Approximate Dynamic Programming Methods for an Inventory Allocation Problem under Uncertainty

Huseyin Topaloglu
School of Operations Research and Industrial Engineering,
Cornell University, Ithaca, NY 14853, USA
topaloglu@orie.cornell.edu

Sumit Kunnumkal
School of Operations Research and Industrial Engineering,
Cornell University, Ithaca, NY 14853, USA

April 15, 2006

Abstract

We propose two approximate dynamic programming methods to optimize the distribution operations of a company manufacturing a certain product at multiple production plants and shipping it to different customer locations for sale. We begin by formulating the problem as a dynamic program. Our first approximate dynamic programming method uses a linear approximation of the value function and computes the parameters of this approximation by using the linear programming representation of the dynamic program. Our second method relaxes the constraints that link the decisions for different production plants. Consequently, the dynamic program decomposes by the production plants. Computational experiments show that the proposed methods are computationally attractive, and in particular, the second method performs significantly better than standard benchmarks.

Supply chain systems with multiple production plants provide protection against demand uncertainty and opportunities for production smoothing by allowing the demand at a particular customer location to be satisfied by different production plants. However, managing these types of supply chains requires careful planning. When planning the distribution of products to the customer locations, one has to consider many factors, such as the current inventory levels, forecasts of future production quantities and forecasts of customer demands. The decisions for different production plants interact and a decision that maximizes the immediate benefit is not necessarily the "best" decision.

In this paper, we consider the distribution operations of a company manufacturing a certain product at multiple production plants and shipping it to different customer locations for sale. A certain amount of production, which is not a decision, occurs at the production plants at the beginning of each time period. Before observing the realization of the demand, the company decides how much product should be shipped to the customer locations. Once a certain amount of product is shipped to a particular customer location, revenue is earned on the sales and shortage cost is incurred on the unsatisfied demand. The product cannot be stored at the customer locations and the left over product at the customer locations is disposed, possibly at a salvage value. The left over product at the production plants is stored until the next time period.

Our work is motivated by the distribution operations of a company processing fresh produce that will eventually be sold at local markets. These markets are set up outdoors for short periods of time, prohibiting the storage of the perishable fresh produce. However, the processing plants are equipped with storage facilities. Depending on the yield of fresh produce, the production quantities at the processing plants fluctuate over time and are not necessarily deterministic.

In this paper, we formulate the problem as a dynamic program and propose two approximate dynamic programming methods. The first method uses a linear approximation of the value function whose parameters are computed by using the linear programming representation of the dynamic program. The second method uses Lagrangian relaxation to relax the constraints that link the decisions for different production plants. As a result of this relaxation, the dynamic program decomposes by the production plants and we concentrate on one production plant at a time.

Our approach builds on previous research. Hawkins (2003) proposes a Lagrangian relaxation method applicable to dynamic programs in which the evolutions of the different components of the state variable are affected by different types of decisions, but these different types of decisions interact through a set of linking constraints. More recently, Adelman & Mersereau (2004) compare the Lagrangian relaxation method of Hawkins (2003) with an approximate dynamic programming method that uses a separable approximation of the value function. The parameters of the separable approximation are computed by using the linear programming representation of the dynamic program. When applied to the inventory allocation problem described above, both of these methods run into computational difficulties. For example, the Lagrangian relaxation method of Hawkins (2003) requires finding a "good" set of Lagrange multipliers by minimizing the so-called dual function. One way of doing this is to solve a linear program, but the number of constraints in this linear program is very large for our problem class. We use constraint generation to iteratively construct the constraints of this linear program, and show that this can be done efficiently because constructing a constraint requires simple sort operations. Another way of finding a "good" set of Lagrange multipliers is to use Benders decomposition to represent the dual function by using a number of cutting planes. We show that we can keep the number of cutting planes at a manageable level by using results from the two-stage stochastic programming literature and constructing a cutting plane requires simple sort operations. The approximate dynamic programming method of Adelman & Mersereau (2004) computes the parameters of the separable value function approximation by solving a linear program whose number of constraints is very large for our problem class. We use constraint generation to iteratively construct the constraints of this linear program and show that constructing a constraint requires solving a min-cost network flow problem. Finally, we show that the value function approximations obtained by the two methods are computationally attractive in the sense that applying the greedy policies characterized by them requires solving min-cost network flow problems.

The approximate dynamic programming field has been active within the past two decades. Most of the work attempts to approximate the value function $V(\cdot)$ by a function of the form $\sum_{k \in \mathcal{K}} \alpha_k \, V_k(\cdot)$, where $\{V_k(\cdot) : k \in \mathcal{K}\}$ are fixed basis functions and $\{\alpha_k : k \in \mathcal{K}\}$ are adjustable parameters. The challenge is to find parameter values $\{\hat{\alpha}_k : k \in \mathcal{K}\}$ such that $\sum_{k \in \mathcal{K}} \hat{\alpha}_k \, V_k(\cdot)$ is a "good" approximation of $V(\cdot)$. Temporal differences and Q-learning use sampled trajectories of the system to find "good" parameter values (Bertsekas & Tsitsiklis (1996)). On the other hand, linear programming-based methods find "good" parameter values by solving a linear program (Schweitzer & Seidmann (1985), de Farias & Van Roy (2003)). Since this linear program contains one constraint for every state-decision pair, it can be very large and is usually solved approximately. Numerous successful applications of approximate dynamic programming appeared in inventory routing (Kleywegt, Nori & Savelsbergh (2002), Adelman (2004)), dynamic fleet management (Powell & Carvalho (1998), Godfrey & Powell (2002), Topaloglu & Powell (2006)), revenue management (Adelman (2005)), marketing (Bertsimas & Mersereau (2005)) and resource allocation under incomplete information (Yost & Washburn (2000)). Of particular interest to us are the papers by Bertsimas & Mersereau (2005), Yost & Washburn (2000) and Adelman (2005). The first two of these papers are applications of the Lagrangian relaxation method of Hawkins (2003), whereas the third one is an application of the linear programming-based method of Adelman & Mersereau (2004).

The literature on inventory allocation under uncertainty is rich and there exist a variety of approaches that are quite different than ours. We refer the reader to Graves, Kan & Zipkin (1981), Tayur, Ganeshan & Magazine (1998) and Zipkin (2000) for detailed treatments of various approaches. Two particularly interesting papers that emerge from this literature are Karmarkar (1981) and Karmarkar (1987), where the author characterizes the form of the optimal policy and give bounds on the optimal objective value for a multiple-period multiple-location newsvendor problem that is similar to the problem considered in this paper.

In this paper, we make the following research contributions. We propose two approximate dynamic programming methods for a stochastic nonstationary multiple-plant multiple-customer inventory allocation problem. Our first method uses a linear approximation of the value function and computes the parameters of this approximation by using the linear programming representation of the dynamic program. We show how to solve this linear program efficiently by using constraint generation. This is one of the few nontrivial linear programming-based approximate dynamic programming methods where the underlying linear program can be solved efficiently. Our second method uses Lagrangian relaxation to relax the constraints that link the decisions for different production plants. We propose two approaches for minimizing the dual function, one of which is based on constraint generation and the other one is based on Benders decomposition. We show that constructing a constraint in constraint generation or constructing a cutting plane in Benders decomposition requires simple sort operations. Finally, we show that the greedy policies obtained by these two approximate dynamic programming methods can be applied by solving min-cost network flow problems. Computational experiments show that our methods yield high-quality solutions. They also provide insights into the conditions that render stochastic models more effective than deterministic ones.

The paper is organized as follows. Section 1 describes the problem and formulates it as a dynamic program. Section 2 describes our first solution method that uses a linear approximation of the value function, whereas Section 3 describes our second solution method that uses Lagrangian relaxation. Section 4 shows that applying the greedy policies obtained by either one of these two solution methods requires solving min-cost network flow problems. Section 5 presents our computational experiments.

1. Problem Formulation

There is a set of plants producing a certain product to satisfy the demand occurring at a set of customer locations. At the beginning of each time period, a certain amount of production occurs at each plant. Due to the randomness in the yield of the production processes, we assume that the production quantities at the plants are random. Before observing the demand at the customer locations, we have to decide how much product to ship from each plant to each customer location.

After shipping the product to the customer locations, we observe the demand. The unsatisfied demand is lost. The left over product at the customer locations is disposed at a salvage value, but the plants can store the product. Our objective is to maximize the total expected profit over a finite horizon. We define the following.

$\mathcal{T} =$ Set of time periods in the planning horizon. We have $\mathcal{T} = \{1, \dots, \tau\}$ for finite τ.

$\mathcal{P} =$ Set of plants.

$\mathcal{C} =$ Set of customer locations.

$c_{ijt} =$ Cost of shipping one unit of product from plant i to customer location j at time period t.

$\rho_{jt} =$ Revenue per unit of product sold at customer location j at time period t.

$\sigma_{jt} =$ Salvage value per unit of unsold product at customer location j at time period t.

$\pi_{jt} =$ Shortage cost of not being able to satisfy a unit of demand at customer location j at time period t.

$h_{it} =$ Holding cost per unit of product held at plant i at time period t.

$Q_{it} =$ Random variable representing the production at plant i at time period t.

$D_{jt} =$ Random variable representing the demand at customer location j at time period t.

We assume that $\rho_{jt} + \pi_{jt} \geq \sigma_{jt}$, the production and demand occur in discrete quantities, and the production and demand random variables for different plants, different customer locations and different time periods are independent of each other and have finite supports. We let U be an upper bound on the production random variables $\{Q_{it} : i \in \mathcal{P},\ t \in \mathcal{T}\}$ and S be an upper bound on the demand random variables $\{D_{jt} : j \in \mathcal{C},\ t \in \mathcal{T}\}$. We define the following decision variables.

$u_{ijt} =$ Amount of product shipped from plant i to customer location j at time period t.

$w_{jt} =$ Total amount of product shipped to customer location j at time period t. That is, we have $w_{jt} = \sum_{i \in \mathcal{P}} u_{ijt}$.

$x_{it} =$ Amount of product held at plant i at time period t.

$r_{it} =$ Beginning inventory at plant i at time period t after observing the production.

We assume that the shipments occur in discrete quantities and the shipment costs are incurred on a per-unit basis. In particular, we do not consider the economies of scale. Furthermore, we assume that we are not allowed to hold more than L units of product at any plant. Therefore, since the production random variables are bounded by U, the inventory at any plant is bounded by $U+L$. We emphasize that the production quantities are modeled as exogenous random variables for ease of notation. We can easily model the production quantities as decision variables by letting $\{q_{it} : i \in \mathcal{P},\ t \in \mathcal{T}\}$ be the decision variables representing the production quantities and $\{\xi_{it} : i \in \mathcal{P},\ t \in \mathcal{T}\}$ be the random variables representing the yield of the production processes. In this case, we need to include the production costs in the list of cost parameters and the production decision variables in the list of decision variables above. The results in the paper continue to hold without almost any modifications.

By suppressing one or more of the indices in the variables defined above, we denote a vector composed of the components ranging over the suppressed indices. For example, we have $Q_t = \{Q_{it} : i \in \mathcal{P}\}$, $r_t = \{r_{it} : i \in \mathcal{P}\}$. We do not distinguish between row and column vectors. We denote the cardinality of set $\mathcal{A}$ by $|\mathcal{A}|$. In the remainder of this section, we define the one-period expected profit function and formulate the problem as a dynamic program.

1.1. One-period expected profit function

If the amount of product shipped to customer location j at time period t is w_{jt} and the demand is D_{jt}, then the obtained profit is

$$F_{jt}(w_{jt}, D_{jt}) = \rho_{jt} \min\{w_{jt}, D_{jt}\} + \sigma_{jt} \max\{w_{jt} - D_{jt}, 0\} - \pi_{jt} \max\{D_{jt} - w_{jt}, 0\}.$$

Letting $F_{jt}(w_{jt}) = \mathbb{E}\big\{F_{jt}(w_{jt}, D_{jt})\big\}$, $F_{jt}(w_{jt})$ is the expected profit obtained at time period t by shipping w_{jt} units of product to customer location j. Using the fact that the random variable D_{jt} takes integer values and $\rho_{jt} + \pi_{jt} \geq \sigma_{jt}$, we can show that $F_{jt}(\cdot)$ is a piecewise-linear concave function with points nondifferentiability being a subset of positive integers. In this case, $F_{jt}(\cdot)$ can be characterized by specifying $F_{jt}(0) = -\pi_{jt}\, \mathbb{E}\big\{D_{jt}\big\}$ and the first differences $F_{jt}(w_{jt}+1) - F_{jt}(w_{jt})$ for all $w_{jt} = 0, 1, \ldots$. The latter can be computed by noting that

$$F_{jt}(w_{jt}+1, D_{jt}) - F_{jt}(w_{jt}, D_{jt}) = \begin{cases} \sigma_{jt} & \text{if } D_{jt} \leq w_{jt} \\ \rho_{jt} + \pi_{jt} & \text{otherwise,} \end{cases}$$

which implies that

$$\begin{aligned} F_{jt}(w_{jt}+1) - F_{jt}(w_{jt}) &= \mathbb{E}\big\{F_{jt}(w_{jt}+1, D_{jt}) - F_{jt}(w_{jt}, D_{jt})\big\} \\ &= \sigma_{jt}\, \mathbb{P}\{D_{jt} \leq w_{jt}\} + \big[\rho_{jt} + \pi_{jt}\big]\, \mathbb{P}\{D_{jt} \geq w_{jt} + 1\}. \end{aligned} \tag{1}$$

Since the random variable D_{jt} is bounded by S, we have $F_{jt}(s+1) - F_{jt}(s) = \sigma_{jt}$ for all $s = S, S+1, \ldots$ by (1). Therefore, letting $f_{jst} = F_{jt}(s+1) - F_{jt}(s)$, we can characterize $F_{jt}(\cdot)$ by specifying $F_{jt}(0)$ and f_{jst} for all $s = 0, \ldots, S$.

1.2. Dynamic programming formulation

Using r_t as the state variable at time period t, we can formulate the problem as a dynamic program. Since the inventory at any plant is bounded by $U+L$, letting $R = U + L$ and $\mathcal{R} = \{0, \ldots, R\}$, we use $\mathcal{R}^{|\mathcal{P}|}$ as the state space. In this case, the optimal policy can be found by computing the value functions through the optimality equation

$$
\begin{aligned}
V_t(r_t) = \max \quad & -\sum_{i\in\mathcal{P}}\sum_{j\in\mathcal{C}} c_{ijt}\, u_{ijt} + \sum_{j\in\mathcal{C}} F_{jt}(w_{jt}) - \sum_{j\in\mathcal{C}} F_{jt}(0) - \sum_{i\in\mathcal{P}} h_{it}\, x_{it} + \mathbb{E}\big\{V_{t+1}(x_t + Q_{t+1})\big\} && (2)\\
\text{subject to} \quad & x_{it} + \sum_{j\in\mathcal{C}} u_{ijt} = r_{it} \qquad \forall\, i \in \mathcal{P} && (3)\\
& \sum_{i\in\mathcal{P}} u_{ijt} - w_{jt} = 0 \qquad \forall\, j \in \mathcal{C} && (4)\\
& x_{it} \le L \qquad \forall\, i \in \mathcal{P} && (5)\\
& u_{ijt},\, w_{jt},\, x_{it} \in \mathbb{Z}_+ \qquad \forall\, i \in \mathcal{P},\ j \in \mathcal{C}. && (6)
\end{aligned}
$$

The reason that we subtract the constant $\sum_{j\in\mathcal{C}} F_{jt}(0)$ from the objective function will be clear in the proof of Lemma 1 below. Constraints (3) are the product availability constraints at the plants, whereas constraints (4) keep track of the amount of product shipped to each customer location. Lemma 1, which we prove in the appendix, gives an alternative representation of problem (2)-(6).

