Bilinear Programming: Applications in the Supply Chain Management

Artyom G. Nahapetyan
Center for Applied Optimization
Industrial and Systems Engineering Department
University of Florida
Gainesville, Florida 32611-6595
Email address: artyom@ufl.edu

1 Introduction

Many problems in the supply chain management can be formulated as a network flow problem with specified arc cost functions. Let $G(N, A)$ represent a network where N and A are the sets of nodes and arcs, respectively, and $f_a(x_a)$ denotes an arc cost function. In the network, there are supply and demand nodes, and the main objective of the problem is to minimize the total cost by satisfying the demand from the available supply. In addition, one can assume that the arc flows are bounded, which corresponds to the cases where a shipment along an arc should not exceed a specified capacity. The mathematical formulation of the problem can be stated as

$$\min_x f(x) = \sum_{a\in A} f_a(x_a) \tag{1}$$

$$\text{s.t.} \qquad Bx = b \tag{2}$$

$$x_a \in [0, \lambda_a] \qquad \forall a \in A \tag{3}$$

where B is the node-arc incident matrix of network G, and b is a suplly/demand vector. In the next section, we discuss two formulations where $f_a(x_a)$ is either a concave piecewise linear or fixed charge function of the arc flow. The concave piecewise linear functions are typically used in the cases where merchandisers encourage to buy more products by offering discounts in the unit price for large orders. The fixed charge functions are used in the cases where regardless the quantity of the shipment it is required to pay a fixed cost to ship along an arc. The fixed cost might be the cost of renting a truck, ship, airplane, or

train to transport goods between nodes of the network. In both cases, we show that the problems are equivalent to a bilinear problem with a disjoint feasible region.

In addition to choosing a proper production level, sometimes managers have to make pricing decisions as well. In particular, one can assume that the satisfied demand is a function of the price, i.e., lower prices generate an additional demand. Such functional relationship between the prices and the satisfied demand is commonly used by economists. However, because of the production capacity restrictions, fixed costs related to the production process, seasonality and other factors, often it is not feasible to satisfy the optimal level of demand, and managers should consider optimal production and inventory levels in combination with pricing decisions to maximize the net profit during a specified time period. One of such problems and an equivalent bilinear formulation are discussed in the next section as well.

In addition to the bilinear formulations of the supply chain problems, in Section 3 we explore the structure of the bilinear problems and discuss difficulties in applying the standard computational methods. Despite the intricacy, the section proposes some heuristic methods to find a near optimum solution to the problems. The solution obtained by a heuristic procedure can also be used to expedite exact algorithms.

2 Formulation

2.1 Concave Piecewise Linear Network Flow Problem

In the problem (1)-(3), assume that $f_a(x_a)$ is a piecewise linear concave function, i.e.,

$$f_a(x_a) = \begin{cases} c_a^1 x_a + s_a^1 (= f_a^1(x_a)) & x_a \in [0, \xi_a^1) \\ c_a^2 x_a + s_a^2 (= f_a^2(x_a)) & x_a \in [\xi_a^1, \xi_a^2) \\ \cdots & \cdots \\ c_a^{n_a} x_a + s_a^{n_a} (= f_a^{n_a}(x_a)) & x_a \in [\xi_a^{n_a-1}, \lambda_a] \end{cases},$$

with $c_a^1 > c_a^2 > \cdots > c_a^{n_a}$. Let $K_a = \{1, 2, \ldots, n_a\}$. Because of the concavity of $f_a(x_a)$, it can be written in the following alternative form

$$f_a(x_a) = \min_{k \in K_a} \{f_a^k(x_a)\} = \min_{k \in K_a} \{c_a^k x_a + s_a^k\}. \tag{4}$$

By introducing additional variables $y_a^k \in [0, 1]$, $k \in K_a$, construct the following bilinear problem.

