Bounding Procedures for Multistage Stochastic Dynamic Networks

Linos F. Frantzeskakis* and Warren B. Powell
Princeton University, School of Engineering and Applied Science, Department of Civil Engineering and Operations Research, Princeton, New Jersey 08544

This paper presents new procedures for obtaining upper and lower bounds in the cost minimization of multistage dynamic networks with random link capacities and contrasts them to existing methods. These methods include adaptation of Jensen's inequality to the problem, Monte-Carlo simulations and an analytical approximation procedure. The new bounding procedures involve a backward pass in the network coupled with a linear approximation at each stage. Some numerical results are presented for networks of substantial size. © 1993 by John Wiley & Sons, Inc.

1. INTRODUCTION

Consider the following N-stage stochastic programming problem:

\[
\min \quad c^T x(1) + \bar{\psi}^2(S(2))
\]

subject to:

\[
A x(1) = T(1)
\]

\[
B x(1) - S(2) = 0
\]

\[
x(1) \preceq \phi(1)
\]

\[
x(1) \succeq 0
\]

We assume that the matrices A and B are m \times n node-arc incidence matrices for a (one-sided) transportation problem and x(1), \phi(1), S(2) are vectors in \mathbb{R}^n, \mathbb{R}^n and \mathbb{R}^m correspondingly. Equation (1.1a) is the flow conservation constraint of the first-stage nodes, that equate the exogenous node supplies T(1) to the sum of the endogenous arcs flows x(1). Equation (1.1b) is a definitional constraint for the vector S(2) which defines the internal supply S(2) of the nodes of the second stage as the result of the internal flows x(1). The arc capacities \phi(1) represent realizations of random upper bounds, which are known for the first time period.

Let \omega = (\omega_1, \omega_2, \ldots, \omega_N) be the outcome of a random vector defined on a probability space (\Omega, F, P) induced in \mathbb{R}^{N-1} with \Omega a convex set. Vector \omega_i is the outcome of period i, and arcs flows of that period will be restricted by random arc capacities \phi(t, \omega_i), with outcomes \phi(t, \omega_i).

The expected recourse function \bar{\psi}^2(S(2)) is defined by:

\[
\bar{\psi}^2(S(2)) = E_\omega[\psi^2(S(2), \phi(2, \omega_2))]
\]

subject to:

\[
A x(2, \omega_2) = T(2) + S(2)
\]

\[
B x(2, \omega_2) - S(3) = 0
\]

where

\[
\psi^2(S(2), \phi(2, \omega_2)) = \min_{x(2, \omega_2) \in \mathbb{R}^n} c^T x(2, \omega_2) + \psi^2(S(3))
\]

subject to:

\[
A x(2, \omega_2) = T(2) + S(2)
\]

\[
B x(2, \omega_2) - S(3) = 0
\]
with the expected recourse function \( \hat{\psi}(S(3)) \) for period 3 defined analogously. Total flow out of nodes in stage 2 must equal the sum of external supplies \( T(2) \) and the internal supplies \( S(2) \) from the previous stage, as specified in equation (1.3a). We assume that \( \hat{\psi}^{*}(SN + 1) = 0 \), thus limiting the problem to \( N \) stages.

Problem (1.1) is an \( N \)-stage stochastic programming problem with a special type of network recourse. Uncertainty enters the problem only through the random arc capacities \( \delta(t), t = 2, 3, \ldots, N \). For stage 1, the upper bound \( \hat{\delta}(1) \) is assumed to be realizations of the random variable \( \delta(1) \). Our goal is to find computable bounds for the optimal value of (1.1) for problems with that type of network structure where the dimensionality of the random vector \( \delta(t) \) may be quite large (e.g. in the thousands). We develop a novel approach for computing bounds for multistage stochastic programs represented by (1.1). The method replaces the expected recourse function \( \hat{\psi}(S(t)) \) with linear functions with the property that, when substituted into (1.3), give approximate recourse functions which are easily computed and bound the function \( \hat{\psi}(S(t)) \). This substitution is applied successively in a backward recursion, allowing upper or lower bounds to be easily computed for large, multistage problems.

The motivation for this problem comes from the Dynamic Vehicle Allocation (DVA) Problem which addresses the problem of dynamically managing a fleet of vehicles over time. The fleet of vehicles must handle loads which represent customer (shipper) demands to carry freight from one city to another. We assume that space has been divided into a set of \( R \) regions (or cities), so that vehicles are assumed to be located at, or moving between, those regions. These problems are faced by trucking companies, railroads, ocean container companies, and one-way truck rental companies. The vector \( T(1) \) describes the initial supplies of vehicles, where \( T(i) \) gives the number of vehicles initially in region \( i \). The vector \( T(t), t > 1 \) describes any deterministic exogenous changes to the size of the vehicle fleet. The random variables \( \delta(t) \) represent forecasted demands for loaded movements between pairs of regions, that is, the total number of vehicles requested by shippers to move from the origin region \( i \) to the destination region \( j \). When a vehicle finishes moving a shipper’s load to the destination region \( j \), it becomes empty and must be either reassigned to another load out of region \( j \), or moved empty from region \( j \) to another region \( k \). For period 1, these demands are assumed known. For later periods, the carrier must anticipate future shipper demands, which can only be forecasted with uncertainty. In truckload trucking, shippers rarely call in more than one day in advance. Thus, decisions regarding the movement of empty and loaded trucks must generally be made without prior knowledge of the future demand environment.

The stochastic DVA was first presented in a stochastic programming framework in Powell [21], which provides a more complete description of the problem, as well as references of prior research into stochastic formulations of the DVA problem. Although in this formulation there is a distinction between the flows of loaded and empty vehicles, the problem is mathematically equivalent to problem (1.1) described above. Frantzeskakis and Powell [10] present a heuristic for solving large-scale stochastic programs of this form; this method depends on approximating the expected recourse functions. They demonstrate that the heuristic performs quite well in rolling horizon experiments. This suggests that the first stage decisions \( x(1) \) may not differ substantially from the decisions that would have been made with perfect information on future demands. However, as indicated in Birge [2], it is possible for the approximate functions to be quite distant but “parallel” to the expected recourse functions. Thus, the quality of the recommended first stage decisions is not a reliable indication of the accuracy of the constructed approximates of the expected recourse functions. In this paper, we seek to derive upper and lower bounds to the optimal value of the stochastic program (1.1) as a means of evaluating heuristic solution techniques. The method developed is computationally efficient for large-scale problems with multiple stages. For industry applications of the DVA problem see Powell [20] and Powell et al. [22].

There is an extensive literature on bounds for stochastic programs; a thorough presentation can be found in Birge and Wets [5]. As presented in Kall and Stoyan [16], Jensen’s inequality provides the standard lower bound on the expectation of a convex function of a random variable. Jensen’s inequality was extended by Perlman [19] to convex vector-valued functions. Using the theory of moment spaces, Madansky [17] obtains an upper bound on the expectation of a convex function of a multivariate random variable with independent components and finite means. Based on similar ideas, Ben-Tal and Hochman [1] obtain sharper upper and lower bounds for the case of convex functions of independent random variables. Huang et al. [13] sequentially apply the conditional form of Jensen and Mandansky bounds to subsets of the domain of the random variables. They use the monotonicity of the above bounds, as indicated in Kall et al. [15], as the partition of the domain becomes finer and show that the bounds converge to the expected value of the convex function.
function and, in the case of simple recourse, so do the optimal decisions. More general convergence results (for the optimal solution) appear in Birge and Wets [5] and in Kall [14].

In the case of dependent random variables, lower bounds are still obtained through the conditional form of Jensen’s inequality, while Madansky’s upper bounds are not available. Marshall and Olkin [18] develop an explicit upper bound for the particular case of a separable convex function, with some additional conditions. Also, Birge and Wallace [3] treat the case where the random right-hand sides of a linear stochastic program are linear transformations of independent random variables. Finally, Gassmann and Ziemba [12] and Frenzen-dorfer [11] provide the extension of the Madansky bound to arbitrary distributions with no assumption of independence.

A different approach approximates the recourse function by a simpler one, whose expectation is easily calculated. Dala [7] approximates the recourse function by a sublinear function and in [8] introduces an upper bound on the expectation of such sublinear functions of random variables defined over the whole Euclidean space without any independence assumptions. The ray approximation procedure, introduced by Birge and Wets [5] uses this sublinearity property to obtain a separable function that majorizes the recourse function.

Similarly, Wallace [24] and Birge and Wallace [4] present piecewise linear bounds on the expectation of the recourse function, for the case of a two-stage problem with network and fixed recourse, respectively. The advantage of this bound is that the number of necessary operations grows polynomially (instead of exponentially) in the number of the random variables. Finally, Birge and Wets [6] extend the ray function approximation procedure and the ideas in Wallace [24] by using the convex hull of separable sublinear functions obtained by linear transformations of the random vectors. It should be emphasized that those bounding procedures involve 2-stage problems and specifically the evaluation of the recourse function for a specific set of first stage actions x.

The outline of the remainder of this paper is as follows. In section 2 we describe the N-stage cost minimization Stochastic Programming problem with Network Recourse for the DVA problem, while Section 3 uses Jensen’s inequality to obtain lower bounds. Section 4 presents a new lower bounding procedure based on a backward recursion, where at each stage linear approximations to the recourse problem are performed, while section 5 introduces an upper bounding procedure based on the same concepts. Section 6 describes a class of alternative Monte-Carlo simulation procedures and an analytical approach that attempts to obtain similar upper bounds with less computational effort. Finally, section 7 describes the experimental design, while section 8 reports computational results and concludes the paper.

2. STOCHASTIC PROGRAMMING FORMULATION

This section presents the N-stage Stochastic Programming problem with network recourse for the stochastic dynamic vehicle allocation (DVA) problem. The original formulation (1.1) is refined, to reflect the existence of unconstrained empty repositioning moves between regions.

