7-2006

Production and inventory management under multiple resource constraints

Kurt M. Bretthauer
Indiana University - Bloomington

Bala Shetty
Texas A & M University - College Station

Siddhartha Syam
Marquette University, siddhartha.syam@marquette.edu

Robert J. Vokurka
Texas A & M University - Corpus Christi

Mathematical and Computer Modelling, Vol. 44, No. 1-2 (July 2006): 85-95. DOI.

Production And Inventory Management Under Multiple Resource Constraints

Kurt M. Bretthauer

Department of Operations and Decision Technologies, Kelley School of Business, Indiana University, Bloomington, IN

Bala Shetty

Department of Information and Operations Management, Lowry Mays College of Business, Texas A&M University, College Station, TX

Siddhartha Syam

Department of Management, College of Business Administration, Marquette University, Milwaukee, WI

Robert J. Vokurka

Department of Economics, Finance, and Decision Sciences, College of Business, Texas A&M University–Corpus Christi, Corpus Christi, TX

Abstract

In this paper we present a model and solution methodology for production and inventory management problems that involve multiple resource constraints. The model formulation is quite general, allowing organizations to handle a variety of multi-item decisions such as determining order quantities, production batch sizes, number of production runs, or cycle times. Resource constraints become necessary to handle interaction among the multiple items. Common types of resource constraints include limits on raw materials, machine capacity, workforce capacity, inventory investment, storage space, or the total number of orders placed. For example, in a production environment, there may be limited workforce capacity and limits on machine capacities for manufacturing various product families. In a purchasing environment where a firm has multiple suppliers, there are often constraints for each supplier, such as the total order from each supplier cannot exceed the volume of the truck. We present efficient algorithms for solving both continuous and integer variable versions of the resource constrained production and inventory management model. The algorithms require the solution of a series of two types of subproblems: one is a nonlinear knapsack problem and the other is a nonlinear problem where the only constraints are lower and upper bounds on the variables. Computational testing of the algorithms is reported and indicates that they are effective for solving large-scale problems.

Keywords

Production and inventory management, Nonlinear optimization, Integer programming

1. Introduction

We present a general model for production and inventory management settings that require multiple resource constraints. For example, we consider production environments where the decision of interest is to determine the number of batches to produce (or batch sizes) for multiple items, and we also discuss multiple supplier environments where the decision of interest is to determine how much to order from each supplier for the case of multiple items. Interaction among the items leads to resource constraints. Here we consider two types of resource constraints: those over all items, and those over disjoint subsets of the items. Examples of resource constraints over all items include limits on workforce capacity, storage space, inventory investment, or the total number of orders placed per year. For the case of constraints on subsets of the items, examples include machine capacities or raw material availabilities for the manufacture of product families, and truck volume limits on the total order from each supplier for purchasing scenarios. Although we focus on applications of the type discussed above, the model is quite general and can handle any production and inventory management decision that exhibits the structure of the general model formulated later in the paper.

The problems considered here are based on two primary assumptions. First, it is assumed that total costs are comprised of a constant term, a linear term, and a reciprocal term. Typically, these terms will represent inventory carrying costs and the cost of placing replenishment orders (or producing batches of product) where this replenishment (production) cost is independent of the size of the order (batch). A second major assumption is that the demand for an item is constant. Other assumptions of the model depend on whether a production or purchasing decision is being considered. Some of these assumptions will include a known and certain lead time; no stockouts; instantaneous replenishment; constant unit cost for items with no discounts; an infinite planning horizon; and demand, lead time, and costs are stationary (i.e., remain fixed over time).

The formulation of the problem leads to a specially structured nonlinear optimization problem. Both continuous and integer variable versions of the problem are addressed. The objective function measures expected total cost per year and, as already mentioned, involves constant, linear and reciprocal terms. The constraints include a single linear constraint that involves all variables, a set of block diagonal linear constraints such that each variable appears in at most one of these constraints, and lower and upper bounds on the variables. Each constraint in the block structured set could represent a limit on production capacity for each machine, or a limit on the shipment volume from each supplier. We present efficient algorithms for solving continuous and integer problems that take advantage of the special structure of the problem. The continuous variable algorithm requires solving a series of box constrained nonlinear subproblems and a series of nonlinear knapsack subproblems. The integer problem is solved with a branch and bound algorithm.

The paper is organized as follows. We begin by reviewing the single resource constraint literature. Then we present the general model with multiple resource constraints. To illustrate the general model, we formulate two specific applications: the number of batches problem where items are produced, and the multiple supplier order quantity problem where items are purchased. Efficient solution methods are developed for continuous and integer problems and extensive computational testing of the algorithms are reported. The last section contains some concluding remarks.

