a set of unacceptable (tabu) solutions, and updating this set each iteration. See Glover and Laguna (1997) for a complete survey on this topic.
Figure 4: $G_1 (T, P^{NS})$ for the instance in Example 1.

The tabu search heuristic we propose is similar to the one discussed in Gendreau et al. (1996b), which has been shown to perform well for vehicle routing problems with stochastic demands with a total expected cost objective. Below we briefly discuss the main components of our heuristic.

**Initial solution.** In the initial solution, we simply place each customer in its own tour.

**Neighborhood structure.** We use a slightly expanded version of the neighborhood $N(p, q, x)$ described in Gendreau et al. (1996b), which we denote by $N_r(p, q, x)$. The neighborhood includes for a given solution, all solutions that are obtained by removing, in turn, one of $q$ randomly selected customers (denoted by $q'$), and reinserting it either immediately after or immediately before each of its $p$ nearest neighbors (denoted by $p'$). When both customers $q'$ and $p'$ are in the same tour and $q'$ precedes $p'$, we consider two alternative ways of inserting
$q'$ before $p'$. For tour

$$
\mathcal{T}_k = \{i_1, i_2, \ldots, q', i_\ell, i_{\ell+1}, \ldots, i_{\ell+\beta}, p', \ldots, i_n\}
$$

we consider both

$$
\mathcal{T}_k = \{i_1, i_2, \ldots, i_\ell, i_{\ell+1}, \ldots, i_{\ell+\beta}, q', p', \ldots, i_n\}
$$

and

$$
\mathcal{T}_k = \{i_1, i_2, \ldots, i_{\ell+\beta}, \ldots, i_{\ell+1}, i_\ell, q', p', \ldots, i_n\},
$$

and randomly select one of them with equal probability. Each generated neighbor is checked for duration-feasibility by solving an appropriately defined adversarial problem. As illustrated in Example 2, when $q'$ and $p'$ are in different tours, it is important to check duration feasibility for both tours, because both removing a customer from and inserting a customer into a tour can increase the maximum total duration. Furthermore, since the order in which a tour is traversed matters (as was also illustrated in Example 2), we evaluate a tour by traversing it in both directions.

**Tabu moves.** If a customer is moved in iteration $\nu$, then any move involving that customer is made tabu until iteration $\nu + \theta$, where $\theta$ is randomly selected in interval $[N - 5, N]$.

**Aspiration criteria.** The search process moves from one iteration to the next considering only non-tabu solutions in the neighborhood of the current solution, unless a tabu solution improves the best solution found thus far.

**Move evaluation.** A move is considered to be improving if the total cost of the resulting set of tours is less than the total cost of the current set of tours.

The total cost $C(x)$ of the set of tours $x$ is given by the objective function of the VRPSD-DC, (1). The penalty term $F_z$ is included if $z > m$, and is added to the total expected duration of all tours. The expected duration of tour $\mathcal{T} = \{1, 2, \ldots, n\}$ is computed for $\mathcal{P}^{NS}$
as follows. Let \( p_i(\delta) \) be the probability that the \( i \)-th customer in the tour has a demand value equal to \( \delta \). Let \( \beta(i, s, q) \) be the probability of having on-board inventory equal to \( q \) after serving the \( i \)-th customer, given recourse states \( s \), where \( s = 1 \) if a recourse action occurs at \( i \) and \( s = 0 \) if not. Furthermore, for ease of notation, define \( \mathcal{Q}(j, k) = \{j, j+1, \cdots, k-1, k\} \), for \( j \) and \( k \) integers such that \( 0 \leq j \leq k \leq Q \).

We can now recursively compute the probabilities \( \beta \). First, for all \( q \in \mathcal{Q}(0, Q) \):

\[
\beta(1, s, q) = \begin{cases} 
    p_1(Q - q) & \text{if } s = 0 \\
    0 & \text{if } s = 1 
\end{cases}
\]

For \( i \geq 2 \) we calculate \( \beta(i, s, q) \) by conditioning on \( s \) and \( q \) for customer \( i - 1 \). For \( s = 0 \), to calculate \( \beta(i, 0, q) \) assume an on-board inventory value of \( \bar{q} \) after serving customer \( i - 1 \), which implies a demand realization of \( \bar{q} - q \geq 0 \) at customer \( i \). Therefore,

\[
\beta(i, 0, q) = \sum_{s \in \{0,1\}} \sum_{\bar{q} \in \mathcal{Q}(q, Q)} \beta(i - 1, s, \bar{q}) \cdot p_i(\bar{q} - q) \quad \text{for} \quad i \in \{2, \cdots, n\}, q \in \mathcal{Q}(0, Q).
\]

To calculate \( \beta(i, s, q) \) for \( s = 1 \), again assume on-board inventory \( \bar{q} \) after serving customer
i − 1. By Observation 1, the demand at customer i is $Q - q > \bar{q}$. Therefore,

$$\beta(i, 1, q) = \sum_{s \in \{0, 1\}} \sum_{\bar{q} \in Q(0, Q - q - 1)} \beta(i - 1, s, \bar{q}) p_i(Q - q)$$

for $i \in \{2, \cdots, n\}$, $q \in Q(1, Q - 1)$, and $\beta(i, 1, Q) = 0$ since no recourse is possible at i if the demand observed at i is zero.

Now let $\pi_i$ be the probability of a tour failure and recourse action at customer i. Then

$$\pi_i = \sum_{q \in Q(0, Q)} \beta(i, 1, q)$$

and the expected additional duration due to taking recourse actions can be calculated as

$$E[\phi(T, P^{NS}, d)] = \sum_{i \in T} \pi_i (l(i, 0) + l(0, i))$$

Note that the total cost $C(w)$ of the set of tours for a neighbor $w \in N_v(p, q, x)$ can be determined by recalculating only the expected cost of the one or two tours that differ from those in x, and by adjusting the penalty term $Fz$ as necessary. Unlike the approach in Gendreau et al. (1996b), we compute the exact expected cost of the complete solution for each feasible potential move.

**Tabu Search Heuristic**

**STEP 1:** *(Initialization)*

Construct solution x with each customer in its own tour; let $C^*$ be the cost of x and let $x^* = x$. Set $p = \min\{N - 1, 5\}$ and $q = \min\{N, 5\}$. Set $t_0 = 0$, $t_1 = 0$ and $t_2 = 50N$, where $t_0$ is the iteration counter, $t_1$ is the number of iterations in which the best solution found thus far has not improved, and $t_2$ is the maximum number of iterations allowed without one of such improvements.

