Journal of the Korean Institute of Industrial Engineers
Vol. 27, No. 3, pp. 274-280, September 2001.

다수상품 흐름문제를 위한 내부점 방법†

임성묵[1†] · 설동렬[2] · 박순달[1]

[1]서울대학교 산업공학과 / [2]Entrue Consulting

Interior Point Methods for Multicommodity Flow Problems

Sungmook Lim[1] · Tongryeol Seol[2] · Soondal Park[1]

[1]Seoul National University, Seoul, 151-742
[2]Entrue Consulting, Seoul, 150-721

In this research, we develop a specialized primal-dual interior point solver for the multicommodity flow problems (MCFP). The Castro's approach that exploits the problem structure is investigated and several aspects that must be considered in the implementation are addressed. First, we show how preprocessing techniques for linear programming(LP) are adjusted for MCFP. Secondly, we develop a procedure that extracts a network structure from the general LP formulated MCFP. Finally, we consider how the special structure of the mutual capacity constraints is exploited. Results of comupational comparison between our solver and a general interior point solver are also included.

Keywords : multicommodity flow problem, interior point method, linear programming

1. 개 요

다수상품 흐름문제(Multicommodity Network Flow Problem)는 대형의 생산계획, 분배 · 수송계획, 사업계획 등의 수리모형화에서 자주 등장하는 형태의 문제이다. 이러한 다수상품 흐름문제는 대부분 많은 제약식과 많은 변수를 포함시키는 대형의 문제이며 일반적인 해법으로는 상당히 풀기 어려운 문제로 인식되어져 왔다(Ahuja, Magnanti and Orlin, 1993). 이 문제의 형태는 최소비용문제(Minimum cost network flow problem)에 네트워크상의 각 호에 흐르는 다수상품의 흐름량 총합에 상한을 부과하는 제약식으로 이루어져 있다. 또한 각 호에 부과되는 비용이 선형인 경우와 비선형인 경우에 따라 선형 다수상품 흐름문제와 비선형 다수상품문제로 나누어진다. 즉, 다수상품 흐름문제는 최소비용문제의 형태를 가지는 네트워크 제약식과 부가제약식으로 이루어져 있다. 네트워크 제약식부분은 현재까지 아주 효율적인 해법이 개발되어 있지만 부가제약식이 첨가됨으로써 다수상품 흐름문제는 어려운 문제가 된다. 따라서 다수상품 흐름문제 내에 내포되어 있는, 비교적 쉬운 네트워크 제약식 형태를 효과적으로 활용하여 전체 풀이 속도를 향상시키는 해법들이 많이 개발되어 있다.

다수상품 흐름문제를 푸는 대표적인 해법으로는 Primal Partitioning Simplex Method(Ahuja, Magnanti and Orlin, 1993)가 있다. 이 방법은 전체 문제의 기저행렬을 네트워크 제약식에 해당하는 부분과 나머지 부분으로 분할하여 기저의 크기를 줄이는 방법이다. 즉, 네트워크 제약식에 대응되는 기저행렬은 삼각기저형태를 이루므로 쉽게 역행렬 연산을 수행할 수 있다는 점을 이용한 것이다. 이 방법으로 구현된 대표적인 전산 프로그램으로 McBride(McBride, 1998)가 개발한 PPRN이 대표적이다. 최근 들어 많이 연구가 진행되고 있는 해법의 형태는 내부점 방법을 이용한 것이다. 내부점 방법은 그 특성상 문제의 크기에 상당히 둔감한 편인데 수십만 개의 제약식과 변수를 가지는 대형의 다수상품 흐름문제를 효과적으로 풀 수 있을 것으로 기대되고 있다. 최소비용문제와 같은 네트워크 문제를 위한 내부점 방법은 현재까지 많이 개발되어 있는데 개선방향을 구할 때 푸는 선형방정식을 선조절자 공액경사법

본 연구는 한국과학재단의 특정기초연구과제(과제번호 98-0200-07-01-2)의 지원을 받았음.
† 연락저자 : 임성묵, 151-742 서울시 관악구 신림9동 서울대학교 공과대학 산업공학과, Fax : 02-880-7183, e-mail : sungmook@orlab.snu.ac.kr
2000년 4월 접수, 2001년 6월 게재 확정.

을 사용하여 해결하는 방법이 주를 이룬다.

내부점 선형계획법에서는 가장 많은 계산이 소요되는 선형 방정식 해법의 효율성을 높이기 위한 연구가 많이 수행되고 있다. 특히, 행렬이 특수한 구조를 가지고 있는 경우에는 이러한 특성을 활용하여 선형방정식의 해법을 효율화하고자 하는 연구들이 수행되었다. Todd(Todd, 1988)는 Karmarkar의 사영법을 사용하는 경우에 행렬의 특수구조를 활용하는 방법에 대해서 연구한 바 있다. Hurd와 Murphy(Hurd and Murphy, 1992)는 활용 가능한 다양한 특수구조를 분석하여 장벽법에서도 Todd의 연구결과를 적용할 수 있음을 보이고 일반적인 방법보다 특수구조를 활용함으로써 수행도를 향상시킬 수 있음을 보였다. Choi와 Goldfarb(Choi and Goldfarb, 1993)는 네트워크 문제에서 나타나는 특수구조를 활용하기 위한 방법으로 행단위 행렬분할 방법을 적용하였다. 최근 Grigoriadis와 Khachiyan (Grigoriadis and Khachiyan, 1995)은 특수구조를 활용하여 다수상품 흐름문제를 위한 변형된 내부점 방법을 제안하였다.