2. Multiple items subject to a single resource constraint

Production and inventory management problems with a single resource constraint have received quite a bit of attention in the literature [1], [2], [3], [4], [5], [6], [7]. The “classic solution technique” to the single constraint, multi-item model is based on the idea that re-order cycle times are independent for each item carried in inventory. Since all of the items carried in an inventory system will eventually peak at the same time, it is the focus of these approaches to ensure that the constraint is not violated at each of these critical junctures. With a single constraint imposed on a convex objective function, the classic solution technique identifies the optimal Lagrange multiplier value for the single constraint [2].

This classic solution technique was improved by Ziegler [7] by establishing bounds on the optimal multiplier and by developing an iterative scheme. Ventura and Klein [6] provided an alternative bounding algorithm. Maloney and Klein [2] developed an algorithm that provides effective bounds on the Lagrange multiplier needed to optimize the n-item inventory system. This algorithm is shown to converge rapidly from its initial bounds to the optimal multiplier [2].

In purchasing environments, because the stocks will normally run out at different times, it is only when the cycles are relatively “in phase” that the constraint has to be considered [8]. This observation led to an improvement approach for order quantity problems [3], [9], [10], [11], [12], [13], [14], [15] that assumes that, regardless of what individual re-order cycle times exist, a joint cycle can be determined. Within this joint cycle, orders of individual items are then time-phased to avoid situations where peak inventories are reached simultaneously (thereby violating the constraint) [2]. Comparisons between the two methods have shown that as the constraint restriction gets tighter, the joint cycle time approach shows greater improvement over the classic Lagrange multiplier method, although neither method is guaranteed to solve the problem with time phasing in an optimal way [4].

Another solution approach is based on the use of individual cycle times that are integer multiples, or power of two multiples, of a base re-order cycle time. This approach provides more flexibility than seen in the joint cycle approach since the base re-order cycle is greater than or equal to the joint cycle [2]. The idea of the base re-order cycle approach has been treated widely in the literature, mainly for the unconstrained problem where the economic advantage of joint replenishment can be realized. Various solution procedures have been documented [13], [16], [17]. However, the computational effort required with this approach is more extensive and implementation more difficult [5].

3. The general model with multiple resource constraints

Many production and inventory management problems require multiple resource constraints, rather than just a single constraint (see the next two sections for examples). The general problem with multiple resource constraints will be formulated as follows.

(1)

$$(P)\mathrm{Min}\sum_{i\in S}(a_i+b_ix_i+c_i/x_i)$$

(2)

$$\mathrm{st}\sum_{i\in S}d_ix_i\le f(\alpha)$$

(3)

$$\sum_{i\in S_k}g_ix_i\le h_k, k=1,\ldots,K$$

(4)

$$l_i\le x_i\le u_i, i\in S$$

(5)

$$(x_i\ \text{integer}, i\in S)$$

Problem (P) is assumed feasible where S is a finite set of indices for the decision variables $x_i \in S$, and $S_0, S_1, S_2, \ldots, S_K$ are disjoint subsets of S that form a partition of S. Let S_0 denote the set of indices in S that are not included in any of the block diagonal constraints in set (3). We assume that cost coefficient $c_i > 0$ for $i \in S$, which implies that the objective function is strictly convex, and we assume that cost coefficients a_i and $b_i \ge 0$ for $i \in S$. In addition, in the resource constraints we assume $d_i \ge 0$ for $i \in S$ and $g_i > 0$ for $i \in S_k, k = 1, \ldots, K$. The parameters f and h_k for $k = 1, \ldots, K$ are positive constants. The lower and upper bounds, l_i and u_i respectively, satisfy $0 \le l_i < u_i$ for $i \in S$. Also, let α denote the Lagrange multiplier for constraint (2) in the continuous variable version of the problem. The above assumptions imply the continuous version of problem (P) is a convex program with a specially structured set of linear constraints.

The interpretation of the decision variables, the cost terms in the objective function, and the resource constraints depend on the particular environment being modeled. If fractional variable values are acceptable, then the integer variable requirements in constraint (5) can be dropped. It is assumed that item demand rates are approximately constant and the cost parameters a_i, b_i, and c_i are time-invariant and independent of x_i for all i. Other additional assumptions depend on the application being considered.

To illustrate the general model, we next discuss two applications: one involves producing items belonging to different product families and the other involves purchasing from multiple suppliers.

