Journal of the Operations Research
Society of Japan
Vol. 31, No. 4, December 1988

BRANCH-AND-BOUND APPROACH FOR A STOCHASTIC PRODUCTION PLANNING PROBLEM WITH CAPACITY CONSTRAINTS

Chang Sup Sung Young Jin Lee
Korea Advanced Institute of Science and Technology

(Received August 15, 1987; Revised March 10, 1988)

Abstract A stochastic production planning problem with a finite number of planning periods is analyzed where cumulative demands up to each period are independent random variables with continuous probability distributions. In the problem, backlogging is permitted and production is restricted by its capacity. Dynamic but linear costs of inventory holding and backlogging, and of production with setup charge are considered. A branch-and-bound algorithm is developed to find an optimal plan within finite searching steps, and its computational effectiveness is evaluated.

1. Introduction

This paper considers a production planning problem with known stochastic demands where the planning horizon is composed of a finite number of planning periods and cumulative demands up to each period have a continuous probability distribution. Capacity restrictions are imposed on production, and unsatisfied demands are backlogged. Furthermore, dynamic but linear costs of inventory holding and backlogging, and of production with setup charge are included. The production problem can then be interpreted as in a similar form to a stochastic programming problem with simple recourse (Ziemba [12]). In fact, the stochastic production problem shall be transformed to an equivalent deterministic problem from which an optimal solution to the original problem is obtained.

The equivalent deterministic problem has an objective function which is neither convex nor concave. Rather, it is a mixture of convex and concave functions which makes it difficult to solve the problem by using usual convex programming or concave programming algorithms. Therefore, this paper suggests a branch-and-bound algorithm which can determine an optimal solution to the deterministic problem within finite searching steps.

The branch-and-bound algorithm is exploited based on the concept of convex envelope (Falk and Hoffman [5]). That is, by transforming the concave part of the objective function into an underestimated linear approximate, an underestimated convex objective function, so-called, convex envelope, is derived. The convex envelope and its associated solution space are then partitioned so as to constitute subproblems. Each of the subproblems leads to construct a sub-tree in the branch-and-bound algorithm for which branching rules and bounding schemes are mechanized to make an efficient optimal solution search for the equivalent deterministic problem (and accordingly for the original problem) within finite steps.

Hadley [6] has discussed a similar problem in a dynamic programming formulation but without incorporating any production capacity restriction. Nevison and Burstein [9] have treated a stochastic production problem with stochastic lead times of demand. They permitted inventory backloggings but did not consider any production capacity. Bitran and Yanasse [3] have considered deterministic approximations to stochastic production planning problems where cumulative demand quantities are random variables and no backlogging is allowed.

In section 2, the stochastic production planning problem is formulated in a mathematical model, and its equivalent deterministic problem is derived. In section 3, a branch-and-bound algorithm is developed. In section 4, a convex programming algorithm is introduced for subproblems corresponding to each sub-tree of the branch-and-bound algorithm. In section 5, the computational performance of the algorithm is evaluated on various criteria by use of its test results on a number of numerical examples. In section 6, concluding remarks are presented.

2. Model Formulation

For the stochastic production problem, the followings are assumed;

(i) The cumulative demand up to each period has a known continuous probability distribution.

(ii) Production at each period is restricted within its capacity.

(iii) Backloggings are permitted.

(iv) Setup charge is imposed on each production, and costs of production, of on-hand inventory holding, and of inventory backlogging are linear.

(v) Planning horizon of periods is finite.

With the assumptions (i)-(v), the stochastic production problem (*SPP*) is formulated as follows;

(*SPP*)

$$\text{Min.} \quad F(x) = E[\sum_{t=1}^{T} \{K_t\delta(x_t) + c_t x_t + h_t^+ I_t^+ + h_t^- I_t^-\}]$$

$$\text{s.t.} \quad x_t + I_{t-1} = d_t + I_t ,$$

$$0 \le x_t \le u_t ,$$

$$I_t = I_t^+ - I_t^- ,$$

$$x_t,\ I_t^+,\ I_t^- \ge 0 \quad \text{for all} \quad t=1,\dots,T,$$

where

T = planning horizon,

x_t = amount of production at period t,

d_t = nonnegative random demand for period t,

I_t = amount of inventory at the end of period t,

I_t^+ = on-hand inventory at the end of period t,

I_t^- = backlogged inventory at the end of period t,

u_t = production capacity at period t,

K_t = setup cost at period t,

c_t = unit production cost at period t,

h_t^+ = inventory holding cost per unit at period t,

h_t^- = inventory backlogging cost per unit at period t,

$\delta(y)$ = 1 if $y > 0$, or 0 otherwise,

$E[\cdot]$ = operation of expectation.

In the model, without loss of generality, it is assumed that the starting inventory $I_0=0$.