$$\min_{x,y} g(x, y) = \sum_{a \in A} \left[\sum_{k \in K_a} c_a^k y_a^k \right] x_a + \sum_{a \in A} \sum_{k \in K_a} s_a^k y_a^k = \sum_{a \in A} \sum_{k \in K_a} f_a^k(x_a) y_a^k \tag{5}$$

$$\text{s.t.} \qquad Bx = b \tag{6}$$

$$\sum_{k \in K_a} y_a^k = 1 \qquad \forall a \in A \tag{7}$$

$$x_a \in [0, \lambda_a], y_a^k \geq 0 \qquad \forall a \in A \text{ and } k \in K_a \tag{8}$$

In [2], the authors show that at any local minima of the bilinear problem, $(\hat{x}, \hat{y})$, $\hat{y}$ is either binary vector or can be used to construct a binary vector with the same objective function value. Although the vector $\hat{y}$ may have a fractional components, the authors note that in practical problems it is highly unlikely. The proof of the theorem below follows directly from (4). Details on the proof as well as transformation of the problem (1)-(3) into (5)-(8) can be found in [2].

Theorem 1 *If (x^*, y^*) is a global optima of the problem (5)-(8) then x^* is a solution of the problem (1)-(3).*

According to the theorem, the concave piecewise linear network flow problem is equivalent to a bilinear problem in a sense that the solution of the later is a solution of the former. It is important to notice that the problem (5)-(8) does not have binary variables, i.e., all variables are continuous. However, at optimum y^* is a binary vector, which makes sure that in the objective only one linear piece is employed.

2.2 Fixed Charge Network Flow Problem

In the case of the fixed charge network flow problem, we assume that the function $f_a(x_a)$ has the following structure.

$$f_a(x_a) = \begin{cases} c_a x_a + s_a & x_a \in (0, \lambda_a] \\ 0 & x_a = 0 \end{cases},$$

Observe that the function is discontinuous at the origin and linear on the interval $(0, \lambda_a]$.

Let $\varepsilon_a \in (0, \lambda_a]$, and define

$$\phi_a^{\varepsilon_a}(x_a) = \begin{cases} c_a x_a + s_a & x_a \in [\varepsilon_a, \lambda_a] \\ c_a^{\varepsilon_a} x_a & x_a \in [0, \varepsilon_a) \end{cases}$$

where $c_a^{\varepsilon_a} = c_a + s_a/\varepsilon_a$. It is easy to see that $\phi_a^{\varepsilon_a}(x_a) = f_a(x_a)$, $\forall x_a \in \{0\} \bigcup [\varepsilon_a, \lambda_a]$ and $\phi_a^{\varepsilon_a}(x_a) < f_a(x_a)$, $\forall x_a \in (0, \varepsilon_a)$, i.e., $\phi_a^{\varepsilon_a}(x_a)$ approximates the function $f_a(x_a)$ from below. (see Figure 1). Let us construct the following concave two-piece linear network flow problem.

$$\min_x \phi^{\varepsilon}(x) = \sum_{a \in A} \phi_a^{\varepsilon_a}(x_a) \tag{9}$$

$$\text{s.t.} \qquad Bx = b, \tag{10}$$

$$x_a \in [0, \lambda_a], \qquad \forall a \in A, \tag{11}$$

where ε denotes the vector of ε_a. Function $\phi^{\varepsilon}(x)$ as well as the problem (9)-(11) depends on the value of the vector ε. In the paper [3], the authors show that for any value of $\varepsilon_a \in (0, \lambda_a]$, a global solution of the problem (9)-(11) provides a lower bound for the fixed charge network flow problem, i.e., $\phi^{\varepsilon}(x^{\varepsilon}) \leq f(x^*)$, where x^{ε} and x^* denote the solutions of the corresponding problems.

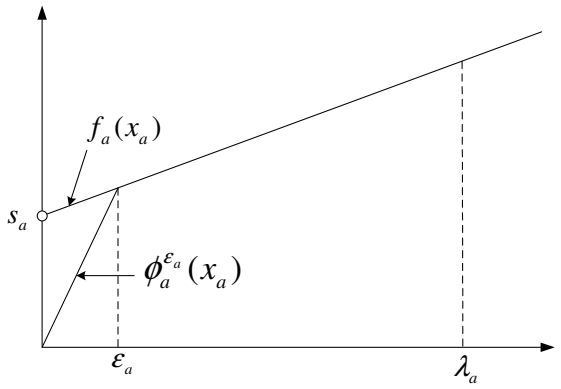


Figure 1: Approximation of function $f_a(x_a)$.