Lemma 1. *The optimality equation*

$$
\begin{aligned}
V_t(r_t) = \max \quad & \sum_{i\in\mathcal{P}}\sum_{j\in\mathcal{C}}\sum_{s=0}^{S} [f_{jst} - c_{ijt}]\, y_{ijst} - \sum_{i\in\mathcal{P}} h_{it}\, x_{it} + \mathbb{E}\big\{V_{t+1}(x_t + Q_{t+1})\big\} && (7)\\
\text{subject to} \quad & x_{it} + \sum_{j\in\mathcal{C}}\sum_{s=0}^{S} y_{ijst} = r_{it} \qquad \forall\, i \in \mathcal{P} && (8)\\
& \sum_{i\in\mathcal{P}} y_{ijst} \le 1 \qquad \forall\, j \in \mathcal{C},\ s = 0,\ldots,S-1 && (9)\\
& x_{it} \le L \qquad \forall\, i \in \mathcal{P} && (10)\\
& x_{it},\, y_{ijst} \in \mathbb{Z}_+ \qquad \forall\, i \in \mathcal{P},\ j \in \mathcal{C},\ s = 0,\ldots,S && (11)
\end{aligned}
$$

is equivalent to the optimality equation in (2)-(6)*. In particular, if* (x_t^*, y_t^*) *is an optimal solution to problem* (7)-(11)*, and we let* $u_{ijt}^* = \sum_{s=0}^{S} y_{ijst}^*$ *and* $w_{jt}^* = \sum_{i\in\mathcal{P}}\sum_{s=0}^{S} y_{ijst}^*$ *for all* $i \in \mathcal{P}$, $j \in \mathcal{C}$*, then* (u_t^*, w_t^*, x_t^*) *is an optimal solution to problem* (2)-(6)*. Furthermore, the optimal objective values of problems* (2)-(6) *and* (7)-(11) *are equal to each other.*

Throughout the paper, we use the optimality equation in (7)-(11). Letting $\mathcal{Y}(r_t)$ be the set of feasible solutions to problem (7)-(11) and

$$
p_t(x_t, y_t) = \sum_{i\in\mathcal{P}}\sum_{j\in\mathcal{C}}\sum_{s=0}^{S} [f_{jst} - c_{ijt}]\, y_{ijst} - \sum_{i\in\mathcal{P}} h_{it}\, x_{it},
$$

this optimality equation can succinctly be written as

$$
V_t(r_t) = \max_{(x_t,y_t)\in\mathcal{Y}(r_t)} \Big\{ p_t(x_t, y_t) + \mathbb{E}\big\{V_{t+1}(x_t + Q_{t+1})\big\} \Big\}. \tag{12}
$$

2. Linear Programming-Based Approximation

In this section, we approximate $V_t(r_t)$ with a linear function of the form $\theta_t + \sum_{i \in \mathcal{P}} \vartheta_{it}\, r_{it}$ and use the linear programming representation of the dynamic program in (12) to decide what values we should pick for $\{\theta_t : t \in \mathcal{T}\}$ and $\{\vartheta_{it} : i \in \mathcal{P},\, t \in \mathcal{T}\}$. Previous research in Powell & Carvalho (1998), Papadaki & Powell (2003) and Adelman (2005) shows that linear value function approximations can yield high-quality solutions in the fleet management, batch service and revenue management settings.

Associating positive weights $\{\alpha(r_1) : r_1 \in \mathcal{R}^{|\mathcal{P}|}\}$ with the initial states, it is well-known that the value functions can be computed by solving the linear program

$$\min \quad \sum_{r_1 \in \mathcal{R}^{|\mathcal{P}|}} \alpha(r_1)\, v_1(r_1) \tag{13}$$

$$\text{subject to} \quad v_t(r_t) \geq p_t(x_t, y_t) + \mathbb{E}\big\{v_{t+1}(x_t + Q_{t+1})\big\}$$

$$\forall\ r_t \in \mathcal{R}^{|\mathcal{P}|},\ (x_t, y_t) \in \mathcal{Y}(r_t),\ t \in \mathcal{T} \setminus \{\tau\} \tag{14}$$

$$v_\tau(r_\tau) \geq p_\tau(x_\tau, y_\tau) \qquad \forall\ r_\tau \in \mathcal{R}^{|\mathcal{P}|},\ (x_\tau, y_\tau) \in \mathcal{Y}(r_\tau), \tag{15}$$

where $\{v_t(r_t) : r_t \in \mathcal{R}^{|\mathcal{P}|},\ t \in \mathcal{T}\}$ are the decision variables. Letting $\{v_t^*(r_t) : r_t \in \mathcal{R}^{|\mathcal{P}|},\ t \in \mathcal{T}\}$ be an optimal solution to this problem, if the weight for a particular initial state r_1 satisfies $\alpha(r_1) > 0$, then we have $v_1^*(r_1) = V_1(r_1)$ (Puterman (1994)). Problem (13)-(15) has $\tau |\mathcal{R}|^{|\mathcal{P}|}$ decision variables and $\sum_{t \in \mathcal{T}} \sum_{r_t \in \mathcal{R}^{|\mathcal{P}|}} |\mathcal{Y}(r_t)|$ constraints, which can both be very large.

To deal with the large number of decision variables, we approximate $V_t(r_t)$ with a linear function of the form $\theta_t + \sum_{i \in \mathcal{P}} \vartheta_{it}\, r_{it}$. To decide what values to pick for $\{\theta_t : t \in \mathcal{T}\}$ and $\{\vartheta_{it} : i \in \mathcal{P},\, t \in \mathcal{T}\}$, we substitute $\theta_t + \sum_{i \in \mathcal{P}} \vartheta_{it}\, r_{it}$ for $v_t(r_t)$ in problem (13)-(15) to obtain the linear program

$$\min \quad \sum_{r_1 \in \mathcal{R}^{|\mathcal{P}|}} \alpha(r_1)\, \theta_1 + \sum_{r_1 \in \mathcal{R}^{|\mathcal{P}|}} \sum_{i \in \mathcal{P}} \alpha(r_1)\, r_{i1}\, \vartheta_{i1} \tag{16}$$

$$\text{subject to} \quad \theta_t + \sum_{i \in \mathcal{P}} r_{it}\, \vartheta_{it} \geq p_t(x_t, y_t) + \theta_{t+1} + \sum_{i \in \mathcal{P}} \big[x_{it} + \mathbb{E}\{Q_{i,t+1}\}\big] \vartheta_{i,t+1}$$

$$\forall\ r_t \in \mathcal{R}^{|\mathcal{P}|},\ (x_t, y_t) \in \mathcal{Y}(r_t),\ t \in \mathcal{T} \setminus \{\tau\} \tag{17}$$

$$\theta_\tau + \sum_{i \in \mathcal{P}} r_{i\tau}\, \vartheta_{i\tau} \geq p_\tau(x_\tau, y_\tau) \qquad \forall\ r_\tau \in \mathcal{R}^{|\mathcal{P}|},\ (x_\tau, y_\tau) \in \mathcal{Y}(r_\tau), \tag{18}$$

where $\{\theta_t : t \in \mathcal{T}\}$, $\{\vartheta_{it} : i \in \mathcal{P},\ t \in \mathcal{T}\}$ are the decision variables. The set of feasible solutions to the problem above is nonempty, since we can obtain a feasible solution $\{\hat{\theta}_t : t \in \mathcal{T}\}$, $\{\hat{\vartheta}_{it} : i \in \mathcal{P},\ t \in \mathcal{T}\}$ by letting $\hat{p}_t = \max_{r_t \in \mathcal{R}^{|\mathcal{P}|}} \big\{\max_{(x_t, y_t) \in \mathcal{Y}(r_t)} p_t(x_t, y_t)\big\}$, $\hat{\theta}_t = \sum_{t'=t}^{\tau} \hat{p}_{t'}$ and $\hat{\vartheta}_{it} = 0$ for all $i \in \mathcal{P}$, $t \in \mathcal{T}$.

The following proposition shows that we obtain upper bounds on the value functions by solving problem (16)-(18). Results similar to Proposition 2 below and Proposition 5 in Section 3 are shown in Adelman & Mersereau (2004) for infinite-horizon problems. Our proofs are for finite-horizon problems and tend to be somewhat simpler.

Proposition 2. *If* $\{\hat{\theta}_t : t \in \mathcal{T}\}$, $\{\hat{\vartheta}_{it} : i \in \mathcal{P},\ t \in \mathcal{T}\}$ *is a feasible solution to problem* (16)-(18), *then we have* $V_t(r_t) \leq \hat{\theta}_t + \sum_{i\in\mathcal{P}} \hat{\vartheta}_{it}\, r_{it}$ *for all* $r_t \in \mathcal{R}^{|\mathcal{P}|}$, $t \in \mathcal{T}$.

Proof. We show the result by induction. It is easy to show the result for the last time period. Assuming that the result holds for time period $t+1$ and using the fact that $\{\hat{\theta}_t : t \in \mathcal{T}\}$, $\{\hat{\vartheta}_{it} : i \in \mathcal{P},\ t \in \mathcal{T}\}$ is feasible to problem (16)-(18), we have

$$\hat{\theta}_t + \sum_{i\in\mathcal{P}} \hat{\vartheta}_{it}\, r_{it} \geq \max_{(x_t,y_t)\in\mathcal{Y}(r_t)} \Big\{ p_t(x_t,y_t) + \mathbb{E}\Big\{ \hat{\theta}_{t+1} + \sum_{i\in\mathcal{P}} \hat{\vartheta}_{i,t+1}\big[x_{it} + Q_{i,t+1}\big] \Big\} \Big\}$$
$$\geq \max_{(x_t,y_t)\in\mathcal{Y}(r_t)} \Big\{ p_t(x_t,y_t) + \mathbb{E}\big\{V_{t+1}(x_t + Q_{t+1})\big\} \Big\} = V_t(r_t). \quad \Box$$

The proposition above also shows that the optimal objective value of problem (16)-(18) is bounded from below by $\sum_{r_1\in\mathcal{R}^{|\mathcal{P}|}} \alpha(r_1)\, V_1(r_1)$, which implies that problem (16)-(18) is bounded.

The number of decision variables in problem (16)-(18) is $\tau + \tau|\mathcal{P}|$, but the number of constraints is still as many as that of problem (13)-(15). We use constraint generation to deal with the large number of constraints. The idea is to iteratively solve a master problem, which has the same objective function and decision variables as problem (16)-(18), but has only a few of the constraints. After solving the master problem, we check if any of constraints (17)-(18) is violated by the solution. If there is one such constraint, then we add this constraint to the master problem and resolve the master problem. In particular, letting $\{\hat{\theta}_t : t \in \mathcal{T}\}$, $\{\hat{\vartheta}_{it} : i \in \mathcal{P},\ t \in \mathcal{T}\}$ be the solution to the current master problem, we solve the problem

$$\max_{r_t\in\mathcal{R}^{|\mathcal{P}|}} \Big\{ \max_{(x_t,y_t)\in\mathcal{Y}(r_t)} \Big\{ p_t(x_t,y_t) + \sum_{i\in\mathcal{P}} \hat{\vartheta}_{i,t+1}\, x_{it} \Big\} - \sum_{i\in\mathcal{P}} \hat{\vartheta}_{it}\, r_{it} \Big\} \tag{19}$$

for all $t \in \mathcal{T} \setminus \{\tau\}$ to check if any of constraints (17) is violated by this solution. Letting $(\hat{r}_t, \hat{x}_t, \hat{y}_t)$ be an optimal solution to problem (19), if we have $p_t(\hat{x}_t, \hat{y}_t) + \sum_{i\in\mathcal{P}} \hat{\vartheta}_{i,t+1}\, \hat{x}_{it} - \sum_{i\in\mathcal{P}} \hat{\vartheta}_{it}\, \hat{r}_{it} > \hat{\theta}_t - \hat{\theta}_{t+1} - \sum_{i\in\mathcal{P}} \mathbb{E}\{Q_{i,t+1}\}\, \hat{\vartheta}_{i,t+1}$, then the constraint

$$\theta_t + \sum_{i\in\mathcal{P}} \hat{r}_{it}\, \vartheta_{it} \geq p_t(\hat{x}_t, \hat{y}_t) + \theta_{t+1} + \sum_{i\in\mathcal{P}} \big[\hat{x}_{it} + \mathbb{E}\{Q_{i,t+1}\}\big]\vartheta_{i,t+1}$$

is violated by the solution $\{\hat{\theta}_t : t \in \mathcal{T}\}$, $\{\hat{\vartheta}_{it} : i \in \mathcal{P},\ t \in \mathcal{T}\}$. We add this constraint to the master problem and resolve the master problem. Similarly, we check if any of constraints (18) is violated by the solution $\{\hat{\theta}_t : t \in \mathcal{T}\}$, $\{\hat{\vartheta}_{it} : i \in \mathcal{P},\ t \in \mathcal{T}\}$ by solving the problem

$$\max_{r_\tau\in\mathcal{R}^{|\mathcal{P}|}} \Big\{ \max_{(x_\tau,y_\tau)\in\mathcal{Y}(r_\tau)} \Big\{ p_\tau(x_\tau, y_\tau) \Big\} - \sum_{i\in\mathcal{P}} \hat{\vartheta}_{i\tau}\, r_{i\tau} \Big\}. \tag{20}$$

We summarize the constraint generation idea in Figure 1.

Fortunately, problems (19) and (20) are min-cost network flow problems, and hence, constraint generation can be done efficiently. To see this, we write problem (19) as

$$
\begin{aligned}
\max \quad & \sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{C}} \sum_{s=0}^{S} [f_{jst} - c_{ijt}] y_{ijst} + \sum_{i \in \mathcal{P}} [\hat{\vartheta}_{i,t+1} - h_{it}] x_{it} - \sum_{i \in \mathcal{P}} \hat{\vartheta}_{it} r_{it} \\
\text{subject to} \quad & (9), (10) \\
& x_{it} + \sum_{j \in \mathcal{C}} \sum_{s=0}^{S} y_{ijst} - r_{it} = 0 && \forall\, i \in \mathcal{P} && (21) \\
& r_{it} \leq R && \forall\, i \in \mathcal{P} && (22) \\
& r_{it},\, x_{it},\, y_{ijst} \in \mathbb{Z}_+ && \forall\, i \in \mathcal{P},\; j \in \mathcal{C},\; s = 0, \ldots, S.
\end{aligned}
$$

We let $\{\eta_{it} : i \in \mathcal{P}\}$ be the slack variables for constraints (22). We define new decision variables $\{\zeta_{jst} : j \in \mathcal{C},\; s = 0, \ldots, S-1\}$, and using these decision variables, we split constraints (9) in the problem above into $\sum_{i \in \mathcal{P}} y_{ijst} - \zeta_{jst} = 0$ and $\zeta_{jst} \leq 1$ for all $j \in \mathcal{C}$, $s = 0, \ldots, S-1$. In this case, the problem above can be written as

$$
\begin{aligned}
\max \quad & \sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{C}} \sum_{s=0}^{S} [f_{jst} - c_{ijt}] y_{ijst} + \sum_{i \in \mathcal{P}} [\hat{\vartheta}_{i,t+1} - h_{it}] x_{it} - \sum_{i \in \mathcal{P}} \hat{\vartheta}_{it} r_{it} && && (23) \\
\text{subject to} \quad & (10), (21) && && (24) \\
& r_{it} + \eta_{it} = R && \forall\, i \in \mathcal{P} && (25) \\
& \sum_{i \in \mathcal{P}} y_{ijst} - \zeta_{jst} = 0 && \forall\, j \in \mathcal{C},\; s = 0, \ldots, S-1 && (26) \\
& \zeta_{jst} \leq 1 && \forall\, j \in \mathcal{C},\; s = 0, \ldots, S-1 && (27) \\
& r_{it},\, x_{it},\, y_{ijst},\, \eta_{it},\, \zeta_{js't} \in \mathbb{Z}_+ && \forall\, i \in \mathcal{P},\; j \in \mathcal{C},\; s = 0, \ldots, S,\; s' = 0, \ldots, S-1. && (28)
\end{aligned}
$$

Defining three sets of nodes $\mathcal{O}_1 = \mathcal{P}$, $\mathcal{O}_2 = \mathcal{P}$ and $\mathcal{O}_3 = \mathcal{C} \times \{0, \ldots, S-1\}$, it is easy to see that problem (23)-(28) is a min-cost network flow problem that takes place over a network with the set of nodes $\mathcal{O}_1 \bigcup \mathcal{O}_2 \bigcup \mathcal{O}_3 \bigcup \{\phi\}$ shown in Figure 2. Corresponding to each decision variable in problem (23)-(28), there exists an arc in the network in Figure 2. The arc corresponding to decision variable r_{it} leaves node $i \in \mathcal{O}_1$ and enters node $i \in \mathcal{O}_2$. The arc corresponding to decision variable x_{it} leaves node $i \in \mathcal{O}_2$ and enters node ϕ. Whenever $s \in \{0, \ldots, S-1\}$, the arc corresponding to decision variable y_{ijst} leaves node $i \in \mathcal{O}_2$ and enters node $(j, s) \in \mathcal{O}_3$. The arc corresponding to decision variable y_{ijSt} leaves node $i \in \mathcal{O}_2$ and enters node ϕ. The arc corresponding to decision variable η_{it} leaves node $i \in \mathcal{O}_1$ and enters node ϕ. Finally, the arc corresponding to decision variable ζ_{jst} leaves node $(j, s) \in \mathcal{O}_3$ and enters node ϕ. Constraints (25), (21) and (26) in problem (23)-(28) are respectively the flow balance constraints for the nodes in $\mathcal{O}_1$, $\mathcal{O}_2$ and $\mathcal{O}_3$. The flow balance constraint for node ϕ is redundant and omitted in problem (23)-(28). The supplies of the nodes in $\mathcal{O}_1$ are R.

We note that problem (20) is also a min-cost network flow problem, since it is a special case of problem (19).