The problem is formulated using a time-space network, which is depicted in Figure 1, where nodes (t, j) refer to the tth stage of the problem and links connect nodes (i, t) to (j, t + 1). A subset of links have random capacities; drawing from the DVA problem we refer to those links as loaded links. Links that have a deterministic finite capacity (and which involve only the first stage of the problem) represent known loads and are referred to as known links, while links with infinite capacity model empty moves and are called empty links. Only nodes that refer to the first stage have positive external supply, and only a super sink node has negative external supply, equal to the total positive external supply. We use the convention that the first time period is t = 1, and, when a time period is mentioned, it is implied that we refer to the beginning of that time period. Revenues are treated as negative costs, in order to formulate the problem as a cost-minimization problem.

The decision variables are:

\[ x_{ij}(1) = \text{the flow on the known link between nodes} \]
\[ (i, 1) \text{ and } (j, 2) \]

\[ x_{ij}(t, \omega) = \text{the flow on the loaded link between nodes} \]
\[ (i, t) \text{ and } (j, t + 1) \text{ under outcome } \omega, \text{ defined for} \]
\[ t = 2, \ldots, N \]

\[ y_{ij}(1) = \text{the flow on the empty link between nodes} \]
\[ (i, 1) \text{ and } (j, 2). \]

\[ y_{ij}(t, \omega) = \text{the flow on the empty link between nodes} \]
\[ (i, t) \text{ and } (j, t + 1) \text{ under outcome } \omega, \text{ defined for} \]
\[ t = 2, \ldots, N \]

\[ S(t) = \text{internal supply of node} \]
\[ (i, t), \text{ for } t = \]
\[ 2, \ldots, N + 1 \]
\[ = \sum_{\omega \in \Omega} x_{ij}(t - 1, \omega, \omega) + y_{ij}(t - 1, \omega, \omega). \]

The cost parameters are given by:

\[ -r g = \text{negative cost on known} \]
\[ (t = 1) \text{ or loaded} \]
\[ (t > 1) \text{ link}. \]
between \((i,t)\) and \((j, t + 1)\), defined for \(t = 1, \ldots, N\).

\(c_t\) = (positive) cost on empty link between \((i,t)\) and \((j,t + 1)\), for \(t = 1, \ldots, N\).

= 0 for \(i = j\) (holding a vehicle until the next period).

Finally, the flows are governed by the following activity variables:

\(\phi_t(t)\) = (random variable) the capacity of the loaded link between nodes \((i,t)\) and \((j,t + 1)\), defined for \(t = 2, \ldots, N\).

\(\phi_t(1)\) = known capacity of the known link between \((i,1)\) and \((j,2)\).

\(T(t)\) = (positive) external supply of node \((i,t)\).

= 0, for \(t > 1\)

\(F\) = total external supply of network,

= \(\sum_{i \in R} T(t)\)

In the model formulations that are presented in the remainder of the paper, we follow the convention that bold letters define a matrix or a vector with known or random elements and a bar on a random variable denotes its expected value.

The minimization of the expected cost in a dynamic network with random arc capacities for a given external supply distribution \(T(1)\) is equivalent to solving the following stochastic programming problem:

\[
\psi(T(1)) = \min_{(i,t), (j,t), S(1), S(2)} \sum_{i \in R} \sum_{(i,t), (j,t)} (-r_{i,j} x_{i,j} + c_{i,j} y_{i,j}) + \bar{v}(S(2))
\]
subject to:

\[ x_i(1) \leq \tilde{d}_i(1), \quad \forall i, j \]  

\[ \sum_{j \in S} (x_i(1) + y_i(1)) = T_i(1), \quad \forall i \]  

\[ \sum_{j \in S} (x_i(1) + y_j(1)) - S_j(2) = 0, \quad \forall j \]  

\[ x(1) > 0, \quad y(1) > 0, \quad S(2) > 0 \]  

where \( \tilde{\psi}(S(2)) \) is the expected recourse function for stage 2, given by:

\[ \tilde{\psi}(S(2)) = E_{\omega_2} [\tilde{\psi}(S(2), \psi(2, \omega_2))] \]  

where for subsequent time periods \( t = 2, \ldots, N \) we define:

\[ \psi(S(t), \psi(t, \omega_2)) = \min_{(S(t), \psi(t, \omega_2))} \sum_{j \in S} \sum_{i \in R_i} (-r_{ij} x_i(t, \omega_2) + c_{ij} y_j(t, \omega_2) + \tilde{\psi}^{t+1}(S(t + 1))) \]  

subject to:

\[ x_i(t, \omega_2) \leq \tilde{d}_i(t, \omega_2), \quad \forall i, j \]  

\[ \sum_{j \in R_i} (x_i(t, \omega_2) + y_j(t, \omega_2)) = S_i(t), \quad \forall i \]  

\[ \sum_{j \in R_i} (x_i(t, \omega_2) + y_j(t, \omega_2)) = S_i(t + 1), \quad \forall j \]  

\[ x(t, \omega_2) > 0, \quad y(t, \omega_2) > 0, \quad S(t + 1) > 0 \]  

with \( \tilde{\psi}^{t+1}(S(t + 1)) \) defined in a similar manner as in (2.2) and with \( \tilde{\psi}^{N+1}(S(N + 1)) \) equal to zero uniformly (since the future after the \( N \)th period is ignored). Note that it is assumed that the realizations of the random variables (capacity of links) for period \( t \) do not become known until the beginning of period \( t \); for this reason the \( N \)-period program is formulated as an \( N \)-stage stochastic program.

With some restrictions, the stochastic program given above solves the stochastic dynamic vehicle allocation problem. Thus, in the remainder of the paper we allow ourselves to use the expression "region \( i \) at time \( t \)" to refer to node \((i, t)\) and the term "vehicle" for the unit of flow in the described network. See [10] for a more detailed description of the DVA problem in the context of stochastic programming.

Strictly speaking, the flows \( x(t, \omega_0) \) of the links at stage \( t \) depend on the outcomes \( \omega_2, \omega_3, \ldots, \omega_{t-1} \) of the previous stages as well. Nevertheless, by the definition of the recourse problem, the flows \( x(t, \omega_0) \) are conditional on the node supplies \( S(t) \) at the beginning of stage \( t \) defined as such, they are independent of the previous history \( \omega_1, \omega_2, \ldots, \omega_{t-1} \).

3. ADAPTATION OF JENSEN'S LOWER BOUND

This section first states Jensen's inequality and then adapts Jensen's bound for the multistage stochastic programming problem (2.1) with its specific type of network recourse.

Jensen's inequality states that if a function \( F(\xi) \) is convex with respect to the random variable \( \xi \), then:

\[ E_{\xi}[F(\xi)] \geq F(\xi) \]  

The objective function of (2.3) does not depend explicitly on the random variables \( \psi(t) \) and is linear in the recourse actions \( x(t) \) and \( y(t) \). Additionally, the constraints are linear in the recourse actions and the random variables. Thus, from parametric analysis (see Rockafellar [23]), the optimal value \( \psi(S(t), \psi(t)) \) of (2.3) is convex with respect to the random variables \( \psi(t) \).

Proposition 1. \( \psi(T(1)) \geq J(T(1)) \) where:

\[ J(T(1)) = \min_{(S(t), \psi(t))} \sum_{t=1}^{N} \sum_{i \in S} \sum_{j \in R_i} (-r_{ij} x_i(t) + c_{ij} y_j(t)) \]  

subject to:

\[ x_i(1) \leq \tilde{d}_i(1), \quad \forall i, j \]  

\[ x_i(t) \leq \tilde{d}_i(t), \quad \forall i, j \]  

\[ \sum_{j \in R_i} (x_i(t) + y_j(t)) = T_i(t), \quad \forall i \]  

\[ \sum_{j \in R_i} (x_i(t) + y_j(t)) = T_i(t + 1), \quad \forall j \]  

\[ x(t) > 0, \quad y(t) > 0, \quad S(t + 1) > 0 \]  

\[ x(t) > 0, \quad y(t) > 0, \quad S(t + 1) > 0 \]  

with \( \tilde{\psi}^{t+1}(S(t + 1)) \) defined in a similar manner as in (2.2) and with \( \tilde{\psi}^{N+1}(S(N + 1)) \) equal to zero uniformly (since the future after the \( N \)th period is ignored). Note that it is assumed that the realizations of the random variables (capacity of links) for period \( t \) do not become known until the beginning of period \( t \); for this reason the \( N \)-period program is formulated as an \( N \)-stage stochastic program.

With some restrictions, the stochastic program given above solves the stochastic dynamic vehicle allocation problem. Thus, in the remainder of the paper we allow ourselves to use the expression "region \( i \) at time \( t \)" to refer to node \((i, t)\) and the term "vehicle" for the unit of flow in the described network. See [10] for a more detailed description of the DVA problem in the context of stochastic programming.
\( x(t) > 0, \ y(t) > 0, \)
\( S(t + 1) > 0, \ t = 1, \ldots , N \)

(3.1d)

**Proof.** See Appendix A.

Problem (3.1) is a multiperiod dynamic network with upper bounds \( \phi_i(t) \) on the loaded movement links. The problem can be solved using standard network codes.

## 4. A NEW LOWER BOUNDING PROCEDURE

In this section we present a new lower bounding procedure, which we refer to as the Successive Linear Approximations Lower Bounding Procedure (SLAP-LB). The algorithm performs a backward pass through the network and at each node \( (i,t) \) of the network constructs a lower bounding approximation of the expected recourse function of stage \( t \), by substituting convex functions by outer linear approximations.

### 4.1 Successive Linear Approximations Lower Bounding Procedure

We start with a rather abstract definition of the lower bounding approximate for period \( t \), as well as its immediate properties. We then describe the procedure for obtaining specific approximate functions (namely the backward recursion) and we finally prove their bounding properties.