I_t^+ and I_t^- are restated as follows;

$$(1) \qquad I_t^+ = \max\{0,\ \sum_{i=1}^{t} (x_i - d_i)\} ,$$

$$(2) \qquad I_t^- = \max\{0,\ \sum_{i=1}^{t} (d_i - x_i)\} .$$

Substituting I_t^+ and I_t^- in (SPP) with (1) and (2), respectively, the following problem (SP) is obtained as an equivalent formulation of (SPP):

(SP)

$$\text{Min.} \quad F(x) = E\Big[\sum_{t=1}^{T} [K_t\delta(x_t) + c_t x_t + h_t^+ \max\{0, \sum_{i=1}^{t}(x_i - d_i)\} \tag{3}$$
$$+ h_t^- \max\{0, \sum_{i=1}^{t}(d_i - x_i)\}]\Big]$$

s.t.

$$0 \le x_t \le u_t, \quad \text{for all} \quad t=1,\dots,T. \tag{4}$$

$F(x)$ in (3) can be rewritten

$$F(x) = \sum_{t=1}^{T} [\{K_t\delta(x_t)+c_t x_t\} + E[h_t^+ \max\{0, \sum_{i=1}^{t}(x_i - d_i)\}$$
$$+ h_t^- \max\{0, \sum_{i=1}^{t}(d_i - x_i)\}]]$$

Letting

$$z_t = (x_1,\dots,x_t),$$
$$P_t(x_t) = K_t\delta(x_t)+c_t x_t, \text{ and}$$
$$Q_t(x_1,\dots,x_t) = E[h_t^+ \max\{0, \sum_{i=1}^{t}(x_i - d_i)\} + h_t^- \max\{0, \sum_{i=1}^{t}(d_i - x_i)\}] = Q_t(z_t),$$

if follows that

$$F(x) = P(x) + Q(x)$$

where

$$P(x) = \sum_{t=1}^{T} P_t(x_t) \text{ and } Q(x) = \sum_{t=1}^{T} Q_t(z_t)$$

Let $g_t(\cdot)$ be the probability density function of the cumulative demand up to period t, $\sum_{i=1}^{t} d_i$, $t=1,\dots,T$. Then an explicit expression of $Q(x)$ is easily derived in terms of $g_t(\cdot)$,

$$Q_t(z_t) = (h_t^+ + h_t^-)\int_0^{\sum_{i=1}^{t} x_i} (\sum_{i=1}^{t} x_i - y) g_t(y)dy + h_t^-[E(\sum_{i=1}^{t} d_i) - \sum_{i=1}^{t} x_i], \tag{5}$$

for all $t=1,\dots,T$.

In (5), the term $\int_0^{Y_t}(Y_t - y)g_t(y)dy$ represents the expected number of on-hand inventory at period t, which may characterize whether or not the expected inventory cost (incurred by both holding and backlogging), $Q_t(z_t)$, at period t can be interpreted as a deterministic function. Since the cumulative production up to period t, $Y_t = \sum_{i=1}^{t} x_i$, is finitely bounded above for all $t=1, \ldots, T$, the integral $\int_0^{Y_t} (Y_t - y) g_t(y) dy$ exists for all Y_t, $t=1,\ldots,T$.

Recall that in problem (SP), the objective function $F(x)$ is decomposed into the production cost $P(x)$ and the expected inventory cost $Q(x)$. And the random variables d_t, $t=1,\ldots,T$, are related only to the function $Q(x)$. Thus, by representing $Q(x)$ in the explicit form of (5) which is not haunted by the random variables d_t, $t=1,\ldots,T$, the overall cost function $F(x)$ takes a deterministic form. This leads to the following equivalent deterministic problem (EDP) of (SPP).

(EDP)

$$(6) \qquad \text{Min.} \quad F(x) = P(x) + Q(x)$$

$$\text{s.t.} \quad 0 \le x_t \le u_t, \quad \text{for all} \quad t=1,\ldots,T,$$

where $Q(x)$ is defined as in (5).

If demands in each period are independent Gamma or normally distributed random variables, then the cumulative demands up to each period are also Gamma or normally distributed, respectively. Thus, from these distributions, the deterministic functions $F(x)$ as required in (6) can analytically be derived from the well-known Gamma-Poisson relationship or by integrating by parts, respectively.

3. Branch-and-Bound Algorithm

In order to formulate a solution algorithm, it shall first be proved that the expected inventory cost function $Q(x)$ is convex. Let Z denote the solution set of problem (EDP);

$$Z = \{ x \mid 0 \le x_t \le u_t,\ t=1,\ldots,T \} .$$

Then Z is a compact convex set.

Theorem 3.1. The expected inventory cost function $Q(x)$ in (5) is convex for all $x \in Z$.

Proof: The integral term $\int_0^{Y_t} (Y_t - y) g_t(y) dy$ in (5) can easily be shown as a convex function for all $x \in Z$ by following the procedure of Hadley ([6], p.89) for all $t=1,\ldots,T$. It follows that $Q(x)$ is the summation of convex functions for all $x \in Z$. This completes the proof.