**STEP 2:** *(Neighborhood search)*

Set $t_0 = t_0 + 1$. Consider all moves in $N_v(p, q, x)$ and build a LIST where all moves are sorted in nondecreasing order of their cost. Let y be the first move in the list. If it is not a tabu move or if $C(y) < C^*$, then let $x = y$; else examine moves in sequence from the LIST until such a move is found, and then make the corresponding assignment to x. If no such move is found, x remains unchanged.

**STEP 3:** *(Incumbent update)*

Let C(x) represent the cost of solution x. If $C(x) < C^*$, set $C^* = C(x)$, $x^* = x$ and $t_1 = 0$. Else, set $t_1 = t_1 + 1$.

If $t_1 < t_2$, go to STEP 2; otherwise go to STEP 4.

**STEP 4:** *(Intensification or termination)*

If $t_2 = 50N$, set $t_1 = 0$, $t_2 = 20N$, $p = \min\{N - 1, 10\}$, $q = N$, $x = x^*$ and go to STEP 2. Otherwise; stop, $x^*$ is the best solution found.

Except for the value of $q$ all parameters are set as in Gendreau et al. (1996b). We chose to initialize $q = \min\{N - 1, 5\}$ as opposed to $q = \min\{N - 1, 5m\}$, where $m$ is the
number of vehicles. These parameter settings yield good results. Furthermore, our goal in this paper is not to design the best possible tabu search algorithm for the VRPSD-DC, but rather to indicate how to handle duration constraints and to illustrate the value of explicitly considering such constraints.

6 Computational Results

In this section, we present a computational study of the VRPSD-DC when tours are operated under recourse policy $P^{NS-E}$. The objective of this study is to assess the effect of including duration constraints in the vehicle routing problem with stochastic demands. We are primarily interested in understanding how routing solutions change when tour duration constraints are enforced. In this study, we focus on problems where each customer in an instance shares the same demand probability distribution.

For a given instance, our starting point, which we refer to as the unconstrained version, is the VRPSD-DC where duration constraints are ignored. The instances are created such that the number of vehicles required when there are no duration constraints is likely to be $\bar{m}$. The actual number $m$ of vehicles required in the solution of the unconstrained version, obtained via our tabu search algorithm, then defines the target number of vehicles for the constrained versions. The maximum duration $D_{\text{max}}$ of the tour in the solution to the unconstrained version with the largest maximum duration is used to set the duration limit $D$ for the constrained versions; $D_{\text{max}}$ is multiplied by a reduction factor $0 < \alpha < 1$ to obtain values of $D$. By setting the penalty value $F$ to a high value, the use of more than $m$ vehicles is discouraged, and thus the fleet size increases only when it is no longer feasible to satisfy the tour duration constraints with $m$ vehicle tours.

In our computational study four factors were varied:

1. **Number of customers**: three problem sizes were considered: $N = 20, 60, \text{ and } 100$ customers.

2. **Demand variability ($\sigma$)**: customer demand is assumed to be discrete uniform, independent, and identically distributed for all customers; hence the demand of any customer $i$ takes values over the same interval $[d(i), \bar{d}(i)]$. We consider three levels of variability, referred to as low, medium, and high:

<table>
<thead>
<tr>
<th></th>
<th>low</th>
<th>medium</th>
<th>high</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d(i)$</td>
<td>4</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>$\bar{d}(i)$</td>
<td>8</td>
<td>9</td>
<td>10</td>
</tr>
</tbody>
</table>

Observe that the expected demand is constant across all levels of demand variability.
3. **Vehicle capacity** ($Q$): two levels of vehicle capacity were considered, $Q_1$ and $Q_2$, determined using expression

$$Q = \frac{N}{\bar{m}} \left( \frac{d(i)}{\bar{m}} + \overline{d(i)} \right).$$

Given $v$, this expression sets $Q$ such that the average demand should be feasibly served by $v$ deterministic tours not constrained by duration. Given values of $N$, $\overline{d(i)}$ and $\overline{d(i)}$, $Q_1$ is determined by this expression by setting $\bar{m} = 6$, and $Q_2$ by setting $\bar{m} = 3$. Observe that $Q_1 < Q_2$ and that in an optimal solution the vehicles are likely to be fully utilized and are likely to perform one recourse.

4. **Reduction factor** ($\alpha$): three levels are considered: 0.75, 0.85, and 0.95.

We use Solomon’s uniform random instance R101 for the Vehicle Routing Problem with Time Windows (see Solomon (1987)) to provide geographical locations of the customers and the depot for all of our test instances. In the usual way, we generate $N$ customer locations by selecting the first $N$ locations (out of the 100) in R101. Travel times between points are determined using the Euclidean distance function. All combinations of $N$, $\sigma$, $Q$, and $\alpha$ were considered. Thus, a total of 72 different instances are solved (18 unconstrained and 54 constrained instances). Each of these 72 instances was solved using the tabu search heuristic described in Section 5. For each instance, the heuristic was run ten times (each time using a different seed for the random number generator) and the solution with the minimum total expected travel time among the ten generated solutions was returned and used in the subsequent analysis.

Before providing the complete set of results, we present some specific examples to build up intuition. Consider an instance with $N = 20$, $Q = 40$, $d(i) = 3$ and $\overline{d(i)} = 9$. Table 2 summarizes the best solution found by the tabu search algorithm for the unconstrained version of this instance, which has a total expected travel time of 345.84. The solution uses three vehicle tours. The maximum durations of the tours result for realizations that require recourse actions at customers 20, 8, and 17; one in each of the three tours. (Note that because the maximum demand realization at a customer is 9 and the vehicle capacity is 40, there cannot be a recourse at any of the first four customers in a tour.) The maximum duration of any tour is $D_{\text{max}} = 173.14$, which implies that the constrained version with reduction factor
\( \alpha = 0.95 \) will have duration limit \( D = 164.4 \). Observe that two out of the three tours in this solution would therefore exceed this duration limit.

Table 3 summarizes the solution of the constrained version of this instance when \( \alpha = 0.95 \), and Figure 7 depicts the solutions to both the unconstrained and constrained versions graphically.