Theorem 2 *(see [3]) For all ε such that $\varepsilon_a \in (0, \lambda_a]$ for all $a \in A$, $\phi^{\varepsilon}(x^{\varepsilon}) \leq f(x^*)$.*

Furthermore, by choosing a sufficiently small value for ε_a one can ensure that both problems have the same solution. Let $\delta = \min\{x_a^v | x^v \in V(x), a \in A, x_a^v > 0\}$, where $V(x)$ denotes the set of vertices of the polyhedra (10)-(11). Observe that δ is the minimum among all positive components of all vectors $x^v \in V(x)$; therefore, $\delta > 0$.

Theorem 3 *(see [3]) For all ε such that $\varepsilon_a \in (0, \delta]$ for all $a \in A$, $\phi^{\varepsilon}(x^{\varepsilon}) = f(x^*)$.*

Theorem 3 proves the equivalence between the fixed charge network flow problem and the concave two-piece linear network flow problem (9)-(11) in a sense that the solution of the later is a solution of the former. As we have seen in the previous section, concave piecewise linear network flow problems are equivalent to bilinear problems. In particular, problem (9)-(11) is equivalent to the following bilinear problem.

$$\min_{x,y} \sum_{a \in A} \left[c_a x_a + s_a\right] y_a + c_a^{\varepsilon_a} x_a \left[1 - y_a\right] \tag{12}$$

$$\text{s.t.} \qquad Bx = b, \tag{13}$$

$$x_a \geq 0, \text{ and } y_a \in [0,1], \qquad \forall a \in A, \tag{14}$$

where $\varepsilon_a \in (0, \delta]$.

2.3 Capacitated Multi-Item Dynamic Pricing Problem

In the problem, we assume that a company during a discrete time period Δ is able to produce different commodities from a set P. In addition, we assume that at each point of time $j \in \Delta$ and for each product $p \in P$ a functional relationship $f_{(p,j)}(d_{(p,j)})$ between the satisfied demand and the price is given, i.e., in order to satisfy the demand $d_{(p,j)}$ of the product p, the price of the product at time j should be equal to $f_{(p,j)}(d_{(p,j)})$. As a

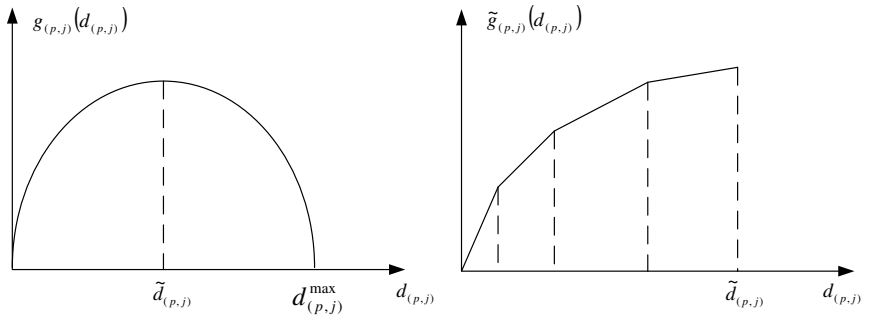
result, the revenue generated from the sales of the product p at time j is $g_{(p,j)}(d_{(p,j)}) = f_{(p,j)}(d_{(p,j)})d_{(p,j)}$. Although we do not specify the function $f_{(p,j)}(d_{(p,j)})$, it should ensure that $g_{(p,j)}(d_{(p,j)})$ is a concave function (see Figure 2a). Because of the concavity of $g_{(p,j)}(d_{(p,j)})$, there exists a point $\tilde{d}_{(p,j)}$, such that the function reaches its maximum, and producing and selling more than $\tilde{d}_{(p,j)}$ is not profitable. Therefore, without lost of generality, we can assume that $d_{(p,j)} \in [0, \tilde{d}_{(p,j)}]$. According to the definition of $g_{(p,j)}(d_{(p,j)})$, it is a concave monotone function on the interval $[0, \tilde{d}_{(p,j)}]$. To avoid nonlinearity in the objective, one can approximate it by a concave piecewise linear function. Doing so, divide $[0, \tilde{d}_{(p,j)}]$ into intervals of equal length, and let $d^k_{(p,j)}$, $k \in \{1, \ldots, N\} \bigcup \{0\} = K \bigcup \{0\}$, denote the end points of the intervals. Then the approximation can be defined as