On the other hand, the expected production cost function $P(x)$ is observed as a concave function for all $x \in Z$, since it is the summation of concave functions $P_t(x_t)$ on Z, $t=1,\ldots,T$.

Now let $W(x)$ be a linear underestimating function of $P(x)$ for all $x \in Z$;

$$(7) \qquad W(x) = \sum_{t=1}^{T} W_t(x_t)$$

where

$$W_t(x_t) = (c_t+K_t/u_t)x_t,$$

for all $x=(x_1,\ldots,x_T) \in Z$ and $t=1,\ldots,T$.

Then, according to Falk and Hoffman [5], $W(x)$ is a convex envelope of $P(x)$, which is characterized as in the next theorem.

Theorem 3.2. Let $W(x)$ be defined as in (7). Then $W(x)$ is a convex envelope of $P(x)$ defined on the convex set Z, which satisfies the conditions

(i) W is a convex function defined over the convex set Z where $W(x) \le P(x)$ for all $x \in Z$, and

(ii) if H is any convex function defined over Z such that

$$H(x) \le P(x) \text{ for all } x \in Z,$$

then $H(x) \le W(x)$ for all $x \in Z$.

Proof: The convexity of $W(x)$ directly follows because $W(x)$ is a linear function for all $x \in Z$. It also holds that

$$W(x) - P(x) = \sum_{t=1}^{T} [K_t\{x_t/u_t - \delta(x_t)\}] \le 0 ,$$

since $0 \le x_t \le u_t$ for all $t=1,\ldots,T$. Therefore, the condition (i) is satisfied. Now, for all $t=1,\ldots,T$,

$$W_t(0) = P_t(0) \text{ and } W_t(u_t) = P_t(u_t) .$$

Moreover, $P_t(x_t)$ is concave for all x_t, $0 \le x_t \le u_t$. Thus, it is evident that the linear function $W_t(x_t)$ satisfies the result of (ii). This completes the proof.

Let $f(x) = W(x) + Q(x)$, where $Q(x)$ is defined as in (5). From Theorem 3.2, it can easily be verified that $f(x)$ is a convex envelope of $F(x)$. Therefore, by substituting $f(x)$ for $F(x)$ in (*EDP*), the following convex programming problem (*CP*) is derived:

(*CP*)

$$\text{Min.} \quad f(x)$$
$$\text{s.t.} \quad x \in Z \; .$$

Moreover, solutions of (*EDP*) and (*CP*) lead to the following relationship.

Theorem 3.3. Let x' and x^* be the optimal solutions of (*CP*) and (*EDP*), respectively. Then, it follows that

$$f(x') \le F(x^*) \; .$$

Proof: By definition, for all $x \in Z$, $f(x) \le F(x)$. Therefore

$$f(x') \le f(x^*) \le F(x^*) \; ,$$

since $x^* \in Z$ and $\min\{f(x) | x \in Z\} = f(x')$. This completes the proof.

Now, our branch-and-bound algorithm is described. Soland [11] has proposed a branch-and-bound algorithm for an optimal facility location problem with concave costs. Recently, Erenguc and Tufekci [4] presented an extended branch-and-bound algorithm for a lot sizing problem with deterministic demands, from which our algorithm is derived similarly.

Let the nodes of the branch-and-bound tree be denoted by N^0, $N^1,\ldots$, with N^0 representing the initial node. Each node N^k represents a subproblem with solution space Z^k defined as a subset of Z and its corresponding objective function f^k which is a convex envelope of F on Z^k. Each Z^k is specified as follows:

$$Z^k = \{x | L_t^k \le x_t \le U_t^k, \; t=1,\ldots,T\} \text{ for all } k \; (k=0,1,2,\ldots) \text{ such}$$

that

(i) $L_t^k = \varepsilon$ and $U_t^k = u_t$, if $t \in A^k$,

(ii) $L_t^k = U_t^k = 0$, if $t \in B^k$, and

(iii) $L_t^k = 0$ and $U_t^k = u_t$, if $t \in J^k$,

where

$I = \{1,\ldots,T\}$,

A^k = the set of pre-specified production periods,

B^k = the set of pre-specified non-production periods,

$J^k = I - A^k \cup B^k$ = the set of undecided periods, and

ε = an arbitrary small positive value provided initially so as to indicate each possible production setup.

At the k-th stage during the algorithm, a node, say, N^r is selected to provide a least lower bound value among all current candidate nodes for further branching. Then at the next stage $k=k+1$, the tree branches from the node N^r to two new nodes N^{2k-1} and N^{2k} corresponding to the solution spaces Z^{2k-1} and Z^{2k}, respectively, such that $Z^{2k-1} \cup Z^{2k} = Z^r$. In the way, the algorithm generates a sequence of feasible points in Z, $(x^0, x^1, \ldots, x^m, \ldots)$ $(m=0, 1, 2, \ldots)$, where x^m is an optimal solution of subproblem at node N^m. Since x^m is a feasible point in Z, $F(x^m)$ becomes an upper bound of $F(x)$ for subproblem at node N^m.

