Supply Chain Problems
Conclusion

Bilinear Reduction Based Algorithm to Solve Network Flow Problems with Concave Cost Functions

Artyom Nahapetyan

University of Florida
Department of Industrial and Systems Engineering

November 9, 2006
Supply Chain Problems

1. Concave Piecewise Linear Network Flow Problem
2. Fixed Charge Network Flow Problem
3. Capacitated Multi-Item Dynamic Pricing Problems

Conclusion
CPLNF Problem

Let

- \( G(N, A) \) represent a network, and
- \( f_a(x_a) \) denote a cost function of arc \( a \).

\[
\min_x \sum_{a \in A} f_a(x_a)
\]

s.t. \( Bx = b \)

\( x_a \in [\lambda^0_a, \lambda^{n_a}_a] \quad \forall a \in A \)

\( B \) - node-arc incident matrix of the network \( G \)
\( f_a(x_a) \) - concave piecewise linear functions
Arc Cost Function

Function $f_a(x_a)$

$$f_a(x_a)$$

$\lambda_a^1, \lambda_a^2, \lambda_a^3, \lambda_a^4$
CPLNF-IP Problem

\[
\min_{x,y} \sum_{a \in A} \sum_{k \in K_a} c_a^k x_a^k + \sum_{a \in A} \sum_{k \in K_a} s_a^k y_a^k
\]

\[Bx = b\]

\[\sum_{k \in K_a} x_a^k = x_a\]

\[\sum_{k \in K_a} \lambda_a^{k-1} y_a^k \leq x_a \leq \sum_{k \in K_a} \lambda_a^k y_a^k,\]

\[\sum_{k \in K_a} y_a^k = 1\]

\[x_a^k \leq My_a^k,\]

\[x_a^k \geq 0,\]

\[y_a^k \in \{0, 1\}\]

where \(x_a^k\) denotes the portion of the total flow that has cost according to the linear function \(f_a^k(x_a) = c_a^k x_a + s_a^k\)
Relaxation of the Binary Variables

CPLNF-R Problem

\[
\min_{x,y} \sum_{a \in A} \sum_{k \in K_a} c_a^k x_a^k + \sum_{a \in A} \sum_{k \in K_a} s_a^k y_a^k
\]

\[
Bx = b
\]

\[
\sum_{k \in K_a} x_a^k = x_a
\]

\[
\sum_{k \in K_a} \lambda_a^{k-1} y_a^k \leq x_a \leq \sum_{k \in K_a} \lambda_a^k y_a^k,
\]

\[
\sum_{k \in K_a} y_a^k = 1
\]

\[
x_a^k = x_a y_a^k, \quad x_a^k \geq 0, \quad y_a^k \geq 0
\]
Relaxation of the Binary Variables

CPLNF-R Problem

\[
\begin{align*}
\min_{x, y} & \quad \sum_{a \in A} \sum_{k \in K_a} c_a^k x_a^k + \sum_{a \in A} \sum_{k \in K_a} s_a^k y_a^k \\
\text{s.t.} & \quad Bx = b \\
& \quad \sum_{k \in K_a} \lambda_{a}^{k-1} y_a^k \leq x_a \leq \sum_{k \in K_a} \lambda_{a}^k y_a^k, \\
& \quad \sum_{k \in K_a} y_a^k = 1 \\
& \quad x_a^k = x_a y_a^k, \quad y_a^k \geq 0
\end{align*}
\]
Relaxation of the Binary Variables

CPLNF-R Problem

\[
\min_{x,y} \sum_{a \in A} \left[ \sum_{k \in K_a} c_{ka} y_{ka} \right] x_a + \sum_{a \in A} \sum_{k \in K_a} s_{ka} y_{ka} \\
Bx = b \\
\sum_{k \in K_a} \lambda_{ka}^{k-1} y_{ka} \leq x_a \leq \sum_{k \in K_a} \lambda_{ka}^{k} y_{ka}, \quad \sum_{k \in K_a} y_{ka} = 1 \\
y_{ka} \geq 0
\]
Theoretical Results

**Lemma**

Any feasible vector of the CPLNF-IP problem is feasible to the CPLNF-R

**Lemma**

Any local optimum of the CPLNF-R problem is either feasible to the CPLNF-IP or leads to a feasible vector of CPLNF-IP with the same objective function value.