Table 3: Metrics for the best solution found for the constrained version (with a total expected travel time of 352.32)

<table>
<thead>
<tr>
<th>Tour</th>
<th>{1,20,9,3,12,4}</th>
<th>{10,11,19,7,8,18,6}</th>
<th>{5,17,16,14,15,2,13}</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixed Duration ( L )</td>
<td>109.90</td>
<td>99.93</td>
<td>102.40</td>
</tr>
<tr>
<td>Expected Duration ( L_E )</td>
<td>118.73</td>
<td>115.03</td>
<td>118.56</td>
</tr>
<tr>
<td>Max Duration ( L )</td>
<td>159.90</td>
<td>152.43</td>
<td>163.23</td>
</tr>
</tbody>
</table>

Figure 7: Best unconstrained solution and best constrained solution (\( \alpha = 0.95 \)).

Observe that by reversing the direction of tour \( \{4,12,3,9,20,1\} \), the maximum duration (now the result of a recourse action at customer 4) drops to 159.90 at the expense of a slight increase in total expected duration. Reversing the order of tour \( \{2,15,14,16,17,5,6,13\} \) and dropping customer 6 leads to a reduction in both the maximum and expected durations. Finally, moving customer 6 after customer 18 in tour \( \{10,11,19,7,8,18\} \) leads to an increase in both the maximum and expected durations, but the increase in maximum duration is small.
Figure 8: The impact of imposing a duration constraint on one of the tours for an instance with 100 customers

Now consider another example instance with $N = 100$, $Q = 100$, $d(i) = 2$ and $\overline{d}(i) = 10$. Figure 8 depicts a tour (on the left) constructed for the unconstrained version of this instance, and a modification of that tour (on the right) that results from solving a constrained version with $D = 269$. Note that the constrained tour visits the same customer set, but alters the visit sequence. For the tour produced by the unconstrained version, the maximum duration results when recourse actions occur at customers 16 and 91. Observe that the adversary takes advantage of the structure of the tour. Given the customer demand distribution, customer 16 is the first opportunity for a recourse action; if the adversary were to wait, the cost of the first recourse would be smaller. Given a recourse action at customer 16, another recourse action is possible at 91; again, waiting for the second recourse would only bring the vehicle closer to the depot making it less costly. When a duration limit is enforced in the constrained version, the structure of the new tour is altered such that two recourse actions cannot occur jointly at two locations so far from the depot.

The tour failure probabilities $\pi_i$ for the two tours are provided in Figure 9. Note that the unconstrained tour has high probabilities when the tour is close to the depot, which of course results from the objective of minimizing expected tour duration. On the other hand, the duration-constrained tour must balance both expected and maximum duration. Therefore, it allows an increase in expected duration so as to ensure that the adversary cannot in any realization obtain recourse actions that all occur at customer locations far away from the depot and that would lead to a violation of the duration constraint. The maximum duration of the unconstrained tour is 276.3 (again, resulting from recourse actions at customers 16 and 91) and its expected duration is 196, while the maximum duration of the constrained tour is 268.3 (with recourse actions at customers 16 and 98) and its expected duration is 199.6. By carefully selecting the order in which customers are visited, the constrained tour
hedges against the demand uncertainty by changing the space of feasible solutions of the adversarial problem, cutting out solutions that result in a larger increase in duration. Note that the price paid for this robustness is about 2%. Such slight increases in expected costs were consistently found in our computational study.

We now present the complete results of our computational study. Table 4 reports for the best solution for each constrained instance the increase in total expected travel time ($\Delta L_E$) over the unconstrained best solution, and the number of required vehicles ($m + z$).

Observe that for 22 out of the 54 constrained instances, the number of vehicles required to serve the customers increases ($z > 0$). Thus, duration limits have an important effect on required fleet size, as expected. Of course, an increase in the number of required vehicles is more likely when the duration limit is more restrictive (i.e., when $\alpha$ is smaller). For $\alpha = 0.75$ the number of required vehicles increases for 83.30% of the instances, for $\alpha = 0.85$ the number of required vehicles increases for 38.90% of the instances, but for $\alpha = 0.95$ none of the instances require more vehicles.

We also observe that the price that must be paid for guaranteeing a maximum tour duration, in terms of the total expected travel time, is likely to be relatively small. There are a few exceptions. When the number of customers is small (20), the vehicle capacity is low ($Q_1$), and the tour duration is severely restricted ($\alpha = 0.75$), then the total expected travel time increases by more than 7% regardless of whether the demand variability is low, medium, or high. Note also that these are the only instances where the number of required vehicles increases by two. In a few cases, the necessity to increase the fleet size can actually lead to a decrease in the total expected travel time. The chance of this happening is greater when the number of customers is large (60 or 100) and the vehicle capacity is large ($Q_2$). Note that in such cases, it would have been cost-effective from the perspective of only minimizing total expected cost to introduce pre-planned detours to the depot in the solution to the
unconstrained problem.

Enforcing tour duration constraints in instances with a small number of customers has a stronger effect (both in terms of the required number of vehicles and the total expected travel time) than in instances with a large number of customers. This suggests that delivery (or pickup) environments in which customers are geographically more concentrated will be less affected by enforcing duration constraints than systems with more geographically dispersed customers.

Similarly, enforcing tour duration constraints in instances with high-capacity vehicles \((i.e., Q_2)\) has a more pronounced impact than in instances with low-capacity vehicles \((i.e., Q_1)\). For low-capacity instances, 74.1% of the constrained versions require the same number of vehicles as their unconstrained versions, while only 44.5% of the high-capacity instances require the same number of vehicles. Interestingly, though, for instances with a large number of customers \((N = 100)\) it does seem to be beneficial to have a fleet of high-capacity vehicles. The explanation may be simple. Although tour duration constraints tend to reduce the benefit of high-capacity vehicles, when customers are highly concentrated more of them can be served even in short duration tours. Thus, high-capacity vehicles may still offer advantages.

Finally, the impact of demand variability seems to be less than the impact of other factors.

7 Conclusions

We have shown how to enforce maximum duration constraints on tours found for the vehicle routing problem with stochastic demands, which is an important idea in practice. Our example computational study shows that while enforcing duration constraints on the test instances considered may come only at a small increase in expected duration, it is certainly more difficult to serve customers with the same number of vehicles. It is important to acknowledge that the study we performed is by no means comprehensive, and the actual impact of duration constraints on the expected travel and vehicle costs for a problem setting will vary. Specifically, additional testing is required to understand how impacts may vary with different customer geographies, or with heterogeneous customers with different demand means and/or variances.

Duration constraints, while important, are only one important timing consideration for stochastic vehicle routing problems. Additional work is needed to also understand how to handle other important practical concerns, such as customer time windows.