Denote by $UB(N^k)$ the upper bound attained at node N^k, so that $UB(N^k) = F(x^k)$. Let $UB_F^k = \min\{F(x^i) \mid i=0,1,\ldots,k\}$ and $UB_x^k = x^h$ for $x^h \in Z$, where $UB_F^k = F(x^h)$. UB_F^k is an upper bound on the optimal solution value of problem (*EDP*), and UB_x^k is the best solution of (*EDP*) found over nodes N^0 through N^k.

Let $LB(N^k)$ denote a lower bound of the optimal value of F defined on Z^k. It is now shown how the lower bound $LB(N^k)$ is calculated. Let the convex envelope of $F(x)$ associated with Z^k be denoted by $f^k(x)$ which satisfies

$$f^k(x) = W^k(x) + Q(x) \tag{8}$$

where

$$W^k(x) = \sum_{t=1}^{T} W_t^k(x_t) \text{ and}$$

$$W_t^k(x_t) = \begin{cases} K^t + c_t x_t, & \text{if } t \in A^k, \\ 0, & \text{if } t \in B^k, \text{ and} \\ (c_t + K_t/u_t)x_t, & \text{if } t \in J^k, \end{cases}$$

for all $t=1,\ldots,T$.

For $k=0$, let A^0 and B^0 be null sets so that $J^0 = I$.

From the result of Theorem 3.3, substituting $f(x)$ and Z with $f^k(x)$ and Z^k, respectively, leads to $f^k(x^k) \leq F(x^*)$, where

$$f^k(x^k) = \min\{f^k(x) \mid x \in Z^k\} \text{ and}$$

$$F(x^*) = \min\{F(x) \mid x \in Z^k\} .$$

Therefore, a lower bound of the minimum of $F(x)$ defined on Z^k is determined such that $LB(N^k) = f^k(x^k)$, where x^k solves the following subproblem (CP^k) at node N^k;

(CP^k)

$$\text{Min. } f^k(x)$$
$$\text{s.t. } x \in Z^k .$$

Now, suppose that, at the end of stage k during the branch-and-bound algorithm, an intermediate node N^r is to be branched further. This will only be the case in which $LB(N^r) < UB_F^{2k}$, where the last numbered node at the stage is N^{2k}. Let x^r be the solution to problem (CP^r) at the corresponding node N^r. Choose any $t \in J^r$, called t^r, which maximizes the difference

$$(9) \qquad P_t(x^r) - W_t(x^r),\ t=1,\dots,T.$$

Then, at the $(k+1)^{st}$ stage, the node N^r is branched to generate two nodes N^{2k+1} and N^{2k+2} according to Z^{2k+1} and Z^{2k+2}, respectively, where Z^{2k+1} and Z^{2k+2} differ from Z^r only at the period t^r which satisfies (9). Theorem 3.4 underlies the determination of such a period t^r.

Theorem 3.4. Assume that $N^{2(k-1)}$ is the last numbered node at the start of the k-th stage during the branch-and-bound algorithm and N^r is one of the intermediate nodes such that $LB(N^r) < UB_F^{2(k-1)}$. Then there exists at least one period t such that

(i) $t \in J^r$,

(ii) $0 \le x_t^r < u_t$, and

(iii) $K_t(1 - x_t^r/u_t) > 0$.

Proof: Since N^r is an intermediate node, J^r has at least one element. Otherwise, it is an already fathomed node which cannot be branched from. Thus, we have

$$LB(N^r) < UB_F^{2(k-1)} \le F(x^r),$$

so that

$$(10) \qquad LB(N^r) = f^r(x^r) < F(x^r).$$

Moreover,

$$\begin{aligned} F(x^r) - f^r(x^r) &= [P(x^r) + Q(x^r) - W^r(x^r) + Q(x^r)] \\ &= P(x^r) - W^r(x^r). \end{aligned}$$

For all $t=1,\dots,T$, from the definition of $W_t^r(x)$ in (8), it follows that

$$(11) \qquad P^t(x^r) - W_t{}^r(x^r) = \begin{cases} 0, & \text{if } t \in A^r \cup B^r, \\ K_t(1 - x_t^r/u_t), & \text{if } t \in J^r. \end{cases}$$

Since $F(x^r) - f^r(x^r) > 0$ from (10), it follows that there exists at least one period t which satisfies the conditions (i)-(iii). This completes the proof.