**Theorem**

A global optimum of the CPLNF-R problem is a solution or leads to a solution of the CPLNF-IP.
Theoretical Results

**Lemma**

Any feasible vector of the CPLNF-IP problem is feasible to the CPLNF-R

**Lemma**

Any local optimum of the CPLNF-R problem is either feasible to the CPLNF-IP or leads to a feasible vector of CPLNF-IP with the same objective function value.

**Theorem**

A global optimum of the CPLNF-R problem is a solution or leads to a solution of the CPLNF-IP.
Theoretical Results

**Lemma**

Any feasible vector of the CPLNF-IP problem is feasible to the CPLNF-R

**Lemma**

Any local optimum of the CPLNF-R problem is either feasible to the CPLNF-IP or leads to a feasible vector of CPLNF-IP with the same objective function value.

**Theorem**

A global optimum of the CPLNF-R problem is a solution or leads to a solution of the CPLNF-IP.
Two Problems

CPLNF-R Problem

\[
\min_{x, y} \sum_{a \in A} \left[ \sum_{k \in K_a} c^k_a y^k_a \right] x_a + \sum_{a \in A} \sum_{k \in K_a} s^k_a y^k_a
\]

\[ Bx = b \]

\[ \sum_{k \in K_a} \lambda^k_a y^k_a \leq x_a \leq \sum_{k \in K_a} \lambda^k_a y^k_a \iff x_a \in [0, \lambda^{n_a}_a] \]

\[ \sum_{k \in K_a} y^k_a = 1 \]

\[ y^k_a \geq 0 \]
CPLNF-R Problem

\[
\begin{align*}
\min_{x,y} & \quad \sum_{a \in A} \left[ \sum_{k \in K_a} c_{a}^{k} y_{a}^{k} \right] x_{a} + \sum_{a \in A} \sum_{k \in K_a} s_{a}^{k} y_{a}^{k} \\
\text{subject to} & \quad Bx = b \\
& \quad \sum_{k \in K_a} \lambda_{a}^{k-1} y_{a}^{k} \leq x_{a} \leq \sum_{k \in K_a} \lambda_{a}^{k} y_{a}^{k} \iff x_{a} \in [0, \lambda_{a}^{n_{a}}] \\
& \quad \sum_{k \in K_a} y_{a}^{k} = 1 \\
& \quad y_{a}^{k} \geq 0
\end{align*}
\]
Two Problems

LP(y) Problem \((y \text{ is fixed})\)

\[
\min_x \sum_{a \in A} \left[ \sum_{k \in K_a} c_{a}^k y_{a}^k \right] x_{a} \\
Bx = b \\
x_{a} \in [0, \lambda_{a}^{na}]
\]
Two Problems

CPLNF-R Problem

\[
\min_{x, y} \sum_{a \in A} \sum_{k \in K_a} \left[ c_a^k x_a + s_a^k \right] y_a^k \\
Bx = b \\
\sum_{k \in K_a} \lambda_a^{k-1} y_a^k \leq x_a \leq \sum_{k \in K_a} \lambda_a^k y_a^k \iff x_a \in [0, \lambda_a^{n_a}] \\
\sum_{k \in K_a} y_a^k = 1 \\
y_a^k \geq 0
\]
Two Problems

CPLNF-R Problem

\[
\begin{aligned}
\min_{x,y} \sum_{a \in A} \sum_{k \in K_a} \left[ c^k_a x_a + s^k_a \right] y^k_a \\
Bx &= b \\
\sum_{k \in K_a} \lambda_a^{k-1} y^k_a &\leq x_a \leq \sum_{k \in K_a} \lambda_a^k y^k_a \iff x_a \in [0, \lambda_a^{n_a}] \\
\sum_{k \in K_a} y^k_a &= 1 \\
y^k_a &\geq 0
\end{aligned}
\]
Two Problems