4. The number of batches problem with resource constraints

Sundararaghavan and Ahmed [18] consider a setting where n different products are produced in a common facility. One of the problems addressed involves determining the minimum cost integer number of batches to produce for each item with a restriction on production capacity. Sundararaghavan and Ahmed assume processing times are independent of the batch sizes and the products are usable only after a complete batch has been produced. Their model is appropriate for producing many chemical products like resins, paints, inks, pharmaceuticals, and dyes [18]. Here, we consider the problem where the set of products S is partitioned into K product families $S_k, k = 1, \ldots, K$. Each product family is produced on a machine with capacity G_k. Also, there is a total workforce capacity of F.

Then, the number of batches problem can be written as follows.

$$(\text{NB})\text{Min}\sum_{i\in S}(A_i x_i + H_i D_i/(2x_i))\ \text{st}\sum_{i\in S} W_i x_i \leq F \sum_{i\in S_k} T_i x_i \leq G_k, k = 1, \ldots, K l_i \leq x_i \leq u_i, i \in S x_i \text{ integer}, i \in S$$

where

x_i = the number of batches of item i to be produced,

A_i = setup cost per batch of item i,

H_i = annual inventory carrying charge per unit of item i,

D_i = units demanded per year for item i,

W_i = workforce capacity consumed per batch of item i,

F = total workforce capacity available,

T_i = machine time consumed per batch of item i,

G_k = machine time available for product family k.

In the objective function of problem (NB), $A_i x_i$ represents yearly setup costs and $H_i D_i/(2x_i)$ represents average yearly inventory holding costs. Note that cost parameter a_i from problem (P) is zero. The production batch size for item i, q_i, is easily computed given x_i via $q_i = D_i/x_i, i = 1, \ldots, n$.

5. The multiple supplier problem

In purchasing there is a dichotomy between single-source and multiple-source philosophies. The debate has intensified in recent years as quality control and just-in-time practitioners advocate single sourcing, while traditional purchasing wisdom upholds the multiple-sourcing approach [19]. Many companies have a policy that states that purchasing must have more than one supplier approved on each item purchased [20]. Other companies are even more confining by having a policy of placing orders, where possible, with several suppliers rather than a single supplier [21].

Table 1 [22] includes arguments given for placing all orders for a given item with one supplier, as well as the arguments for diversification of suppliers. In recent years, there is a growing trend to reduce the number of suppliers. The main motivation of supplier reduction is that it is felt that it may be more difficult for a company to properly train a large number of suppliers in MRP II and JIT/TQC and achieve the quality levels required. Also, with fewer suppliers for an item, communications are improved, there is more opportunity for joint problem solving, and it is easier to get the suppliers involved earlier in the product development process [20]. Given the advantages of JIT, there remain situations where other factors indicate that multiple suppliers are needed. Required physical distance, reliability of transportation or the supplier, and seasonality are examples of such situations. Some products that fit these criteria are oil, natural gas, and agricultural products. See Table 1 for the advantages of single and multiple sourcing.

Table 1. Advantages and disadvantages of single sourcing (Source: Leenders, Fearon, and England [22])

	Advantages of single sourcing		Advantages of multiple sourcing
•	The supplier may be the exclusive owner of certain essential patents or processes and, therefore, be the only possible source.	•	Knowing that competitors are getting some of the business may tend to keep the supplier more alert to the need of giving good prices and service.
•	A given supplier may be so outstanding in the quality of product or in the service provided as to preclude serious consideration of buying elsewhere.	•	Assurance of supply is increased. Should fire, strikes, breakdowns, or accidents occur to any one supplier, deliveries can still be obtained from others.
•	The order may be so small as to make it just not worthwhile, if only because of added clerical expense, to divide it.	•	Even should floods, railway strikes, or other widespread occurrences develop which may affect all suppliers to some extent, the chances of securing at least a part of the goods are increased.
•	Concentrating purchases may make possible certain discounts or lower freight rates that could not be had otherwise.	•	Some companies diversify their purchases because they do not want to become the sole support of one company, with the responsibility that such a position entails.
•	The supplier is more cooperative, more interested, and more willing to please having all the buyer's business.	•	Assigning orders to several suppliers gives a company a greater degree of flexibility, because it can call on the unused capacity of all the suppliers instead of on only one.
•	A special case arises when the purchase of an item involves a die, tool, mold charge, or costly setup. The expense of duplicating this equipment or setup is likely to be substantial.	•	It has been common practice among the majority of buyers to use more than one source, especially on the important items.
•	When all orders are placed with one supplier, deliveries may be more easily scheduled.		
•	The use of just-in-time production or stockless buying or systems contracting provides many advantages which are not possible to obtain unless business is concentrated with one or at best a very few suppliers.		