Based on the results of Theorem 3.4, at the k-th stage Z^r is divided into two subsets Z^{2k-1} and Z^{2k}. These subsets are specified by their index sets A^{2k-1} and B^{2k-1} and A^{2k} and B^{2k}, respectively, such that

(i) $A^{2k-1} = A^r \cup \{t^r\}$ and $B^{2k-1} = B^r$, and

(ii) $A^{2k} = A^r$ and $B^{2k} = B^r \cup \{t^r\}$,

where

$$t^r = \mathrm{argmax}\{K_t(1 - x_t^r/u_t) \mid 0 \le x_t^r < u_t, \text{ for all } t \in J^r\} .$$

As the algorithm proceeds, a list of nodes that need be further branched from is maintained. This list is called the candidate list. Now, the branch-and-bound algorithm is formulated as in Steps (i), (ii), and (iii).

Step (i). Let k=0=p. Set N^0 with Z^0=Z, and A^0=B^0=ϕ. Solve $f^0(x)$ for x^0 to give $LB(N^0)=f^0(x^0)$, $UB(N^0)=F(x^0)$, $UB_F^0=UB(N^0)$, and $UB_x^0=x^0$. Add N^0 to the candidate list. Go to step (iii).

Step (ii). If there is no node in the candidate list, then terminate and obtain an optimal solution of (*EDP*). Otherwise, find r such that $LB(N^r)$ is the minimum of $LB(N^i)$ among all intermediate nodes N^i in the candidate list. If $LB(N^r) \ge UB_F^{2k}$, then terminate and obtain an optimal solution of (*EDP*), UB_x^{2k}. Otherwise, set p=r and go to step (iii).

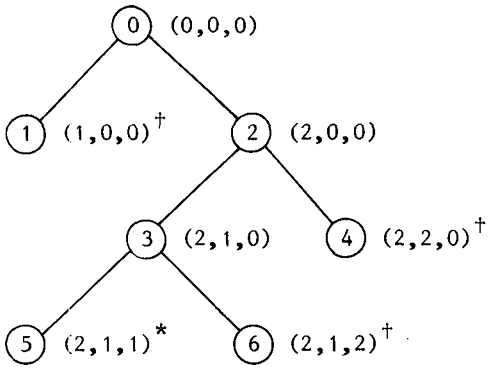
Step (iii). Let k=k+1. Branch from node N^p to generate two new nodes N^{2k-1} and N^{2k} added to the candidate list. Find $LB(N^{2k-1})$ and $LB(N^{2k})$. Update UB_F^{2k} and UB_x^{2k}.

Step (iv). (Fathoming tests)

Step (iv-a). (Completeness test)
Fathom a node N^i in the candidate list such that $A^i \cup B^i = I$.

Step (iv-b). (Bound test)
Fathom a node N^i such that $LB(N^i) \ge UB_F^{2k}$. Go to Step (ii).

The above branch-and-bound algorithm is illustrated by Figure 1 depicting the entire tree of subproblems for T=3. It should be clear that, if the algorithm terminates at stage k, then UB_x^{2k} (the last numbered node N^{2k}) is an optimal solution of (*EDP*). The algorithm terminates within finite steps, since at most (2^T+1) nodes are necessary to be examined for the optimal solution.



* = optimal node, † : fathomed node, (·,·,·) : period state, where 0 = undecided period, 1 = production period, and 2 = non-production period.

Figure 1. The Tree of Subproblems for $T = 3$

4. Convex Programming Algorithm for Each Subproblem

T each node N^k in the branch-and-bound algorithm, the subproblem (CP^k) is corresponded. Recall that

(CP^k)

$$\text{Min.} \quad f^k(x) = W^k(x) + Q(x)$$

$$\text{s.t.} \quad x \in Z^k .$$

By definition, $W^k(x)$ is a continuous differentiable convex function defined on Z^k, and $Q(x)$ in (5) is also a continuous differentiable function for all $x \in Z^k$. Let $\nabla f^k(x) = (\nabla f_1^k(x),\ldots,\nabla f_T^k(x))$ where $\nabla f_i^k(x) = \partial f^k(x)/\partial x_i$, $i=1,\ldots,T$. Then it follows that $\nabla f_i^k(x) = \partial W^k(x)/\partial x_i + \partial Q(x)/\partial x_i$ for all $i=1,\ldots,T$, where

$$\partial W^k(x)/\partial x_i = \begin{cases} c_i, & \text{if } i \in A^k , \\ 0, & \text{if } i \in B^k , \text{ and} \\ c_i+K_i/u_i, & \text{if } i \in J^k , \end{cases}$$

and

$\partial Q(x)/\partial x_i$ is given in Theorem 4.1.

Theorem 4.1. For all $i=1,\ldots,T$,

$$\partial Q(x)/\partial x_i = \sum_{t=i}^{T} [(h_t^+ + h_t^-)G_t(Y_t) - h_t^-] ,$$

where

$G_t(\cdot)$ is the distribution function of $\sum_{i=1}^{t} d_i$, for all $t=1,\ldots,T$.