**LP($x$) Problem (x is fixed)**

\[
\min_y \sum_{a \in A} \sum_{k \in K_a} \left[ c^k_a x_a + s^k_a \right] y^k_a \\
\sum_{k \in K_a} y^k_a = 1, \quad y^k_a \geq 0
\]

- A binary variable that satisfies the inequality is a solution of the problem.

\[
\sum_{k \in K_a} \lambda^{k-1} y^k_a \leq x_a \leq \sum_{k \in K_a} \lambda^k y^k_a
\]
Two Problems

LP($x$) Problem ($x$ is fixed)

$$\min_y \sum_{a \in A} \sum_{k \in K_a} \left[c_a^k x_a + s_a^k\right] y_a^k$$

$$\sum_{k \in K_a} y_a^k = 1, \quad y_a^k \geq 0$$

- A binary variable that satisfies the inequality is a solution of the problem.

$$\sum_{k \in K_a} \lambda_a^{k-1} y_a^k \leq x_a \leq \sum_{k \in K_a} \lambda_a^k y_a^k$$
Dynamic Cost Updating Procedure (DCUP)

**DCUP: Iteratively Solves** $LP(x)$ and $LP(y)$

**Step 1:** Let $y^0$ denote the initial vector of $y^k_a$, where $y^0_{10} = 1$ and $y^k_a = 0$, $\forall k \in K_a$, $k \neq 1$. $m \leftarrow 1$.

**Step 2:** Let $x^m = \text{argmin}\{LP(y^{m-1})\}$, and $y^m = \text{argmin}\{LP(x^m)\}$.

**Step 3:** If $y^m = y^{m-1}$ then stop. Otherwise, $m \leftarrow m + 1$ and go to Step 2.
Theoretical Results

**Theorem**

Given any initial binary vector $y^0$, DCUP converges in a finite number of iterations.

**Theorem**

Let $(x^*, y^*)$ be the solution returned by DCUP. If $y^*$ is a unique solution of the LP$(x^*)$ problem then $(x^*, y^*)$ is a local minimum of CPLNF-R.
Theoretical Results

**Theorem**

Given any initial binary vector $y^0$, DCUP converges in a finite number of iterations.

**Theorem**

Let $(x^*, y^*)$ be the solution returned by DCUP. If $y^*$ is a unique solution of the LP$(x^*)$ problem then $(x^*, y^*)$ is a local minimum of CPLNF-R.
Supply Chain Problems

Conclusion

Concave Piecewise Linear Network Flow Problem
Fixed Charge Network Flow Problem
Capacitated Multi-Item Dynamic Pricing Problems

DSSP

\[ f_a(x_a) \]

\[ x_a \]

Artyom Nahapetyan INFORMS 2006 - Supply Chain Optimization
Supply Chain Problems

Conclusion

Concave Piecewise Linear Network Flow Problem
Fixed Charge Network Flow Problem
Capacitated Multi-Item Dynamic Pricing Problems

DSSP
Supply Chain Problems

Conclusion

Concave Piecewise Linear Network Flow Problem
Fixed Charge Network Flow Problem
Capacitated Multi-Item Dynamic Pricing Problems

DSSP

\[ f_a(x_a) \]

\[ \lambda_a^1 \quad \lambda_a^2 \quad \lambda_a^3 \quad \lambda_a^4 \quad x_a \]
Supply Chain Problems

Conclusion

Concave Piecewise Linear Network Flow Problem
Fixed Charge Network Flow Problem
Capacitated Multi-Item Dynamic Pricing Problems

DSSP

\[ f_a(x_a) \]

\[ x_a^1 \quad x_a^2 \quad x_a^3 \quad x_a^4 \]
Supply Chain Problems

Conclusion

Concave Piecewise Linear Network Flow Problem
Fixed Charge Network Flow Problem
Capacitated Multi-Item Dynamic Pricing Problems

DSSP

\[ f_a(x_a) \]

\[ \lambda_a^1 \quad \lambda_a^2 \quad \lambda_a^3 \quad \lambda_a^4 \quad x_a \]

Test Problems

Parameters of the Problems

- Demand: U[10,20], U[20,30], or U[30,40]
- Number of linear pieces: 5 or 10.
- Total number of problems sets is 30.
- There are 30 problems per problem set. (In total 900 problems)
Test Problems

Computational Results

- DCUP provides a better solution than DSSP in about 50% of the test problems.