3. Lagrangian Relaxation-Based Approximation

This section proposes a method to approximate the value functions that is based on the observation that if we relax constraints (9) in problem (7)-(11), then the set of feasible solutions to this problem decomposes by the elements of $\mathcal{P}$. Associating positive Lagrange multipliers $\{\lambda_{jst} : j \in \mathcal{C},\ s = 0, \ldots, S-1,\ t \in \mathcal{T}\}$ with these constraints, this suggests solving the optimality equation

$$
\begin{aligned}
V_t^{\lambda}(r_t) = \max \quad & \sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{C}} \sum_{s=0}^{S} \big[f_{jst} - c_{ijt}\big] y_{ijst} - \sum_{i \in \mathcal{P}} h_{it}\, x_{it} \\
& \qquad + \sum_{j \in \mathcal{C}} \sum_{s=0}^{S-1} \lambda_{jst} \Big[1 - \sum_{i \in \mathcal{P}} y_{ijst}\Big] + \mathbb{E}\big\{V_{t+1}^{\lambda}(x_t + Q_{t+1})\big\} && (29) \\
\text{subject to} \quad & (8), (10), (11) && (30) \\
& y_{ijst} \le 1 \qquad \forall\, i \in \mathcal{P},\ j \in \mathcal{C},\ s = 0, \ldots, S-1, && (31)
\end{aligned}
$$

where we use the superscript λ to emphasize that the solution depends on the Lagrange multipliers. Recalling constraints (9), we note that constraints (31) would be redundant in problem (7)-(11), but we add them to problem (29)-(31) to tighten the relaxation. For ease of notation, we let

$$
\begin{aligned}
\mathcal{Y}_i(r_{it}) = \Big\{ (x_{it}, y_{it}) \in \mathbb{Z}_+ \times \mathbb{Z}_+^{|\mathcal{C}|(1+S)} : \quad & x_{it} + \sum_{j \in \mathcal{C}} \sum_{s=0}^{S} y_{ijst} = r_{it} \\
& x_{it} \le L \\
& y_{ijst} \le 1 \qquad \forall\, j \in \mathcal{C},\ s = 0, \ldots, S-1 \Big\},
\end{aligned}
$$

$$
p_{it}(x_{it}, y_{it}) = \sum_{j \in \mathcal{C}} \sum_{s=0}^{S} \big[f_{jst} - c_{ijt}\big] y_{ijst} - h_{it}\, x_{it},
$$

where we use $y_{it} = \{y_{ijst} : j \in \mathcal{C},\ s = 0, \ldots, S\}$. In this case, problem (29)-(31) can be written as

$$
\begin{aligned}
V_t^{\lambda}(r_t) = \max \quad & \sum_{i \in \mathcal{P}} p_{it}(x_{it}, y_{it}) + \sum_{j \in \mathcal{C}} \sum_{s=0}^{S-1} \lambda_{jst} \Big[1 - \sum_{i \in \mathcal{P}} y_{ijst}\Big] + \mathbb{E}\big\{V_{t+1}^{\lambda}(x_t + Q_{t+1})\big\} && (32) \\
\text{subject to} \quad & (x_{it}, y_{it}) \in \mathcal{Y}_i(r_{it}) \qquad \forall\, i \in \mathcal{P}. && (33)
\end{aligned}
$$

The benefit of this method is that the optimality equation in (32)-(33) decomposes into $|\mathcal{P}|$ optimality equations, each involving a one-dimensional state variable.

Proposition 3. *If $\{V_{it}^{\lambda}(r_{it}) : r_{it} \in \mathcal{R},\ t \in \mathcal{T}\}$ is a solution to the optimality equation*

$$
V_{it}^{\lambda}(r_{it}) = \max_{(x_{it}, y_{it}) \in \mathcal{Y}_i(r_{it})} \Big\{ p_{it}(x_{it}, y_{it}) - \sum_{j \in \mathcal{C}} \sum_{s=0}^{S-1} \lambda_{jst}\, y_{ijst} + \mathbb{E}\big\{V_{i,t+1}^{\lambda}(x_{it} + Q_{i,t+1})\big\} \Big\} \qquad (34)
$$

for all $i \in \mathcal{P}$, then we have

$$
V_t^{\lambda}(r_t) = \sum_{t'=t}^{\tau} \sum_{j \in \mathcal{C}} \sum_{s=0}^{S-1} \lambda_{jst'} + \sum_{i \in \mathcal{P}} V_{it}^{\lambda}(r_{it}).
$$

Proof. We show the result by induction. It is easy to show the result for the last time period. Assuming that the result holds for time period $t+1$, the objective function in (32) can be written as

$$\sum_{i \in \mathcal{P}} p_{it}(x_{it}, y_{it}) + \sum_{j \in \mathcal{C}} \sum_{s=0}^{S-1} \lambda_{jst} \Big[1 - \sum_{i \in \mathcal{P}} y_{ijst}\Big] + \sum_{t'=t+1}^{\tau} \sum_{j \in \mathcal{C}} \sum_{s=0}^{S-1} \lambda_{jst'} + \sum_{i \in \mathcal{P}} \mathbb{E}\big\{V^{\lambda}_{i,t+1}(x_{it} + Q_{i,t+1})\big\}.$$

The result follows by noting that both the objective function above and the feasible set of problem (32)-(33) decompose by the elements of $\mathcal{P}$. □

Therefore, the optimality equation in (32)-(33) can be solved by concentrating on one plant at a time. The optimality equation in (34) involves a one-dimensional state variable, but it requires solving an optimization problem involving $1 + |\mathcal{C}|(1 + S)$ decision variables. The following result shows that this optimality equation can be solved efficiently.

Lemma 4. *Problem* (34) *can be solved by a sort operation.*

Proof. Using backward induction on time periods, it is easy to show that $V^{\lambda}_{i,t+1}(r_{i,t+1})$ is a concave function of $r_{i,t+1}$ in the sense that $V^{\lambda}_{i,t+1}(r_{i,t+1} - 1) + V^{\lambda}_{i,t+1}(r_{i,t+1} + 1) \leq 2\, V^{\lambda}_{i,t+1}(r_{i,t+1})$ for all $r_{i,t+1} = 1, \ldots, R-1$. This implies that $\mathbb{E}\big\{V^{\lambda}_{i,t+1}(x_{it} + Q_{i,t+1})\big\}$ is also a concave function of x_{it}. Therefore, letting $\Delta_{i\ell,t+1} = \mathbb{E}\big\{V^{\lambda}_{i,t+1}(\ell + 1 + Q_{i,t+1})\big\} - \mathbb{E}\big\{V^{\lambda}_{i,t+1}(\ell + Q_{i,t+1})\big\}$ for all $\ell = 0, \ldots, L-1$ and associating the decision variables $\{z_{i\ell,t+1} : \ell = 0, \ldots, L-1\}$ with these first differences, problem (34) can explicitly be written as

$$\begin{aligned}
\max \quad & \sum_{j \in \mathcal{C}} \sum_{s=0}^{S} \big[f_{jst} - c_{ijt}\big] y_{ijst} - h_{it}\, x_{it} - \sum_{j \in \mathcal{C}} \sum_{s=0}^{S-1} \lambda_{jst}\, y_{ijst} + \mathbb{E}\big\{V^{\lambda}_{i,t+1}(Q_{i,t+1})\big\} + \sum_{\ell=0}^{L-1} \Delta_{i\ell,t+1}\, z_{i\ell,t+1} \\
\text{subject to} \quad & x_{it} + \sum_{j \in \mathcal{C}} \sum_{s=0}^{S} y_{ijst} = r_{it} \\
& x_{it} - \sum_{\ell=0}^{L-1} z_{i\ell,t+1} = 0 && (35) \\
& y_{ijst} \leq 1 \qquad \forall\, j \in \mathcal{C},\ s = 0, \ldots, S-1 && (36) \\
& z_{i\ell,t+1} \leq 1 \qquad \forall\, \ell = 0, \ldots, L-1 && (37) \\
& x_{it},\, y_{ijst},\, z_{i\ell,t+1} \in \mathbb{Z}_+ \qquad \forall\, j \in \mathcal{C},\ s = 0, \ldots, S,\ \ell = 0, \ldots, L-1.
\end{aligned}$$

Dropping the constant term in the objective function and using constraint (35) to substitute $\sum_{\ell=0}^{L-1} z_{i\ell,t+1}$ for x_{it}, the problem above becomes

$$\begin{aligned}
\max \quad & \sum_{j \in \mathcal{C}} \sum_{s=0}^{S} \big[f_{jst} - c_{ijt}\big] y_{ijst} - \sum_{j \in \mathcal{C}} \sum_{s=0}^{S-1} \lambda_{jst}\, y_{ijst} + \sum_{\ell=0}^{L-1} \big[\Delta_{i\ell,t+1} - h_{it}\big] z_{i\ell,t+1} \\
\text{subject to} \quad & (36), (37) \\
& \sum_{j \in \mathcal{C}} \sum_{s=0}^{S} y_{ijst} + \sum_{\ell=0}^{L-1} z_{i\ell,t+1} = r_{it} \\
& y_{ijst},\, z_{i\ell,t+1} \in \mathbb{Z}_+ \qquad \forall\, j \in \mathcal{C},\ s = 0, \ldots, S,\ \ell = 0, \ldots, L-1.
\end{aligned}$$

The result follows by noting that the problem above is a knapsack problem where each item consumes one unit of space. $\square$

Proposition 5 shows that the optimality equation in (32)-(33) provides upper bounds on the value functions.

Proposition 5. *We have $V_t(r_t) \leq V_t^{\lambda}(r_t)$ for all $r_t \in \mathcal{R}^{|\mathcal{P}|}$, $t \in \mathcal{T}$.*

Proof. We show the result by induction. It is easy to show the result for the last time period. We assume that the result holds for time period $t+1$, fix $r_t \in \mathcal{R}^{|\mathcal{P}|}$ and let $(\hat{x}_t, \hat{y}_t) = \operatorname{argmax}_{(x_t,y_t)\in\mathcal{Y}(r_t)} \Big\{ p_t(x_t, y_t) + \mathbb{E}\big\{V_{t+1}(x_t + Q_{t+1})\big\} \Big\}$. We have

$$
\begin{aligned}
V_t(r_t) &= \sum_{i\in\mathcal{P}} p_{it}(\hat{x}_{it}, \hat{y}_{it}) + \mathbb{E}\big\{V_{t+1}(\hat{x}_t + Q_{t+1})\big\} \\
&\leq \sum_{i\in\mathcal{P}} p_{it}(\hat{x}_{it}, \hat{y}_{it}) + \sum_{j\in\mathcal{C}} \sum_{s=0}^{S-1} \lambda_{jst} \Big[1 - \sum_{i\in\mathcal{P}} \hat{y}_{ijst}\Big] + \mathbb{E}\big\{V_{t+1}^{\lambda}(\hat{x}_t + Q_{t+1})\big\} \leq V_t^{\lambda}(r_t),
\end{aligned}
$$

where the first inequality follows from the induction assumption and the fact that $\lambda_{jst} \geq 0$ for all $j \in \mathcal{C}$, $s = 0, \ldots, S-1$ and $(\hat{x}_t, \hat{y}_t) \in \mathcal{Y}(r_t)$. Noting the objective function of problem (32)-(33), the second inequality follows from the fact that $(\hat{x}_{it}, \hat{y}_{it}) \in \mathcal{Y}_i(r_{it})$ for all $i \in \mathcal{P}$. $\square$

By Proposition 5, we have $V_1(r_1) \leq V_1^{\lambda}(r_1)$ for any set of positive Lagrange multipliers. If the initial state is known, then we can solve $\min_{\lambda\geq 0} V_1^{\lambda}(r_1)$ for a particular initial state r_1 to obtain the tightest possible bound on the value function at r_1. If, however, the initial state is not known and we need to obtain a "good" approximation to the value function for all possible initial states, then we can associate positive weights $\{\alpha(r_1) : r_1 \in \mathcal{R}^{|\mathcal{P}|}\}$ with the initial states. Assuming that $\sum_{r_1\in\mathcal{R}^{|\mathcal{P}|}} \alpha(r_1) = 1$ without loss of generality, we can obtain a possibly tight bound on the value function by solving the problem

$$
\begin{aligned}
\min_{\lambda\geq 0} \Big\{ \sum_{r_1\in\mathcal{R}^{|\mathcal{P}|}} \alpha(r_1)\, V_1^{\lambda}(r_1) \Big\} &= \min_{\lambda\geq 0} \Big\{ \sum_{r_1\in\mathcal{R}^{|\mathcal{P}|}} \sum_{t\in\mathcal{T}} \sum_{j\in\mathcal{C}} \sum_{s=0}^{S-1} \alpha(r_1)\, \lambda_{jst} + \sum_{r_1\in\mathcal{R}^{|\mathcal{P}|}} \sum_{i\in\mathcal{P}} \alpha(r_1)\, V_{i1}^{\lambda}(r_{i1}) \Big\} \\
&= \min_{\lambda\geq 0} \Big\{ \sum_{t\in\mathcal{T}} \sum_{j\in\mathcal{C}} \sum_{s=0}^{S-1} \lambda_{jst} + \sum_{r_1\in\mathcal{R}^{|\mathcal{P}|}} \sum_{i\in\mathcal{P}} \sum_{\hat{r}_{i1}\in\mathcal{R}} \mathbf{1}(r_{i1} = \hat{r}_{i1})\, \alpha(r_1)\, V_{i1}^{\lambda}(\hat{r}_{i1}) \Big\} \\
&= \min_{\lambda\geq 0} \Big\{ \sum_{t\in\mathcal{T}} \sum_{j\in\mathcal{C}} \sum_{s=0}^{S-1} \lambda_{jst} + \sum_{i\in\mathcal{P}} \sum_{\hat{r}_{i1}\in\mathcal{R}} \beta_i(\hat{r}_{i1})\, V_{i1}^{\lambda}(\hat{r}_{i1}) \Big\}, \qquad (38)
\end{aligned}
$$

where the first equality follows from Proposition 3, $\mathbf{1}(\cdot)$ is the indicator function and we let $\beta_i(\hat{r}_{i1}) = \sum_{r_1\in\mathcal{R}^{|\mathcal{P}|}} \mathbf{1}(r_{i1} = \hat{r}_{i1})\, \alpha(r_1)$. The objective function of problem (38) is called the dual function. In the following two sections, we propose two methods to minimize the dual function.

3.1. Constraint generation

In this section, we formulate problem (38) as a linear program that has a large number of constraints. Similar to Section 2, we deal with the large number of constraints by using constraint generation.

Since $\{V_{it}^{\lambda}(r_{it}) : r_{it} \in \mathcal{R},\ t \in \mathcal{T}\}$ is a solution to the optimality equation in (34), for any set of Lagrange multipliers λ, $\sum_{i \in \mathcal{P}} \sum_{r_{i1} \in \mathcal{R}} \beta_i(r_{i1})\, V_{i1}^{\lambda}(r_{i1})$ can be computed by solving the linear program

$$\min \quad \sum_{i \in \mathcal{P}} \sum_{r_{i1} \in \mathcal{R}} \beta_i(r_{i1})\, v_{i1}(r_{i1})$$

$$\text{subject to} \quad v_{it}(r_{it}) \geq p_{it}(x_{it}, y_{it}) - \sum_{j \in \mathcal{C}} \sum_{s=0}^{S-1} \lambda_{jst}\, y_{ijst} + \mathbb{E}\big\{v_{i,t+1}(x_{it} + Q_{i,t+1})\big\}$$
$$\forall\ r_{it} \in \mathcal{R},\ (x_{it}, y_{it}) \in \mathcal{Y}_i(r_{it}),\ i \in \mathcal{P},\ t \in \mathcal{T} \setminus \{\tau\} \quad (39)$$
$$v_{i\tau}(r_{i\tau}) \geq p_{i\tau}(x_{i\tau}, y_{i\tau}) - \sum_{j \in \mathcal{C}} \sum_{s=0}^{S-1} \lambda_{js\tau}\, y_{ijs\tau} \qquad \forall\ r_{i\tau} \in \mathcal{R},\ (x_{i\tau}, y_{i\tau}) \in \mathcal{Y}_i(r_{i\tau}),\ i \in \mathcal{P}, \quad (40)$$

where $\{v_{it}(r_{it}) : r_{it} \in \mathcal{R},\ i \in \mathcal{P},\ t \in \mathcal{T}\}$ are the decision variables. Therefore, we can find an optimal solution to problem (38) by solving the linear program

$$\min \quad \sum_{t \in \mathcal{T}} \sum_{j \in \mathcal{C}} \sum_{s=0}^{S-1} \lambda_{jst} + \sum_{i \in \mathcal{P}} \sum_{r_{i1} \in \mathcal{R}} \beta_i(r_{i1})\, v_{i1}(r_{i1}) \quad (41)$$
$$\text{subject to} \quad (39), (40) \quad (42)$$
$$\lambda_{jst} \geq 0 \qquad \forall\ j \in \mathcal{C},\ s = 0, \ldots, S-1,\ t \in \mathcal{T}, \quad (43)$$

where $\{v_{it}(r_{it}) : r_{it} \in \mathcal{R},\ i \in \mathcal{P},\ t \in \mathcal{T}\}$, $\{\lambda_{jst} : j \in \mathcal{C},\ s = 0, \ldots, S-1,\ t \in \mathcal{T}\}$ are the decision variables. It is easy to see that the set of feasible solutions to the problem above is nonempty. Furthermore, Proposition 5 and (38) show that the optimal objective value of this problem is bounded from below by $\sum_{r_1 \in \mathcal{R}^{|\mathcal{P}|}} \alpha(r_1)\, V_1(r_1)$.