References


Appendix - Recourse policy $\mathcal{P}^{NS-E}$

In this appendix, we present recourse conditions for $\mathcal{P}^{NS-E}$ and show how to calculate the expected duration of a tour operated under $\mathcal{P}^{NS-E}$.

Recall that $\mathcal{P}^{NS-E}$ denotes the following non-splittable recourse policy on a fixed sequence $\mathcal{T}$ with two types of recourse actions:

Type I: take a recourse action at customer $i$ restocking at the depot and then going back to $i$ if and only if $d(i)$ is strictly greater than onboard inventory. The recourse action is taken before satisfying any of the demand of $i$.

Type II: take a recourse action for customer $i$ immediately after serving customer $i - 1$, restocking at the depot before traveling to $i$ if and only if onboard inventory after serving $i - 1$ is zero.

Observation 2 Consider a fixed sequence $\mathcal{T} = \{1, \cdots, n\}$ operated using recourse policy $\mathcal{P}^{NS-E}$, and assume a demand realization $d$ such that there is a recourse of type I at customer $i \in \mathcal{T}$; then $d(i) \geq 2$.

Observation 3 Consider a fixed sequence $\mathcal{T} = \{1, \cdots, n\}$ operated using recourse policy $\mathcal{P}^{NS-E}$, and assume a demand realization $d$ such that there is a recourse (type I or II) for customer $i \in \mathcal{T}$; then the on-board inventory after $i$ is served is $Q - d(i)$.

The following three lemmas will be useful in the proofs to follow; the second two are clear and are presented without proof.

Lemma 5 Consider a fixed sequence $\mathcal{T} = \{1, \cdots, n\}$ operated using recourse policy $\mathcal{P}^{NS-E}$, and assume a demand realization $d \in \mathcal{U}$ such that the $(r - 1)$-th recourse action occurred for $j \in \mathcal{T}$. If the $r$-th occurs for $i \in \mathcal{T}$ ($j < i$) and it is a Type I recourse action then

$$d(i) \geq d(i/j) \equiv \max \left\{ 2, Q + 1 - \sum_{\ell=j}^{i-1} d(\ell) \right\}.$$ 

Proof. For $\mathcal{P}^{NS-E}$ such demand realization $d$ must satisfy

$$\sum_{\ell=j}^{i-1} d(\ell) \leq Q - 1 \quad (5)$$

$$\sum_{\ell=j}^{i} d(\ell) \geq Q + 1 \quad (6)$$

Since $d \in \mathcal{U}$ then $d(\ell) \leq \overline{d}(\ell)$ and summing over $\ell$

$$\sum_{\ell=j}^{i-1} d(\ell) \leq \sum_{\ell=j}^{i-1} \overline{d}(\ell).$$
The above expression together with (5) implies
\[
\sum_{\ell=j}^{i-1} d(\ell) \leq \min \left\{ Q - 1, \sum_{\ell=j}^{i-1} \bar{d}(\ell) \right\}
\]

From (6)
\[
d(i) \geq Q + 1 - \sum_{\ell=j}^{i-1} d(\ell) \\
\geq Q + 1 - \min \left\{ Q - 1, \sum_{\ell=j}^{i-1} \bar{d}(\ell) \right\} \\
\geq \max \left\{ 2, Q + 1 - \sum_{\ell=j}^{i-1} \bar{d}(\ell) \right\}.
\]

Observe that \(d(i/j)\) is a lower bound on the demand of customer \(i\) if a Type I recourse is taken for customer \(i\) given a recourse action (of either type) for customer \(j\) \((j < i)\). Furthermore, observe that the occurrence of a Type II recourse for customer \(i\) is independent of the value of \(d(i)\).

**Lemma 6** Consider vectors \(\underline{d}, \bar{d} \in \mathbb{Z}_n^+\) such that \(\underline{d} \leq \bar{d}\) and \(\bar{d}(n) > 1\). If
\[
\sum_{\ell=1}^{n-1} d(\ell) \leq M - 1 \quad \text{and} \quad \sum_{\ell=1}^{n} \bar{d}(\ell) \geq M + 1,
\]
where \(M < \infty\) is an integer constant, then there exists a vector \(d \in \mathbb{Z}_n^+\) such that \(\underline{d} \leq d \leq \bar{d}\) that satisfies
\[
\sum_{\ell=1}^{n-1} d(\ell) \leq M - 1 \quad \text{and} \quad \sum_{\ell=1}^{n} d(\ell) \geq M + 1.
\]

**Lemma 7** Consider vectors \(\underline{d}, \bar{d} \in \mathbb{Z}_n^+\) such that \(\underline{d} \leq \bar{d}\) and \(\bar{d}(n) > 0\). If
\[
\sum_{\ell=1}^{n-1} d(\ell) \leq M - 1 \quad \text{and} \quad \sum_{\ell=1}^{n} d(\ell) \leq M \quad \text{and} \quad \sum_{\ell=1}^{n} \bar{d}(\ell) \geq M,
\]
where \(M < \infty\) is an integer constant, then there exists a vector \(d \in \mathbb{Z}_n^+\) such that \(\underline{d} \leq d \leq \bar{d}\) that satisfies
\[
\sum_{\ell=1}^{n-1} d(\ell) \leq M - 1 \quad \text{and} \quad \sum_{\ell=1}^{n} d(\ell) = M.
\]
We now derive recourse conditions for $\mathcal{P}^{NS-E}$.

**Proposition 5 (Recourse condition Type I - Type I)** Consider a fixed sequence $\mathcal{T} = \{1, \cdots, n\}$ operated under recourse policy $\mathcal{P}^{NS-E}$. Assume there exists a demand realization $d \in \mathcal{U}$ such that the $(r-1)$-th recourse occurs for $j$ and the $r$-th occurs for $i > j$ for $r > 0$. Furthermore, assume the $r$-th is a Type I recourse action, then the $(r+1)$-th recourse can occur for $k > i$ and be of Type I if and only if

\[
d(k) \geq 2 \text{ and } \sum_{\ell=1}^k d(\ell) \geq Q + 1 \text{ and } \max\{d(i/j), d(i)\} + \sum_{\ell=i+1}^{k-1} d(\ell) \leq Q - 1
\]

**Proof.**

(⇒) The first condition follows immediately from Observation 2. For a recourse of Type I to occur at $k$ after a recourse of Type I at $i$, we have to have that the demand realization $d \in \mathcal{U}$ satisfies:

\[
\sum_{\ell=i}^{k-1} d(\ell) \leq Q - 1 \tag{7}
\]

and

\[
\sum_{\ell=i}^{k} d(\ell) \geq Q + 1. \tag{8}
\]