Proof: Based on the differentiation rule of a definite integral with respect to a parameter Y_t (Heyman and Sobel [8], pp.518-519),

$$\partial Q(x)/\partial x_i = \sum_{t=1}^{T} \partial Q_t(z_t)/\partial x_i .$$

$$\begin{aligned}\partial Q_t(z_t)/\partial x_i &= \partial[(h_t^+ + h_t^-)\{\int_0^{Y_t}(Y_t - y)g_t(y)dy\} + h_t^-\{E(\sum_{j=1}^{t} d_j) - Y_t\}]/\partial x_i \\ &= (h_t^+ + h_t^-)\partial[\int_0^{Y_t}(Y_t - y)g_t(y)dy - h_t^-]/\partial x_i \\ &= \begin{cases} 0, & \text{if } t=1,\ldots,i-1 \text{ and} \\ (h_t^+ + h_t^-)G_t(Y_t) - h_t^-, & \text{if } t=i,\ldots,T. \end{cases}\end{aligned}$$

This completes the proof.

Since (CP^k) is a usual convex programming problem, general solution algorithms can be adapted for the convex programming. In specific, the algorithm of Frank and Wolfe [2] is employed here to solve the subproblem (CP^k), since it guarantees generally the convergence of the solution to the Kuhn-Tucker point. The Frank-Wolfe algorithm is given below where for brevity of expression the stage index k is deleted:

Step 0. Find an initial feasible point x in Z.

Step 1. Calculate $\nabla f(x)$ and let for all $t=1,\ldots,T$,

$$Y_t = \begin{cases} U_t, & \text{if } \nabla f_t(x) < 0 , \\ L_t, & \text{if } \nabla f_t(x) \geq 0 . \end{cases}$$

Step 2. Find the scalar value λ^* where
$\lambda^* = \text{argmin}\{f(\lambda x + (1-\lambda)y) \mid 0 \leq \lambda \leq 1\}$.

Step 3. Let $y^* = \lambda^* x + (1-\lambda^*)y$. If $|f(y^*) - f(x)| \leq \varepsilon$, then terminate.
Otherwise, update $x = y^*$ and go to Step 1.

In Step 3, the scalar value λ^* can be found by using any available one-dimensional search method, for example, the well-known Golden Section method or the Newton method. In this paper, an efficient search method, so-called, Brent's method, is rather employed which intends parabolic interpolation to find the optimal value of scalar λ^* (Press et al. [10]).

For each subproblem (CP^k), an initial feasible solution can easily be provided, since the solution set Z^k is defined over the ranges $L_t^k \le x_t \le U_t^k$ for all $t=1,\ldots,T$, and so any x_t value can be selected as the initial solution between L_t^k and U_t^k. Thus, incorporating the algorithm of Frank and Wolfe [2] into the branch-and-bound algorithm, a sequence of feasible points $(x^0, x^1, \ldots, x^k, \ldots)$ $(k=0, 1, 2,\ldots)$ is obtained where x^k is an optimal solution at each subproblem at node N^k. This leads to an optimal solution of (EDP) by following the whole procedure of the branch-and-bound algorithm over such all possible nodes.

5. Computational Effectiveness of the Branch-and-Bound Algorithm

To examine the effectiveness of the proposed branch-and-bound algorithm a number of numerical examples are generated by varying the problem parameters. And then by solving the problems, the computational performance is evaluated on various criteria. This computational experiment is designed based on the study of Baker et al.'s [1] where they have treated a similar problem with deterministic demands, stationary costs and capacities.

(a) Generation of numerical examples

In order to draw valid conclusions, a number of variations on problem parameters are considered. The problem parameters include demand pattern, planning horizon, cost variations in ratio, and capacity level.

For the demand pattern, two cases, stationary and seasonal demands, are considered. For the stationary demand, the expected value of demand at each period t, $D_t = E[d_t]$, is generated from the uniform interval [100, 300] and its variance is given as $(Var(d_t))^{1/2} = 0.1D_t$. For the seasonal demand, the demand is given:

(12) $$D_t = \mu + a\cdot\sin[2\pi(t+b/4)/b],\ t=1,\ldots,T,$$

where

T = planning horizon,

μ = average demand during the planning horizon T,

a = amplitude of the seasonality, and

b = length of seasonal cycle, in periods.

For each test problem with a planning horizon T, the average demand μ is generated from the uniform interval [100, 300]. Then with the fixed value of μ, the expected value of demand at period t $(t=1,\ldots,T)$ is given by the equation (12) at $a = 0.5\mu$ and $b = T$. For this seasonal demand, the variance of demand at period t is given to increase with time:

$$(Var(d_t))^{1/2} = \sigma(1 + \alpha t), \ t=1,\ldots,T,$$

where $\sigma = 0.1\mu$ and $\alpha = 0.01$.

The aforementioned two cases of the demand pattern imply that demands at each period follow normal probability distributions with the corresponding expected demands and variances.

For the planning horizon T, three cases are considered: $T = 6$, 12, and 24 (in periods).