**Definition of Functions \( \overline{V}, L \) and \( \overline{L} \)**

For stage \( t + 1 \), let \( \overline{V}^{t+1} \) be a linear function bounding the exact recourse function \( \overline{\phi}^{t+1} \) from below:

\[
\overline{V}^{t+1}(S(t + 1)) \leq \overline{V}^{t+1}(S(t + 1)) \quad \forall S(t + 1)
\]

(4.1a)

with

\[
\overline{V}^{t+1}(S(t + 1)) = \sum_{\alpha k} \overline{V}^{t+1}(S(t + 1)) \quad \forall S(t + 1)
\]

(4.1b)

where for each region \( j \) we have a linear function defined as:

\[
\overline{V}_j^{t+1}(s) = v_j(t + 1)s + b_j(t + 1), \quad s = 1, \ldots , F \]

(4.1c)

Then, for stage \( t \) define the function:

\[
L(S(t), \phi(t), a) = \min_{\{(\phi(t), a) \in \partial, (\phi(t), a) \geq 0 \}} \sum_{\alpha k} \overline{V}^{t+1}(S(t + 1))
\]

subject to:

\[
x_j(t, a) \leq \phi_j(t, a) \quad \forall j \in R \quad \forall j \in R
\]

(4.2a)

\[
\sum_{\alpha k} [x_j(t, a) + y_j(t, a)] = S_j(t) \quad \forall j \in R
\]

(4.2b)

Obviously, \( L \) depends on the choice of the components \( \overline{V}_j^{t+1} \) of the lower bounding function \( \overline{V}^{t+1} \), or (equivalently) on the choice of parameters \( v_j(t + 1) \). Let us also define:

\[
\overline{L}(S(t)) = E_{\epsilon}[L(S(t), \phi(t), a)]
\]

(4.3)

**Property I: Separability of \( L \) and \( \overline{L} \).** Problem (4.2) decomposes by region \( j \):

\[
L(S(t), \phi(t), a) = \sum_{\alpha k} L_j(S_j(t), \phi_j(t), a)
\]

(4.4a)

\[
\overline{L}(S(t)) = \sum_{\alpha k} \overline{L}_j(S_j(t))
\]

(4.4b)

where function \( L(s, \phi(t), a) \) is the optimal value of:

\[
L(s, \phi(t), a) = \min_{\{(\phi(t), a) \in \partial, (\phi(t), a) \geq 0 \}} \sum_{\alpha k} [(-r_j + v_j(t + 1))x_j(t, a) + (c_j + v_j(t + 1))y_j(t, a)]
\]

subject to:

\[
x_j(t, a) \leq \phi_j(t, a) \quad \forall j \in R
\]

(4.5a)

\[
\sum_{\alpha k} [x_j(t, a) + y_j(t, a)] = s
\]

(4.5b)

**Property II: Convexity of \( L \) and \( \overline{L} \).** From parametric analysis, \( L(s, \phi(t)) \) is convex in \( s \), \( \forall t, \phi \). Moreover, since the expectation operator is linear, \( \overline{L}(s) \) is convex in \( s \), \( \forall t \).

**Description of Backward Recursion**

Consider the recourse problem (2.3) for stage \( t = N \). Since the future after stage \( t = N + 1 \) is ignored, the second term \( \overline{\phi}^{t+1} \) of the objective, as well as constraint (2.3c), are redundant. Thus, problem (2.3) for \( t = N \) decomposes by region \( i \). Define:

\[
\psi_i^N(S(N), \phi(N), a) = \min_{\{(\phi(N), a) \in \partial, (\phi(N), a) \geq 0 \}} \sum_{\alpha k} [(-r_i + v_i(N + 1))x_i(N, a) + (c_i + v_i(N + 1))y_i(N, a)]
\]

(4.6)
subject to:

\[ x_j(N,a_{\omega_k}) \leq d_j(N,a_{\omega_k}), \quad \forall j \quad (4.6a) \]

\[ \sum_j (x_j(N,a_{\omega_k}) + y_j(N,a_{\omega_k})) = S(N) \quad (4.6b) \]

The optimal solution of this problem is clearly a greedy one; this allows us to calculate the regional expected recourse functions:

\[ \tilde{\psi}_j(S(N)) = E_x(\psi_j(S(N),d(N,a_{\omega_k}))) \quad (4.7) \]

which are clearly convex. For more details, see [9].

Essentially, the recourse problem at any stage \( t \) is complicated because of the nonlinearity \( \tilde{\psi}^{-1} \) of the objective. Its nonlinearity forces the nodes at stage \( t \) to make joint decisions on their outbound flows to achieve global optimality. If the second term \( \tilde{\psi}^{-1} \) of the objective were linear, the recourse problem (2.3) (and the recourse function \( \psi \) and the expected recourse function \( \tilde{\psi} \)) would be separable and easy to solve.

The following backward recursive scheme is introduced, in an effort to utilize the above property of the problem. At the last stage \( t = N \) and for all regions \( j \), let \( \bar{V}_j \) be a linear support of the exact expected recourse function \( \tilde{\psi}_j \) which satisfies:

\[ \bar{V}_j(S) \geq \tilde{\psi}_j(S) \quad \forall s \quad (4.8) \]

Thus, function \( \bar{V}_j \) defined by (4.1b) satisfies the lower bounding property (4.1a):

\[ \bar{V}_j(S) \geq \tilde{\psi}_j(S) \quad \forall s \quad (4.9) \]

At step \( t = N - 1 \), the functions \( \bar{V}_j \) (or specifically the slopes \( v_j(N) \)) define through (4.5) the convex approximate functions \( \tilde{L}_j^{-1} \). As we prove next, the (separable) expected approximate functions \( \tilde{L}_j^{-1} \) bound the exact recourse function \( \phi_j^{-1} \) from below.

Choosing \( \bar{V}_j^{-1} \) to be tangential linear approximations of \( \tilde{L}_j^{-1} \) guarantees the necessary properties (4.2a) and (4.2b). At step \( t = N - 2 \), use of \( \bar{V}_j^{-1} \) provides convex functions \( \tilde{L}_j^{-1} \) which again bound the exact function \( \phi_j^{-1} \) from below, and so forth.

Property III. Lower Bounding Property of \( \tilde{L} \). Under this backward recursive procedure, the lower bounding property of \( \tilde{L} \) is provided by the following proposition.

**Proposition 2.** For \( t = 2, \ldots, N - 1 \),

\[ \tilde{V}(S(t)) \leq \tilde{L}(S(t)) + \sum_{j=1}^{N} \sum_{t+1}^{N} b_j(t) \quad (4.10) \]

Proof: See Appendix B.

Property IV. Function \( \tilde{L} \) can be Calculated Numerically. The approximate function \( \tilde{L} \) is the result of substituting the expected recourse function \( \tilde{\psi}^{-1} \) by the linear function \( \tilde{V} \) (and ignoring the constant terms). Thus, similarly to the discussion of problem (4.6), the approximate function \( \tilde{L} \) can be expressed as:

\[ \tilde{L}(S(t)) = \sum_{i=1}^{R} \bar{\theta}_i(t) \quad \forall i = 1, \ldots, R, k \quad (4.11) \]

where the terms \( \bar{\theta} \) can be easily calculated, as shown in Appendix C.

**Resulting First-Stage Problem:**

This section gives a backward recursive scheme that relies on the substitution of convex functions by linearizations and preserves the property of bounding the exact recourse function from below. We should emphasize that functions \( \tilde{L}_j \) depend on the slopes \( v_j(N) \) of all of the future linear approximations, that is on all \( v_j(t) \). The procedure starts from the last period to be considered \( (t = N) \). After \( N \) - 2 backward steps, the piecewise linear, convex functions \( \tilde{L}_j \) can be presented in a network context by multiple links; thus there is no need for approximating them. The objective function becomes:

\[ \psi(T(1)) = \min_{(s(1) \in S, \ldots, s(1) \in S, \ldots, s(N) \in S)} \sum_{j \in R} \sum_{\omega_k \in \Omega} \left[ -t_j x_j(1) + c_j y_j(1) \right] \]

\[ + \sum_{j \in R} \tilde{V}_j(s(2)) = \min_{(s(1) \in S, x_j(1) \in S, y_j(1) \in S)} \sum_{j \in R} \left[ -t_j x_j(1) + c_j y_j(1) \right] \]

\[ + \sum_{j \in R} \tilde{L}_j(s(2)) + \sum_{j \in R} \sum_{t+1}^{N} b_j(t) \]

Thus, the optimal objective value \( \psi(T(1)) \) of the DVA problem is bounded from below by

\[ \psi(T(1)) \geq \tilde{L}(T(1)) + \sum_{j \in R} \sum_{t+1}^{N} b_j(t) \quad (4.12) \]

that is, the sum of the constant term \( \sum_{j \in R} \sum_{t+1}^{N} b_j(t) \) and the result \( L(T(1)) \) defined below:

\[ L(T(1)) = \min_{(s(1) \in S, x_j(1) \in S, y_j(1) \in S)} \sum_{j \in R} \sum_{t+1}^{N} \left[ -t_j x_j(1) + c_j y_j(1) \right] \]

\[ + \sum_{j \in R} \sum_{t+1}^{N} \bar{\theta}_j(t) \quad (4.13) \]
subject to:

\[ x_{ij}(1) \leq \tilde{\phi}_{ij}(1) \]  \hspace{1cm} (4.13a)

\[ \sum_{j \in R} (x_{ij}(1) + y_{ij}(1)) = T_i(1) \hspace{0.5cm} \forall i \in R \]  \hspace{1cm} (4.13b)

\[ \sum_{j \in R} (x_{ij}(1) + y_{ij}(1)) - S_j(2) = 0 \hspace{0.5cm} \forall j \in R \]  \hspace{1cm} (4.13c)

Problem (4.13) can be solved by the cost minimization of a network similar to the one depicted in Figure 2. The network includes links connecting nodes \((i, j)\) to the supersink SS; these links are called stochastic links and they represent the value of each additional vehicle in that region at that time. Thus, the stochastic links have upper bounds equal to one and (for the \(k^{th}\) link) cost \(\Theta_{ik}(2)\), as obtained in Appendix C.