The second condition follows from (8) and $d(\ell) \leq \overline{d}(\ell)$ for all $\ell$. Finally, the third condition follows from (7), $d(\ell) \leq \overline{d}(\ell)$ for all $\ell$, and Lemma 5, because together these imply that

\[
\max\{d(i/j), d(i)\} + \sum_{\ell=i+1}^{k-1} d(\ell) \leq Q - 1.
\]

(⇐) Let $d'(1) = \max\{d(i/j), d(i)\}$, $d'(\ell) = d(i + \ell - 1)$ for $\ell = 2, \cdots, k - i + 1$ and $\overline{d}'(\ell) = \overline{d}(i + \ell - 1)$ for $\ell = 1, \cdots, k - i + 1$. The result now follows from Lemma 6 (with $m = Q, n = k - i + 1, \overline{d} = \overline{d}'$, and $\overline{d} = \overline{d}$).

Observe that Proposition 5 can be used to determine the customers where the second Type I recourse action can occur given that the first Type I recourse occurs for customer $i$ by setting $r = 1$ and $j = 0$ since it is assumed that $\overline{d}(0) = \overline{d}(0) = 0$.

**Proposition 6 (Recourse condition Type I - Type II)** Consider a fixed sequence $\mathcal{T} = \{1, \cdots, n\}$ operated using recourse policy $\mathcal{P}^{NS-E}$. Assume there exists a demand realization $d \in \mathcal{U}$ such that the $(r-1)$-th recourse occurs for $j$ and the $r$-th occurs for $i > j$ for $r > 0$. If the $r$-th is a Type I recourse action, then the $(r+1)$-th recourse can occur for $k > i$ and be of Type II if and only if

\[
\sum_{\ell=i}^{k-1} \overline{d}(\ell) \geq Q \text{ and } \max\{d(i/j), d(i)\} + \sum_{\ell=i+1}^{k-1} d(\ell) \leq Q - 1
\]

\[
\max\{d(i/j), d(i)\} + \sum_{\ell=i+1}^{k-2} d(\ell) \leq Q - 1.
\]

32
Proof.

(⇒) For a recourse of Type II to occur at \( k \) after a recourse of Type I at \( i \), we have to have that the demand realization \( d \in \mathcal{U} \) satisfies:

\[
\sum_{\ell=i}^{k-2} d(\ell) \leq Q - 1 \tag{9}
\]

and

\[
\sum_{\ell=i}^{k-1} d(\ell) = Q. \tag{10}
\]

The first condition follows from (10) and \( d(\ell) \leq d'(\ell) \) for all \( \ell \). The second condition follows from Lemma 5, (10), and \( d(\ell) \leq d'(\ell) \) for all \( \ell \) since together they imply

\[
\max \{d(i/j), d(i)\} + \sum_{\ell=i+1}^{k-1} d(\ell) \leq \sum_{\ell=i+1}^{k-1} d(\ell) = Q.
\]

Finally, third condition follows from Lemma 5, (9), and \( d(\ell) \leq d'(\ell) \) for all \( \ell \) since together they imply

\[
\max \{d(i/j), d(i)\} + \sum_{\ell=i+1}^{k-2} d(\ell) \leq \sum_{\ell=i+1}^{k-2} d(\ell) \leq Q - 1.
\]

(⇐) Let \( d'(1) = \max \{d(i/j), d(i)\} \), \( d'(\ell) = d(i + \ell - 1) \) for \( \ell = 2, \ldots , k - i \) and \( d'(\ell) = d(i + \ell - 1) \) for \( \ell = 1, \ldots , k - i \). The result now follows from Lemma 7 (with \( M = Q \), \( n = k - i \), \( \overline{d} = \overline{d}' \), and \( \overline{d} = \overline{d}' \)).

The following results are presented without proof because they are practically identical to those of either Proposition 5 or Proposition 6.

Proposition 7 (Recourse condition Type II - Type II) Consider a fixed sequence \( \mathcal{T} = \{1, \cdots , n\} \) operated using recourse policy \( P^{NS-E} \). Assume there exists a demand realization \( d \in \mathcal{U} \) such that the \( r \)-th recourse occurs for \( i \) for \( r > 0 \). Assume the \( r \)-th is a Type II recourse action, then the \((r+1)\)-th recourse can occur for \( k > i \) and be of Type II if and only if

\[
\sum_{\ell=i}^{k-2} d(\ell) \leq Q - 1 \quad \text{and} \quad \sum_{\ell=i}^{k-1} d(\ell) \leq Q \quad \text{and} \quad \sum_{\ell=i}^{k-1} \overline{d}(\ell) \geq Q.
\]

Proposition 8 (Recourse condition Type II - Type I) Consider a fixed sequence \( \mathcal{T} = \{1, \cdots , n\} \) operated using recourse policy \( P^{NS-E} \). Assume there exists a demand realization \( d \in \mathcal{U} \) such that the \( r \)-th recourse occurs for \( i \) for \( r > 0 \). Assume the \( r \)-th is a Type II recourse action, then the \((r+1)\)-th recourse can occur for \( k > i \) and be of Type I if and only if
if
\[ \sum_{\ell=1}^{k-1} d(\ell) \leq Q - 1 \quad \text{and} \quad \sum_{\ell=1}^{k} \bar{d}(\ell) \geq Q + 1 \quad \text{and} \quad \bar{d}(k) \geq 2. \]

**Proposition 9 (First Recourse Type I)** Consider a fixed sequence \( \mathcal{T} = \{1, \cdots, n\} \) operated using recourse policy \( \mathcal{P}^{NS-E} \). There exists a demand realization \( d \in \mathcal{U} \) such that the first recourse occurs for \( i \in \mathcal{T} \) and be of Type I if and only if
For \( i > 1 \):
\[ \sum_{\ell=1}^{i-1} d(\ell) \leq Q - 1 \quad \text{and} \quad \sum_{\ell=1}^{i} \bar{d}(\ell) \geq Q + 1 \quad \text{and} \quad \bar{d}(i) \geq 2. \]

Note that a Type I recourse cannot occur for customer 1.