$$\tilde{g}_{(p,j)}(\lambda_{(p,j)}) = \sum_{k=1}^{N} g^k_{(p,j)} \lambda^k_{(p,j)},$$

where $g^k_{(p,j)} = g_{(p,j)}(d^k_{(p,j)}) = f_{(p,j)}(d^k_{(p,j)})d^k_{(p,j)}$, $\sum_{k=0}^{N} \lambda^k_{(p,j)} = 1$, and $\lambda^k_{(p,j)} \geq 0, \forall p \in P$, $j \in \Delta$ (see Figure 2b).

Let $x^k_{(p,i,j)}$ denote the amount of product p that is produced at time i and sold at time j using the unit price $g^k_{(p,j)}/d^k_{(p,j)} = f^k_{(p,j)} = f_{(p,j)}(d^k_{(p,j)})$. In addition, let $y_{(p,i)}$ denote a binary variable, which equals one if $\sum_k \sum_j x^k_{(p,i,j)} > 0$ and zero otherwise. Costs associated with the production process include inventory costs $c^{in}_{(p,i,j)}$, production costs $c^{pr}_{(p,i)}$, and setup costs $c^{st}_{(p,i)}$. At last, let C_i represent the production capacity at time i, which is "shared" by all products. Using those definitions, one can construct a linear mixed integer formulation of the problem. Below we provide a simplified formulation of the problem, where the variables $\lambda_{(p,j)}$ are eliminated from the formulation. For the details on the mathematical formulation of the problem and its simplification we refer to [4].

$$\max_{x,y} \sum_{p \in P} \sum_{i \in \Delta} \left[\sum_{j \in \Delta | i \leq j} \sum_{k \in K} q^k_{(p,i,j)} x^k_{(p,i,j)} - c^{st}_{(p,i)} y_{(p,i)} \right] \tag{15}$$



a) The Revenue Function b) Approximation of the Revenue Function

Figure 2: The revenue function and its approximation.

$$\sum_{p\in P}\sum_{j\in\Delta|i\le j}\sum_{k\in K} x^k_{(p,i,j)} \le C_i, \qquad \forall i \in \Delta, \tag{16}$$

$$\sum_{j\in\Delta|i\le j}\sum_{k\in K} x^k_{(p,i,j)} \le C_i y_{(p,i)}, \qquad \forall p \in P \text{ and } i \in \Delta, \tag{17}$$

$$\sum_{k\in K}\sum_{i\in\Delta|i\le j} \frac{x^k_{(p,i,j)}}{d^k_{(p,j)}} \le 1, \qquad \forall p \in P \text{ and } j \in \Delta, \tag{18}$$

$$x^k_{(p,i,j)} \ge 0, y_{(p,i)} \in \{0,1\}, \qquad \forall p \in P, i, j \in \Delta \text{ and } k \in K, \tag{19}$$

where $q^k_{(p,i,j)} = f^k_{(p,j)} - c^{in}_{(p,i,j)} - c^{pr}_{(p,i)}$.

Let $X = \{x | x \ge 0$ and $x^k_{(p,i,j)}$ be feasible to (16) and (18)$\}$, and $Y = [0,1]^{|P||\Delta|}$. Consider the following bilinear problem.

$$\max_{x\in X, y\in Y} \varphi(x,y) = \sum_{p\in P}\sum_{i\in\Delta}\left[\sum_{j\in\Delta|i\le j}\sum_{k\in K} q^k_{(p,i,j)} x^k_{(p,i,j)} - c^{st}_{(p,i)}\right] y_{(p,i)} \tag{20}$$

Theorem 4 *(see [4]) A global maxima of the bilinear problem (20) is a solution or can be transformed into a solution of the problem (15)-(19).*

3 Methods

In the previous section, we have discussed several problems arising in the supply chain management. To solve the bilinear formulations of the problems, one can employ techniques applicable for general bilinear problems. In particular, a cutting plain algorithm proposed by Konno can be applied to find a global solution of the problems. In addition, he proposes an iterative procedure, which converges to a local minimum of the problem in a finite number of iterations. For details on the procedure, which is also known as "mountain climbing" procedure (MCP), and the cutting plain algorithm we refer to the paper [1] or Bilinear Programming section of this encyclopedia.