The number of decision variables in problem (41)-(43) is $\tau|\mathcal{P}||\mathcal{R}| + \tau|\mathcal{C}|S$, which is manageable, but the number of constraints is $\sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{P}} \sum_{r_{it} \in \mathcal{R}} |\mathcal{Y}_i(r_{it})|$, which can be very large. We use constraint generation to deal with the large number of constraints, where we iteratively solve a master problem that has the same objective function and decision variables as problem (41)-(43), but has only a few of the constraints. The idea is very similar to the one in Section 2.

In particular, letting $\{\hat{v}_{it}(r_{it}) : r_{it} \in \mathcal{R},\ i \in \mathcal{P},\ t \in \mathcal{T}\}$, $\{\hat{\lambda}_{jst} : j \in \mathcal{C},\ s = 0, \ldots, S-1,\ t \in \mathcal{T}\}$ be the solution to the current master problem, we solve the problem

$$\max_{(x_{it}, y_{it}) \in \mathcal{Y}_i(r_{it})} \left\{ p_{it}(x_{it}, y_{it}) - \sum_{j \in \mathcal{C}} \sum_{s=0}^{S-1} \hat{\lambda}_{jst}\, y_{ijst} + \mathbb{E}\big\{\hat{v}_{i,t+1}(x_{it} + Q_{i,t+1})\big\} \right\} \quad (44)$$

for all $r_{it} \in \mathcal{R}$, $i \in \mathcal{P}$, $t \in \mathcal{T} \setminus \{\tau\}$ to check if any of constraints (39) in problem (41)-(43) is violated by this solution. Letting $(\hat{x}_{it}, \hat{y}_{it})$ be an optimal solution to problem (44), if we have

$p_{it}(\hat{x}_{it}, \hat{y}_{it}) - \sum_{j \in \mathcal{C}} \sum_{s=0}^{S-1} \hat{\lambda}_{jst}\, \hat{y}_{ijst} + \mathbb{E}\{\hat{v}_{i,t+1}(\hat{x}_{it} + Q_{i,t+1})\} > \hat{v}_{it}(r_{it})$, then the constraint

$$v_{it}(r_{it}) \geq p_{it}(\hat{x}_{it}, \hat{y}_{it}) - \sum_{j \in \mathcal{C}} \sum_{s=0}^{S-1} \lambda_{jst}\, \hat{y}_{ijst} + \mathbb{E}\{v_{i,t+1}(\hat{x}_{it} + Q_{i,t+1})\}$$

is violated by the solution $\{\hat{v}_{it}(r_{it}) : r_{it} \in \mathcal{R},\ i \in \mathcal{P},\ t \in \mathcal{T}\}$, $\{\hat{\lambda}_{jst} : j \in \mathcal{C},\ s = 0, \ldots, S-1,\ t \in \mathcal{T}\}$. We add this constraint to the master problem and resolve the master problem. Similarly, we solve the problem $\max_{(x_{i\tau}, y_{i\tau}) \in \mathcal{Y}_i(r_{i\tau})} \Big\{ p_{i\tau}(x_{i\tau}, y_{i\tau}) - \sum_{j \in \mathcal{C}} \sum_{s=0}^{S-1} \hat{\lambda}_{js\tau}\, y_{ijs\tau} \Big\}$ for all $r_{i\tau} \in \mathcal{R}$, $i \in \mathcal{P}$ to check if any of constraints (40) in problem (41)-(43) is violated by the solution $\{\hat{v}_{it}(r_{it}) : r_{it} \in \mathcal{R},\ i \in \mathcal{P},\ t \in \mathcal{T}\}$, $\{\hat{\lambda}_{jst} : j \in \mathcal{C},\ s = 0, \ldots, S-1,\ t \in \mathcal{T}\}$.

If $\hat{v}_{i,t+1}(r_{i,t+1})$ is a concave function of $r_{i,t+1}$ in the sense that

$$\hat{v}_{i,t+1}(r_{i,t+1} - 1) + \hat{v}_{i,t+1}(r_{i,t+1} + 1) \leq 2\, \hat{v}_{i,t+1}(r_{i,t+1}) \tag{45}$$

for all $r_{i,t+1} = 1, \ldots, R-1$, then problem (44) has the same form as the problem considered in Lemma 4 and can be solved by a sort operation. This makes constraint generation very efficient. In general, it is not guaranteed that the solution to the master problem will satisfy (45). To ensure that (45) is satisfied, we add the constraints

$$v_{it}(r_{it} - 1) + v_{it}(r_{it} + 1) \leq 2\, v_{it}(r_{it}) \qquad \forall\ r_{it} = 1, \ldots, R-1,\ i \in \mathcal{P},\ t \in \mathcal{T} \tag{46}$$

to the master problem before we begin constraint generation. By the following lemma, adding constraints (46) to problem (41)-(43) does not change its optimal objective value. Therefore, we can add constraints (46) to the master problem without disturbing the validity of constraint generation.

Lemma 6. *Adding constraints* (46) *to problem* (41)-(43) *does not change its optimal objective value.*

Proof. We let $\{v^*_{it}(r_{it}) : r_{it} \in \mathcal{R},\ i \in \mathcal{P},\ t \in \mathcal{T}\}$, $\{\lambda^*_{jst} : j \in \mathcal{C},\ s = 0, \ldots,\ S-1,\ t \in \mathcal{T}\}$ be an optimal solution to problem (41)-(43) and $\{V^{\lambda^*}_{it}(r_{it}) : r_{it} \in \mathcal{R},\ t \in \mathcal{T}\}$ be obtained by solving the optimality equation in (34) with the set of Lagrange multipliers λ^*. As mentioned in the proof of Lemma 4, we have $V^{\lambda^*}_{it}(r_{it} - 1) + V^{\lambda^*}_{it}(r_{it} + 1) \leq 2\, V^{\lambda^*}_{it}(r_{it})$ for all $r_{it} = 1, \ldots, R-1$, $i \in \mathcal{P}$, $t \in \mathcal{T}$. We now show that $\{V^{\lambda^*}_{it}(r_{it}) : r_{it} \in \mathcal{R},\ i \in \mathcal{P},\ t \in \mathcal{T}\}$, $\{\lambda^*_{jst} : j \in \mathcal{C},\ s = 0, \ldots, S-1,\ t \in \mathcal{T}\}$ is also an optimal solution to problem (41)-(43) and this establishes the result.

Since $\{V^{\lambda^*}_{it}(r_{it}) : r_{it} \in \mathcal{R},\ t \in \mathcal{T}\}$ solves the optimality equation in (34), we have

$$V^{\lambda^*}_{it}(r_{it}) \geq p_{it}(x_{it}, y_{it}) - \sum_{j \in \mathcal{C}} \sum_{s=0}^{S-1} \lambda^*_{jst}\ y_{ijst} + \mathbb{E}\{V^{\lambda^*}_{i,t+1}(x_{it} + Q_{i,t+1})\}$$

for all $r_{it} \in \mathcal{R}$, $(x_{it}, y_{it}) \in \mathcal{Y}_i(r_{it})$, $i \in \mathcal{P}$, $t \in \mathcal{T}$. Therefore, $\{V^{\lambda^*}_{it}(r_{it}) : r_{it} \in \mathcal{R},\ i \in \mathcal{P},\ t \in \mathcal{T}\}$, $\{\lambda^*_{jst} : j \in \mathcal{C},\ s = 0, \ldots, S-1,\ t \in \mathcal{T}\}$ is feasible to problem (41)-(43). We now show by induction that

$v^*_{it}(r_{it}) \geq V^{\lambda^*}_{it}(r_{it})$ for all $r_{it} \in \mathcal{R},\ i \in \mathcal{P},\ t \in \mathcal{T}$, which implies that $\{V^{\lambda^*}_{it}(r_{it}) : r_{it} \in \mathcal{R},\ i \in \mathcal{P},\ t \in \mathcal{T}\}$, $\{\lambda^*_{jst} : j \in \mathcal{C},\ s = 0, \ldots, S-1,\ t \in \mathcal{T}\}$ is an optimal solution to problem (41)-(43). It is easy to show the result for the last time period. Assuming that the result holds for time period $t+1$ and noting constraints (39) in problem (41)-(43), we have

$$v^*_{it}(r_{it}) \geq \max_{(x_{it}, y_{it}) \in \mathcal{Y}_i(r_{it})} \left\{ p_{it}(x_{it}, y_{it}) - \sum_{j \in \mathcal{C}} \sum_{s=0}^{S-1} \lambda^*_{jst}\, y_{ijst} + \mathbb{E}\big\{ v^*_{i,t+1}(x_{it} + Q_{i,t+1}) \big\} \right\}$$

$$\geq \max_{(x_{it}, y_{it}) \in \mathcal{Y}_i(r_{it})} \left\{ p_{it}(x_{it}, y_{it}) - \sum_{j \in \mathcal{C}} \sum_{s=0}^{S-1} \lambda^*_{jst}\, y_{ijst} + \mathbb{E}\big\{ V^{\lambda^*}_{i,t+1}(x_{it} + Q_{i,t+1}) \big\} \right\} = V^{\lambda^*}_{it}(r_{it}). \quad \square$$

3.2. Benders decomposition

We begin this section by showing that $\sum_{i \in \mathcal{P}} \sum_{r_{i1} \in \mathcal{R}} \beta_i(r_{i1})\, V^{\lambda}_{i1}(r_{i1})$ is a convex function of λ. This result allows us to solve problem (38) through Benders decomposition, which represents the dual function by using a number of cutting planes that are constructed iteratively.

We use some new notation in this section. In particular, we let $x^{\lambda}_{it}(r_{it})$, $\{y^{\lambda}_{ijst}(r_{it}) : j \in \mathcal{C},\ s = 0, \ldots, S\}$ be an optimal solution to problem (34). We use the superscript λ and the argument r_{it} to emphasize that the solution depends on the Lagrange multipliers and the state. We let V^{λ}_{it} and λ_{jt} respectively be the vectors $\{V^{\lambda}_{it}(r_{it}) : r_{it} \in \mathcal{R}\}$ and $\{\lambda_{jst} : s = 0, \ldots, S-1\}$. We let $Y^{\lambda}_{ijt} = \{y^{\lambda}_{ijst}(r_{it}) : r_{it} \in \mathcal{R},\ s = 0, \ldots, S-1\}$ be the $|\mathcal{R}| \times S$-dimensional matrix whose (r_{it}, s)-th component is $y^{\lambda}_{ijst}(r_{it})$. Finally, we let P^{λ}_{it} be the $|\mathcal{R}| \times |\mathcal{R}|$-dimensional matrix whose $(r_{it}, r_{i,t+1})$-th component is $\mathbb{P}\{x^{\lambda}_{it}(r_{it}) + Q_{i,t+1} = r_{i,t+1}\}$.

The following proposition shows that $\sum_{i \in \mathcal{P}} \sum_{r_{i1} \in \mathcal{R}} \beta_i(r_{i1})\, V^{\lambda}_{i1}(r_{i1})$ is a convex function of λ.

Proposition 7. *For any two sets of Lagrange multipliers λ and $\hat{\lambda}$, we have*

$$V^{\hat{\lambda}}_{it} \geq V^{\lambda}_{it} - \sum_{j \in \mathcal{C}} Y^{\lambda}_{ijt} \big[\hat{\lambda}_{jt} - \lambda_{jt}\big] - P^{\lambda}_{it} \sum_{j \in \mathcal{C}} Y^{\lambda}_{ij,t+1} \big[\hat{\lambda}_{j,t+1} - \lambda_{j,t+1}\big]$$
$$- P^{\lambda}_{it}\, P^{\lambda}_{i,t+1} \sum_{j \in \mathcal{C}} Y^{\lambda}_{ij,t+2} \big[\hat{\lambda}_{j,t+2} - \lambda_{j,t+2}\big] - \ldots - P^{\lambda}_{it}\, P^{\lambda}_{i,t+1} \ldots P^{\lambda}_{i,\tau-1} \sum_{j \in \mathcal{C}} Y^{\lambda}_{ij\tau} \big[\hat{\lambda}_{j\tau} - \lambda_{j\tau}\big]$$

for all $i \in \mathcal{P}$, $t \in \mathcal{T}$.

Proof. We show the result by induction. It is easy to show the result for the last time period. We assume that the result holds for time period $t+1$. Letting $y^{\lambda}_{it}(r_{it}) = \{y^{\lambda}_{ijst}(r_{it}) : j \in \mathcal{C},\ s = 0, \ldots, S\}$, since $(x^{\lambda}_{it}(r_{it}), y^{\lambda}_{it}(r_{it}))$ is an optimal solution to problem (34), we have

$$V^{\lambda}_{it}(r_{it}) = p_{it}(x^{\lambda}_{it}(r_{it}), y^{\lambda}_{it}(r_{it})) - \sum_{j \in \mathcal{C}} \sum_{s=0}^{S-1} \lambda_{jst}\, y^{\lambda}_{ijst}(r_{it}) + \mathbb{E}\big\{ V^{\lambda}_{i,t+1}(x^{\lambda}_{it}(r_{it}) + Q_{i,t+1}) \big\}$$

$$V^{\hat{\lambda}}_{it}(r_{it}) \geq p_{it}(x^{\lambda}_{it}(r_{it}), y^{\lambda}_{it}(r_{it})) - \sum_{j \in \mathcal{C}} \sum_{s=0}^{S-1} \hat{\lambda}_{jst}\, y^{\lambda}_{ijst}(r_{it}) + \mathbb{E}\big\{ V^{\hat{\lambda}}_{i,t+1}(x^{\lambda}_{it}(r_{it}) + Q_{i,t+1}) \big\}.$$

Subtracting the first expression from the second one, we obtain

$$V_{it}^{\hat{\lambda}}(r_{it}) - V_{it}^{\lambda}(r_{it}) \geq -\sum_{j \in \mathcal{C}} \sum_{s=0}^{S-1} y_{ijst}^{\lambda}(r_{it}) \big[\hat{\lambda}_{jst} - \lambda_{jst}\big] \\ + \mathbb{E}\big\{V_{i,t+1}^{\hat{\lambda}}(x_{it}^{\lambda}(r_{it}) + Q_{i,t+1}) - V_{i,t+1}^{\lambda}(x_{it}^{\lambda}(r_{it}) + Q_{i,t+1})\big\}. \quad (47)$$

The expectation on the right side above can be written as

$$\sum_{r_{i,t+1} \in \mathcal{R}} \mathbb{P}\{x_{it}^{\lambda}(r_{it}) + Q_{i,t+1} = r_{i,t+1}\} \big[V_{i,t+1}^{\hat{\lambda}}(r_{i,t+1}) - V_{i,t+1}^{\lambda}(r_{i,t+1})\big],$$

which implies that (47) can be written in matrix notation as

$$V_{it}^{\hat{\lambda}} - V_{it}^{\lambda} \geq -\sum_{j \in \mathcal{C}} Y_{ijt}^{\lambda} \big[\hat{\lambda}_{jt} - \lambda_{jt}\big] + P_{it}^{\lambda} \big[V_{i,t+1}^{\hat{\lambda}} - V_{i,t+1}^{\lambda}\big]. \quad (48)$$

The result follows by using the induction assumption

$$V_{i,t+1}^{\hat{\lambda}} \geq V_{i,t+1}^{\lambda} - \sum_{j \in \mathcal{C}} Y_{ij,t+1}^{\lambda} \big[\hat{\lambda}_{j,t+1} - \lambda_{j,t+1}\big] - P_{i,t+1}^{\lambda} \sum_{j \in \mathcal{C}} Y_{ij,t+2}^{\lambda} \big[\hat{\lambda}_{j,t+2} - \lambda_{j,t+2}\big] \\ - \ldots - P_{i,t+1}^{\lambda} P_{i,t+2}^{\lambda} \ldots P_{i,\tau-1}^{\lambda} \sum_{j \in \mathcal{C}} Y_{ij\tau}^{\lambda} \big[\hat{\lambda}_{j\tau} - \lambda_{j\tau}\big]$$

in (48) and noting that the matrix P_{it}^{λ} has positive entries. □

Letting $\Pi_{ijt}^{\lambda} = P_{i1}^{\lambda} P_{i2}^{\lambda} \ldots P_{i,t-1}^{\lambda} Y_{ijt}^{\lambda}$ with $\Pi_{ij1}^{\lambda} = Y_{ij1}^{\lambda}$, we have

$$V_{i1}^{\hat{\lambda}} \geq V_{i1}^{\lambda} - \sum_{j \in \mathcal{C}} \Pi_{ij1}^{\lambda} \big[\hat{\lambda}_{j1} - \lambda_{j1}\big] - \sum_{j \in \mathcal{C}} \Pi_{ij2}^{\lambda} \big[\hat{\lambda}_{j2} - \lambda_{j2}\big] - \ldots - \sum_{j \in \mathcal{C}} \Pi_{ij\tau}^{\lambda} \big[\hat{\lambda}_{j\tau} - \lambda_{j\tau}\big]$$

by Proposition 7. In this case, letting β_i be the vector $\{\beta_i(r_{i1}) : r_{i1} \in \mathcal{R}\}$, we obtain

$$\sum_{i \in \mathcal{P}} \sum_{r_{i1} \in \mathcal{R}} \beta_i(r_{i1})\, V_{i1}^{\hat{\lambda}}(r_{i1}) = \sum_{i \in \mathcal{P}} \beta_i\, V_{i1}^{\hat{\lambda}} \geq \sum_{i \in \mathcal{P}} \beta_i\, V_{i1}^{\lambda} - \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{C}} \beta_i\, \Pi_{ijt}^{\lambda} \big[\hat{\lambda}_{jt} - \lambda_{jt}\big]. \quad (49)$$

Therefore, $\sum_{i \in \mathcal{P}} \sum_{r_{i1} \in \mathcal{R}} \beta_i(r_{i1})\, V_{i1}^{\lambda}(r_{i1})$ has a subgradient, and hence, Theorem 3.2.6 in Bazaraa, Sherali & Shetty (1993) implies that $\sum_{i \in \mathcal{P}} \sum_{r_{i1} \in \mathcal{R}} \beta_i(r_{i1})\, V_{i1}^{\lambda}(r_{i1})$ is a convex function of λ.