**Proposition 10 (First Recourse Type II)** Consider a fixed sequence \( \mathcal{T} = \{1, \cdots, n\} \) operated using recourse policy \( \mathcal{P}^{NS-E} \). There exists a demand realization \( d \in \mathcal{U} \) such that the first recourse occurs for \( i \in \mathcal{T} \) and be of Type II if and only if
For \( i > 2 \):
\[ \sum_{\ell=1}^{i-2} d(\ell) \leq Q - 1 \quad \text{and} \quad \sum_{\ell=1}^{i-1} d(\ell) \leq Q \quad \text{and} \quad \sum_{\ell=1}^{i-1} \bar{d}(\ell) \geq Q. \]

Note that a Type II recourse cannot occur for the first or second customers in the tour.

For \( \mathcal{P}^{NS-E} \), a Type II recourse for customer \( i \) is a regenerative point in the sense that it always leads to a full vehicle inventory state prior to visiting \( i \). A Type I recourse action for \( i \) is not such a point, however all the information required by the adversary to determine where the next recourse can take place is captured fully by both \( i \) and the customer location of the previous recourse action \( j \). Therefore, for \( \mathcal{P}^{NS-E} \) the adversarial problem can be solved on a digraph where node \( (i/j) \) represents taking a Type I recourse action for customer \( i \) given a recourse action for customer \( j \), while node \( (i) \) represents taking a Type II recourse action at \( i - 1 \) for customer \( i \). Note that the type of recourse action for customer \( j \) is not required to be known. An arc between two nodes represents the existence of a demand realization that generates consecutive recourse actions for the corresponding customers.

**Solving the adversarial problem for recourse policy \( \mathcal{P}^{NS-E} \)**

Let \( G_{2}(\mathcal{T}, \mathcal{P}^{NS-E}) = (\mathcal{N}, \mathcal{A}) \) denote the digraph used to solve the adversarial problem for policy \( \mathcal{P}^{NS-E} \). The node set is given by:
\[ \mathcal{N} = \{s\} \cup \{t\} \cup \mathcal{N}' \cup \mathcal{N}''. \]

Node set component \( \mathcal{N}' \) is defined as
\( \mathcal{N}' = \{(1,i/0) | i \in \mathcal{T} \setminus \{1\}\} \cup \{(r,i/j) | r \in \{2, \cdots, R\}, i \in \{r+1, \cdots, n\}, j \in \{r, \cdots, i-1\}\}, \)

where node \((1,i/0)\) is associated with having the first Type I recourse for customer \(i\), and where \((r,i/j)\) is associated with having the \(r\)-th recourse of Type I for customer \(i\) given that the \((r-1)\)-th occurred for customer \(j\). Node set component \(\mathcal{N}''\) is defined as

\( \mathcal{N}'' = \{(1,i) | i \in \mathcal{T} \setminus \{1,2\}\} \cup \{(r,i) | r \in \{2, \cdots, R\}, i \in \{r+1, \cdots, n\}\}, \)

where node \((1,i)\) is associated with having the first Type II recourse for customer \(i\) and where \((r,i)\) is associated with having the \(r\)-th recourse be of Type II for customer \(i\).

The arc set \(\mathcal{A}\) is defined using the recourse conditions. Let

\[
\mathcal{A} = \bar{\mathcal{A}} \cup \mathcal{A}_I^I \cup \mathcal{A}_I^I,
\]

where

\[
\bar{\mathcal{A}} = \mathcal{A}_s^I \cup \mathcal{A}_s^I \cup \mathcal{A}_{r,r+1}^I \cup \mathcal{A}_{r,r+1}^I \cup \mathcal{A}_{r,r+1}^I \cup \mathcal{A}_{r,r+1}^I.
\]

We now define the component sets used above. The first set,

\( \mathcal{A}_I^I = \{(s,(1,i/0)) | (1,i/0) \in \mathcal{N}' \text{ for any } i \text{ satisfying Proposition 9}\}, \)

is associated with the first Type I recourse action and the cost of such arcs is the additional travel time of a recourse action for \(i\): \(l(i,0) + l(0,i)\). The second set,

\( \mathcal{A}_s^I = \{(s,(1,i)) | (1,i) \in \mathcal{N}'' \text{ for any } i \text{ satisfying Proposition 10}\}, \)

is associated with the first Type II recourse action and the cost of such arcs is the additional travel time of a recourse action for \(i\): \(l(i-1,0) + l(0,i) - l(i-1,i)\). The third set,

\( \mathcal{A}_{r,r+1}^I = \{(r,i/j),(r+1,k/i)) | (r,i/j) \in \mathcal{N}', (r+1,k/i) \in \mathcal{N}' \text{ and } i,j,k \text{ satisfying Proposition 5}\}, \)

is associated with a Type I recourse for customer \(k\) preceded by a Type I recourse for customer \(i\) \((i < k)\) which is in turn preceded by a recourse (of either type) for customer \(j\) \((j < i)\) with cost \(l(k,0) + l(0,k)\). The fourth set,

\( \mathcal{A}_{r,r+1}^{I,I} = \{(r,i/j),(r+1,k)) | (r,i/j) \in \mathcal{N}', (r+1,k) \in \mathcal{N}'' \text{ and } i,j,k \text{ satisfying Proposition 6}\}, \)

is associated with a Type II recourse for customer \(k\) preceded by a Type I recourse for customer \(i\) \((i < k)\) again in turn preceded by a recourse (of either type) for customer \(j\) \((j < i)\) with cost \(l(k-1,0) + l(0,k) - l(k-1,k)\). The fifth set,
\( \mathcal{A}_{II}^{r+1} = \{(r, i), (r+1, k) \mid (r, i) \in \mathcal{N}^r, (r+1, k) \in \mathcal{N}^r \text{ and } i, k \text{ satisfying Proposition } 7 \} \),

is associated with a Type II recourse for customer \( k \) preceded by a Type II recourse for customer \( i \) \((i < k)\) with cost \( l(k-1,0) + l(0,k) - l(k-1,k) \). The sixth set,

\( \mathcal{A}_{II,l}^{r+1} = \{(r, i), (r+1, k/i) \mid (r, i) \in \mathcal{N}^r, (r+1, k/i) \in \mathcal{N}^r \text{ and } i, k \text{ satisfying Proposition } 8 \} \),

is associated with a Type I recourse for customer \( k \) preceded by a Type II recourse for customer \( i \) \((i < k)\) with cost \( l(k,0) + l(0,k) \). The seventh set,

\( \mathcal{A}_{I}^t = \{(r, i/j), t \mid (r, i/j) \in \mathcal{N}^t \text{ and } \text{indeg}(r, i/j) > 0, \text{outdeg}(r, i/j) = 0 \text{ on } (\mathcal{N}, \bar{A}) \} \),

connects recourse nodes when necessary to the sink, where \( \text{indeg}(v) \) and \( \text{outdeg}(v) \) denotes the indegree and outdegree respectively of node \( v \). The cost of such arcs is 0. Finally, the eighth set,

\( \mathcal{A}_{II}^t = \{(r, i), t \mid (r, i) \in \mathcal{N}^t \text{ and } \text{indeg}(r, i) > 0, \text{outdeg}(r, i) = 0 \text{ on } (\mathcal{N}, \bar{A}) \} \),

also connects recourse nodes when necessary to the sink with cost 0.