- Both algorithms provide the same solution in about 30% of the test problems.

- DCUP spends 2-5 times less CPU time than DSSP.

- DCUP converges using less number of iterations than DSSP.
Description of the Problem

**FCNF Problem**

\[
\min_x f(x) = \sum_{a \in A} f_a(x_a)
\]

s.t. \( Bx = b \),

\( x_a \in [0, \lambda_a] \),

where \( f_a(x_a) = \begin{cases} 
  c_a x_a + s_a & x_a \in (0, \lambda_a] \\
  0 & x_a = 0
\end{cases} \)
Supply Chain Problems
Conclusion

Concave Piecewise Linear Network Flow Problem
Fixed Charge Network Flow Problem
Capacitated Multi-Item Dynamic Pricing Problems

ε-Approximation

$\mathbf{s}_a$

$f_a(\mathbf{x}_a)$

$\lambda_a$
Supply Chain Problems
Conclusion

Consistency

Concave Piecewise Linear Network Flow Problem
Fixed Charge Network Flow Problem
Capacitated Multi-Item Dynamic Pricing Problems

$\varepsilon$-Approximation

$\varphi_a(s_a, x_a)$

$\varepsilon_a$

$\lambda_a$

Artyom Nahapetyan
INFORMS 2006 - Supply Chain Optimization
ε-Approximation

CPLNF(ε) Problem

\[
\min_x \phi^\varepsilon(x) = \sum_{a \in A} \phi_a^\varepsilon(x_a)
\]

\text{s.t. } Bx = b, \quad x_a \in [0, \lambda_a],

where \( \varepsilon \) is a vector of \( \varepsilon_a \).
Let

- \( x^* = \arg \min (FCNF) \),
- \( x^\varepsilon = \arg \min (CPLNF(\varepsilon)) \), and
- \( \delta = \min \{ x^\varepsilon_v | x^\varepsilon_v \in V, a \in A, x^\varepsilon_v > 0 \} \), where \( V \) represents the set of vertices of the feasible region.

Theorem

For all \( \varepsilon \) such that \( \varepsilon_a \in (0, \lambda_a] \), \( \forall a \in A \), \( \phi^\varepsilon(x^\varepsilon) \leq f(x^*) \).

Theorem

For all \( \varepsilon \) such that \( \varepsilon_a \in (0, \delta] \), \( \forall a \in A \), \( \phi^\varepsilon(x^\varepsilon) = f(x^*) \).
Theoretical Results

Let

- \( x^* = \text{argmin}(FCNF) \),
- \( x^\varepsilon = \text{argmin} \ (CPLNF(\varepsilon)) \), and
- \( \delta = \min\{x^\nu_a \mid x^\nu \in V, a \in A, x^\nu_a > 0\} \), where \( V \) represents the set of vertices of the feasible region.

**Theorem**

For all \( \varepsilon \) such that \( \varepsilon_a \in (0, \lambda_a] \), \( \forall a \in A \), \( \phi^\varepsilon(x^\varepsilon) \leq f(x^*) \).

**Theorem**

For all \( \varepsilon \) such that \( \varepsilon_a \in (0, \delta] \), \( \forall a \in A \), \( \phi^\varepsilon(x^\varepsilon) = f(x^*) \).
Theoretical Results

Let

- $x^* = \text{argmin}(\text{FCNF})$,
- $x^\varepsilon = \text{argmin}(\text{CPLNF}(\varepsilon))$, and
- $\delta = \min \{x^\varepsilon_a | x^\varepsilon \in V, a \in A, x^\varepsilon_a > 0\}$, where $V$ represents the set of vertices of the feasible region.