Our use of Benders decomposition to solve problem (38) bears close resemblance to Benders decomposition for two-stage stochastic programs (Ruszczynski (2003)). We iteratively solve a master problem of the form

$$\begin{aligned} \min \quad & \sum_{t \in \mathcal{T}} \sum_{j \in \mathcal{C}} \sum_{s=0}^{S-1} \lambda_{jst} + v && (50) \\ \text{subject to} \quad & v \geq \sum_{i \in \mathcal{P}} \beta_i\, V_{i1}^{\lambda^n} - \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{C}} \beta_i\, \Pi_{ijt}^{\lambda^n} \left[\lambda_{jt} - \lambda_{jt}^n\right] \qquad \forall\, n = 1, \ldots, k-1 && (51) \\ & \lambda_{jst} \geq 0 \qquad \forall\, j \in \mathcal{C},\ s = 0, \ldots, S-1,\ t \in \mathcal{T} && (52) \end{aligned}$$

at iteration k, where v, $\{\lambda_{jst} : j \in \mathcal{C},\ s = 0, \ldots, S-1,\ t \in \mathcal{T}\}$ are the decision variables. Constraints (51) are the cutting planes that represent the function $\sum_{i \in \mathcal{P}} \sum_{r_{i1} \in \mathcal{R}} \beta_i(r_{i1})\, V_{i1}^{\lambda}(r_{i1})$ and they are constructed iteratively by using the solution to the master problem and the subgradient provided by (49). Since the decision variable v only appears on the left side of constraints (51), it is easy to see that the set of feasible solutions to the master problem is nonempty. We have $k = 1$ at the first iteration. Therefore, the master problem does not include cutting planes at the first iteration and its optimal objective value is $-\infty$. In practical implementations, we add the constraint $v \geq -M$ to the master problem at the first iteration, where M is a large but finite number. This ensures that the optimal objective value of the master problem is bounded from below by $-M$.

We let (v^k, λ^k) be the solution to the master problem at iteration k. After solving the master problem, we compute $V_{i1}^{\lambda^k}(r_{i1})$ for all $r_{i1} \in \mathcal{R}$, $i \in \mathcal{P}$. By Lemma 4, this can be done very efficiently. From (51), we always have

$$v^k = \max_{n \in \{1,\ldots,k-1\}} \Big\{ \sum_{i \in \mathcal{P}} \beta_i\, V_{i1}^{\lambda^n} - \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{C}} \beta_i\, \Pi_{ijt}^{\lambda^n} \big[\lambda_{jt}^k - \lambda_{jt}^n\big] \Big\} \leq \sum_{i \in \mathcal{P}} \beta_i\, V_{i1}^{\lambda^k} = \sum_{i \in \mathcal{P}} \sum_{r_{i1} \in \mathcal{R}} \beta_i(r_{i1})\, V_{i1}^{\lambda^k}(r_{i1}),$$

where the inequality follows from (49). If we have $v^k = \sum_{i \in \mathcal{P}} \sum_{r_{i1} \in \mathcal{R}} \beta_i(r_{i1})\, V_{i1}^{\lambda^k}(r_{i1})$, then for any set of positive Lagrange multipliers λ, we have

$$\begin{aligned}
\sum_{t \in \mathcal{T}} \sum_{j \in \mathcal{C}} \sum_{s=0}^{S-1} \lambda_{jst}^k + \sum_{i \in \mathcal{P}} \sum_{r_{i1} \in \mathcal{R}} \beta_i(r_{i1})\, V_{i1}^{\lambda^k}(r_{i1}) &= \sum_{t \in \mathcal{T}} \sum_{j \in \mathcal{C}} \sum_{s=0}^{S-1} \lambda_{jst}^k + v^k \\
&= \sum_{t \in \mathcal{T}} \sum_{j \in \mathcal{C}} \sum_{s=0}^{S-1} \lambda_{jst}^k + \max_{n \in \{1,\ldots,k-1\}} \Big\{ \sum_{i \in \mathcal{P}} \beta_i\, V_{i1}^{\lambda^n} - \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{C}} \beta_i\, \Pi_{ijt}^{\lambda^n} \big[\lambda_{jt}^k - \lambda_{jt}^n\big] \Big\} \\
&\leq \sum_{t \in \mathcal{T}} \sum_{j \in \mathcal{C}} \sum_{s=0}^{S-1} \lambda_{jst} + \max_{n \in \{1,\ldots,k-1\}} \Big\{ \sum_{i \in \mathcal{P}} \beta_i\, V_{i1}^{\lambda^n} - \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{C}} \beta_i\, \Pi_{ijt}^{\lambda^n} \big[\lambda_{jt} - \lambda_{jt}^n\big] \Big\} \\
&\leq \sum_{t \in \mathcal{T}} \sum_{j \in \mathcal{C}} \sum_{s=0}^{S-1} \lambda_{jst} + \sum_{i \in \mathcal{P}} \beta_i\, V_{i1}^{\lambda} = \sum_{t \in \mathcal{T}} \sum_{j \in \mathcal{C}} \sum_{s=0}^{S-1} \lambda_{jst} + \sum_{i \in \mathcal{P}} \sum_{r_{i1} \in \mathcal{R}} \beta_i(r_{i1})\, V_{i1}^{\lambda}(r_{i1}),
\end{aligned}$$

where the first inequality follows from the fact that (v^k, λ^k) is an optimal solution to the master problem and the second inequality follows from (49). Thus, if we have $v^k = \sum_{i \in \mathcal{P}} \sum_{r_{i1} \in \mathcal{R}} \beta_i(r_{i1})\, V_{i1}^{\lambda^k}(r_{i1})$, then λ^k is an optimal solution to problem (38) and we stop. On the other hand, if we have $v^k < \sum_{i \in \mathcal{P}} \sum_{r_{i1} \in \mathcal{R}} \beta_i(r_{i1})\, V_{i1}^{\lambda^k}(r_{i1})$, then we construct the constraint

$$v \geq \sum_{i \in \mathcal{P}} \beta_i\, V_{i1}^{\lambda^k} - \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{C}} \beta_i\, \Pi_{ijt}^{\lambda^k} \big[\lambda_{jt} - \lambda_{jt}^k\big], \tag{53}$$

add it to the master problem, increase k by 1 and resolve the master problem. We summarize the Benders decomposition idea in Figure 3.

By using an argument similar to the one used to show the convergence of Benders decomposition for two-stage stochastic programs, we can show that the Benders decomposition method described above terminates after a finite number of iterations with an optimal solution to problem (38).

Proposition 8. *The Benders decomposition method for solving problem* (38) *terminates after a finite number of iterations with an optimal solution.*

Proof. There are a finite number of solutions in $\mathcal{Y}_i(r_{it})$. Therefore, there is a finite set of $|\mathcal{R}| \times S$-dimensional matrices such that Y^λ_{ijt} takes a value in this set for any λ. Similarly, there exists a finite set of $|\mathcal{R}| \times |\mathcal{R}|$-dimensional matrices such that P^λ_{it} takes a value in this set for any λ. Consequently, there exists a finite set of S-dimensional vectors such that $\beta_i \, \Pi^\lambda_{ijt}$ takes a value in this set for any λ.

Letting p^λ_{it} be the vector $\{p_{it}(x^\lambda_{it}(r_{it}), y^\lambda_{it}(r_{it})) : r_{it} \in \mathcal{R}\}$, we write (34) in matrix notation as

$$V^\lambda_{it} = p^\lambda_{it} - \sum_{j \in \mathcal{C}} Y^\lambda_{ijt} \, \lambda_{jt} + P^\lambda_{it} \, V^\lambda_{i,t+1}.$$

Using this expression and backward induction on time periods, it is easy to show that

$$V^\lambda_{i1} = p^\lambda_{i1} + P^\lambda_{i1} \, p^\lambda_{i2} + P^\lambda_{i1} \, P^\lambda_{i2} \, p^\lambda_{i3} + \ldots + P^\lambda_{i1} \, P^\lambda_{i2} \ldots P^\lambda_{i,\tau-1} \, p^\lambda_{i\tau} - \sum_{t \in \mathcal{T}} \sum_{j \in \mathcal{C}} \Pi^\lambda_{ijt} \, \lambda_{jt}. \tag{54}$$

Since there are a finite number of solutions in $\mathcal{Y}_i(r_{it})$, there exists a finite set of $|\mathcal{R}|$-dimensional vectors such that p^λ_{it} takes a value in this set for any λ. This implies that there exists a finite set of scalars such that $\beta_i \big[p^\lambda_{i1} + P^\lambda_{i1} \, p^\lambda_{i2} + P^\lambda_{i1} \, P^\lambda_{i2} \, p^\lambda_{i3} + \ldots + P^\lambda_{i1} \, P^\lambda_{i2} \ldots P^\lambda_{i,\tau-1} \, p^\lambda_{i\tau}\big]$ takes a value in this set for any λ. Using (53) and (54), we write the constraint added to the master problem at iteration k as

$$v \geq \sum_{i \in \mathcal{P}} \beta_i \big[p^{\lambda^k}_{i1} + P^{\lambda^k}_{i1} \, p^{\lambda^k}_{i2} + P^{\lambda^k}_{i1} \, P^{\lambda^k}_{i2} \, p^{\lambda^k}_{i3} + \ldots + P^{\lambda^k}_{i1} \, P^{\lambda^k}_{i2} \ldots P^{\lambda^k}_{i,\tau-1} \, p^{\lambda^k}_{i\tau}\big] - \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{C}} \beta_i \, \Pi^{\lambda^k}_{ijt} \, \lambda_{jt}.$$

There exist a finite number of possible values for $\beta_i \, \Pi^{\lambda^k}_{ijt}$ and $\beta_i \big[p^{\lambda^k}_{i1} + P^{\lambda^k}_{i1} \, p^{\lambda^k}_{i2} + P^{\lambda^k}_{i1} \, P^{\lambda^k}_{i2} \, p^{\lambda^k}_{i3} + \ldots + P^{\lambda^k}_{i1} \, P^{\lambda^k}_{i2} \ldots P^{\lambda^k}_{i,\tau-1} \, p^{\lambda^k}_{i\tau}\big]$. Therefore, the constraint added to the master problem at iteration k is one of the finitely many possible constraints. Using the finiteness of the number of possible cutting planes, we can show the result by following the same argument in Ruszczynski (2003) that is used to show the finite convergence of Benders decomposition for two-stage stochastic programs. $\square$

One problem with Benders decomposition is that the number of cutting planes in the master problem can grow large. We note that since a new cutting plane is added at each iteration, the objective value of the master problem does not decrease from one iteration to the next. If we drop the loose cutting planes that are satisfied as strict inequalities from the master problem, then the objective value still does not decrease from one iteration to the next. It turns out that we can indeed drop the loose cutting planes from the master problem and still ensure that the Benders decomposition method terminates after a finite number of iterations with an optimal solution.

Lemma 9. *Assume that we drop the loose cutting planes from the master problem at iteration k whenever the objective value of the master problem at iteration k is strictly larger than the objective* *value at iteration $k-1$. In this case, the Benders decomposition method for solving problem* (38) *terminates after a finite number of iterations with an optimal solution.*

Proof. The proof follows from an argument similar to the one in Ruszczynski (2003) and uses the finiteness of the number of possible cutting planes. $\square$

In general, dropping all loose cutting planes is not a good idea, since the dropped cutting planes may have to be reconstructed at the later iterations. In our computational experiments, we drop the cutting planes that remain loose for 50 consecutive iterations.

We note that we are not restricted with the methods given in Sections 3.1 and 3.2. Since the objective function of problem (38) is a convex function with subgradients characterized as in Proposition 7, there exist other methods, such as subgradient search, to minimize this nonsmooth convex function. For our problem class, Benders decomposition yields small master problems that can be solved easily. However, subgradient search may be a more attractive option for large applications where solving the master problem becomes problematic.

4. Applying the Greedy Policy

This section shows that applying the greedy policies characterized by the value function approximations requires solving min-cost network flow problems.

Letting $\{\theta_t^* : t \in \mathcal{T}\}$, $\{\vartheta_{it}^* : i \in \mathcal{P},\ t \in \mathcal{T}\}$ be an optimal solution to problem (16)-(18), the value function approximations obtained by the method given in Section 2 are of the form $\theta_t^* + \sum_{i \in \mathcal{P}} \vartheta_{it}^* \, r_{it}$. On the other hand, letting $\{\lambda_{jst}^* : j \in \mathcal{C},\ s = 0, \ldots, S-1,\ t \in \mathcal{T}\}$ be an optimal solution to problem (38) and noting Proposition 3, the value function approximations obtained by the method given in Section 3 are of the form

$$\sum_{t'=t}^{\tau} \sum_{j \in \mathcal{C}} \sum_{s=0}^{S-1} \lambda_{jst'}^* + \sum_{i \in \mathcal{P}} V_{it}^{\lambda^*}(r_{it}).$$

As mentioned in the proof of Lemma 4, $V_{it}^{\lambda^*}(r_{it})$ is a concave function of r_{it}. Therefore, the value function approximations obtained by the methods given in Sections 2 and 3 are of the form $\hat{\theta}_t + \sum_{i \in \mathcal{P}} \hat{V}_{it}(r_{it})$, where $\hat{\theta}_t$ is a constant and $\hat{V}_{it}(r_{it})$ is a concave function of r_{it}.