Let \( L(\mathcal{G}_2(\mathcal{T}, \mathcal{P}^{NS-E})) \) be the length of the longest \( s - t \) path in the network. If there does not exist an \( s - t \) path, then by definition \( L(\mathcal{G}_2(\mathcal{T}, \mathcal{P}^{NS-E})) = 0 \).

**Theorem 2** For a tour \( \mathcal{T} \) operated using policy \( \mathcal{P}^{NS-E} \), \( \Phi(\mathcal{T}, \mathcal{P}^{NS-E}) = L(\mathcal{G}_2(\mathcal{T}, \mathcal{P}^{NS-E})) \). Furthermore, \( L(\mathcal{G}_2(\mathcal{T}, \mathcal{P}^{NS-E})) \) can be computed in \( O(n^4) \) time by solving a longest \( s - t \) path problem on \( \mathcal{G}_2(\mathcal{T}, \mathcal{P}^{NS-E}) \).

**Proof.** By construction \( \Phi(\mathcal{T}, \mathcal{P}^{NS-E}) \) is equal to the length of the longest \( s - t \) path in \( \mathcal{G}_2(\mathcal{T}, \mathcal{P}^{NS-E}) \). Observe that the number of nodes belonging to \( \mathcal{N}^r \) in the \( r \)-th layer of the graph for \( r > 1 \) is bounded by \( \sum_{i=1}^{n-r} i = \frac{(n-r)(n-r+1)}{2} \); for \( r = 1 \) it is bounded by \( n \). Also, the number of nodes belonging to \( \mathcal{N}^n \) in the \( r \)-th layer of the graph is bounded by \( n \); this implies that \( |\mathcal{A}_I^r| \) is \( O(n) \) and \( |\mathcal{A}_{II}^r| \) is \( O(n) \). \( R \) is bounded by \( n \); therefore, the number of nodes in \( \mathcal{N}^r \) is \( O(n^3) \), and the number of nodes in \( \mathcal{N}^n \) is \( O(n^2) \); this further implies that \( |\mathcal{A}_I^r| \) is \( O(n^3) \) and \( |\mathcal{A}_{II}^r| \) is \( O(n^2) \).

Each node \((r, i/j)\) belonging to \( \mathcal{N}^r \) in the \( r \)-th layer of the graph is connected to at most \( n \) nodes in \( \mathcal{N}^r \) and to at most \( n \) nodes in \( \mathcal{N}^n \) in the \((r+1)\)-th layer. Similarly, each node \((r, i)\) belonging to \( \mathcal{N}^n \) in the \( r \)-th layer of the graph is connected to at most \( n \) nodes in \( \mathcal{N}^r \), and to at most \( n \) nodes in \( \mathcal{N}^n \) in the \((r+1)\) layer. Therefore, \( |\mathcal{A}_I^{r+1}| \) is \( O(n^4) \), \( |\mathcal{A}_{II}^{r+1}| \) is \( O(n^3) \), \( |\mathcal{A}^{r+1}_{I,l}| \) is \( O(n^4) \) and \( |\mathcal{A}_{II,l}^{r+1}| \) is \( O(n^3) \). It is concluded then that \( |\mathcal{A}| \) is \( O(n^4) \) and the
complexity result follows because the digraph is acyclic.

*Calculating the expected duration of a tour operated with recourse policy $P^{NS-E}$*

Consider a tour $T = \{1, 2, \ldots, n\}$ and let $p_i(\delta)$ be the probability that the $i$-th customer in the tour has a demand value equal to $\delta$. At any customer $i$ there are three mutually exclusive and exhaustive states denoted by $s$:

$$s = \begin{cases} 0 & \text{if no recourse occurs for } i \\ 1 & \text{if a Type I recourse occurs for } i \\ 2 & \text{if a Type II recourse occurs for } i + 1 \end{cases}$$

Denote set $S = \{0, 1, 2\}$ representing the states defined above, and let $\beta(i, s, q)$ be the probability of having on-board inventory equal to $q$ when arriving to customer $i + 1$ after serving the $i$-th customer in $T$. Observe that the value of $q$ includes the updated on-board inventory if a Type II recourse action is initiated for customer $i + 1$ (i.e., $s = 2$ for customer $i$), therefore $q \in \{1, 2, \ldots, Q\}$. For ease of notation, we define $Q(j, k) = \{j, j + 1, \ldots, k - 1, k\}$, for $j$ and $k$ integers such that $1 \leq j \leq k \leq Q$.

It is clear that under our assumptions for any $q \in Q(1, Q)$,

$$\beta(1, s, q) = \begin{cases} p_1(Q - q) & \text{if } s = 0 \\ 0 & \text{otherwise} \end{cases}$$

For $i \geq 2$ we calculate $\beta(i, s, q)$ by conditioning on the elements of state space $S$ and the value of $q$ for the $i - 1$-th customer in $T$. Consider the three cases associated with each element of $S$. For $s = 0$, to calculate $\beta(i, 0, q)$ assume an on-board inventory value of $\bar{q}$ after serving customer $i - 1$, then an on-board inventory value of $q$ after serving customer $i$ implies a demand realization $\bar{q} - q$ at customer $i$. Therefore,

$$\beta(i, 0, q) = \sum_{s \in S} \sum_{\bar{q} \in Q(1,q)} \beta(i - 1, s, \bar{q}) p_i(\bar{q} - q) \text{ for } i \in \{2, \ldots, n\}, q \in Q(1, Q)$$

In order to calculate $\beta(i, s, q)$ for $s = 1$, assume an on-board inventory $\bar{q}$ after serving customer $i - 1$, which by Observation 3 implies a demand realization such that $Q - q > \bar{q}$ at customer $i$. Therefore,

$$\beta(i, 1, q) = \sum_{s \in S} \sum_{\bar{q} \in Q(1,Q-q-1)} \beta(i - 1, s, \bar{q}) p_i(Q - q) \text{ for } i \in \{2, \ldots, n\}, q \in Q(1, Q - 2)$$