**Theorem**

*For all $\varepsilon$ such that $\varepsilon_a \in (0, \lambda_a]$, $\forall a \in A$, $\phi^\varepsilon(x^\varepsilon) \leq f(x^*)$.***

**Theorem**

*For all $\varepsilon$ such that $\varepsilon_a \in (0, \delta]$, $\forall a \in A$, $\phi^\varepsilon(x^\varepsilon) = f(x^*)$.***
Adaptive Dynamic Cost Updating Procedure
Adaptive Dynamic Cost Updating Procedure

\[ f_a(x_a) \]

\[ \phi_a(x_a) \]

\[ S_a \]

\[ \lambda_a \]

\[ \lambda_2 / 2 \]
Adaptive Dynamic Cost Updating Procedure
Test Problems

Parameters of the Problems

- Setup cost: U[50,100], U[100,200], or U[200,400]
- Slope: U[1,5], U[10,20], or U[30,40]
- Demand: U[30,50]
- Total number of problems sets is 36.
- There are 30 problems per problem set. (In total 1080 problems)
## Test Problems

### Computational Results

- ADCUP provides a better solution than DSSP
  - Small problems - 62%, and
  - Large problems - 98%.

- Both algorithms provide the same solution
  - Small problems - 28%, and
  - Large problems - 0%.

- ADCUP spends 2-8 times less CPU time than DSSP.

- ADCUP converges using less number of iterations than DSSP.
Problem Description

Given Data:

- Set of products.
- Unit and setup costs.
- Inventory costs.
- Production capacities.
- Demand is a function of the price.

Problem

Find a production and pricing policy, which maximizes the profit.
**Problem Description**

**Given Data:**
- Set of products.
- Unit and setup costs.
- Inventory costs.
- Production capacities.
- Demand is a function of the price.

**Problem**
Find a production and pricing policy, which maximizes the profit.
Price-Demand and Profit-Demand Relationships

\[ g(d) = f(d)d \]
Formulation

**CMDP Problem**

\[
\begin{align*}
\max_{x,y,d} & \sum_{p \in P} \sum_{j \in \Delta} g(p,j) \left( \sum_{i \in \Delta | i \leq j} x(p,i,j) \right) \\
- & \sum_{p \in P} \sum_{i,j \in \Delta | i \leq j} \left[ c_{in}(p,i,j) + c_{pr}(p,i) \right] x(p,i,j) - \sum_{p \in P} \sum_{i \in \Delta} c_{st}(p,i) y(p,i) \\
\text{s.t.} & \sum_{p \in P} \sum_{j \in \Delta | i \leq j} x(p,i,j) \leq C_i, \sum_{j \in \Delta | i \leq j} x(p,i,j) \leq C_i y(p,i), \\
& x(p,i,j) \geq 0, y(p,i) \in \{0, 1\}
\end{align*}
\]
Approximation
Approximate Formulation

**ACMDP Problem**

\[
\max_{x, y, \lambda} \sum_{p \in P} \sum_{j \in \Delta} \sum_{k \in K} g_{(p,j)}^k \lambda^k_{(p,j)}
\]

\[
- \sum_{p \in P} \sum_{i,j \in \Delta| i \leq j} \left[ c_{(p,i,j)}^{in} + c_{(p,i)}^{pr} \right] x_{(p,i,j)} - \sum_{p \in P} \sum_{i \in \Delta} c_{(p,i)}^{st} y_{(p,i)},
\]

s.t. \[
\sum_{p \in P} \sum_{j \in \Delta| i \leq j} x_{(p,i,j)} \leq C_i, \quad \sum_{j \in \Delta| i \leq j} x_{(p,i,j)} \leq C_i y_{(p,i)},
\]

\[
\sum_{i \in \Delta| i \leq j} x_{(p,i,j)} = \sum_{k \in K} d_{(p,j)}^k \lambda^k_{(p,j)}, \quad \sum_{k=0}^N \lambda^k_{(p,j)} = 1,
\]

\[
x_{(p,i,j)} \geq 0, \lambda^k_{(p,j)} \geq 0, y_{(p,i)} \in \{0, 1\}.
\]
Approximate Formulation