The greedy policy characterized by the value function approximations $\{\hat{\theta}_t + \sum_{i \in \mathcal{P}} \hat{V}_{it}(r_{it}) : t \in \mathcal{T}\}$ makes the decisions at time period t by solving the problem

$$\max_{(x_t, y_t) \in \mathcal{Y}(r_t)} \left\{ p_t(x_t, y_t) + \hat{\theta}_t + \sum_{i \in \mathcal{P}} \mathbb{E}\big\{ \hat{V}_{i,t+1}(x_{it} + Q_{i,t+1}) \big\} \right\}.$$

Since $\mathbb{E}\{\hat{V}_{i,t+1}(x_{it}+Q_{i,t+1})\}$ is a concave function of x_{it}, letting $\Delta_{i\ell,t+1} = \mathbb{E}\{\hat{V}_{i,t+1}(\ell+1+Q_{i,t+1})\} - \mathbb{E}\{\hat{V}_{i,t+1}(\ell + Q_{i,t+1})\}$ for all $\ell = 0,\dots,L-1$ and associating the decision variables $\{z_{i\ell,t+1} : \ell = 0,\dots,L-1\}$ with these first differences, the problem above can be written as

$$\begin{aligned}
\max \quad & \sum_{i\in\mathcal{P}}\sum_{j\in\mathcal{C}}\sum_{s=0}^{S}\big[f_{jst} - c_{ijt}\big] y_{ijst} - \sum_{i\in\mathcal{P}} h_{it}\, x_{it} + \hat{\theta}_t + \sum_{i\in\mathcal{P}} \mathbb{E}\{\hat{V}_{i,t+1}(Q_{i,t+1})\} + \sum_{i\in\mathcal{P}}\sum_{\ell=0}^{L-1} \Delta_{i\ell,t+1}\, z_{i\ell,t+1} \\
\text{subject to} \quad & x_{it} + \sum_{j\in\mathcal{C}}\sum_{s=0}^{S} y_{ijst} = r_{it} && \forall\, i\in\mathcal{P} && (55) \\
& x_{it} - \sum_{\ell=0}^{L-1} z_{i\ell,t+1} = 0 && \forall\, i\in\mathcal{P} && (56) \\
& \sum_{i\in\mathcal{P}} y_{ijst} \le 1 && \forall\, j\in\mathcal{C},\ s=0,\dots,S-1 && (57) \\
& z_{i\ell,t+1} \le 1 && \forall\, i\in\mathcal{P},\ \ell=0,\dots,L-1 && (58) \\
& x_{it},\, y_{ijst},\, z_{i\ell,t+1} \in \mathbb{Z}_+ && \forall\, i\in\mathcal{P},\ j\in\mathcal{C},\ s=0,\dots,S,\ \ell=0,\dots,L-1.
\end{aligned}$$

We define the new decision variables $\{\zeta_{jst} : j\in\mathcal{C},\ s=0,\dots,S-1\}$, and using these decision variables, we split constraints (57) into $\sum_{i\in\mathcal{P}} y_{ijst} - \zeta_{jst} = 0$ and $\zeta_{jst} \le 1$ for all $j\in\mathcal{C}$, $s=0,\dots,S-1$. In this case, dropping the constant terms in the objective function, the problem above can be written as

$$\begin{aligned}
\max \quad & \sum_{i\in\mathcal{P}}\sum_{j\in\mathcal{C}}\sum_{s=0}^{S}\big[f_{jst} - c_{ijt}\big] y_{ijst} - \sum_{i\in\mathcal{P}} h_{it}\, x_{it} + \sum_{i\in\mathcal{P}}\sum_{\ell=0}^{L-1} \Delta_{i\ell,t+1}\, z_{i\ell,t+1} && && (59) \\
\text{subject to} \quad & (55), (56), (58) && && (60) \\
& \sum_{i\in\mathcal{P}} y_{ijst} - \zeta_{jst} = 0 && \forall\, j\in\mathcal{C},\ s=0,\dots,S-1 && (61) \\
& \zeta_{jst} \le 1 && \forall\, j\in\mathcal{C},\ s=0,\dots,S-1 && (62) \\
& x_{it},\, y_{ijst},\, z_{i\ell,t+1},\, \zeta_{js't} \in \mathbb{Z}_+ && \forall\, i\in\mathcal{P},\ j\in\mathcal{C},\ s=0,\dots,S, \\
& && \qquad \ell=0,\dots,L-1,\ s'=0,\dots,S-1. && (63)
\end{aligned}$$

Defining three sets of nodes $\mathcal{O}_1 = \mathcal{P}$, $\mathcal{O}_2 = \mathcal{P}$ and $\mathcal{O}_3 = \mathcal{C}\times\{0,\dots,S-1\}$, it is easy to see that problem (59)-(63) is a min-cost network flow problem that takes place over a network with the set of nodes $\mathcal{O}_1 \bigcup \mathcal{O}_2 \bigcup \mathcal{O}_3 \bigcup \{\phi\}$ shown in Figure 4. Corresponding to each decision variable in problem (59)-(63), there exists an arc in the network in Figure 4. The arc corresponding to decision variable x_{it} leaves node $i\in\mathcal{O}_1$ and enters node $i\in\mathcal{O}_2$. Whenever $s\in\{0,\dots,S-1\}$, the arc corresponding to decision variable y_{ijst} leaves node $i\in\mathcal{O}_1$ and enters node $(j,s)\in\mathcal{O}_3$. The arc corresponding to decision variable y_{ijSt} leaves node $i\in\mathcal{O}_1$ and enters node ϕ. The arc corresponding to decision variable $z_{i\ell,t+1}$ leaves node $i\in\mathcal{O}_2$ and enters node ϕ. Finally, the arc corresponding to decision variable ζ_{jst} leaves node $(j,s)\in\mathcal{O}_3$ and enters node ϕ. Constraints (55), (56) and (61) in problem (59)-(63) are respectively the flow balance constraints for the nodes in $\mathcal{O}_1$, $\mathcal{O}_2$ and $\mathcal{O}_3$. The supplies of the nodes in $\mathcal{O}_1$ are $\{r_{it} : i\in\mathcal{P}\}$.

5. Computational Experiments

In this section, we numerically test the performances of the approximate dynamic programming methods given in Sections 2 and 3.

5.1. Experimental setup and benchmark strategy

All of our test problems involve 41 customer locations spread over a 1000×1000 region. We let $c_{ijt} = \bar{c}\, d_{ij}$, where $\bar{c}$ is the shipping cost applied on a per-mile basis and d_{ij} is the Euclidean distance from plant i to customer location j. From (1), the expected profit function at customer location j at time period t depends on σ_{jt} and $\rho_{jt} + \pi_{jt}$, and hence, we let $\pi_{jt} = 0$ without loss of generality. For all $i \in \mathcal{P}$, $j \in \mathcal{C}$, $t \in \mathcal{T}$, we sample σ_{jt}, ρ_{jt} and h_{it} from the uniform distributions over $[0.5\,\bar{\sigma}, 1.5\,\bar{\sigma}]$, $[0.5\,\bar{\rho}, 1.5\,\bar{\rho}]$ and $[0.5\,\bar{h}, 1.5\,\bar{h}]$ respectively. In all of our test problems, we let $\bar{c} = 1.6$, $\bar{\rho} = 1000$ and $\bar{h} = 20$. We vary the other parameters to obtain test problems with different characteristics.

We model the production random variables through mixtures of uniformly distributed random variables. In particular, we let $Q_{it} = \sum_{n=1}^{N} \mathbf{1}(X_{it} = n)\, U_{it}^n$, where X_{it} is uniformly distributed over $\{1, \ldots, N\}$ and U_{it}^n is uniformly distributed over $\{a_{it}^n, \ldots, b_{it}^n\}$ for all $n = 1, \ldots, N$. This allows us to change the variance of Q_{it} in any way we like by changing N and $\{(a_{it}^n, b_{it}^n) : n = 1, \ldots, N\}$. Furthermore, we can accurately approximate any random variable with a discrete distribution and a finite support by using mixtures of uniformly distributed random variables. When presenting the results, we give the coefficients of variation of the production random variables.

The benchmark strategy we use is the so-called rolling horizon method. For a given rolling horizon length K, this method solves an optimization problem that spans K time periods and uses the point forecasts of the future production quantities. In particular, if the state vector at time period t is r_t, then the rolling horizon method makes the decisions by solving the problem

$$\max \quad -\sum_{t'=t}^{t+K-1} \sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{C}} c_{ijt'}\, u_{ijt'} + \sum_{t'=t}^{t+K-1} \sum_{j \in \mathcal{C}} F_{jt'}(w_{jt'}) - \sum_{t'=t}^{t+K-1} \sum_{i \in \mathcal{P}} h_{it'}\, x_{it'} \tag{64}$$

$$\text{subject to} \quad \sum_{j \in \mathcal{C}} u_{ijt} + x_{it} = r_{it} \qquad \forall\, i \in \mathcal{P} \tag{65}$$

$$\sum_{j \in \mathcal{C}} u_{ijt'} + x_{it'} - x_{i,t'-1} = \mathbb{E}\{Q_{it'}\} \qquad \forall\, i \in \mathcal{P},\ t' = t+1, \ldots, t+K-1 \tag{66}$$

$$\sum_{i \in \mathcal{P}} u_{ijt'} - w_{jt'} = 0 \qquad \forall\, j \in \mathcal{C},\ t' = t, \ldots, t+K-1 \tag{67}$$

$$x_{it'} \leq L \qquad \forall\, i \in \mathcal{P},\ t' = t, \ldots, t+K-1 \tag{68}$$

$$u_{ijt'},\, w_{jt'},\, x_{it'} \in \mathbb{R}_+ \qquad \forall\, i \in \mathcal{P},\ j \in \mathcal{C},\ t' = t, \ldots, t+K-1. \tag{69}$$

(If we have $t + K - 1 > \tau$, then we substitute τ for $t + K - 1$ in the problem above.) Although this problem includes decision variables for time periods $t, \ldots, t+K-1$, we only implement the decisions for time period t and solve a similar problem to make the decisions for time period $t+1$. The rolling horizon method is expected to give better solutions as K increases. For our test problems, increasing K beyond 8 time periods provides marginal improvements in the objective value.

We let $\alpha(r_1) = 1/|\mathcal{R}|^{|\mathcal{P}|}$ for all $r_1 \in \mathcal{R}^{|\mathcal{P}|}$. Setup runs showed that changing these weights does not noticeably affect the performances of the methods given in Sections 2 and 3.

5.2. Computational results

In Section 3, we give two methods to minimize the dual function. Setup runs showed that the Benders decomposition method in Section 3.2 is significantly faster than the constraint generation method in Section 3.1. Therefore, we use the Benders decomposition method to minimize the dual function.

However, the Benders decomposition method has slow tail performance in the sense that the improvement in the objective value of the master problem slows down as the iterations progress. We deal with this difficulty by solving problem (38) only approximately. In particular, letting λ^* be an optimal solution to problem (38) and (v^k, λ^k) be an optimal solution to the master problem at iteration k, we have

$$
\begin{aligned}
\sum_{t\in\mathcal{T}}\sum_{j\in\mathcal{C}}\sum_{s=0}^{S-1}\lambda_{jst}^k + v^k &= \sum_{t\in\mathcal{T}}\sum_{j\in\mathcal{C}}\sum_{s=0}^{S-1}\lambda_{jst}^k + \max_{n\in\{1,\dots,k-1\}}\left\{\sum_{i\in\mathcal{P}}\beta_i\, V_{i1}^{\lambda^n} - \sum_{t\in\mathcal{T}}\sum_{i\in\mathcal{P}}\sum_{j\in\mathcal{C}}\beta_i\,\Pi_{ijt}^{\lambda^n}\left[\lambda_{jt}^k - \lambda_{jt}^n\right]\right\} \\
&\le \sum_{t\in\mathcal{T}}\sum_{j\in\mathcal{C}}\sum_{s=0}^{S-1}\lambda_{jst}^* + \max_{n\in\{1,\dots,k-1\}}\left\{\sum_{i\in\mathcal{P}}\beta_i\, V_{i1}^{\lambda^n} - \sum_{t\in\mathcal{T}}\sum_{i\in\mathcal{P}}\sum_{j\in\mathcal{C}}\beta_i\,\Pi_{ijt}^{\lambda^n}\left[\lambda_{jt}^* - \lambda_{jt}^n\right]\right\} \\
&\le \sum_{t\in\mathcal{T}}\sum_{j\in\mathcal{C}}\sum_{s=0}^{S-1}\lambda_{jst}^* + \sum_{i\in\mathcal{P}}\beta_i\, V_{i1}^{\lambda^*} = \sum_{t\in\mathcal{T}}\sum_{j\in\mathcal{C}}\sum_{s=0}^{S-1}\lambda_{jst}^* + \sum_{i\in\mathcal{P}}\sum_{r_{i1}\in\mathcal{R}}\beta_i(r_{i1})\, V_{i1}^{\lambda^*}(r_{i1}) \\
&\le \sum_{t\in\mathcal{T}}\sum_{j\in\mathcal{C}}\sum_{s=0}^{S-1}\lambda_{jst}^k + \sum_{i\in\mathcal{P}}\sum_{r_{i1}\in\mathcal{R}}\beta_i(r_{i1})\, V_{i1}^{\lambda^k}(r_{i1}),
\end{aligned}
$$

where the first inequality follows from the fact that (v^k, λ^k) is an optimal solution to the master problem, the second inequality follows from (49) and the third inequality follows from the fact that λ^* is an optimal solution to problem (38). Therefore, the first and last terms in the chain of inequalities above give lower and upper bounds on the optimal objective value of problem (38). In Figure 5, we plot the percent gap between the lower and upper bounds as a function of the iteration number k for a particular test problem, along with the total expected profit that is obtained by the greedy policy characterized by the value function approximations $\{\sum_{t'=t}^{\mathcal{T}}\sum_{j\in\mathcal{C}}\sum_{s=0}^{S-1}\lambda_{jst'}^k + \sum_{i\in\mathcal{P}} V_{it}^{\lambda^k}(r_{it}) : t\in\mathcal{T}\}$. This figure shows that the Benders decomposition method has slow tail performance, but the quality of the greedy policy does not improve after the first few iterations. Consequently, we stop the Benders decomposition method when the percent gap between the lower and upper bounds is less than 10%. This does not noticeably affect the quality of the greedy policy. Such slow tail performance is also reported in Yost & Washburn (2000). Magnanti & Wong (1981) and Ruszczynski (2003) show that choosing the cutting planes "carefully" and using regularized Benders decomposition or trust region methods may remedy this difficulty.

We summarize our results in Tables 1-4. In these tables, the first column gives the characteristics of the test problems, where τ is the length of the planning horizon, $|\mathcal{P}|$ is the number of plants, P is the number of plants that serves a particular customer location (we assume that each customer location is served by the closest P plants), $\bar{\sigma}$ is the average salvage value, $\bar{Q}$ is the total expected production quantity (that is, $\bar{Q} = \mathbb{E}\{\sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{P}} Q_{it}\}$) and $\bar{V}$ is the average coefficient of variation of the production random variables (that is, $\bar{V}$ is the average of $\{\sqrt{Var(Q_{it})}/\mathbb{E}\{Q_{it}\} : i \in \mathcal{P},\ t \in \mathcal{T}\}$). The second set of columns give the performance of the linear programming-based method (LP). Letting $\{\theta_t^* + \sum_{i \in \mathcal{P}} \vartheta_{it}^* \, r_{it} : t \in \mathcal{T}\}$ be the value function approximations obtained by LP, the first one of these columns gives the ratio of the total expected profit that is obtained by the greedy policy characterized by these value function approximations to the total expected profit obtained by the 8-period rolling horizon method (RH). To estimate the total expected profit that is obtained by the greedy policy, we simulate the behavior of the greedy policy for 500 different samples of $\{Q_{it} : i \in \mathcal{P},\ t \in \mathcal{T}\}$. The second column gives the number of constraints added to the master problem. The third column gives the CPU seconds needed to solve problem (16)-(18). The fourth and fifth columns give what percents of the CPU seconds are spent on solving the master problem and constructing the constraints. The third set of columns give the performance of the Lagrangian relaxation-based method (LG). Letting $\{\sum_{t'=t}^{\tau} \sum_{j \in \mathcal{C}} \sum_{s=0}^{S-1} \lambda_{jst'}^* + \sum_{i \in \mathcal{P}} V_{it}^{\lambda^*}(r_{it}) : t \in \mathcal{T}\}$ be the value function approximations obtained by LG, the first one of these columns gives the ratio of the total expected profit that is obtained by the greedy policy characterized by these value function approximations to the total expected profit obtained by RH. The second and third columns give the number of cutting planes and the CPU seconds needed to solve problem (38) with 10% optimality gap. The fourth and fifth columns give what percents of the CPU seconds are spent on solving the master problem and constructing the cutting planes. Letting λ^0 be a trivial feasible solution to problem (38) consisting of all zeros, the sixth column gives the ratio of the total expected profit that is obtained by the greedy policy characterized by the value function approximations $\{\sum_{t'=t}^{\tau} \sum_{j \in \mathcal{C}} \sum_{s=0}^{S-1} \lambda_{jst'}^0 + \sum_{i \in \mathcal{P}} V_{it}^{\lambda^0}(r_{it}) : t \in \mathcal{T}\}$ to the total expected profit obtained by RH. Consequently, the gap between the columns labeled "Prf" and "In Prf" shows the significance of finding a near-optimal solution to problem (38).

There are several observations that we can make from Tables 1-4. On a majority of the test problems, LP performs worse than RH, whereas LG performs better than RH. (Almost all of the differences are statistically significant at the 5% level.) The CPU seconds and the number of constraints for LP show less variation among different test problems than the CPU seconds and the number of cutting planes for LG. Comparing the columns labeled "Prf" and "In Prf" shows that finding a near-optimal solution to problem (38) significantly improves the quality of the greedy policy obtained by LG. For some test problems, the greedy policy characterized by the value function approximations $\{\sum_{t'=t}^{\tau}\sum_{j\in\mathcal{C}}\sum_{s=0}^{S-1}\lambda^0_{jst'}+\sum_{i\in\mathcal{P}}V^{\lambda^0}_{it}(r_{it}) : t\in\mathcal{T}\}$ performs better than LP. Therefore, simply ignoring the constraints that link the decisions for different plants can provide better policies than using linear value function approximations. Nevertheless, this approach is not a good idea in general. The last row in Table 3 shows that the total expected profit obtained by this approach can be almost half of the total expected profit obtained by RH.