We present formulations for both the single and multiple sourcing problems. In the single-sourcing problem, the set of items S is partitioned into k disjoint groups $S_k, k = 1, \ldots, K$. Each item from set S_k is ordered from supplier k, and not ordered from any other suppliers. The single-sourcing problem (SS) can be formulated as follows.

$$(\mathrm{SS})\mathrm{Min} \sum_{i \in S} (P_i D_i + H P_i x_i/2 + A_i D_i/x_i) \text{ st} \sum_{i \in S} P_i x_i \leq F \sum_{i \in S_k} T_i x_i \leq G_k \text{ ,} k = 1, \ldots, K l_i \leq x_i \leq u_i, i$$
$$\in S x_i \text{ integer}, i \in S$$

where

x_i = the order quantity of item i,

P_i = the unit variable purchase cost of item i,

D_i = units demanded per year for item i,

H = annual inventory carrying charge as a percentage of unit purchase cost P_i,

A_i = fixed cost of ordering item i,

F = upper limit on inventory investment,

T_i = shipping space required for each item i,

G_k = shipping space allotted for each order from supplier k.

The objective function minimizes variable purchase costs plus holding costs plus fixed ordering costs. The constraints set an upper limit of F on total inventory investment and an upper limit of G_k on the truck volume for shipments from supplier k.

The multiple-sourcing problem (MS) can be formulated as follows.

$$(\mathrm{MS})\mathrm{Min} \sum_{j=1}^{m} \sum_{i=1}^{n} (P_{ij}D_{ij} + HP_{ij}x_{ij}/2 + A_{ij}D_{ij}/x_{ij}) \text{ st} \sum_{j=1}^{m} \sum_{i=1}^{n} P_{ij}x_{ij} \le F \sum_{i} T_i x_{ij} \le G_j j = 1, \dots, m l_{ij} \le x_{ij} \le u_{ij}, i = 1, \dots, n; j = 1, \dots, m x_{ij} \text{ integer}, i = 1, \dots, n; j = 1, \dots, m$$

where

x_{ij} = order quantity of item i from supplier $j, i = 1, \dots, n; j = 1, \dots, m$

n = number of items,

m = number of suppliers,

P_{ij} = unit variable purchase cost of item i from supplier j,

D_{ij} = units demanded per year for item i from supplier j,

H = annual inventory carrying charge as a percentage of unit purchase cost P_{ij},

A_{ij} = fixed cost of ordering item i from supplier j,

F = upper limit on inventory investment,

T_i = shipping space required for each item i,

G_j = shipping space allotted for each order from supplier j,

The constraint $\sum_{i} T_i x_{ij} \le G_j$ places a space limitation on each order from supplier j. This will optimize the shipping space and time for items ordered from each supplier. Other possible constraints could be developed depending on the particular space, order cost, or other limitation desired for each order from any supplier. Both problems (SS) and (MS) have the structure of the general problem (P).

6. Solution methodology for the continuous problem

To solve the continuous variable version of problem (P), we present an efficient algorithm that requires the solution of a series of box constrained nonlinear subproblems and a series of nonlinear knapsack subproblems. The box constrained subproblems are trivial to solve in closed form. We present an efficient method for solving the nonlinear knapsack subproblems.

Consider the following Lagrangian dual of the continuous version of problem (P) with respect to constraint (2).

$$(D)\mathrm{Max}\theta(\alpha)\mathrm{st}\alpha \geq 0.$$

In problem (D), θ(α) is defined as follows.

$$\theta(\alpha) = \mathrm{Min}\sum_{i\in S}(a_i + b_i x_i + c_i/x_i) + \alpha\left(\sum_{i\in S} d_i x_i - f\right)\mathrm{st}\sum_{i\in S_k} g_i x_i \leq h_k, k = 1, \dots, K l_i \leq x_i \leq u_i, i \in S.$$

For a given α, note that $\theta(\alpha)$ decomposes into one box constrained convex subproblem and K convex knapsack subproblems. The box constrained subproblem is of the following form.

$$(\mathrm{PB})\mathrm{Min}\sum_{i\in S_0}(a_i + (b_i + \alpha d_i)x_i + c_i/x_i) - \alpha f \mathrm{st} l_i \leq x_i \leq u_i, i \in S_0.$$

Problem (PB) is trivial to solve with optimal solution x_i^{PB} for $i \in S_0$ given below.

$$x_i^{\mathrm{PB}} = \begin{cases} l_i & \text{if } (c_i/(b_i + \alpha d_i))^{0.5} \leq l_i \\ (c_i/(b_i + \alpha d_i))^{0.5} & \text{if } l_i < (c_i/(b_i + \alpha d_i))^{0.5} < u_i \\ u_i & \text{if } (c_i/(b_i + \alpha d_i))^{0.5} \geq u_i. \end{cases}$$

The K convex knapsack subproblems are of the following form.

$$(P_k)\mathrm{Min}\sum_{i\in S_k}(a_i + (b_i + \alpha d_i)x_i + c_i/x_i)\ \mathrm{st}\sum_{i\in S_k} g_i x_i \leq h_k l_i \leq x_i \leq u_i, i \in S_k.$$

In the next section we present a method for solving the knapsack problems $(\mathrm{P}_1), (\mathrm{P}_2), \dots,\ (\mathrm{P}_K)$.