Below, we discuss problem specific difficulties of applying the above mentioned algorithms and some effective heuristic procedures, which are able to provide a near optimum solution using negligible computer resources. The MCP, which is used by the heuristics to find a local minimum/maximum of the problems, is very fast due to a special structure of both LP problems employed by the procedure. However, to obtain a high quality solution, in some problems it is necessary to solve a sequence of approximate problems. The bilinear formulations of the supply chain problems typically have many local minima. Therefore, cutting plain algorithms may require many cuts to converge. By combining the heuristic procedures with the cutting plain algorithm, one can reduce the number of cuts by generating deep cuts.

One of the main properties of a bilinear problem with a disjoint feasible region is that by fixing vectors x or y to a particular value, the problem reduces to a linear one. The

"mountain climbing" procedure employs this property and iteratively solves two linear problems by fixing the corresponding vectors to the solution of the corresponding linear programs. In the case of concave piecewise linear network flow problem, given the vector $\hat{x}$, the problem (5)-(8) can be decomposed into $|A|$ problems,

$$\min_{\{y_a^k | k \in K_a\}} \sum_{k \in K_a} [c_a^k \hat{x}_a + s_a^k] y_a^k$$

$$\text{s.t.} \qquad \sum_{k \in K_a} y_a^k = 1, \qquad y_a^k \geq 0 \quad \forall k \in K_a.$$

Furthermore, it can be shown that a solution of the problem is a binary vector, which has to satisfy the inequality

$$\sum_{k \in K_a} \xi_a^{k-1} y_a^k \leq \hat{x}_a \leq \sum_{k \in K_a} \xi_a^k y_a^k.$$

As a result, one can employ a search technique by assigning $y_a^{\hat{k}} = 1$ if $\xi_a^{\hat{k}-1} \leq \hat{x}_a \leq \xi_a^{\hat{k}}$ and $y_a^k = 0$, $\forall k \in K_a$, $k \neq \hat{k}$. On the other hand, by fixing the vector y to the value of the constructed vector $\hat{y}$, the problem (5)-(8) reduces to the following network flow problem.

$$\min_x \sum_{a \in A} \left[\sum_{k \in K_a} c_a^k \hat{y}_a^k \right] x_a$$

$$\text{s.t.} \qquad Bx = b, \qquad x_a \geq 0, \qquad \forall a \in A$$

Observe that $\sum_{k \in K_a} c_a^k \hat{y}_a^k = c_a^{\hat{k}}$, and different vectors $\hat{y}$ change the cost vector in the problem.

Although the MCP converges to a local minimum, it can provide a near optimum solution for the problem (5)-(8) if the initial vector $\hat{y}$ is such that $\hat{y}_a^{n_a} = 1$ and $\hat{y}_a^k = 0$, $\forall k \in K_a$, $k \neq n_a$. The effectiveness of the procedure is partially due to the fact that in the supply chain problems $f_a(x_a)$ is an increasing function. In addition, the procedure requires less computer resources to converge because both linear problems are relatively easy to solve. A detailed description of the procedure, properties of the linear problems, and computational experiments can be found in [2].

In the case of fixed charge network flow problems, it is not obvious how to choose the vector ε. Theorem 3 guarantees the equivalence between the fixed charge network flow problem and the bilinear problem (12)-(14) if $\varepsilon_a \in (0, \delta]$. However, according to the definition, it is necessary to find all vertices of the feasible region to compute the value of δ, which is computationally expensive. Even if the correct value of δ is known, typically it is a very small number. As a result, the value of ε_a is close to zero, and $c_a^{\varepsilon_a}$ is very large compared to the value of c_a. The later creates some difficulties for finding a global solution of the bilinear problem. In particular, the MCP may converge to a local minimum, which is far from being a global solution.