In addition to a lower bound, the procedure also provides a ranking of the outbound links at all nodes of the network. We refer to this ranking as a policy and denote it by \(\delta\), to be consistent with the literature. The more successful this ranking is to represent the future dispatch decisions, the tighter the produced lower bound is expected to be. After this general description of the algorithm, in the next section we give more details on the construction of the linear approximations that are performed at each step of the algorithm.

4.2 Developing the Linear Approximations

As is apparent from the presentation of the SLAP-LB algorithm, the substitution at each node \((i, t)\) of the approximate expected recourse function \(\bar{L}_i\) by any linear support \(\bar{V}_i\) would suffice to allow the necessary backward recursion and produce the bounding property. Nevertheless, the performance of the algorithm is expected to rely heavily on the choice of the linear approximations:

\[ \bar{V}_i(s) = v_i(t)s + b_i(t) \hspace{0.5cm} \forall i \in R, \hspace{0.5cm} \forall t \geq 3 \]  \hspace{1cm} (4.14)

This section addresses this issue of the choice of parameters \(v_i(t)\) and \(b_i(t)\) that define the function \(\bar{V}_i(s)\) of the variable \(s = S_i(t)\).

It would be reasonable for this linear approximation to be fitted around an approximation of the expected vehicle supply \(s = S_i(t)\) of region \(i\) at period \(t\), \(m_i(t) = E[S_i(t)]\). That is:

\[ \bar{L}_i(m_i(t)) = \bar{V}_i(m_i(t)) = v_i(t)m_i(t) + b_i(t) \]  \hspace{1cm} (4.15a)

and

\[ v_i(t) = \frac{\delta L_i(s)}{\delta s} \bigg|_{s=m_i(t)} \]  \hspace{1cm} (4.15b)
These formulas uniquely define the parameters of the linear approximation, once $m(t)$ is defined.

One source of information for estimating $m(t)$ in advance is the forecasted frequencies of outbound and inbound loaded movements of region $i$ on a weekday corresponding to day $t$. These frequencies are typical of the operation of an actual carrier. The total outbound flow of region $i$, though, depends not only on the vehicle supply of $i$ at that day, but also on the demand for loads out of $i$; similarly for the total inbound flow. Our approach is to take a weighted average of the total inbound and outbound flow of the region at that day:

$$m(t) = \left[ q \sum_{i \in R} \delta_{ii}(t - 1) + (1 - q) \sum_{j \in R} \delta_{ji}(t) \right] K, \quad \forall t \in R, \forall t \geq 3$$

(4.16)

where $K$ is a normalizing factor that can be assumed to be equal to one for the time being. Details on the actual value of this factor are given in section 7.

Regions with high outbound and low inbound flow are expected to attract empty vehicles from other regions to satisfy their vehicle deficit. One would expect that, since the historical empty flows are not available, the estimate of $m(t)$ should depend heavily on the outbound flow: thus, a value of $q$ approaching zero would be more appropriate for these regions. Alternatively, regions with low outbound and high inbound flow are not expected to attract any empty vehicles. As a result, the total inbound flow is a good estimation of $m(t)$ and a value of $q$ approaching 1 would be appropriate for these regions. Nevertheless, in our experiments we used uniformly the value $q = 0.5$.

In our procedure, an estimate for the expected value $m(t)$ is first obtained by (4.16). Then, since the approximate expected recourse function is piecewise linear, the choice for the average value of a vehicle in region $i$ at time $t$ is the value of the $(m(t))^{th}$ vehicle in that region at that time:

$$v(t) = \hat{\delta}_{ii}(t) \quad \forall t \in R, \forall t \geq 3$$

(4.17)

This choice is depicted in Figure 3.

### 4.3 Iterative Version of SLAP-LB

In the procedure described in section 4.2, the closer the estimate $m(t)$ is to the expected internal supply $S_i(t)$ of node $(i,t)$, the better our approximation is expected to perform (producing a tighter lower bound). The estimate obtained in (4.16), while reasonable, is not necessarily close to the expected vehicle supply.

Thus, we developed an iterative procedure, which at each iteration obtains new estimates of the internal node supplies. These new linearization points provide different marginal values $v(t + i)$; this may alter the ranking of the outbound links of nodes $(i,\tau)$, $\tau \leq t$ and hence will produce a new optimal solution and a new lower bound on $\psi(T(t))$.

Initially, $m(t)$, is obtained by (4.16). Then the SLAP-LB procedure provides the recommended first-period dispatch decisions $\mathbf{x}(t)$ and $\mathbf{y}(t)$, as well as a ranking $\delta$ of the outbound links at each node of the network. Using this information, we can perform a heuristic forward pass through the network and obtain new estimates of the node supplies; these estimates can then be used in the next iteration of the backward recursion of SLAP-LB. See [9] for more details on the implementation of this iterative scheme.

### 5. A NEW UPPER BOUNDING PROCEDURE

In the previous section, the substitution of the approximate expected recourse function $L_i$ at each node $(i,t)$ by a linear function that bounds it from below provided a lower bound on the optimal total expected cost, $\psi(T(t))$. Similarly, in a procedure we refer to as SLAP-UB, substituting convex functions (which are defined later in this section) with linear functions that bound them from above produces an upper bound on $\psi(T(t))$.

**Definition of Functions $\bar{W}, U$ and $\bar{U}$:**

For stage $t + 1$, let $W^{t+1}$ be a separable linear function bounding the exact expected recourse function $\psi^{t+1}$ from above:
\[ \overline{W}^{t+1}(S(t + 1)) \geq \overline{W}^{t+1}(S(t + 1)) \forall S(t + 1) \]  \hspace{1cm} (5.1a)

with

\[ \overline{W}^{t+1}(S(t + 1)) = \sum_{j \in R} \overline{W}^{t+1}_j(S(t + 1)) \]  \hspace{1cm} (5.1b)

where for each region \( j \) we have a linear function defined as:

\[ \overline{W}^{t+1}_j(s) = w_j(t + 1)s, \quad s = 1, \ldots, F \]  \hspace{1cm} (5.1c)

Then, for stage \( t \) let us define the function:

\[ U^t(S(t), \phi(t, o_t)) = \min_{[x_i, \phi_i(t, o_t)] \in \mathcal{F}_t} \sum_{j \in R} [x_j(t, o_t) + y_j(t, o_t)] \]  \hspace{1cm} (5.2)

subject to:

\[ x_j(t, o_t) \leq \phi_j(t, o_t) \quad \forall i \in R \quad \forall j \in R \]  \hspace{1cm} (5.2a)

\[ \sum_{j \in R} [x_j(t, o_t) + y_j(t, o_t)] = S_j(t) \quad \forall i \in R \]  \hspace{1cm} (5.2b)

Obviously, \( U^t \) depends on the choice of the components \( \overline{W}^{t+1}_j \) of the upper bounding function \( \overline{W}^{t+1} \), or (equivalently) on the choice of parameters \( w_j(t + 1) \). Let us also define:

\[ \overline{U}^t(S(t)) = E_r[U^t(S(t), \phi(t, o_t))] \]

**Property I: Separability of \( U \) and \( \overline{U} \).** Problem (5.2) decomposes by region \( i \):

\[ U^t(S(t), \phi(t, o_t)) = \sum_{j \in R} U^t_j(S_j(t), \phi(t, o_t)) \]  \hspace{1cm} (5.3a)

\[ \overline{U}^t(S(t)) = \sum_{j \in R} \overline{U}^t_j(S_j(t)) \]  \hspace{1cm} (5.3b)

where function \( U_j(s, \phi(t, o_t)) \) is the optimal value of:

\[ U_j(s, \phi(t, o_t)) = \min_{[x, \phi(t, o_t)] \in \mathcal{F}_t} \sum_{j \in R} [x_j(t, o_t) + y_j(t, o_t)] \]  \hspace{1cm} (5.4)

subject to:

\[ x_j(t, o_t) \leq \phi_j(t, o_t) \quad \forall j \in R \]  \hspace{1cm} (5.4a)

\[ \sum_{j \in R} [x_j(t, o_t) + y_j(t, o_t)] = s \]  \hspace{1cm} (5.4b)

**Property II: Convexity of \( U \) and \( \overline{U} \).** From parametric analysis, \( U_j(s, \phi(t)) \) is convex in \( s \), \( \forall t, \phi \). Moreover, since the expectation operator is linear, \( \overline{U}^t(s) \) is convex in \( s \), \( \forall t \).

**Closest Inner Linearization:**

In the case of a convex function \( f(x) \), \( x \in (a,b) \) the "closest" inner linearization is defined by the points \((a,f(a))\) and \((b,f(b))\). Figure 4 shows schematically this linearization of a function where \( a = 0 \) and \( f(a) = 0 \).

**Description of Backward Recursion**

The following backward recursive scheme is introduced, similar to the one in SLAP-LB. At the last stage \( t = N \) and for all regions \( j \), let \( \overline{W}^N_j \) be the "closest" inner linearization of the exact expected recourse function \( \overline{W}^N_j \). In particular, the extreme choice for the supply \( S_j(N) \) of the node \((j, N)\) is:

\[ S_j(N) = F \]  \hspace{1cm} (5.5a)

which implies that the whole network supply \( F \) is present at node \((j, N)\). Since \( \overline{W}^N_j \) is the expected cost of the \( k \)th unit (vehicle) at node \((j, N)\), the average cost of all \( F \) units is:

\[ w_f(N) = (1/F) \sum_{k=1}^{F} \overline{\theta}_f(N) \]  \hspace{1cm} (5.5b)
Then, the convexity of the expected recourse function \( \tilde{\psi}_N(x) \) gives:

\[
\tilde{\psi}_N(x) \leq \overline{W}_N(s) = w_j(N)x, \quad \forall x = 1, \ldots, N (5.6)
\]

Then, function \( \overline{W}_N \) defined by (5.1a) and (5.6) satisfies the upper bounding property (5.1a):

\[
\overline{W}_N(S(N)) \geq \tilde{\psi}_N(S(N))
\]

At step \( t = N - 1 \), the functions \( \tilde{\psi}_N \) (or specifically of the slopes \( w_j(N) \)) define through (5.2) the convex approximate functions \( \overline{U}_N \). As stated next, the (separable) expected approximate functions \( \overline{U}_N \) bound the exact recourse function \( \tilde{\psi}_N \) from above.