Because $ci > 0$ for all $i \in S, \theta(\alpha)$ has a unique optimal solution for each α. Therefore, the objective function of the dual problem (D) is concave and differentiable [23, Section 6.3]. The optimal value of α can be obtained by performing a simple line search. Each trial value of α requires solving the subproblem (PB) and the K subproblems $(P_1), \dots, (P_K)$. Let α^* denote the optimal value of α. The optimal solution to (PB) and $(P_1), \dots, (P_K)$ with $\alpha = \alpha^*$ provides the optimal solution to the continuous variable version of problem (P).

Performing a line search to obtain α^* requires as input lower and upper bounds on α^*. Therefore, we next present a result in Proposition 1 below that provides an upper bound on α^*. Zero can be used as a lower bound on α^*. Consider the following relaxation of problem (P) where constraint sets (3), (5) have not been included.

$$(\mathrm{PR})\mathrm{Min}\sum_{i\in S}(a_i + b_i x_i + c_i/x_i)\ \mathrm{st}\sum_{i\in S} d_i x_i \leq f(\gamma) l_i \leq x_i \leq u_i, i \in S.$$

Let $\gamma^* \geq 0$ denote the optimal value of the Lagrange multiplier γ for the knapsack constraint in problem (PR).

Proposition 1

$$\alpha^* \leq \gamma^*$$

Proof

Consider the following expression for xi written as a function of the multiplier γ for problem (PR).

(6)

$$x_i^{\mathrm{PR}}(\gamma)=\begin{cases} l_i & \text{if } (c_i/(b_i+\gamma d_i))^{0.5}\le l_i \\ (c_i/(b_i+\gamma d_i))^{0.5} & \text{if } l_i<(c_i/(b_i+\gamma d_i))^{0.5}<u_i \\ u_i & \text{if } (c_i/(b_i+\gamma d_i))^{0.5}\ge u_i. \end{cases}$$

Based on the Karush–Kuhn–Tucker (KKT) conditions, Eq. (6) is one of a set of necessary and sufficient conditions for an optimal solution to the convex problem (PR) [24]. Therefore, the above expression for $x_i^{\mathrm{PR}}(\gamma), i\in S$, must be satisfied in an optimal solution to problem (PR).

We also need the following Lagrangean relaxation of the continuous version of problem (P) where we have dualized constraint set (3) with nonnegative multipliers $\beta_1,\dots,\beta_K$.

$$(\mathrm{PL})\mathrm{Min}\sum_{i\in S}(a_i+b_ix_i+c_ix_i+\beta_ig_ix_i)-\sum_{k=1}^{K}\ \beta_kh_k\mathrm{st}\sum_{i\in S}\ d_ix_i\le f(\alpha)l_i\le x_i\le u_i, i\in S$$

In problem (PL), $\beta_i=\beta_k$ for all $i\in S_k, k=1,\dots,K, \beta_i=0$ for all $i\in S_0$, and, as in the continuous version of problem (P), α denotes the multiplier for the knapsack constraint in problem (PL). Again based on the KKT conditions, the following expression $x_i^{\mathrm{PL}}(\alpha)$ written as a function of $\alpha\ge 0$ is one of a set of necessary and sufficient conditions for an optimal solution to the convex problem (PL).

(7)

$$x_i^{\mathrm{PL}}(\alpha)=\begin{cases} l_i & \text{if } (c_i/(b_i+\alpha d_i+\beta_ig_i))^{0.5}\le l_i \\ (c_i/(b_i+\alpha d_i+\beta_ig_i))^{0.5} & \text{if } l_i<(c_i/(b_i+\alpha d_i+\beta_ig_i))^{0.5}<u_i \\ u_i & \text{if } (c_i/(b_i+\alpha d_i+\beta_ig_i))^{0.5}\ge u_i. \end{cases}$$

We now prove the proposition by considering the two cases $\alpha^*=0$ or $\alpha^*>0$. Case (i): $\alpha^*=0$. The inequality $\alpha^*\le\gamma^*$ is trivially true in this case. Case (ii): $\alpha^*>0$. Assuming the β_i's are at their optimal values, complementary slackness and Eqs. (6), (7) imply $f=\sum_{i\in S}\ d_ix_i^{\mathrm{PL}}(\alpha^*)\le\sum_{i\in S}\ d_ix_i^{\mathrm{PR}}(\alpha^*)$.
Because $\sum_{i\in S}d_ix_i^{\mathrm{PR}}(\gamma)$ is a nonincreasing function of γ, the inequality $\sum_{i\in S}\ d_ix_i^{\mathrm{PR}}(\alpha^*)\ge f$ implies we must have $\alpha^*\le\gamma^*$. □

Therefore, obtaining the upper bound γ∗ on α^* requires solving one continuous nonlinear knapsack problem of the form of (PR).