To overcome those difficulties, [3] proposes a procedure where it gradually decreases the value of ε (see Algorithm 1). The algorithm starts from an initial value for the vector ε,

Algorithm 1 :

Step 1: Let $\varepsilon_a \leftarrow \lambda_a$, $x_a^0 \leftarrow 0$, $y_a^0 \leftarrow 0$, and $m \leftarrow 1$.
Step 2: Find a local minimum of the problem (12)-(14) using the MCP. Let (x^m, y^m) denote the solution found by the algorithm.
Step 3: If $\exists a \in A$ such that $x_a^m \in (0, \varepsilon_a^m)$ then $\varepsilon_a \leftarrow \alpha\varepsilon_a$, $m \leftarrow m+1$, and go to step 2. Otherwise, stop.

i.e., $\varepsilon_a = \lambda_a$. After constructing the corresponding bilinear problem, it employs the MCP to find a local minimum of the problem. If the stopping criteria is not satisfied, the value of ε is updated, i.e., $\varepsilon_a = \alpha\varepsilon_a$ where $\alpha \in (0,1)$, and the algorithm again solves the updated bilinear problem using the current solution as an initial vector for the MCP.

The choice of α has a direct influence on the CPU time of the algorithm and the quality of the solution. Specifically, if the value of α is closer to one, then due to the fact that ε decreases slowly, the algorithm requires many iterations to stop. On the other hand, if the values of the parameter is closer to zero, it may worsen the quality of the solution. A proper choice of the parameter depends on the problem, and it should be chosen by trials and errors. In the paper [3], the authors test the algorithm on various randomly generated test problems and found satisfactory to choose $\alpha = 0.5$.

As for the stopping criteria, it is possible to show that the solution of the final bilinear problem is the solution of the fixed charge network flow problem if on Step 2 one is able to find a global solution of the corresponding bilinear problems. For details on the numerical experiments, stopping criteria and other properties of the algorithm, we refer to [3].

In the problems with pricing decisions, one may also experience some difficulties to employ the MCP for finding a near optimum solution. To explore the properties of the problem, consider the following two linear problems, which are constructed from the problem (20) by fixing either vector x or y to the value of the vector $\hat{x}$ or $\hat{y}$, respectively.

$$LP_1: \quad \max_{y \in Y} \sum_{p \in P} \sum_{i \in \Delta} \left[\sum_{j \in \Delta | i \le j} \sum_{k \in K} q^k_{(p,i,j)} \hat{x}^k_{(p,i,j)} - c^{st}_{(p,i)} \right] y_{(p,i)}$$

$$LP_2: \quad \max_{x \in X} \sum_{p \in P} \sum_{i \in \Delta} \sum_{j \in \Delta | i \le j} \sum_{k \in K} \left[q^k_{(p,i,j)} \hat{y}_{(p,i)} \right] x^k_{(p,i,j)}.$$

The MCP solves iteratively LP_1 and LP_2 problems, where the solution of the first problem is used to fix the corresponding vector in the second problem. However, if one of the components of the vector y equals to zero during one of the iterations, e.g., $\hat{y}_{(p,i)} = 0$, then in the second problem coefficients of the corresponding variables $x^k_{(p,i,j)}$ are equal to zero as well. As a result, changes in the values of those variables do not have any influence on the objective function value. Furthermore, because the products "share" the capacity and other products may have positive coefficients in the objective, it is likely that at optimum of LP_2, $\hat{x}^k_{(p,i,j)} = 0$, $\forall j \in \Delta$, $k \in K$. From the later, it follows that $\hat{y}_{(p,i)} = 0$ during the next iteration, and one concludes that if some products are eliminated from the problem during the iterative process, the MCP does not consider them again. Therefore, it is likely that the solution returned by the algorithm is far from being a global one. To avoid zero coefficients in the objective of LP_2, [4] proposes an approximation to the problem (20),

which can be used in the MCP to find a near optimum solution.