The average ratio of production to inventory holding cost and inventory backlogging cost are given by, respectively:

$$c_t/h_t^+ = 10, \ c_t/h_t^- = 2/3.$$

And these costs are generated from the following uniform intervals by allowing 50% deviations from the midpoints, 10, 1, and 15, respectively:

$$c_t = [5, 15], \ h_t^+ = [0.5, 1.5], \ h_t^- = [7.5, 22.5]$$

Since the effectiveness of the branch-and-bound algorithm may greatly depend on the cost approximation procedure, a wide range of setup costs and capacities are considered.

For the setup cost, three levels are chosen such that

(1) low level: $K_t = [46.875, 140.625]$

(2) medium level: $K_t = [421.875, 1265.625]$

(3) high level: $K_t = [1687.500, 5062.500]$

The midpoints of these intervals, 93.75, 843.75, and 3375.00, are figured out in correspondence to economic order quantities of 200, 600, and 1200 units, respectively. Note that if the demands are assumed deterministic with the fixed demand level 200 units at each period, then the corresponding planning horizons will include 1, 3, and 6 periods, respectively. The formula for computing the economic order quantity Q^* with stationary demand rate d, setup cost K, inventory holding cost h^+ and backlogging cost h^- is given by (see, Hax and Candea [7]):

$$Q^* = (2Kd(h^+ + h^-)/(h^+ h^-))^{1/2}.$$

For the capacities, with the given value of the expected demand D_t, $t=1,\ldots,T$, three levels are considered:

(1) low level: $u_t = [U_1, U_2]$

(2) medium level: $u_t = [U_2, U_3]$

(3) high level: $u_t = [U_3, U_4]$

where $U_1 = \sum_{t=1}^{T} D_t/T$, $U_2 = \mathrm{Max}\{D_t |\ t=1,\ldots,T\}$, $U_3 = (U_1+U_2)/2$, and $U_4 = \sum_{t=1}^{T} D_t$.

With reference to each of parameter combinations (demand pattern, planning horizon, level of setup cost and capacity), five probabilistic replications shall be generated randomly. It follows that the performance of the algorithm need be tested on a total of 2×3×3×3×5 (= 270) numerical examples.

The solution algorithm is programmed in PASCAL and run on a 32-bit Micro VAX II computer with the parameter $\varepsilon = 0.01$.

(b) Performance of the algorithm

The computational performance of the algorithm sahll be evaluated with respect to the following criteria:

(1) Tree size - the total number of subproblems examined for each numerical example.

(2) Elimination effectiveness of the fathoming test - the percentage of subproblems fathomed by the bound test and the completeness test in Step (iv) of the algorithm in Section 3.

(3) Depth effectiveness of the bound test - the number of undecided periods at the node fathomed by the bound test. If the node is fathomed with M undecided periods, then 2^{M-1} nodes are implicitly eliminated.

(4) Computing time in second - the central processing unit time to solve each problem.

With respect to the demand pattern, the algorithm performs on all criteria better for the stationary demand case than for the seasonal demand one. The computational results as summarized in Table 1 show that on the average the algorithm solves fewer number of subproblems, eliminates greater number of subproblems at higher levels in the tree, and spends fewer computing times, for the numerical problems with the stationary demand pattern than with the seasonal demand pattern.

In general, the tree size and the computing time may increase with the planning horizon T. However, since the cost function is more tightly approximated to at the low levels of setup cost and capacity, it can be expected that at each corresponding low level many subproblems might be eliminated by the bound test. This intuition is reflected by the computational results in Table 1 and Table 2. In Table 1, the number of examined subproblems increases with the planning horizon T and the level of capacity, while the level of setup cost does not seem to affect the tree size as significantly as the planning horizon T and the level of capacity.

Table 1. Number of Examined Subproblems and Computing Times (in Second) Averaged over All Demand Patterns and Replications.

	Capacity Level		T=6			T=12			T=24		
			L	M	H	L	M	H	L	M	H
Setup Cost Level	L	A	8.2	13.0	13.8	19.4	29.4	36.6	116.8	79.2	85.8
		B	1.7	1.6	1.0	3.5	5.4	4.2	39.3	18.3	18.0
	M	A	12.2	14.2	15.6	22.6	29.6	56.4	89.6	207.6	263.2
		B	1.4	2.0	1.7	5.1	5.3	7.0	23.8	54.5	64.6
	H	A	12.0	14.6	14.4	26.2	35.2	35.4	55.6	193.2	196.4
		B	1.8	2.0	2.0	6.1	7.9	6.7	36.8	40.2	50.3

L = low, *M* = medium, *H* = high, *A* = number of examined subproblems, *B* = computing time in second.

Table 2. Percentage of Subproblems Eliminated by the Bound Test and Number of Levels Up from Bottom of Tree at Elimination Averaged over All Demand Patterns and Replications.