7. A method for solving the knapsack subproblems

In this section we present an algorithm for solving the nonlinear knapsack subproblems $(P_k), k=1,\dots,K$ and problem (PR). The method, hereafter referred to as the *multiplier search algorithm*, solves the nonlinear knapsack problem via a one-dimensional search for the optimal Lagrange multiplier of the knapsack constraint. It requires finding the root of one nonlinear equation and is described briefly here. For more details, see Bretthauer et al. [1].

In problem (P_k), let λ denote the nonnegative Lagrange multiplier for the knapsack constraint $\sum_{i\in S_k}\ g_ix_i\le h_k$, let w_i denote the Lagrange multiplier for $x_i\le u_i$, and let v_i denote the Lagrange multiplier for $x_i\ge l_i$. Consider the following expressions for x_i, w_i, and v_i written as a function of λ.

$x_i(\lambda)=max\{min((c_i/((b_i+\alpha d_i)+\lambda g_i))^{0.5},u_i),l_i\}$for all $i\in S_k$

$w_i(\lambda)=max\{-(b_i+\alpha d_i)+c_i/\underset{i}{\overset{2}{u}}-\lambda g_i,0\}$for all $i\in S_k$

$v_i(\lambda) = max\{(b_i + \alpha d_i) - c_i / l_i^2 + \lambda g_i, 0\}$for all $i \in S_k$.

Bretthauer et al. [1] show that, for any nonnegative λ, the above expressions for $x_i(\lambda), w_i(\lambda)$, and $v_i(\lambda)$ satisfy all of the KKT conditions of the nonlinear knapsack problem (P_k) except the knapsack constraint $\sum_{i \in S_k} g_i x_i \leq h_k$ and the following complementary slackness condition.

(8)

$$\lambda\left(\sum_{i \in S_k} g_i x_i - h_k\right) = 0.$$

Thus, determining the optimal solution to problem (P_k) requires finding a nonnegative λ value, call it λ^*, that yields a solution also satisfying the knapsack constraint and the above complementary slackness condition. Once we know λ^* we can easily calculate the optimal decision variable values by substituting λ^* into the equations for $x_i(\lambda), w_i(\lambda)$, and $v_i(\lambda)$ for $i \in S_k$.

Let $g(\lambda) = \sum_{i \in S_k} g_i x_i(\lambda)$. Because each $x_i(\lambda)$ is a nonincreasing function of λ, $g(\lambda)$ is also a nonincreasing function of λ. We now present an algorithm for determining the optimal Lagrange multiplier value λ^*.

Algorithm for λ^*

(1) Set $\lambda = 0$. If $g(\lambda) \leq h_k$, then terminate with $\lambda^* = 0$.

(2) Otherwise, from $\lambda^* \geq 0$ and Eq. (8), we know $\lambda^* > 0$ and $g(\lambda) = h_k$. Solve the single nonlinear equation $g(\lambda) = h_k$ for the one unknown variable λ. Set this value of λ equal to λ^*.

Assuming problem (P_k) is feasible, the algorithm clearly returns a nonnegative value for λ^* that satisfies the two remaining KKT conditions $\sum_{i \in S_k} g_i x_i \leq h_k$ and $\lambda\left(\sum_{i \in S_k} g_i x_i - h_k\right) = 0$.

8. Solution methodology for the integer problem

We solve the integer problem (P) with a standard branch and bound algorithm [25]. Each subproblem in the branch and bound tree is of the form of the continuous relaxation of (P). The continuous subproblems differ only in the lower and upper bounds on the variables, and are solved with the algorithm presented in the previous two sections.