**ACMDP Problem**

\[
\begin{align*}
\max_{x,y} \ & \sum_{p \in P} \sum_{i \in \Delta} \left[ \sum_{j \in \Delta | i \leq j} \sum_{k \in K} q^{k}_{(p,i,j)} x^{k}_{(p,i,j)} - c^{st}_{(p,i)} y_{(p,i)} \right] \\
\text{s.t.} \ & \sum_{p \in P} \sum_{j \in \Delta | i \leq j} \sum_{k \in K} x^{k}_{(p,i,j)} \leq C_{i}, \ \sum_{j \in \Delta | i \leq j} \sum_{k \in K} x^{k}_{(p,i,j)} \leq C_{i} y_{(p,i)}, \ \sum_{k \in K} \sum_{i \in \Delta | i \leq j} \frac{x^{k}_{(p,i,j)}}{d^{k}_{(p,j)}} \leq 1, \ x^{k}_{(p,i,j)} \geq 0, \ y_{(p,i)} \in \{0, 1\},
\end{align*}
\]

where \( q^{k}_{(p,i,j)} = f^{k}_{(p,j)} - c^{in}_{(p,i,j)} - c^{pr}_{(p,i)} \)
Objective Function

\[ \sum_{j \in \Delta t \leq j \in K} \sum_{k \in K} q_{(p, i, j)}^k x_{(p, i, j)}^k - c_{(p, i)}^{st} y_{(p, i)} \]

\[ \sum_{j \in \Delta t \leq j \in K} \sum_{k \in K} x_{(p, i, j)}^k \]
Objective Function

\[ \sum_{j \in \Delta i \leq j} \sum_{k \in K} q_{(p,i,j)}^k x_{(p,i,j)}^k - c_{(p,i)}^s v_{(p,i)} \]

\[ \sum_{j \in \Delta i \leq j} \sum_{k \in K} x_{(p,i,j)}^k \]
ACMDP-B Problem

\[
\max_{x,y} \sum_{p \in P} \sum_{i \in \Delta} \left[ \sum_{j \in \Delta} \sum_{i \leq j} \sum_{k \in K} q_{(p,i,j)}^k x_{(p,i,j)}^k - c_{(p,i)}^{st} \right] y(p,i) = \varphi(x, y)
\]

\[
x \in X \text{ and } y \in Y,
\]

where \( Y = [0, 1]^{|P|\cdot|\Delta|} \) and

\[
X = \{ x \mid x_{(p,i,j)}^k \geq 0, \sum_{p,k,j \mid i \leq j} x_{(p,i,j)}^k \leq C_i, \sum_{k,i \mid i \leq j} \frac{x_{(p,i,j)}^k}{d_{(p,j)}^k} \leq 1 \}
\]
Theoretical Results

**Theorem**

Any local maximum of the ACMDP-B problem is feasible or leads to a feasible solution of the ACMDP problem with the same objective function value.

**Theorem**

A global maximum of the ACMDP-B problem is a solution or leads to a solution of the ACMDP problem.
Theoretical Results

**Theorem**

Any local maximum of the ACMDCP-B problem is feasible or leads to a feasible solution of the ACMDCP problem with the same objective function value.

**Theorem**

A global maximum of the ACMDCP-B problem is a solution or leads to a solution of the ACMDCP problem.
Procedure 1

Two LPs

\[ \text{LP}(x) : \] 
\[
\max_{y \in Y} \sum_{p \in P} \sum_{i \in \Delta} \left[ \sum_{j : i \leq j} \sum_{k \in K} q^k_{(p,i,j)} x^k_{(p,i,j)} - c^st_{(p,i)} \right] y_{(p,i)}
\]

\[ \text{LP}(y) : \] 
\[
\max_{x \in X} \sum_{p \in P} \sum_{i \in \Delta} \sum_{j : i \leq j} \sum_{k \in K} \left[ q^k_{(p,i,j)} y_{(p,i)} \right] x^k_{(p,i,j)}
\]
Procedure 1

**Step 1:** Let $y^0$ denote an initial binary vector, where $y_{(p,i)} = 1$. $m \leftarrow 1$.

**Step 2:** Let $x^m = \text{argmax}\{LP(y^{m-1})\}$, and $y^m = \text{argmax}\{LP(x^m)\}$.

**Step 3:** If $y^m = y^{m-1}$ then stop. Otherwise, $m \leftarrow m + 1$ and go to Step 2.
Procedure 1

Disadvantage

The quality of the solution is not good enough.