Choosing \( \overline{U}_N \) to be the closest inner linear approximations of \( \overline{U}_N \) guarantees the necessary properties (5.1a) and (5.1b). At step \( t = N - 2 \), use of \( \tilde{\psi}_N \) provides convex functions \( \overline{U}_N \) which again bound the exact function \( \tilde{\psi}_N \) from above.

**Property III: Upper Bounding Property of \( \overline{U} \).** Under this backward recursive scheme, the lower bounding property of \( \overline{U} \) is provided by the following proposition.

**Proposition 3:** For \( t = 2, \ldots, N - 1 \),

\[
\tilde{\psi}(S(t)) \leq \overline{U}(S(t)) (5.7)
\]

**Proof:** By induction, similar to the proof of (4.10).

**Property IV: Function \( \overline{U} \) can be calculated numerically.** The approximate function \( \overline{U} \) is the result of substituting in (2.3) the expected recourse function \( \tilde{\psi}(x) \) by the linear function \( \overline{U} \). Similarly with (4.11) we have that the approximate function \( \overline{U} \) can be expressed as:

\[
\overline{U}(S(t)) = \sum_{k=1}^{s(t)} \lambda_k x \quad \forall i = 1, \ldots, R, k (5.8)
\]

where the terms \( \lambda_k \) are derived exactly as the terms \( \beta_k \) are in Appendix C, by using the slopes \( w(t+1) \) instead of the slopes \( w(t) \).

**Resulting First-Stage Problem:**

This section introduces a backward recursive scheme that relies on the substitution of convex functions by their "closest" inner linearizations and preserves the property of bounding the exact recourse function from above. We should emphasize that the functions \( \overline{U} \) depend on the slopes (virtually) all of the future linear approximations, that is on all \( w(\tau), \forall \tau > t \). Similarly as in SLAP-LB, when the first stage is reached we have:

\[
\psi(T(1)) \leq \overline{U}(T(1)) (5.9)
\]

where again \( \overline{U}(T(1)) \) corresponds to problem (4.13) for the extreme choice of slopes \( w(3) \). This procedure provides a rigorous upper bound on \( \psi(T(1)) \).

In Figure 5 we depict schematically how the SLAP-LB and SLAP-UB procedures bound the objective function value, across time. At stage \( N \), we show the original expected recourse function \( \tilde{\psi}_N \) for region \( j \), as well as the linear approximations that the two procedures use. Once the linear approximations of the expected recourse functions of stage \( N \) are performed, the problem of stage \( N - 1 \) can be solved parametrically (in terms of the node supplies \( S(N - 1) \)). At stage \( N - 1 \) we depict the unknown exact expected recourse func-

**Fig. 5. Bounding the expected recourse function from above and below, across time.**
tion $\tilde{N}^{-1}(S(N-1))$. We also show the approximate functions $\tilde{L}^N$ and $\tilde{U}^N$ obtained by the procedures SLAP-LB and SLAP-UB respectively. From the development of the procedures it is clear that $\tilde{N}^{-1}$ lies between the two approximate functions, as the figure indicates.

If the exact function $N^{-1}$ were known, by applying to it the two bounding procedures we would get the approximate functions $\tilde{L}^N$ and $\tilde{U}^N$ that would, once again, bound function $N^{-1}$. Instead, we apply the SLAP-LB procedure to the approximate function $\tilde{L}^N$ to obtain function $\tilde{L}^{N^{-1}}$ which is still a lower bounding function to $N^{-1}$ (but obviously looser than $\tilde{L}^N$). Similarly, we apply the SLAP-UB procedure to the approximate function $\tilde{U}^N$ to get a looser upper bounding function $\tilde{U}^{N^{-1}}$.

After $N-2$ such steps and the solution of the resulting simple network, the two bounding procedures produce upper and lower bounds to the objective value $\psi(T(1))$ of the original stochastic programming problem (2.1). It is clear from Figure 4, however, that these bounds (especially the upper bound) are not expected to be tight.

6. STATISTICAL METHODS FOR OBTAINING UPPER BOUNDS

By its construction, procedure SLAP-UB is expected to produce a loose upper bound; thus, in this section, we resort to Monte-Carlo methods to obtain a statistical estimate of an upper bound. The Monte-Carlo approach is computationally demanding, so the next section introduces an analytical approximation of the expected upper bound that can be computed far more quickly.

Consider a fixed dispatch policy $\delta$; that is, a ranking (or preference order) of the outbound links at each node of the network. Adopting such a policy establishes a one-to-one correspondence between an outcome $\phi(\omega) = (\delta(2,\omega_2), \ldots, \delta(N,\omega_N))$ of the link capacities (demands) and the link flows of the network (for given initial conditions $T(1)$ and $d(1)$).

In this section, we focus on the optimal solution of the network for an individual outcome of the link capacities, once a fixed dispatch policy $\delta$ is adopted. Thus, for the flows on the period of time $t > 1$, we appropriately refine the notation as follows:

$x(t,\omega,\delta) =$ the flow on the loaded links of period $t > 1$ that result from the use of dispatch policy $\delta$ under outcome $\omega$.

$y(t,\omega,\delta) =$ similarly, for the flows on empty links.

$S_j(t,\omega,\delta) =$ the supply of node $(j,t)$, $t > 2$ when policy $\delta$ is used under outcome $\phi(\omega)$.

$$S_j(t,\omega,\delta) = \sum_{\epsilon \in \epsilon^j} [x_j(t-1,\omega,\delta) + y_j(t-1,\omega,\delta)].$$

$$\phi(\delta)(T(1),\omega,\delta) = \text{the total cost from using } x(1), y(1) \text{ and policy } \delta \text{ for } t = 2, \ldots, N \text{ to dispatch a fleet with the initial allocation } T(1), \text{ under outcome } \omega. \text{ Then:}$$

$$\phi(\delta)(T(1),\omega,\delta) = \sum_{\epsilon \in \epsilon^j} \sum_{\epsilon \in \epsilon^j} [-r_\epsilon x_j(1) + c_\epsilon y_j(1)]$$

$$+ \sum_{\epsilon \in \epsilon^j} \sum_{\epsilon \in \epsilon^j} [-r_\epsilon x_j(t,\omega,\delta) + c_\epsilon y_j(t,\omega,\delta)].$$

$$\tilde{\phi}(T(1),\delta) = \text{the total expected cost from using policy } \delta \text{ to dispatch (for } N \text{ periods) a fleet with the initial allocation } T(1).$$

$$\tilde{\phi}(T(1),\delta) = E_\omega[\phi(\delta)(T(1),\omega,\delta)].$$

The use of the SLAP-UB algorithm provides us with first-stage actions that are typically suboptimal. Even if they were optimal, use of a typically suboptimal policy $\delta$ results in feasible suboptimal future flows. Thus:

$$\psi(\delta)(T(1)) \leq \tilde{\phi}(T(1),\delta).$$

Instead of obtaining the exact value of the expectation on the right-hand side, we use a statistical estimate of the total expected costs obtained by using policy $\delta$.

The next two subsections present two methods that provide such statistical estimates by using as fixed policy $\delta$ the ranking of the links obtained by the backward recursion of the SLAP-LB algorithm.

6.1 Monte-Carlo Simulations

This section describes the use of Monte-Carlo simulations to obtain

$$\tilde{\psi}_{MC}(T(1),\delta)$$

which is an estimate of $\tilde{\phi}(T(1),\delta)$.

To conduct Monte-Carlo simulations we randomly sample a number $L$ of realizations $\omega_1, \omega_2, \ldots, \omega_L$. First, the flows $x(1)$ and $y(1)$ are implemented. Then, the dispatch policy $\delta$ applied to each realization $\omega$ results in the flows $x(t,\omega',\delta)$ and $y(t,\omega',\delta)$ and the internal supplies $S_j(t,\omega',\delta)$. These flows yield the total costs $\psi(\omega',\delta)$ obtained by (6.2a).

From this sample of total costs, we can compute the sample mean:

$$\tilde{\psi}_{MC}(T(1),\delta,L) = \frac{1}{L} \sum_{i=1}^{L} \psi(\delta)(T(1),\omega_i,\delta).$$
and the sample standard deviation $s(T(1), \delta, L)$. Then, the standard deviation of the sample mean is:

$$s(T(1), \delta, L) = \sqrt{1/(L^2)} \cdot s(T(1), \delta, L)$$  \hspace{1cm} (6.4b)

Additionally, the sample mean of the supply of each node $(i,t)$ is:

$$S_{i,MC}(t, \delta, L) = (1/L) \sum_{l=1}^{L} S_i(t, \omega^l, \delta)$$  \hspace{1cm} (6.4c)

We performed Monte-Carlo simulations for $L = 100$ outcomes $\omega^l$. The relative error of the sample of the objective function values (that is the width of the 95% confidence interval divided by the sample mean) was on the order of 0.5%, indicating that no further simulations were necessary. Thus, $\hat{\Phi}_{MC}(T(1), \delta, L)$ is a reliable statistical estimate of $\Phi(T(1), \delta)$.

The Monte-Carlo simulations have the disadvantage of requiring substantial computational effort in order to achieve reliable estimates of the random quantities of interest. Thus we resort to an analytical procedure in which estimates can be computed more easily; this procedure is described in the following section.