Table 2. Problem (SS)—Problem Set A, continuous variables

Number of variables	**Number of constraints**	**Variables per constraint**	**Average solution time (CPU seconds)**
100	5	20	<0.1
200	5	40	<0.1
400	5	80	<0.1
1,000	5	200	0.1
200	20	10	<0.1
1,000	20	50	<0.1
10,000	5	2000	0.8
25,000	5	5000	1.8

Table 3. Problem (SS)—Problem Set B, continuous variables

Number of	**Number of**	**Variables per**	**Average solution**

variables	constraints	constraint	time (CPU seconds)
100	5	20	<0.1
200	5	40	<0.1
400	5	80	<0.1
1,000	5	200	0.1
200	20	10	<0.1
1,000	20	50	<0.1
10,000	5	2000	0.9
25,000	5	5000	1.8

9. Computational results

Here we describe the computational testing done to evaluate the performance of the algorithm. Both continuous and integer variable problems for the single-sourcing problem (SS) described in Section 5 were generated. Twenty problems of each size were solved, except for the last two rows of Table 4 where ten problems of each size were solved. Also, the integer problems were solved to within 0.05% of optimality, except where noted in the tables. That is, the branch and bound algorithm terminates when $(\mathrm{UB}^t - \mathrm{LB}^t)/\mathrm{UB}^t \leq 0.0005$, where LB^t and UB^t denote lower and upper bounds on the optimal objective value of the problem at iteration t of the algorithm. The algorithms were implemented in Fortran 90 (733 MHz Pentium II desktop computer, 192 MB), and solution times do not include input or output operations.

Table 4. Problem (SS)—Problem Set A, integer variables

Number of variables	Number of constraints	Variables per constraint	Average no. of nodes in tree	Average solution time (CPU seconds)
75	5	15	266	1.1
100	5	20	30	0.2
125	5	25	201	1.4
150	5	30	41	0.4
200	5	40	21	0.2
400	5	80	33	0.9
1000	5	200	10	0.5
200[a]	20	10	3293	42.1
1000[a]	40	25	899	45.6

[a]Percent of optimality for termination = 0.07% (rather than 0.05% as in other problems).

The parameters for Problem Set A were uniformly distributed values from the following intervals: $P_i \in [10,20], D_i \in [50,100], A_i \in [20,40], T_i \in [10,20], l_i$, and $u_i \in [5,25]$, and $H = 0.10$. Problem Set B was generated from the following intervals: $P_i \in [10,15], D_i \in [25,100], A_i \in [125,400], T_i \in [10,15], l_i$ and $u_i \in [5,25]$, and $H = 0.10$. The constraint right hand sides were generated last to guarantee problem feasibility. Table 2, Table 3 present the computational results for the continuous variable versions of Problem Sets A and B, respectively. Table 4, Table 5 report the results for the integer variable Problem Sets A and B. The number of constraints refers to the number of block diagonal constraints K.

Table 5. Problem (SS)—Problem Set B, integer variables

Number of variables	Number of constraints	Variables per constraint	Average no. of nodes in tree	Average solution time (CPU seconds)
100	5	20	2467	14.6
400	5	80	7876	236.0

The computational results for the continuous version of problem (SS) were very similar between Problem Sets A and B. In both cases, we were able to solve large-scale problems with up to 25,000 variables in less than an average of 2 CPU seconds per problem. However, as can be seen in Table 4, Table 5, the integer version of Problem Set B was much more difficult than the integer version of Problem Set A.

Also, the results in Table 4 suggest that, at least for the problems solved, increasing the number of constraints while holding the number of integer variables constant seems to increase problem difficulty quite a bit (see the 200 variable problems with 5 and 20 constraints, and the 1000 variable problems with 5 and 40 constraints). Therefore, we performed further testing to determine how the number of integer variables and the number of constraints impact problem difficulty. In Table 6, we fix the number of integer variables at 200, and vary the number of block diagonal constraints from 5 to 40. In Table 7, we fix the number of block diagonal constraints at 10 and vary the number of integer variables from 50 to 500. Problem parameters were generated from the same intervals as in Problem Set A. Fifteen problems of each size were solved to within 0.1% of optimality.

Table 6. Problem (SS)—Problem Set A, fixed number of integer variables

Number of variables	**Number of constraints**	**Variables per constraint**	**Average no. of nodes in tree**	**Average solution time (CPU seconds)**
200	5	40	15	0.2
200	10	20	157	2.6
200	20	10	3,079	44.6
200	25	8	44,432	662.5
200	40	5	253,990	2766.4

Table 7. Problem (SS)—Problem Set A, fixed number of constraints

Number of variables	**Number of constraints**	**Variables per constraint**	**Average no. of nodes in tree**	**Average solution time (CPU seconds)**
50	10	5	1327	4.5
100	10	10	293	2.0
200	10	20	157	2.4
300	10	30	84	1.9
400	10	40	25	0.7
500	10	50	336	13.6

The results in Table 6 indicate that, for a fixed number of integer variables, increasing the number of block diagonal constraints causes a large increase in the number of nodes in the branch and bound tree and solution time. In Table 7, for a fixed number of constraints, increasing the number of integer variables seems to actually decrease the number of nodes in the tree (except for the 500 variable problems). But by looking at all the results in Table 6, Table 7 together, they suggest that the number of variables per block diagonal constraint can have a large impact on the number of nodes and solution time. In both Table 6, Table 7, the number of nodes and solution time increased as the number of variables per constraint decreased, except for a couple of values in Table 7.