	Capacity Level		T=6			T=12			T=24		
			L	M	H	L	M	H	L	M	H
Seutp Cost Level	L	A	55.2	45.0	42.9	49.1	45.0	45.5	50.4	47.8	48.8
		B	3.7	3.3	2.9	7.2	6.1	5.2	13.3	11.4	13.2
	M	A	46.5	38.9	37.8	48.6	45.6	44.3	49.6	49.0	48.2
		B	3.2	2.9	2.9	6.7	5.9	5.5	12.8	11.7	10.4
	H	A	47.0	39.5	39.7	47.7	44.2	44.8	51.3	48.4	48.9
		B	3.1	2.8	2.8	6.3	5.4	5.4	14.4	11.0	11.6

A: percentage of eliminated subproblems, *B*: levels up from bottom of tree.

Table 3. The Fraction of the Worst Tree Size to 2^T+1.

T	Worst size	Fraction
6	33	0.508
12	141	0.034
24	557	(less than 0.001)

Even if the tree size increases with the planning horizon, Table 3 shows that the fraction of the worst tree size to the maximum possible size 2^T+1 for each T decreases rapidly as T increases.

The number of fathomed nodes by the completeness test is very few. For example, the maximum number fathomed among all examined problems is 8. However, more than 40% of examined nodes are eliminated by the bound test as summarized in Table 2. This implies that the algorithm eliminates about half the examined nodes until it finds a solution. Computational results in Table 2 show that the average depth level of elimination is about half the planning horizon.

In summary, this evaluation shows that the algorithm performs weakly as the planning horizon and the levels of setup cost and capacity increase. However, the ratio of tree size to its maximum possible size decreases rapidly as the planning horizon increases. And the fathoming test gives rise to eliminate many subproblems at around half the level of the tree throughout all cases. The computing time takes less than 10 seconds for the planning horizons T = 6, 12.

6. Conclusion

A stochastic production planning problem with a finite planning horizon, where cumulative demands up to each period are random variables, is analyzed based on the solution characteristics of its equivalent deterministic problem. A branch-and-bound algorithm is proposed, which employs a cost approximation procedure of the objective cost function to the convex function at each branching stage. A convex programming algorithm is then incorporated in the branch-and-bound algorithm for lower bound calculations at each subproblem. The performance evaluation experiment concludes that the algorithm may work practically for reasonably sized problems.

The application area of the branch-and-bound algorithm proposed in this paper can include stochastic programming problems with simple recourse, since the associated recourse program part can be expressed in a deterministic program so that an equivalent deterministic program (to the whole program) composed of two deterministic programs can be derived in the similar form of this paper. Further applications may include stochastic transportation-location problems, and capacity expansion problems with random demands.

As a further research, the authors have been considering a decomposition approach of nonlinear programming as another treatment of the concave part (fixed charge) of the cost function.

References

[1] Baker, K.R., Dixon, P., Magazine, M.J., and Silver, E.A.: An Algorithm for the Dynamic Lot-Size Problem with Time-Varying Production Capacity Constraints. *Management Science*, Vol.24 (1978), 1710-1720.

[2] Bazaraa, M.S. and Shetty, C.M.: *Nonlinear Programming*. John Wiley & Sons, New York, 1979.

[3] Bitran, G.R. and Yanasse, H.H.: Deterministic Approximations to Stochastic Production Problems. *Operations Research*, Vol.32 (1984), 999-1018.

[4] Erenguc, S.S. and Tufekci, S.: A Branch and Bound Algorithm for a Single-Item Multi-Source Dynamic Lot Sizing Problem with Capacity Constraints. *IIE Transactions*, Vol.19 (1987), 73-80.

[5] Falk, J.E. and Hoffman, K.R.: A Successive Underestimation Method for Concave Minimization Problems. *Mathematics of Operations Research*, Vol.1 (1976), 251-259.

[6] Hadley, G.: *Nonlinear and Dynamic Programming*. Addison-Wesley Publishing Co., Massachusetts, 1970.

[7] Hax, A.C. and Candea, D.: *Production and Inventory Management*. Pretice-Hall Inc., Englewood-Cliffs, New Jersey, 1984.

[8] Heyman, D.P. and Sobel, M.J.: *Stochastic Models in Operations Research, Volume II: Stochastic Optimization*. McGraw-Hill Book Co., New York, 1984.

[9] Nevison, C. and Burstein, M.: The Dynamic Lot-Size Model with Stochastic Lead Times. *Management Science*, Vol.30 (1984), 100-109.

[10] Press, W.H., Brian, P.F., Teukolsky, P.A., and Vetterling, W.T.: *Numerical Recipes, The Art of Scientific Computing*. Cambridge University Press, 1986.

[11] Soland, R.M: Optimal Facility Location with Concave Costs. *Operations Research*, Vol.22 (1974), 373-382.

[12] Ziemba, W.T.: *Stochastic Programs with Simple Recourse, in Mathematical Programming in Theory and Practice*. Hammer, P.L. and Zoutendijk, G. (eds.) North-Holland Publishing Co., 1974.

Chang Sup SUNG: Department of Industrial Engineering, KAIST, P.O. Box 150, Cheongryang, Seoul 131, Korea