### 6.2 Analytical Approximation Bound

In this section we outline an analytical procedure, which provides an estimate:

$$\tilde{\Phi}(T(1), \delta)$$

for $\Phi(T(1), \delta)$, in a computationally easier way. It uses the same dispatch policy $\delta$ as the Monte-Carlo simulations, but attempts to obtain expected flows and node supplies through analytical techniques such as probability calculation and distribution fitting.

Section 6.2.1 treats the problem of estimating the expected value and the variance of the flow on a specific link out of a node, when its internal supply is known or random with a known distribution. Section 6.2.2 presents how such a probability distribution of the node supply may be obtained and gives an overview of the analytical approximation procedure.

#### 6.2.1 Mean and Variance of the Flow on a Specific Link

Consider the outbound link of node $(i,t)$ that has rank $n$ under policy $\delta$; let its destination node be $(j,t+1)$. In this section we obtain an analytical approximation of the expected flow $E_x|\lambda(t, \omega, \delta)]$ on that link.

In this section we simplify the notation by dropping indices $i$, $j$ and $t$. Thus, let us denote:

$$S(\omega, \delta) = (S_i(t, \omega, \delta))$$

the internal supply of node $(i,t)$.

$$n_i = n_i(\delta)$$

the rank of the most attractive empty link out of node $(i,t)$.

Under the assumptions of SLAP-LB (see Appendix C), a loaded or empty link that under policy $\delta$ has rank $n > n_i$, has zero flow almost surely and zero expected flow.

Let us first consider that $n < n_i$, that is the link under consideration is a (loaded) link ranking higher than the best ranking empty link. Then, let us define:

$$x(\omega, \delta) = (x_i(t, \omega, \delta))$$

the flow of that specific link (out of node $(i,t)$) with destination node $(j,t+1)$ under outcome $\omega$, when the dispatch policy is $\delta$,

$$\Phi(\omega) = (\Phi_i(\omega))$$

the capacity of that link.

$$P_i = \text{Prob}[\Phi = k], \quad P_k = \text{Prob}[\Phi = k]$$

the total capacity of the loaded links ranking higher that $n$ under $\delta$.

$$\tilde{\phi}(\omega, \delta) = \sum_{k=1}^{n-1} \phi_k(\omega, \delta)$$

where $\phi_k(\omega, \delta) = \Phi_i(\omega)$ if, under policy $\delta$, the $k^{th}$ ranking link at node $(i,t)$ is a loaded link with destination $(j,t+1)$.

Then, by definition:

$$x(\omega, \delta) = \min\{S(\omega, \delta) - \tilde{\phi}(\omega, \delta), \Phi(\omega)\}$$  \hspace{1cm} (6.5)

Since the capacities on different links are independent, so are the two components under the minimum operator. Let us first assume that node $(i,t)$ has a deterministic vehicle supply $s$. Then:

$$\text{Prob}[x(\omega, \delta) = k | S(\delta) = s] = \text{Prob}[s - \phi(\delta) = k | \Phi(\delta) \geq k]$$  \hspace{1cm} (6.6)

and

$$\text{Prob}[x(\omega, \delta) = k | S(\delta) = s] = \tilde{P}_{s-k}(\delta)P_{s-k}$$  \hspace{1cm} (6.7)

Let us now consider $n = n_i$, that is the link under consideration is the highest ranking empty link under $\delta$. Its flow is defined as:
so that for \( k > 0 \):

\[
\text{Prob}[x(\delta) = k | S(\delta) = s] = \text{Prob}[s - \bar{\phi}(\delta) = k] = \bar{p}_s(\delta)
\]

while for \( k = 0 \):

\[
\text{Prob}[x(\delta) = 0 | S(\delta) = s] = \text{Prob}[\bar{\phi}(\delta) \geq s] = \bar{P}_s(\delta) + \bar{p}_s(\delta)
\]

The distributions of the link capacities \( \phi_n(\delta) \) with \( m > n \) (under policy \( \delta \)) are convolved to produce the distribution of the random variable \( \bar{\phi}(\delta) \). If we assume that \( \phi_n(\delta) \) follows a Poisson distribution with mean \( \bar{\delta}_n(\delta) \), then \( \bar{\phi}(\delta) \) follows a Poisson distribution with mean:

\[
E[\bar{\phi}(\omega, \delta)] = \sum_{n=1}^{m-1} \bar{\delta}_n(\delta)
\]

Thus the distribution of the link flow \( x(\omega, \delta) \) can be computed by (6.7), (6.10) and (6.11) for all outbound links of node \((i,t)\), provided that it has a deterministic internal supply \( s \). For the general case where the internal supply, \( S(\delta) = S(t, \delta) \), of node \((i,t)\) is random with a known distribution:

\[
q(\delta) = \text{Prob}[S(\delta) = s]
\]

then

\[
\text{Prob}[x(\delta) = k] = \sum_{s=0}^{\infty} q(\delta) \text{Prob}[x(\delta) = k | S(\delta) = s]
\]

(6.13)

gives the unconditional distribution for the flow on the link, while:

\[
x(\delta) = \sum_{k=0}^{\infty} k \sum_{s=0}^{\infty} q(\delta) \text{Prob}[x(\delta) = k | S(\delta) = s]
\]

(6.14)

provides the expected value of the flow \( x(\delta) \) on the outbound links of node \((i,t)\); a similar expression provides the variance of the flow.

6.2.2 Distribution of the Internal Supply of a Node

In this subsection we develop an approximation of the distribution \( q(\delta) \) of the internal supply \( S(\delta) = S(t, \delta) \), of node \((i,t)\) (that was assumed known in the previous subsection).

Under our assumption that the network consists only of links connecting nodes \((j,t-1)\) to \((i,t)\) with \( t = 2, \ldots, N - 1 \), we have that for any outcome \( \omega \):

\[
S(t, \omega, \delta) = \sum_{j \in \mathcal{K}} [x_j(t-1, \omega, \delta) + y_j(t-1, \omega, \delta)]
\]

(6.15)

and taking expectations of both sides we get:

\[
\bar{S}(t, \delta) = \sum_{j \in \mathcal{K}} [\bar{x}_j(t-1, \delta) + \bar{y}_j(t-1, \delta)]
\]

(6.16)

Approximation I: the first approximation of our method is the assumption that the flows into \((i,t)\) from different nodes \((j,t-1)\) are assumed to be independent random variables. Thus:

\[
\text{Var}[S(t, \delta)] = \sum_{j \in \mathcal{K}} \text{Var}[x_j(t-1, \delta)] + \text{Var}[y_j(t-1, \delta)]
\]

(6.17)

where the expected values and the variances of \( x_j(t, \delta) \) and \( y_j(t, \delta) \) are calculated as described in the previous subsection.

Approximation II: the second approximation of this analytical procedure is that a probability distribution is fitted on the internal supply \( S(t, \delta) \) of node \((i,t)\); this distribution should be consistent with the expected value and the variance computed in (6.16) and (6.17). We chose this distribution to be either a negative binomial or a Poisson distribution; for more details on the fittings of the distributions, see [9].

Thus, a distribution \( q(\delta) \) of the internal supply \( s(t, \delta) \) can be obtained for any node \((i,t)\) provided that (under the given policy \( \delta \)) the expected value and the variance of the flows on links emanating from nodes of the previous period are available. This motivates a forward pass through the nodes of the network.

The complete analytical approximation procedure is outlined below:

**Step 0.** Set \( t = 1 \). For each node \((i,1)\):
- dispatch on known and empty links according to the recommended actions \( x(1) \) and \( y(1) \).

**Step 1.** Set \( t = 2 \). For each node \((i,2)\):
- a) compute \( S(2) \) from (2.1c).
- b) calculate the distribution of flow on the loaded and empty links by using formulas (6.7), (6.9) or (6.10).
- c) calculate the expected value and the variance of the flow on those links.
STEP 2. Set \( t = t + 1 \). For each node \((i,t)\):

a) compute the expected value and the variance of \( S(2, \delta) \) from (6.16) and (6.17).

b) fit a Negative Binomial or a Poisson distribution to obtain \( q_\delta \).

c) calculate the distribution of flow on the loaded and empty links by using formulas (6.7), (6.9), (6.10), and (6.13).

d) calculate the expected value and the variance of the flow on those links.

STEP 3. If \( t < N \), then go to Step 2.

Else, terminate.

After the termination of this procedure, the analytical approximation of the statistical upper bound can be calculated by:

\[
\bar{\phi}_{u}(T(1), \delta) = \sum_{i} \sum_{j} \sum_{k} \left[ -r_{ij} \bar{q}_{ij}(t, \delta) + r_{ij} \bar{q}_{ij}(t, \delta) \right] \tag{6.18}
\]

where \( \bar{q}_{ij}(t, \delta) \) and \( \bar{q}_{ij}(t, \delta) \) are the expected flows on the corresponding loaded and empty options respectively that are calculated in section 6.2.1. and the subscript \( A \) indicates that they are the product of the analytical approximation procedure.

If the analytical procedure described in this section is based on valid approximations, then the bound obtained by (6.18) is expected to be within the 95% confidence interval of the sample mean \( \bar{\phi}_{u}(T(1), \delta) \). The comparison of the two methods is further discussed in section 8.1.

7. EXPERIMENTAL DESIGN

In this section, we give details about the design of our experiments and specifically about the data requirements for the methods.

Links of the network may connect nodes \((i,t)\) and \((j,t + \tau)\); then, the travel time between the regions \( i \) and \( j \) or the length of the link is equal to \( \tau_{ij} \). In our experiments, the travel times and costs of movements between regions came from the configuration of the service network of a major motor carrier.