10. Concluding remarks

We have presented a model and solution methodology for resource constrained production and inventory management problems. The model is quite general, but we focused on multi-item settings where either the number of production runs or order quantities were the decision of interest. The model can also handle production environments where the decision variables represent production batch sizes or cycle times. Resource constraints on such things as inventory investment, machine capacity, and truck volume were shown to have a special structure, and efficient methods for solving the resulting nonlinear optimization problems were presented. Extensive computational testing indicated that the algorithm is able to solve large-scale problems in reasonable computing time.

References

[1] K. Bretthauer, B. Shetty, S. Syam, S. White, **A model for resource constrained production and inventory management** Decision Sciences, 25 (1994), pp. 561-580

[2] B. Maloney, C. Klein, **Constrained multi-item inventory systems: an implicit approach** Computers and Operations Research, 20 (1993), pp. 639-649

[3] E. Page, R. Paul, **Multi-product inventory situations with one restriction** Operational Research Quarterly, 27 (1976), pp. 815-834

[4] M. Rosenblatt, **Multi-item system with budgetary constraint: a comparison between the lagrangian and the fixed cycle approach** International Journal of Production Research, 19 (1981), pp. 331-339

[5] M. Rosenblatt, U. Rothblum. **On the single resource capacity problem for multi-item inventory systems** Operations Research, 38 (1990), pp. 686-693

[6] J. Ventura, C. Klein. **A note on multi-item inventory systems with limited capacity** Operations Research Letters, 7 (1988), pp. 71-75

[7] H. Ziegler. **Solving certain singly constrained convex optimization problems in production planning** Operations Research Letters, 1 (1982), pp. 246-252

[8] D. Phillips, A. Ravindran, J. Solberg. **Operations Research: Principles and Practice** Wiley, New York (1976)

[9] L. Krone. **A note on economic lot sizes for multi-purpose equipment** Management Science, 10 (1964), pp. 461-464

[10] W. Maxwell. **The scheduling of economic lot sizes** Naval Research Logistics Quarterly, 11 (1964), pp. 89-124

[11] J. Parsons. **A note on Krone's economic lot-size formulas** Management Science, 12 (1965), pp. 314-315

[12] E. Homer. **Space-limited aggregate inventory with phased deliveries** Journal of Industrial Engineering, 17 (1966), pp. 327-337

[13] E. Silver. **A simple method of determining order quantities in joint replenishments under deterministic demand** Management Science, 22 (1976), pp. 1351-1361

[14] K. Zoller. **Deterministic multi-item inventory systems with limited capacity** Management Science, 24 (1977), pp. 451-455

[15] R. Hartley, L. Thomas. **The deterministic, two-product, inventory system with capacity constraint** Journal of the Operational Research Society, 33 (1982), pp. 1013-1020

[16] M. Kaspi, M. Rosenblatt. **An improvement of Silver's algorithm for the joint replenishment problem** IIE Transactions, 15 (1983), pp. 264-267

[17] R. Roundy. **98%-effective integer-ratio lot-sizing for one-warehouse multi-retailer systems** Management Science, 31 (1985), pp. 1416-1430

[18] P.S. Sundararaghavan, M.U. Ahmed. **Inventory control and scheduling of multiple products in a common facility** Decision Sciences, 20 (1989), pp. 476-491

[19] E. Scheuing. **Purchasing Management** Prentice Hall, Englewood Cliffs, NJ (1989)

[20] J. Schorr. **Purchasing in the 21st Century: A Guide to State-of-the-Art Techniques and Strategies** Oliver Wight Publications, Inc., Essex Junction, VT (1992)

[21] J. Westing, I. Fine, G. Zenz. **Purchasing Management: Materials in Motion** Wiley, New York (1976)

[22] M. Leenders, H. Fearon, W. England. **Purchasing and Materials Management** Richard D. Irwin, Inc., Homewood, IL (1985)

[23] M.S. Bazaraa, H.D. Sherali, C.M. Shetty. **Nonlinear Programming: Theory and Algorithms** (2nd edition), Wiley, New York (1993)

[24] K.M. Bretthauer, B. Shetty. **The nonlinear resource allocation problem** Operations Research, 43 (1995), pp. 670-683

[25] G.L. Nemhauser, L.A. Wolsey. **Integer and Combinatorial Optimization** Wiley, New York (1988)