- Procedure 1 converges to a local maximum of the problem.

- If $y^m_{(p,i)} = 0$ then in all following iterations $x^k_{(p,i,j)} = 0$, $\forall j \in \Delta$, $i \leq j$, and $p \in P$.

- As a result, the local maximum can be far from being a global one.
Approximation of the Objective Function

\[ E(p_j) \]
Approximation of the Objective Function
Approximation of the Objective Function

\[ \sum_{(i,j)} \alpha^2 \varepsilon_{(i,j)} \]

\[ \sum_{(i,j)} \varepsilon_{(i,j)} \]
Approximation of the Objective Function

\[ \varepsilon_{(p,j)} \]

\[ \alpha^3 \varepsilon_{(p,j)} \]
Procedure 2

**Step 1:** Let \( \varepsilon(p,i) \) be a sufficiently large number, and \( y^0 \) be such that \( y^0_{(p,i)} = 1, \forall p \in P \) and \( i \in \Delta \). \( m \leftarrow 0 \).

**Step 2:** Construct the approximation problem and run Procedure 1 to find a local maximum of the problem, where \( y^m \) is an initial binary vector. Let \( (x^{m+1}, y^{m+1}) \) denote the local maximum.

**Step 3:** If \( \exists p \in P \) and \( i \in \Delta \) such that
\[
\sum_{j \in \Delta | i \leq j} \sum_{k \in K} q^k_{(p,i,j)} x^{(m+1)k}_{(p,i,j)} - c^{st}_{(p,i)} \leq \varepsilon^m_{(p,i)} \text{ and }
\sum_{j \in \Delta | i \leq j} \sum_{k \in K} x^{(m+1)k}_{(p,i,j)} > 0
\]
then \( \varepsilon \leftarrow \alpha \varepsilon \), \( m \leftarrow m + 1 \) and go to Step 2. Otherwise, stop.
How Big is $\varepsilon(p,i)$?

**Procedure 3**

For all $i \in \Delta$ and $p \in P$ assign the available capacity first to the variables with a higher value of $q^k_{(p,i,j)}$
Supply Chain Problems
Concave Piecewise Linear Network Flow Problem
Fixed Charge Network Flow Problem
Capacitated Multi-Item Dynamic Pricing Problems

Test Problems

Parameters of the Problems

- Number of products: $|P| = 5$, 10, or 20.
- Planning horizon: $|\Delta| = 12$, or 52.
- Capacity: $C_i = |P| \cdot U[10, 100]$, $|P| \cdot U[50, 150]$, $|P| \cdot U[100, 200]$, or $|P| \cdot U[150, 250]$.
- Costs: $c^{pr}_{(p,i)} = U[20, 40]$, $c^{st}_{(p,i)} = U[600, 1000]$, and $c^{in}_{(p,i)} = U[4, 8]$.
- Maximum price/demand: $U[70, 90]/U[500, 1000]$.
- Profit per unit of investment: $\beta \in [0.7, 1.3]$.
- The value of the parameter $\alpha$: 1/2, 2/3, 9/10.
- Total number of problem sets is 24.
- There are 10 problems per problem set. In total 240 problems.
Test Problems

Computational Results

- Quality of the solution:
  - Procedure 1 - 1-8.3%, and
  - Procedure 2 - <1.2%.

- CPU time:
  - Procedure 1 - 0.2-35 sec, and
  - Procedure 2
    - $\alpha = 1/2$ - 0.5-58 sec,
    - $\alpha = 2/3$ - 0.8-78 sec, and
    - $\alpha = 9/10$ - 1.8-257 sec.

- A higher value of $\alpha$ provides a slightly better solution.
Bilinear reduction technique is very effective because:
- they are continuous, i.e. no binary variables,
- a global solution of the problems is a solution of the initial MIP formulation, and
- there are fast algorithms converging to a local minimum.

The techniques can be used in heuristic algorithms.

A combination of the heuristic procedure with a cutting plain algorithm can provide an exact solution.

Because of its general structure the technique can be applied to other problems with a similar structure.
Questions?