For this case it is clear that the values of $\beta(i, 1, Q)$ and $\beta(i, 1, Q - 1)$ are equal to zero, because they both imply a Type I recourse for customer $i$ after observing a demand realization of 0 and 1 respectively, which cannot occur under this policy.
Conditioning on an on-board inventory $\bar{q}$ after serving customer $i - 1$, a state $s = 2$ for customer $i$ implies a demand realization of $\bar{q}$ at this customer; therefore,

$$
\beta(i, 2, q) = \sum_{s \in S} \sum_{\bar{q} \in \mathbb{Q}(1,Q)} \beta(i - 1, s, \bar{q}) p_i(\bar{q}) \quad \text{for } i \in \{2, \ldots, n - 1\}, \; q = Q
$$

It is clear that $\beta(i, 2, q) = 0$ for any $q < Q$, and also that $\beta(n, 2, Q) = 0$ since a Type II recourse action is never initiated after serving the last customer in the tour.

Let $\pi_i^I = \text{probability of a type I recourse action for customer } i$ and $\pi_i^{II} = \text{a type II recourse action for customer } i + 1$. Therefore,

$$
\pi_i^I = \sum_{q \in \mathbb{Q}(1,Q)} \beta(i, 1, q)
$$

$$
\pi_i^{II} = \sum_{q \in \mathbb{Q}(1,Q)} \beta(i, 2, q)
$$

and the expected additional duration due to taking recourse actions can be calculated as

$$
\mathbb{E}[\phi(T, P^{NS-E}, d)] = \sum_{i \in T} \pi_i^I [l(i, 0) + l(0, i)] + \sum_{i \in T} \pi_i^{II} [l(i, 0) + l(0, i + 1) - l(i, i + 1)].
$$
<table>
<thead>
<tr>
<th>$(N, Q, \sigma, m)$</th>
<th>$\alpha$</th>
<th>$\Delta L_E$</th>
<th>$m + z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(20, Q_1, \text{High}, 6)$</td>
<td>0.95</td>
<td>2.34%</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>0.85</td>
<td>4.57%</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>0.75</td>
<td>8.82%</td>
<td>8</td>
</tr>
<tr>
<td>$(20, Q_1, \text{Med}, 6)$</td>
<td>0.95</td>
<td>0.38%</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>0.85</td>
<td>2.58%</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>0.75</td>
<td>7.02%</td>
<td>8</td>
</tr>
<tr>
<td>$(20, Q_1, \text{Low}, 6)$</td>
<td>0.95</td>
<td>1.06%</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>0.85</td>
<td>1.10%</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>0.75</td>
<td>8.96%</td>
<td>8</td>
</tr>
<tr>
<td>$(20, Q_2, \text{High}, 3)$</td>
<td>0.95</td>
<td>1.33%</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>0.85</td>
<td>1.96%</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>0.75</td>
<td>1.96%</td>
<td>4</td>
</tr>
<tr>
<td>$(20, Q_2, \text{Med}, 3)$</td>
<td>0.95</td>
<td>1.87%</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>0.85</td>
<td>2.21%</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>0.75</td>
<td>2.65%</td>
<td>4</td>
</tr>
<tr>
<td>$(20, Q_2, \text{Low}, 3)$</td>
<td>0.95</td>
<td>1.50%</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>0.85</td>
<td>2.55%</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>0.75</td>
<td>3.71%</td>
<td>4</td>
</tr>
<tr>
<td>$(60, Q_1, \text{High}, 6)$</td>
<td>0.95</td>
<td>0.74%</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>0.85</td>
<td>1.35%</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>0.75</td>
<td>1.35%</td>
<td>6</td>
</tr>
<tr>
<td>$(60, Q_1, \text{Med}, 6)$</td>
<td>0.95</td>
<td>0.83%</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>0.85</td>
<td>0.83%</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>0.75</td>
<td>0.83%</td>
<td>6</td>
</tr>
<tr>
<td>$(60, Q_1, \text{Low}, 6)$</td>
<td>0.95</td>
<td>0.44%</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>0.85</td>
<td>0.44%</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>0.75</td>
<td>0.44%</td>
<td>6</td>
</tr>
<tr>
<td>$(60, Q_2, \text{High}, 3)$</td>
<td>0.95</td>
<td>0.18%</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>0.85</td>
<td>-1.03%</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>0.75</td>
<td>-0.75%</td>
<td>4</td>
</tr>
<tr>
<td>$(60, Q_2, \text{Med}, 3)$</td>
<td>0.95</td>
<td>4.88%</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>0.85</td>
<td>-0.34%</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>0.75</td>
<td>0.73%</td>
<td>4</td>
</tr>
<tr>
<td>$(60, Q_2, \text{Low}, 3)$</td>
<td>0.95</td>
<td>1.12%</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>0.85</td>
<td>0.87%</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>0.75</td>
<td>4.26%</td>
<td>4</td>
</tr>
<tr>
<td>$(100, Q_1, \text{High}, 6)$</td>
<td>0.95</td>
<td>0.69%</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>0.85</td>
<td>1.60%</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>0.75</td>
<td>-0.31%</td>
<td>7</td>
</tr>
<tr>
<td>$(100, Q_1, \text{Med}, 6)$</td>
<td>0.95</td>
<td>1.01%</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>0.85</td>
<td>2.86%</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>0.75</td>
<td>-0.41%</td>
<td>7</td>
</tr>
<tr>
<td>$(100, Q_1, \text{Low}, 6)$</td>
<td>0.95</td>
<td>1.18%</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>0.85</td>
<td>1.18%</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>0.75</td>
<td>5.22%</td>
<td>7</td>
</tr>
<tr>
<td>$(100, Q_2, \text{High}, 3)$</td>
<td>0.95</td>
<td>0.67%</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>0.85</td>
<td>1.28%</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>0.75</td>
<td>-1.66%</td>
<td>4</td>
</tr>
<tr>
<td>$(100, Q_2, \text{Med}, 3)$</td>
<td>0.95</td>
<td>0.17%</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>0.85</td>
<td>0.61%</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>0.75</td>
<td>-1.15%</td>
<td>4</td>
</tr>
<tr>
<td>$(100, Q_2, \text{Low}, 3)$</td>
<td>0.95</td>
<td>0.17%</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>0.85</td>
<td>0.89%</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>0.75</td>
<td>-0.74%</td>
<td>4</td>
</tr>
</tbody>
</table>