Our computational results complement the findings in Adelman & Mersereau (2004) in an interesting fashion. Adelman & Mersereau (2004) show that if the linear programming-based method uses nonlinear approximations of the value functions, then it provides tighter upper bounds on the value functions than does the Lagrangian relaxation-based method. However, for our problem class, if the linear programming-based method uses nonlinear approximations of the value functions, then constraint generation requires solving integer programs, which can be computationally prohibitive. Consequently, although Adelman & Mersereau (2004) show that the linear programming-based method is superior to the Lagrangian relaxation-based method when it uses a nonlinear "approximation architecture," our computational results indicate that the linear programming-based method along with a linear "approximation architecture" can be inferior to the Lagrangian relaxation-based method.

We proceed to examine Tables 1-4 in detail. Table 1 shows the results for problems with different values of P. For each value of P, we use low, moderate and high values for the coefficients variation of the production random variables. As the number of plants that can serve a particular customer location increases, the performance gap between LG and RH diminishes. This is due to the fact that if a customer location can be served by a large number of plants, then it is possible to make up an inventory shortage in one plant by using the inventory in another plant. In this case, it is not crucial to make the "correct" inventory allocation decisions and RH performs almost as well as LG. It is also interesting to note that the performance of the greedy policy characterized by the value function approximations $\{\sum_{t'=t}^{\tau}\sum_{j\in\mathcal{C}}\sum_{s=0}^{S-1}\lambda^0_{jst'}+\sum_{i\in\mathcal{P}}V^{\lambda^0}_{it}(r_{it}) : t\in\mathcal{T}\}$ gets better as P decreases. This shows that if P is small, then simply ignoring the constraints that link the decisions for different plants can provide good policies. Finally, the performance gap between LG and RH gets larger as the coefficients of variation of the production random variables get large.

Table 2 shows the results for problems with different values of $\bar{\sigma}$. As the salvage value increases, a large portion of the inventory at the plants is shipped to the customer locations to exploit the high salvage value and the incentive to store inventory decreases. This reduces the value of a dynamic programming model that carefully balances the inventory holding decisions with the shipment decisions, and the performance gap between LG and RH diminishes.

Table 3 shows the results for problems with different values of $\bar{Q}$. As the total expected production quantity increases, the product becomes more abundant and it is not crucial to make the "correct" inventory allocation decisions. As a result, the performance gap between LG and RH diminishes.

Finally, Table 4 shows the results for problems with different dimensions. The CPU seconds and the number of constraints for LP increase as τ or $|\mathcal{P}|$ increases. However, the CPU seconds and the number of cutting planes for LG do not change in a systematic fashion. (This has been the case for many other test problems we worked on.) Nevertheless, as shown in Figure 5, the quality of the greedy policy obtained by LG is quite good even after a few iterations and problem (38) does not have to be solved to optimality. This observation is consistent with that of Cheung & Powell (1996), where the authors carry out only a few iterations of a subgradient search algorithm to obtain a good lower bound on the recourse function arising from a multi-period stochastic program.

6. Conclusion

We presented two approximate dynamic programming methods for an inventory allocation problem under uncertainty. Computational experiments showed that the Lagrangian relaxation-based method performs significantly better than the linear programming-based method and the rolling horizon method. It appears that a model that explicitly uses the full distributions of the production random variables can yield better decisions than the linear programming-based method and the rolling horizon method, which use only the expected values of the production random variables (see problems (16)-(18) and (64)-(69)). The magnitude of the improvement obtained by the Lagrangian relaxation-based method over the other methods depends on the problem parameters. Tables 1-4 indicate that the Lagrangian relaxation-based method is particularly useful when a customer location can be served by a few plants, when the salvage value for the product is low, when the product is scarce and when the variability in the production quantities is high.

The Lagrangian relaxation-based method offers promising research opportunities. There are many dynamic programs where the evolutions of the different components of the state variable are affected by different types of decisions and these different types of decisions interact through a few linking constraints. For example, almost every problem that involves dynamic allocation of a fixed amount of resource to independent activities is of this nature. It is interesting to see what improvement the Lagrangian relaxation-based method will provide over other solution methods in different application settings.

7. Acknowledgements

The authors acknowledge the comments of two anonymous referees that tightened the presentation significantly. This work was supported in part by National Science Foundation grant DMI-0422133.

8. Appendix

Proof of Lemma 1. Since $F_{jt}(\cdot)$ is a piecewise-linear concave function with points of nondifferentiability being a subset of positive integers, noting that $f_{jst} = \sigma_{jt}$ for all $s = S, S+1, \ldots$ and associating the decision variables $\{z_{jst} : s = 0, \ldots, S\}$ with the first differences of $F_{jt}(\cdot)$, problem (2)-(6) can be written as

$$
\begin{aligned}
V_t(r_t) = \max \quad & -\sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{C}} c_{ijt}\, u_{ijt} + \sum_{j \in \mathcal{C}} \sum_{s=0}^{S} f_{jst}\, z_{jst} - \sum_{i \in \mathcal{P}} h_{it}\, x_{it} + \mathbb{E}\big\{V_{t+1}(x_t + Q_{t+1})\big\} \\
\text{subject to} \quad & (3), (5) \\
& \sum_{i \in \mathcal{P}} u_{ijt} - \sum_{s=0}^{S} z_{jst} = 0 && \forall\, j \in \mathcal{C} \\
& z_{jst} \le 1 && \forall\, j \in \mathcal{C},\ s = 0, \ldots, S-1 \\
& u_{ijt},\, x_{it},\, z_{jst} \in \mathbb{Z}_+ && \forall\, i \in \mathcal{P},\ j \in \mathcal{C},\ s = 0, \ldots, S.
\end{aligned}
$$

(See Nemhauser & Wolsey (1988) for more on embedding piecewise-linear concave functions in optimization problems.) Defining the new decision variables $\{y_{ijst} : i \in \mathcal{P},\ j \in \mathcal{C},\ s = 0, \ldots, S\}$ and substituting $\sum_{s=0}^{S} y_{ijst}$ for u_{ijt}, the problem above becomes

$$
\begin{aligned}
V_t(r_t) = \max \quad & -\sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{C}} \sum_{s=0}^{S} c_{ijt}\, y_{ijst} + \sum_{j \in \mathcal{C}} \sum_{s=0}^{S} f_{jst}\, z_{jst} - \sum_{i \in \mathcal{P}} h_{it}\, x_{it} + \mathbb{E}\big\{V_{t+1}(x_t + Q_{t+1})\big\} && (70) \\
\text{subject to} \quad & (5) && (71) \\
& x_{it} + \sum_{j \in \mathcal{C}} \sum_{s=0}^{S} y_{ijst} = r_{it} \qquad \forall\, i \in \mathcal{P} && (72) \\
& \sum_{i \in \mathcal{P}} \sum_{s=0}^{S} y_{ijst} - \sum_{s=0}^{S} z_{jst} = 0 \qquad \forall\, j \in \mathcal{C} && (73) \\
& z_{jst} \le 1 \qquad \forall\, j \in \mathcal{C},\ s = 0, \ldots, S-1 && (74) \\
& x_{it},\, y_{ijst},\, z_{jst} \in \mathbb{Z}_+ \qquad \forall\, i \in \mathcal{P},\ j \in \mathcal{C},\ s = 0, \ldots, S. && (75)
\end{aligned}
$$

By Lemma 10 below, we can substitute $\sum_{i \in \mathcal{P}} y_{ijst}$ for z_{jst} in problem (70)-(75), in which case constraints (73) become redundant and the result follows. $\square$

Lemma 10. *There exists an optimal solution (x_t^*, y_t^*, z_t^*) to problem (70)-(75) that satisfies $\sum_{i \in \mathcal{P}} y_{ijst}^* = z_{jst}^*$ for all $j \in \mathcal{C}$, $s = 0, \ldots, S$.*

Proof of Lemma 10. We let (x_t^*, y_t^*, z_t^*) be an optimal solution to problem (70)-(75), $\mathcal{I}^+ = \{(j,s) : \sum_{i\in\mathcal{P}} y_{ijst}^* > z_{jst}^*\}$ and $\mathcal{I}^- = \{(j,s) : \sum_{i\in\mathcal{P}} y_{ijst}^* < z_{jst}^*\}$. If we have $|\mathcal{I}^+| + |\mathcal{I}^-| = 0$, then we are done. Assume that we have $|\mathcal{I}^+| + |\mathcal{I}^-| > 0$. We now construct another optimal solution $(\hat{x}_t, \hat{y}_t, \hat{z}_t)$ with $|\hat{\mathcal{I}}^+| + |\hat{\mathcal{I}}^-| < |\mathcal{I}^+| + |\mathcal{I}^-|$, where we use $\hat{\mathcal{I}}^+ = \{(j,s) : \sum_{i\in\mathcal{P}} \hat{y}_{ijst} > \hat{z}_{jst}\}$ and $\hat{\mathcal{I}}^- = \{(j,s) : \sum_{i\in\mathcal{P}} \hat{y}_{ijst} < \hat{z}_{jst}\}$. This establishes the result.

Assume that $(j', s') \in \mathcal{I}^+$. Since (x_t^*, y_t^*, z_t^*) satisfies constraints (73), there exists s'' such that $(j', s'') \in \mathcal{I}^-$. (If we assume that $(j', s'') \in \mathcal{I}^-$, then there exists s' such that $(j', s') \in \mathcal{I}^+$ and the proof remains valid.) We let $\delta = \sum_{i\in\mathcal{P}} y_{ij's't}^* - z_{j's't}^* > 0$ and assume that $\delta \le z_{j's''t}^* - \sum_{i\in\mathcal{P}} y_{ij's''t}^*$. We pick $i_1, \ldots, i_n \in \mathcal{P}$ such that $y_{i_1j's't}^* + \ldots + y_{i_nj's't}^* \ge \delta$ and $y_{i_1j's't}^* + \ldots + y_{i_{n-1},j's't}^* < \delta$.

We let $\hat{x}_{it} = x_{it}^*$, $\hat{z}_{jst} = z_{jst}^*$ for all $i \in \mathcal{P},\ j \in \mathcal{C},\ s = 0, \ldots, S$ and

$$\hat{y}_{ijst} = \begin{cases} 0 & \text{if } i \in \{i_1, \ldots, i_{n-1}\},\ j = j',\ s = s' \\ y_{i_1j's't}^* + \ldots + y_{i_nj's't}^* - \delta & \text{if } i = i_n,\ j = j',\ s = s' \\ y_{ij's''t}^* + y_{ij's't}^* & \text{if } i \in \{i_1, \ldots, i_{n-1}\},\ j = j',\ s = s'' \\ y_{i_nj's''t}^* - y_{i_1j's't}^* - \ldots - y_{i_{n-1},j's't}^* + \delta & \text{if } i = i_n,\ j = j',\ s = s'' \\ y_{ijst}^* & \text{otherwise.} \end{cases} \tag{76}$$

It is easy to check that $\sum_{s=0}^S \hat{y}_{ijst} = \sum_{s=0}^S y_{ijst}^*$ for all $i \in \mathcal{P},\ j \in \mathcal{C}$, which implies that $(\hat{x}_t, \hat{y}_t, \hat{z}_t)$ is feasible to problem (70)-(75) and yields the same objective value as (x_t^*, y_t^*, z_t^*). Therefore, $(\hat{x}_t, \hat{y}_t, \hat{z}_t)$ is an optimal solution. Furthermore, (76) implies that

$$\sum_{i\in\mathcal{P}} \hat{y}_{ijst} = \begin{cases} z_{j's't}^* & \text{if } j = j',\ s = s' \\ \sum_{i\in\mathcal{P}} y_{ij's''t}^* + \delta & \text{if } j = j',\ s = s'' \\ \sum_{i\in\mathcal{P}} y_{ijst}^* & \text{otherwise.} \end{cases}$$

Since we have $\hat{z}_{jst} = z_{jst}^*$ for all $j \in \mathcal{C},\ s = 0, \ldots, S$ and $\sum_{i\in\mathcal{P}} \hat{y}_{ijst} = \sum_{i\in\mathcal{P}} y_{ijst}^*$ whenever $(j,s) \notin \{(j',s'), (j',s'')\}$, the elements of $\hat{\mathcal{I}}^+$ and $\hat{\mathcal{I}}^-$ are respectively the same as the elements of $\mathcal{I}^+$ and $\mathcal{I}^-$, except possibly for (j',s') and (j',s''). Since we have $\sum_{i\in\mathcal{P}} \hat{y}_{ij's't} = z_{j's't}^* = \hat{z}_{j's't}$, we have $(j',s') \notin \hat{\mathcal{I}}^+$ and $(j',s') \notin \hat{\mathcal{I}}^-$. Finally, since we have $\sum_{i\in\mathcal{P}} \hat{y}_{ij's''t} = \sum_{i\in\mathcal{P}} y_{ij's''t}^* + \delta \le \sum_{i\in\mathcal{P}} y_{ij's''t}^* + z_{j's''t}^* - \sum_{i\in\mathcal{P}} y_{ij's''t}^* = \hat{z}_{j's''t}$, we have $(j',s'') \notin \hat{\mathcal{I}}^+$. Therefore, we have $|\hat{\mathcal{I}}^+| = |\mathcal{I}^+| - 1$ and $|\hat{\mathcal{I}}^-| \le |\mathcal{I}^-|$. The proof for the case $\delta > z_{j's''t}^* - \sum_{i\in\mathcal{P}} y_{ij's''t}^*$ follows from a similar argument. □

References

Adelman, D. (2004), 'A price-directed approach to stochastic inventory routing', *Operations Research* **52**(4), 499–514.

Adelman, D. (2005), Dynamic bid-prices in revenue management, Technical report, The University of Chicago, Graduate School of Business.

Adelman, D. & Mersereau, A. J. (2004), Relaxations of weakly coupled stochastic dynamic programs, Working paper, Graduate School of Business, The University of Chicago, Chicago, IL.

Bazaraa, M. S., Sherali, H. D. & Shetty, C. M. (1993), *Nonlinear Programming: Theory and Algorithms*, second edn, John Wiley & Sons, Inc., New York.

Bertsekas, D. P. & Tsitsiklis, J. N. (1996), *Neuro-Dynamic Programming*, Athena Scientific, Belmont, MA.

Bertsimas, D. J. & Mersereau, A. J. (2005), Adaptive interactive marketing to a customer segment, Working paper, Graduate School of Business, The University of Chicago, Chicago, IL.

Cheung, R. K. M. & Powell, W. B. (1996), 'Models and algorithms for distribution problems with uncertain demands', *Transportation Science* **30**(1), 43–59.

de Farias, D. P. & Van Roy, B. (2003), 'The linear programming approach to approximate dynamic programming', *Operations Research* **51**(6), 850–865.

Godfrey, G. A. & Powell, W. B. (2002), 'An adaptive, dynamic programming algorithm for stochastic resource allocation problems I: Single period travel times', *Transportation Science* **36**(1), 21–39.

Graves, S. C., Kan, A. H. G. R. & Zipkin, P. H., eds (1981), *Handbooks in Operations Research and Management Science, Volume on Logistics of Production and Inventory*, North Holland, Amsterdam.

Hawkins, J. (2003), A Lagrangian Decomposition Approach to Weakly Coupled Dynamic Optimization Problems and its Applications, PhD thesis, Massachusetts Institute of Technology, Cambridge, MA.

Karmarkar, U. S. (1981), 'The multiperiod multilocation inventory problems', *Operations Research* **29**, 215–228.

Karmarkar, U. S. (1987), 'The multilocation multiperiod inventory problem: Bounds and approximations', *Management Science* **33**(1), 86–94.

Kleywegt, A. J., Nori, V. S. & Savelsbergh, M. W. P. (2002), 'The stochastic inventory routing problem with direct deliveries', *Transportation Science* **36**(1), 94–118.

Magnanti, T. L. & Wong, R. T. (1981), 'Accelerating Benders decomposition: Algorithmic enhancement and model selection criteria', *Operations Research* **29**(3), 464–484.

Nemhauser, G. & Wolsey, L. (1988), *Integer and Combinatorial Optimization*, John Wiley & Sons, Inc., Chichester.

Papadaki, K. & Powell, W. B. (2003), 'An adaptive dynamic programming algorithm for a stochastic multiproduct batch dispatch problem', *Naval Research Logistics* **50**(7), 742–769.

Powell, W. B. & Carvalho, T. A. (1998), 'Dynamic control of logistics queueing network for large-scale fleet management', *Transportation Science* **32**(2), 90–109.

Puterman, M. L. (1994), *Markov Decision Processes*, John Wiley and Sons, Inc., New York.

Ruszczynski, A. (2003), Decomposition methods, *in* A. Ruszczynski & A. Shapiro, eds, '*Handbook in Operations Research and Management Science*, Volume on *Stochastic Programming*', North Holland, Amsterdam.

Schweitzer, P. & Seidmann, A. (1985), 'Generalized polynomial approximations in Markovian decision processes', *Journal of Mathematical Analysis and Applications* **110**, 568–582.