For the distribution of the random capacity of link connecting nodes \((i,t)\) and \((j,t + \tau)\) we used actual historical data on the average frequency of loaded moves from region \( i \) to region \( j \) that depart on the weekday that corresponds to the period \( t \). These data are representative of the demand environment of a carrier with a fleet size of 680 vehicles, which we will call the base fleet size. The assumption of a weekly cycle in the random demands is neither necessary nor restrictive, but it is quite realistic for the case of the DVA problem; the data obtained by the motor carrier enforced the validity of this assumption. The capacities of the loaded links were assumed to follow a Poisson distribution with means equal to the historical average frequency of the corresponding loaded move.

The performance of the bounding procedures is expected to depend on the fleet size and the initial vehicle allocation. For example, the SLAP-UB model may yield better results for bigger fleet sizes and fewer nodes per period, since the expected internal supply of a node may be closer to the fleet size \( F \) and the approximation (5.5a) may not be so extreme. We experimented with five different vectors of external supply \( T(1) \), that differed in the total external supply (fleet size). The fleet sizes used were 100, 250, 500, 750 and 1000 units. Within each of those five experiments, all bounding procedures had the same initial allocation \( T(1) \) and the same capacities of the known links \( \phi(1) \). The demand environment was unchanged and thus the loaded links had the same expected upper bounds, \( \phi \).

Because of this discrepancy (increased fleet size for the same demands), the estimate of the internal supply of a node (in (4.16)) was scaled up or down for fleet sizes higher or lower than the base fleet size respectively. So, the normalizing factor \( K \) used in (4.16) was: \( K = F/680 \).

8. NUMERICAL RESULTS

In this section we present the results of the experiments on the performance of the bounding procedures. We first comment on the goodness-of-fit between the analytical approximation procedure and the Monte-Carlo simulations. Next we report the upper and lower bounds obtained from the introduced procedures and contrast them to the standard bounds in the literature.

8.1 Comparison of Statistical Methods

In the development of the analytical procedure, there are two key assumptions for the approximation of the node supply \( S(t, \delta, \phi) \). First, the flows on the links with the same destination are assumed to be independent random variables, and second is that the random node supply follows a negative binomial or a Poisson distribution with the corresponding mean.

The goodness-of-fit is tested by comparing the expected supply of node \((i,t)\) obtained by the analytical approach \( S_{L}(t, \delta) \) to the average supply of that node in the Monte-Carlo simulations \( S_{MC}(t, \delta, L) \).

The results of this comparison are shown in Figure 6 for fleet sizes of 100, 500 and 1000. For each fleet size and at time stages \( t = 4 \), \( t = 11 \) and \( t = 18 \), we
Fig. 6. Comparison of the node supplies obtained by the two methods.

obtain a plot of points (one point for each region) where the coordinate axes refer to the internal supply of the corresponding node obtained by the Monte-Carlo simulations and the Analytical Approximation procedure. In figure 6a, we show an example of those plots (for a fleet of 100 vehicles and time stage \( t = 4 \)), while in figure 6b we show the regression line and correlation coefficient for each case. A perfect fit between the two methods corresponds to a regression line with constant term equal to zero and with slope equal to 1 (points aligned on the 45 degrees line).

We believe that the numerical results justify our assumptions; the fit was good even for time stages well into the future. This suggests that the two central assumptions of the Analytical Approximation procedure are justified.

### 8.2 Comparison of the Upper Bounds

In Table I we report the upper bounds obtained by the Monte-Carlo simulations (\( \Phi_{1uc} \) in column a1), the analytical approximation (\( \Phi_{1} \) in column b) and by the SLAP-UB procedure (U in column c) for the different fleet sizes. Column (a2) gives the standard deviation of the mean \( \tilde{\pi}_{1}(1, \tilde{8}, L) \) of the sample of the Monte-Carlo simulations. This column, multiplied by 1.96
TABLE I. Upper Bounds obtained by the alternative procedures

<table>
<thead>
<tr>
<th>Fleet size</th>
<th>Monte-Carlo simul. (a1)</th>
<th>(a2)</th>
<th>Analyt. approx. (b)</th>
<th>SLAP-UB model (c)</th>
<th>(b)-(a)/a (%) (d)</th>
<th>(c)-(a)/a (%) (e)</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>283.6</td>
<td>0.97</td>
<td>284.3</td>
<td>178.3</td>
<td>+0.2</td>
<td>37.1</td>
</tr>
<tr>
<td>250</td>
<td>641.2</td>
<td>2.00</td>
<td>1146.4</td>
<td>313.2</td>
<td>+0.5</td>
<td>51.1</td>
</tr>
<tr>
<td>500</td>
<td>1452.2</td>
<td>2.71</td>
<td>1314.0</td>
<td>518.5</td>
<td>+0.2</td>
<td>60.5</td>
</tr>
<tr>
<td>750</td>
<td>1311.3</td>
<td>3.02</td>
<td>1341.0</td>
<td>553.9</td>
<td>+0.4</td>
<td>58.5</td>
</tr>
</tbody>
</table>

This gives the width of the 95% confidence interval of the mean.

The rigorous upper bound obtained by SLAP-UB is admittedly quite loose, as expected. Its difference from the results of the Monte-Carlo simulations is shown in column (c) and ranges from 37.1% to 51.1%. A reason for the size of this deviation is in the choice of the extreme point $S(t) = F$ used for the "closest" inner linearization. Even if the total internal supply of the nodes $(i,t)$ of the time stage $t$ is equal to the total fleet size $F$, it is unreasonable to assume that all the units may be concentrated in a single region $i$: the higher the number $R$ of nodes in each stage is, the more unreasonable this extreme point becomes. Additionally, since the network contains links of length greater than one, the total internal supply of nodes $(i,t)$ of the time stage $t$ is not equal to the fleet size; the longer the average length of the links of the network, the looser the upper bound is expected to be.

The difference between the performance of the costly Monte-Carlo simulations and the analytical approximation is shown in column (d). For any fleet size, the bound obtained by the analytical approximation is slightly higher (0.1% to 0.5%) than the one obtained by the Monte-Carlo simulations but is always within the 95% confidence interval. Thus, we can claim that the two methods yield essentially equivalent upper bounds.

In Table III we present typical CPU times on a MICROVAX-2 for the alternative models; the analytical approximation required about 5% of the computational effort of the Monte-Carlo simulations.

8.3 Comparison of the Lower Bounds

In Table II we present (for the different fleet sizes) the lower bounds obtained by Jensen's inequality (a), the SLAP-LB procedure (b) and by the iterative SLAP-LB procedure (c1). In column (c2) we report the iteration that produced the best lower bound.

The comparison between the results of SLAP-LB and Jensen's bound is reported in column (d). The new procedure performed better for very small fleet sizes (2.3% for 100), while it did worse for larger fleet sizes (3.5% for 500, up to 13.9% for 1000). The iterative SLAP-LB procedure performed slightly better than SLAP-LB: it outperformed Jensen's for fleet sizes of up to 250 units (4.2% for 100 and 1.4% for 250), while when it did worse the percentages were smaller (1.3% for 500, up to 11.7% for 1000).

Table III reports typical run times on a MICROVAX-2. For our particular problem with $R = 60$ regions and a planning horizon of $N = 21$ time periods, the network for the Jensen's bound has 1260 nodes and 82,065 links, is created in about 2 minutes and is optimized by a standard network simplex routine in 27 minutes.

TABLE II. Lower Bounds obtained by the alternative procedures

<table>
<thead>
<tr>
<th>Fleet size</th>
<th>Jensen bound (a)</th>
<th>SLAP-LB model (b)</th>
<th>ITER. SLAP-LB (5 iter.) (c)</th>
<th>(b)-(a)/a (%) (d)</th>
<th>(c)-(a)/a (%) (e)</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>-360.3</td>
<td>-352.0</td>
<td>-345.1</td>
<td>1.1%</td>
<td>2.3%</td>
</tr>
<tr>
<td>250</td>
<td>-807.3</td>
<td>-813.5</td>
<td>-796.5</td>
<td>1.3%</td>
<td>0.8%</td>
</tr>
<tr>
<td>500</td>
<td>-1408.7</td>
<td>-1458.0</td>
<td>-1427.6</td>
<td>1.3%</td>
<td>+3.5%</td>
</tr>
<tr>
<td>750</td>
<td>-1833.3</td>
<td>-1967.4</td>
<td>-1932.7</td>
<td>1.3%</td>
<td>+3.3%</td>
</tr>
<tr>
<td>1000</td>
<td>-1907.2</td>
<td>-2354.6</td>
<td>-2308.2</td>
<td>1.3%</td>
<td>+13.9%</td>
</tr>
</tbody>
</table>
minutes. The data for our experiments excluded quite a few links; a complete network with links of length one would have $2^60^60^21 = 151,200$ links.

On the other hand, the SLAP-LB network has 7,175 links, is created in less than a minute and is optimized in 1 minute by the same network simplex routine. Nevertheless, the overhead of the backward pass is about 80% of the total time required for the complete SLAP-LB procedure. If the data included links between all pairs of nodes the effect on the network would be much smaller than it is on Jensen network. The SLAP-LB procedure is faster than the Jensen’s bounding procedure by a factor of 2.9 and this comparison may be accentuated in larger networks. The iterative SLAP-LB procedure takes (per iteration) 30% more CPU time than a single SLAP-LB procedure because of the forward pass that updates the estimates of the node supplies.

In conclusion, the contribution of this research is the introduction of a new class of bounding procedures for stochastic dynamic networks with random arc capacities (or, equivalently for the N-stage stochastic programming problem with a special type of network recourse). These procedures rely on a backward pass over the network and at each stage $t$ the approximate expected recourse functions of the nodes of the next time period are bounded above and below by some linear approximations. The work presented in this paper is an extension of earlier work by Frantzeskakis and Powell [10] in the area of dynamic networks with random arc capacities. In that paper, the authors also rely on a backward recursion over the network but they focus instead on the quality of the first-stage recommended actions obtained by the procedure and perform experiments with a Rolling Horizon Procedure to contrast those actions to the ones recommended by other models. Instead, in this paper, we focus on estimating whether the representation of the future inside the model is satisfactory; essentially, we obtain bounds on how well or bad our model can perform in a random environment.