To construct the approximate problem, let

$$\varphi^1_{(p,i)}(x_{(p,i)}) = \sum_{j\in\Delta|i\le j}\sum_{k\in K} q^k_{(p,i,j)}x^k_{(p,i,j)} - c^{st}_{(p,i)}, \text{ and}$$

$$\varphi^2_{(p,i)}(x_{(p,i)}) = \frac{\varepsilon_{(p,i)}}{\varepsilon_{(p,i)} + c^{st}_{(p,i)}}\sum_{j\in\Delta|i\le j}\sum_{k\in K} q^k_{(p,i,j)}x^k_{(p,i,j)},$$

where $\varepsilon_{(p,i)} > 0$, and $x_{(p,i)}$ is the vector of $x^k_{(p,i,j)}$. Using those functions, construct the following bilinear problem

$$\max_{x\in X, y\in Y} \varphi^\varepsilon(x,y) = \sum_{p\in P}\sum_{i\in\Delta}\left[\varphi^1_{(p,i)}(x_{(p,i)})y_{(p,i)} + \varphi^2_{(p,i)}(x_{(p,i)})(1-y_{(p,i)})\right], \tag{21}$$

where the feasible region is the same as in the problem (20). The authors show that $\varphi^\varepsilon(x,y)$ approximates the function $\varphi(x,y)$ from above.

Theorem 5 *(see [4]) There exists a sufficiently small $\varepsilon > 0$ such that a solution of the problem (20) is a solution of the problem (21).*

Algorithm 2 starts from a sufficiently large value of $\varepsilon_{(p,i)}$ and finds a local maximum of the corresponding bilinear problem (21) using the MCP. If the stopping criteria is not satisfied then it updates the value of ε to $\alpha\varepsilon$, updates the bilinear problem (21), and employs the MCP to find a better solution. Similar to the fixed charge network flow problem, the choice of α has a direct influence on the CPU time of the algorithm and the quality of the returned solution. The running time of the algorithm and the quality of the solution for the different values of α are studied in [4].

In addition to α, one has to find a proper initial value for the parameter $\varepsilon_{(p,i)}$. Ideally, it should be equal to the maximum profit that can be generated by producing only product p at time i. However, it requires solving a linear problem for each pair $(p,i) \in P \times \Delta$, which is computationally expensive. On the other hand, it is not necessary to find an exact solution of those LPs, and one might consider a heuristic procedure which provides a quality solution within a reasonable time. One of such procedures is discussed in [4].

Algorithm 2 :

Step 1: Let $\varepsilon_{(p,i)}$ be a sufficiently large number, $y^0_{(p,i)} = 1$, $\forall p \in P$, $i \in \Delta$, and $m \leftarrow 0$.
Step 2: Construct the approximation problem (21), and find a local maximum of the problem using the MSP. Let (x^{m+1}, y^{m+1}) denote the solution returned by the algorithm.
Step 3: If $\exists p \in P$ and $i \in \Delta$ such that $\sum_{j\in\Delta|i\le j}\sum_{k\in K} q^k_{(p,i,j)}x^{(m+1)k}_{(p,i,j)} - c^{st}_{(p,i)} \le \varepsilon^m_{(p,i)}$ and $\sum_{j\in\Delta|i\le j}\sum_{k\in K} x^{(m+1)k}_{(p,i,j)} > 0$ then $\varepsilon \leftarrow \alpha\varepsilon$, $m \leftarrow m+1$ and go to Step 2. Otherwise, stop.

References

[1] Konno H. A Cutting Plane Algorithm for Solving Bilinear Programs. Mathematical Programming 1976;11:14-27.

[2] Nahapetyan A, Pardalos P. A Bilinear Relaxation Based Algorithm for Concave Piecewise Linear Network Flow Problems. Journal of Industrial and Management Optimization 3(1), pp. 71-85, 2007.

[3] Nahapetyan A, Pardalos P. Adaptive Dynamic Cost Updating Procedure for Solving Fixed Charge Network Flow Problems. Computational Optimization and Applications, submitted for publication.

[4] Nahapetyan A, Pardalos P. A Bilinear Reduction Based Algorithm for Solving Capacitated Multi-Item Dynamic Pricing Problems. to appear in the Computers and Operations Research journal, 2007.