Tayur, S., Ganeshan, R. & Magazine, M., eds (1998), *Quantitative Models for Supply Chaing Management*, Kluwer Academic Publishers, Norwell, Massachusetts.

Topaloglu, H. & Powell, W. B. (2006), 'Dynamic programming approximations for stochastic, time-staged integer multicommodity flow problems', *INFORMS Journal on Computing* **18**(1), 31–42.

Yost, K. A. & Washburn, A. R. (2000), 'The LP/POMDP marriage: Optimization with imperfect information', *Naval Research Logistics Quarterly* **47**, 607–619.

Zipkin, P. (2000), *Foundations of Inventory Management*, McGraw-Hill, Boston, MA.

Step 1. Initialize the sets $\{\mathcal{N}_t^1 : t \in \mathcal{T}\}$ to empty sets. Initialize the iteration counter k to 1.

Step 2. Solve the master problem at iteration k

$$\begin{aligned}
\min \quad & \sum_{r_1 \in \mathcal{R}^{|\mathcal{P}|}} \alpha(r_1)\, \theta_1 + \sum_{r_1 \in \mathcal{R}^{|\mathcal{P}|}} \sum_{i \in \mathcal{P}} \alpha(r_1)\, r_{i1}\, \vartheta_{i1} \\
\text{subject to} \quad & \theta_t + \sum_{i \in \mathcal{P}} r_{it}^n\, \vartheta_{it} \geq p_t(x_t^n, y_t^n) + \theta_{t+1} + \sum_{i \in \mathcal{P}} \big[x_{it}^n + \mathbb{E}\{Q_{i,t+1}\}\big]\, \vartheta_{i,t+1} \\
& \hspace{20em} \forall\, n \in \mathcal{N}_t^k,\ t \in \mathcal{T} \setminus \{\tau\} \\
& \theta_\tau + \sum_{i \in \mathcal{P}} r_{i\tau}^n\, \vartheta_{i\tau} \geq p_\tau(x_\tau^n, y_\tau^n) \hspace{10em} \forall\, n \in \mathcal{N}_\tau^k.
\end{aligned}$$

Let $\{\theta_t^k : t \in \mathcal{T}\}$, $\{\vartheta_{it}^k : i \in \mathcal{P},\ t \in \mathcal{T}\}$ be an optimal solution to this problem.

Step 3. For all $t \in \mathcal{T} \setminus \{\tau\}$, solve problem (19) with $\hat{\vartheta}_{it} = \vartheta_{it}^k$ and $\hat{\vartheta}_{i,t+1} = \vartheta_{i,t+1}^k$ for all $i \in \mathcal{P}$. Letting (r_t^k, x_t^k, y_t^k) be an optimal solution to this problem, if we have

$$p_t(x_t^k, y_t^k) + \sum_{i \in \mathcal{P}} \vartheta_{i,t+1}^k\, x_{it}^k - \sum_{i \in \mathcal{P}} \vartheta_{it}^k\, r_{it}^k > \theta_t^k - \theta_{t+1}^k - \sum_{i \in \mathcal{P}} \mathbb{E}\{Q_{i,t+1}\}\, \vartheta_{i,t+1}^k,$$

then let $\mathcal{N}_t^{k+1} = \mathcal{N}_t^k \bigcup \{k\}$. Otherwise, let $\mathcal{N}_t^{k+1} = \mathcal{N}_t^k$.

Step 4. Solve problem (20) with $\hat{\vartheta}_{i\tau} = \vartheta_{i\tau}^k$ for all $i \in \mathcal{P}$. Letting $(r_\tau^k, x_\tau^k, y_\tau^k)$ be an optimal solution to this problem, if we have

$$p_\tau(x_\tau^k, y_\tau^k) - \sum_{i \in \mathcal{P}} \vartheta_{i\tau}^k\, r_{i\tau}^k > \theta_\tau^k,$$

then let $\mathcal{N}_\tau^{k+1} = \mathcal{N}_\tau^k \bigcup \{k\}$. Otherwise, let $\mathcal{N}_\tau^{k+1} = \mathcal{N}_\tau^k$.

Step 5. If we have $\mathcal{N}_t^{k+1} = \mathcal{N}_t^k$ for all $t \in \mathcal{T}$, then $\{\theta_t^k : t \in \mathcal{T}\}$, $\{\vartheta_{it}^k : i \in \mathcal{P},\ t \in \mathcal{T}\}$ is an optimal solution to problem (16)-(18) and stop. Otherwise, increase k by 1 and go to Step 2.

Figure 1: Constraint generation method to solve problem (16)-(18).

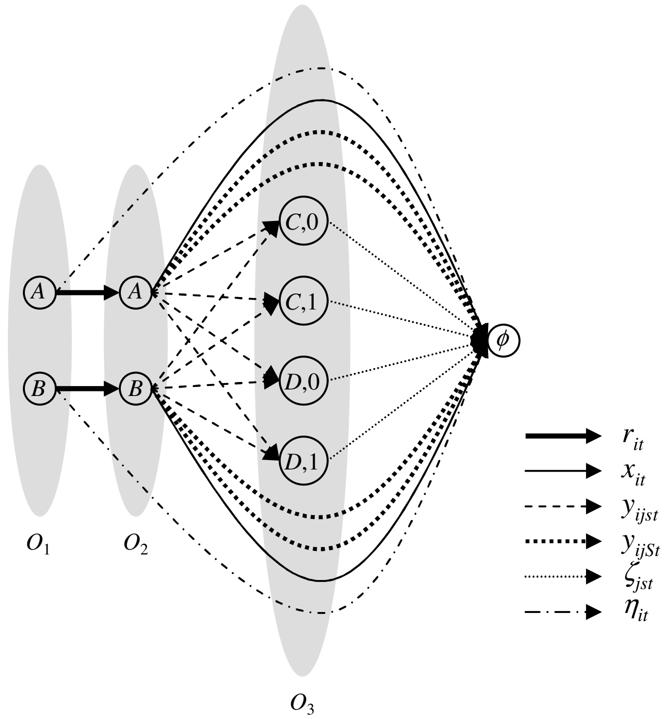

Figure 2: Problem (23)-(28) is a min-cost network flow problem. In this figure, we assume that $\mathcal{P} = \{A, B\}$, $\mathcal{C} = \{C, D\}$ and $S = 2$.

Step 1. Initialize the iteration counter k to 1.

Step 2. Solve the master problem (50)-(52). Let (v^k, λ^k) be an optimal solution to this problem.

Step 3. Compute $\{V_{i1}^{\lambda^k}(r_{i1}) : r_{i1} \in \mathcal{R},\ i \in \mathcal{P}\}$ by solving the optimality equation in (34) with $\lambda = \lambda^k$.

Step 4. If we have $v^k = \sum_{i \in \mathcal{P}} \sum_{r_{i1} \in \mathcal{R}} \beta_i(r_{i1})\, V_{i1}^{\lambda^k}(r_{i1})$, then λ^k is an optimal solution to problem (38) and stop. Otherwise, add constraint (53) to the master problem (50)-(52), increase k by 1 and go to Step 2.

Figure 3: Benders decomposition method to solve problem (38).

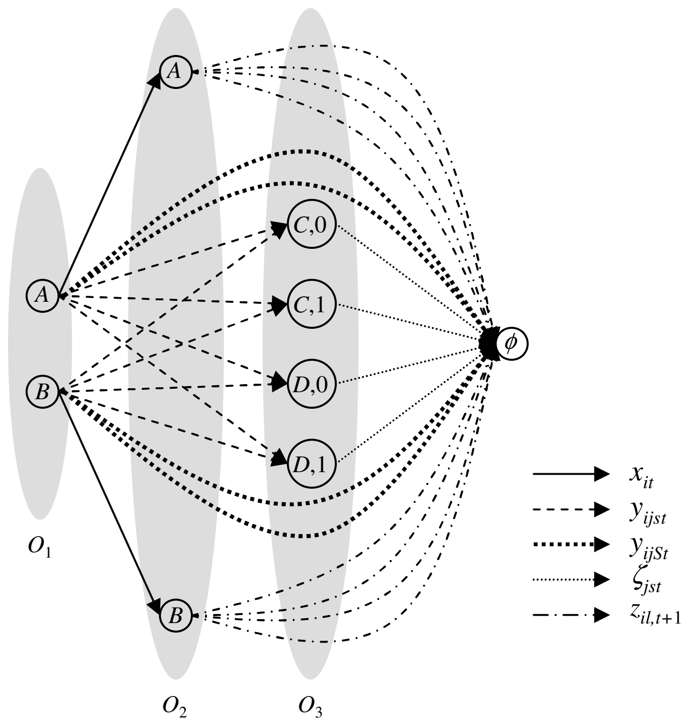

Figure 4: Problem (59)-(63) is a min-cost network flow problem. In this figure, we assume that $\mathcal{P} = \{A, B\}$, $\mathcal{C} = \{C, D\}$, $S = 2$ and $L = 4$.

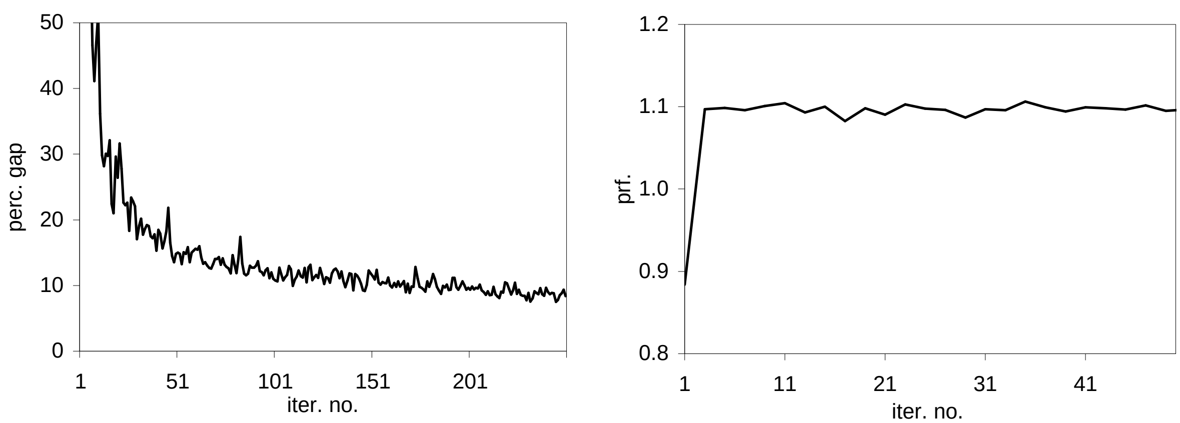

Figure 5: Percent gap between the lower and upper bounds on the optimal objective value of problem (38) (on the left side) and total expected profit that is obtained by the greedy policy characterized by the value function approximations $\{\sum_{t'=t}^{\tau} \sum_{j\in\mathcal{C}} \sum_{s=0}^{S-1} \lambda_{jst'}^{k} + \sum_{i\in\mathcal{P}} V_{it}^{\lambda^k}(r_{it}) : t \in \mathcal{T}\}$ (on the right side).

Problem parameters	LP					LG					
$(\tau, \lvert\mathcal{P}\rvert, P, \bar{\sigma}, \bar{Q}, \bar{V})$	Prf	#Cn	Cpu	%Mp	%Cg	Prf	#Ct	Cpu	%Mp	%Cg	In Prf
$(14, 9, 2, -200, 2000, 0.90)$	0.96	153	21	41	59	1.04	28	10	64	36	1.00
$(14, 9, 4, -200, 2000, 0.90)$	0.97	129	20	39	60	1.02	187	374	74	26	0.96
$(14, 9, 8, -200, 2000, 0.90)$	0.97	137	25	34	66	1.02	259	578	74	26	0.95
$(14, 9, 2, -200, 2000, 0.95)$	0.94	130	18	42	58	1.08	28	9	65	35	1.01
$(14, 9, 4, -200, 2000, 0.95)$	0.95	172	26	38	62	1.04	114	129	70	30	0.92
$(14, 9, 8, -200, 2000, 0.95)$	0.96	153	28	33	67	1.04	278	493	71	29	0.91
$(14, 9, 2, -200, 2000, 0.99)$	0.92	135	19	44	56	1.25	28	8	65	34	1.11
$(14, 9, 4, -200, 2000, 0.99)$	0.92	174	26	37	63	1.10	110	115	69	31	0.88
$(14, 9, 8, -200, 2000, 0.99)$	0.91	143	26	32	68	1.09	275	432	69	31	0.86

Table 1: Computational results for problems with different numbers of plants that serve a customer location.

Problem parameters	LP					LG					
$(\tau, \lvert\mathcal{P}\rvert, P, \bar{\sigma}, \bar{Q}, \bar{V})$	Prf	#Cn	Cpu	%Mp	%Cg	Prf	#Ct	Cpu	%Mp	%Cg	In Prf
$(14, 9, 4, 100, 2000, 0.90)$	0.99	139	20	37	63	1.01	10	4	69	31	0.99
$(14, 9, 4, 0, 2000, 0.90)$	0.98	130	19	39	61	1.01	17	8	69	30	0.99
$(14, 9, 4, -200, 2000, 0.90)$	0.97	129	20	39	60	1.02	187	374	74	26	0.96
$(14, 9, 4, -300, 2000, 0.90)$	0.83	143	22	38	62	1.09	736	4334	87	13	0.77
$(14, 9, 4, 100, 2000, 0.95)$	0.99	201	29	36	64	1.01	11	4	70	29	0.99
$(14, 9, 4, 0, 2000, 0.95)$	0.99	177	26	37	63	1.01	17	7	69	31	0.98
$(14, 9, 4, -200, 2000, 0.95)$	0.95	172	26	38	62	1.04	114	129	70	30	0.92
$(14, 9, 4, -300, 2000, 0.95)$	0.90	142	21	36	64	1.11	605	2225	80	20	0.77
$(14, 9, 4, 100, 2000, 0.99)$	0.99	131	20	39	61	1.01	11	4	70	30	0.99
$(14, 9, 4, 0, 2000, 0.99)$	0.98	139	21	39	61	1.01	18	7	67	33	0.99
$(14, 9, 4, -200, 2000, 0.99)$	0.92	174	26	37	63	1.10	110	115	69	31	0.88
$(14, 9, 4, -300, 2000, 0.99)$	0.95	152	23	36	64	1.06	582	1968	79	21	0.91

Table 2: Computational results for problems with different salvage values.

Problem parameters	LP					LG					
$(\tau, \lvert\mathcal{P}\rvert, P, \bar{\sigma}, \bar{Q}, \bar{V})$	Prf	#Cn	Cpu	%Mp	%Cg	Prf	#Ct	Cpu	%Mp	%Cg	In Prf
$(14, 9, 4, -200, 1000, 0.91)$	0.93	142	17	41	59	1.07	412	428	76	24	1.06
$(14, 9, 4, -200, 2000, 0.90)$	0.97	129	20	39	60	1.02	187	374	74	26	0.96
$(14, 9, 4, -200, 4000, 0.87)$	0.94	162	32	33	67	0.99	52	133	68	32	0.90
$(14, 9, 4, -200, 1000, 0.96)$	0.88	131	16	42	58	1.14	402	334	73	27	1.13
$(14, 9, 4, -200, 2000, 0.95)$	0.95	172	26	38	62	1.04	114	129	70	30	0.92
$(14, 9, 4, -200, 4000, 0.95)$	0.90	141	29	34	66	1.04	56	137	68	32	0.81
$(14, 9, 4, -200, 1000, 0.98)$	0.76	144	17	42	57	1.50	363	232	69	31	1.45
$(14, 9, 4, -200, 2000, 0.99)$	0.92	174	26	37	63	1.10	110	115	69	31	0.88
$(14, 9, 4, -200, 4000, 0.98)$	0.76	142	29	33	66	0.98	47	103	68	32	0.54

Table 3: Computational results for problems with different total expected production quantities.

Problem parameters	LP					LG					
$(\tau, \lvert\mathcal{P}\rvert, P, \bar{\sigma}, \bar{Q}, \bar{V})$	Prf	#Cn	Cpu	%Mp	%Cg	Prf	#Ct	Cpu	%Mp	%Cg	In Prf
$(8, 9, 4, -200, 1000, 0.92)$	0.79	115	7	26	73	1.28	332	112	72	28	1.21
$(14, 9, 4, -200, 2000, 0.90)$	0.97	129	20	39	60	1.02	187	374	74	26	0.96
$(21, 9, 4, -200, 3000, 0.93)$	0.91	176	40	41	58	1.04	830	17911	99	1	0.90
$(14, 4, 4, -200, 2000, 0.91)$	0.85	40	3	6	94	1.12	1637	6090	92	8	1.11
$(14, 9, 4, -200, 2000, 0.90)$	0.97	129	20	39	60	1.02	187	374	74	26	0.96
$(14, 14, 4, -200, 2000, 0.93)$	0.91	216	67	67	33	0.98	523	4175	96	4	0.93

Table 4: Computational results for problems with different dimensions.