The introduction of the bounding approximate expected recourse functions facilitates the decomposition by node of the problem of stage $t$ and allows the computation of the approximate expected recourse function of the nodes of that period. In both the case of the lower and the upper bound, the backward pass results in a small network, whose optimal solution bounds the objective function value of the N-stage stochastic programming problem. The bounding procedures do not necessarily perform better than the ones existing in the literature. Nevertheless, they provide rigorous upper and lower bounds and constitute a different approach to the problem. Moreover, there is a lot of fine-tuning still to be done, since certain values of parameters (e.g., in (4,16), for example) as well as functional forms, such as (4,16) and the step-size rule, were set by intuition and not after an extensive experimental search.

Additionally, in the process of obtaining upper bounds we develop and validate a new analytical approximation procedure which yields equally strict statistical estimates of an upper bound as the Monte-Carlo simulations with far less computational effort.

The authors would like to acknowledge the helpful comments of three anonymous referees in improving the presentation of the paper. This research is supported by the National Science Foundation under Presidential Young Investigator Award, grant no. CES-8451466 and DDM 910-2134.

Appendix A: Proof of Proposition 1.

Since the recourse function $\psi(S(t),d(t))$ is convex with respect to the random variables $d(t)$, Jensen’s inequality for stage $N$ gives:

$$\psi^N(S(N)) \geq \psi^N(S(N), \tilde{d}(N)) \quad (A.1)$$

Then, the objective function of the optimization problem for period $N - 1$ for a given realization satisfies the following inequality:

$$\psi^{N-1}(S(N-1), d(N-1))$$

$$= \min_{(s_{N-1}, \lambda_{N-1})} \sum_{t \in R} \sum_{x_{t} \in R} (-t_{x}x_{t}(N-1))$$

$$+ \phi_{S}(N-1) + \phi_{\tilde{d}}(S(N))$$

$$\geq \min_{(s_{N-1}, \lambda_{N-1})} \sum_{t \in R} \sum_{x_{t} \in R} (-t_{x}x_{t}(N-1))$$

$$+ \phi_{S}(N-1) + \psi^N(S(N), \tilde{d}(N))$$

subject to the constraints (2.3a), (2.3b), (2.3c) for $t = N-1$.

But we have by definition that:
\[
\psi^N(S(N), \overline{V}(N)) = \min_{(x(N), s(N))} \sum_{i \in R} \sum_{j \in R} [-r_{ij} x_{ij}(N) + c_{ij} y_{ij}(N)]
\]  
subject to:

\[
\begin{align*}
\psi_{ij}(N) &\leq \phi_{ij}(N), \quad \forall i \in R \quad (A.3a) \\
\sum_{j \in R} (x_{ij}(N) + y_{ij}(N)) &= S_i(N), \quad \forall i \in R \quad (A.3b)
\end{align*}
\]

Combining (A.2) and (A.3):

\[
\psi^{-1}(S(N-1), \phi(N-1)) = \min_{\psi_{Y^{-1}}(\phi)} \sum_{i \in R} \sum_{j \in R} [-r_{ij} x_{ij}(N-1) + c_{ij} y_{ij}(N)]
\]

subject to:

\[
\begin{align*}
x_{ij}(N-1) &\leq \phi_{ij}(N-1), \quad \forall i \in R \quad (A.4a) \\
x_{ij}(N-1) &\leq \overline{\phi}_{ij}(N-1), \quad \forall i \in R \quad (A.4b) \\
\sum_{j \in R} (x_{ij}(N-1) + y_{ij}(N-1)) &= s_i(N-1), \quad \forall i \in R \quad (A.4c) \\
\sum_{i \in R} (x_{ij}(N-1) + y_{ij}(N-1)) &= \sum_{i \in R} x_{ij}(N-1) + y_{ij}(N-1) \leq \overline{\phi}(N-1), \quad \forall j \in R \quad (A.4d)
\end{align*}
\]

where \(XY^{-1} = \{x(N-1), y(N-1), s(N), \phi(N-1)\}\). By telescoping, we derive (3.1), that is the optimal value \(\psi(t(1))\) of problem (2.1) is bounded by the result of the cost minimization of a possibly large network with link capacities \(\overline{\phi}_{ij}(t)\), \(\forall i \in R\), \(\forall j \in R\) and with external supply \(T(t)\) at the first period nodes \((i, 1)\).

**Appendix B: Proof of Proposition 2.**

By induction; let us first prove it for \(t = N - 1\):

\[
\begin{align*}
\psi^{N-1}(S(N-1), \phi(N-1)) &= \min_{(x(N-1), s(N-1))} \sum_{i \in R} \sum_{j \in R} [-r_{ij} x_{ij}(N-1) + c_{ij} y_{ij}(N-1)] + \overline{\psi}(S(N)) \\
\text{subject to (2.3a), (2.3b) and (2.3c) for } t = N - 1
\end{align*}
\]

We intend to prove it for stage \(t\):

\[
\begin{align*}
\psi(t, \phi(t)) &= \min_{(x(t), s(t))} \sum_{i \in R} \sum_{j \in R} [-r_{ij} x_{ij}(t) + c_{ij} y_{ij}(t)] + \overline{\psi}(S(t))
\end{align*}
\]

subject to (2.3a), (2.3b) and (2.3c)

\[
\begin{align*}
\psi^{t+1}(S(t + 1)) &= \min_{(x(t+1), s(t+1))} \sum_{i \in R} \sum_{j \in R} [-r_{ij} x_{ij}(t) + c_{ij} y_{ij}(t)] + \overline{\psi}(S(t + 1)) \\
\text{subject to (2.3a), (2.3b) and (2.3c)}
\end{align*}
\]
subject to (2.3a), (2.3b) and (2.3c)

\[ \begin{align*}
&\geq \min \sum_{t=0}^{\infty} \sum_{s=0}^{\infty} (-r_0 s_0(t) + c_0 y_0(t)) \\
&+ \sum_{j=0}^{J-1} \mathbf{V}_j S(t+1) + \sum_{j=0}^{J-1} b_j(t) \tag{B.2c}
\end{align*} \]

subject to (2.3a), (2.3b) and (2.3c)

\[ \begin{align*}
&\geq \min \sum_{t=0}^{\infty} \sum_{s=0}^{\infty} (-r_0 s_0(t) + c_0 y_0(t)) \\
&+ \sum_{j=0}^{J-1} [v_j(t+1) \sum_{i=0}^{N} (x_i(t) + y_j(t)) + b_j(t+1)] \\
&+ \sum_{i=0}^{N} b_i(t) \tag{B.2f}
\end{align*} \]

subject to (2.3a) and (2.3b)

\[ L S(t), \phi(t) + \sum_{t=0}^{\infty} \sum_{s=0}^{\infty} b(s(t) \tag{B.2g} \]

where in (B.2b) we used the statement (B.2a) of the proposition for \( t + 1 \), while in (B.2c) and (B.2d) we used the separability and convexity of \( L \). In (B.2c) we used the definition of functions \( \mathbf{V}_j S \) and (2.3c), while in (B.2f) we collected terms. The last term of (B.2g) is the collection of all the constant terms of the linearizations performed in the later periods that are ignored from the perspective of finding the optimal solution of \( L \) at each stage \( t \).

Appendix C: Exhibition of Property IV.

The structure of the network implies that there are at most 2R links emanating from node \((i,t)\). For the link connecting nodes \((i,t)\) to \((j,t+1)\) we define the approximate (or exact for \( t = N \) link contribution) as the sum of the link's direct cost \((-r_0 \text{ or } c_0 \text{ for loaded or empty links correspondingly}) \) plus the (negative) term \( v(t+1) \) of the link's destination. We then rank the links emanating from \((i,t)\) in a decreasing order of their contribution and define:

\[ g(t,n) = \text{the contribution of the } n^{th} \text{ ranked} \]

\[ = -r_0 + v(t+1) \text{ if the } n^{th} \text{ ranked link is a} \]

\[ = c_0 + v(t+1) \text{ if the } n^{th} \text{ ranked link is an} \]

\[ = \text{empty link to node } (j,t+1). \]

The optimal solution to the problem (4.2) of finding \( L(k,\phi(t),n_0) \) of node \((i,t)\) with supply \( S_i(t) = k \) is obvious. Each unit \( k \) in node \((i,t)\) is moved over the highest ranked link \( n \) with residual capacity (under \( \phi(t),n_0 \)) after the first \( k - 1 \) units are dispatched. In the presence of randomness in the link capacities, let us now introduce:

\[ d_i(t,n) = \text{the probability that the } k^{th} \text{ unit in } (i,t) \]

\[ \text{will take the } n^{th} \text{ ranked link}. \]

\[ \Theta_i(t) = \text{approximate (or exact for } t = N) \text{ expected} \]

\[ \text{cost of the } k^{th} \text{ vehicle in } i \text{ at time } t. \]

Since the unbounded holding (inventory) links have zero cost, the \( \Theta_i(t) \) are nonpositive quantities.

These dispatch probabilities \( d_i(t,n) \) for node \((i,t)\) can be derived as described in Appendix of [10], as long as distributions of the random link capacities are provided and the independence assumptions hold. Note that the highest ranking empty (unbounded) link (with rank \( n_j \)) is used by all the units remaining after the dispatch on higher ranking (\( n \leq n_j \)) links is completed. Then, the (approximate) expected cost of the \( k^{th} \) supply unit in \((i,t)\) is given by:

\[ \Theta_i(t) = \sum_{n=1}^{2R} g(t,n) d_i(t,n) \tag{C.1} \]

and

\[ L(S_i(t)) = \sum_{i=1}^{2R} \Theta_i(t) \quad \forall i = 1, \ldots, R_k \tag{C.2} \]

REFERENCES


Received August 1989
Accepted September 1992