네트워크 문제의 경우에는 네트워크의 특성을 활용하면서 단체법을 수행하는 네트워크 단체법이 잘 개발되어 있어서 일반적으로 대형의 문제에 대해서도 효과적으로 문제를 풀어낼 수 있다. 그러나 문제의 크기가 매우 큰 경우에 대해서는 네트워크 문제의 특성을 활용하도록 개발된 내부점 방법을 적용했을 때에 더 효율적임이 보고되어 있다(Mehrotra and Wang, 1995; Resende and Pardalos, 1995; Resende and Veiga, 1993; Todd, 1988). 최소비용 흐름문제 또는 다수상품 흐름문제에 대해서는 일반적으로 변수의 개수가 10만 개 이상인 경우에 대해서 내부점 방법이 효과적이었다(McBride, 1998).

본 논문에서는 일반적인 내부점 방법에서 연립선형방정식을 푸는 과정을 다수상품 흐름문제의 특성을 활용하도록 수정하여 해법의 효율성을 높이는 방법을 다룬다.

2. 다수상품 흐름문제와 내부점 방법

유방향 네트워크 $G=(N,E)$를 고려하자. 여기서 N은 $m+1$개의 점들의 집합이고, E는 n개의 호의 집합이다. K는 네트워크 G를 통해서 흐르게 될 k개의 상품의 집합이라고 하자. 이제 다수상품 흐름문제는 다음과 같은 선형계획법 모형으로 표현될 수 있다.

$$\begin{aligned} \min \quad & \sum_{i=1}^{k} \boldsymbol{c}^{(i)T}\boldsymbol{x}^{(i)} \\ \text{s.t.} \quad & A_N \boldsymbol{x}^{(i)} = \boldsymbol{b}^{(i)} \\ & \sum_{i=1}^{k} \boldsymbol{x}^{(i)} \le \boldsymbol{b}_{mc} \\ & 0 \le \boldsymbol{x}^{(i)} \le \boldsymbol{u}^{(i)} \end{aligned} \tag{1}$$

단, $\boldsymbol{x}^{(i)} \in R^n$, $\boldsymbol{c}^{(i)} \in R^n$은 각각 상품 $i=1,\cdots,k$에 대한 흐름 및 비용 벡터이다. $A_N \in R^{m\times n}$은 각 열이 호에 대응되고 행이 점에 대응되는 node-arc incidence 행렬이다. 시점과 종점에 대응되는 행에 대해서는 비영요소가 오직 하나 존재한다. 용량한계 제약식에 대해서 여유변수 $\boldsymbol{s}_{mc}$를 도입하면 다음과 같다.

$$\begin{aligned} \min \quad & \sum_{i=1}^{k} \boldsymbol{c}^{(i)T}\boldsymbol{x}^{(i)} \\ \text{s.t.} \quad & A_N \boldsymbol{x}^{(i)} = \boldsymbol{b}^{(i)} \\ & \sum_{i=1}^{k} \boldsymbol{x}^{(i)} + \boldsymbol{s}_{mc} = \boldsymbol{b}_{mc} \\ & 0 \le \boldsymbol{x}^{(i)} \le \boldsymbol{u}^{(i)} \\ & 0 \le \boldsymbol{s}_{mc} \le \boldsymbol{b}_{mc} \end{aligned} \tag{2}$$

다수상품 흐름문제의 계수행렬 A는 다음과 같은 구조로 되어 있다.

$$A = \begin{array}{|c|c|c|c|c|} \hline A_N & & & & \\ \hline & A_N & & & \\ \hline & & \ddots & & \\ \hline & & & A_N & \\ \hline I & I & \cdots & I & I \\ \hline \end{array}$$

단, $I \in R^{n\times n}$인 단위행렬이고 비어 있는 부분은 모두 0이다. 한편, Θ는 다음과 같다.

$$\Theta = \begin{array}{|c|c|c|c|} \hline \Theta^{(1)} & & & \\ \hline & \ddots & & \\ \hline & & \Theta^{(k)} & \\ \hline & & & \Theta_{mc} \\ \hline \end{array}$$

단, $\Theta^{(i)} \in R^{n\times n}$, $i=1,\cdots,k$이고 $\Theta_{mc} \in R^{n\times n}$이다. 각각 상품 i의 흐름 $\boldsymbol{x}^{(i)}$와 여유변수 $\boldsymbol{s}_{mc}$에 대응된다. 이때에 $A\Theta A^T$는 다음과 같이 표현된다.

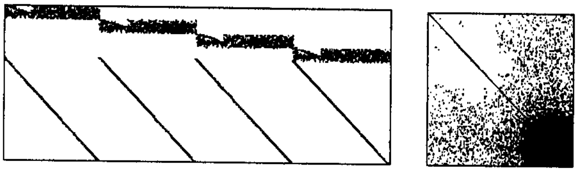
$$A\Theta A^T = \begin{array}{|c|c|c|c|} \hline A_N\Theta^{(1)}A_N^T & & & A_N\Theta^{(k)}A_N^T \\ \hline & \ddots & & \vdots \\ \hline & & A_N\Theta^{(k)}A_N^T & A_N\Theta^{(k)} \\ \hline \Theta^{(1)}A_N^T & \cdots & \Theta^{(k)}A_N^T & \Theta_{mc} + \sum_{i=1}^{k}\Theta^{(i)} \\ \hline \end{array}$$

$$= \begin{array}{|c|c|} \hline B & C \\ \hline C^T & D \\ \hline \end{array}$$

위와 같이 행렬을 분할하여 표현하면 내부점 방법에서 풀어야 하는 정규방정식 형태의 연립선형방정식 $A\Theta A^T \Delta y = \overline{b}$은 다음과 같이 된다.

$$\begin{pmatrix} B & C \\ C^T & D \end{pmatrix}\begin{pmatrix} \Delta \boldsymbol{y}_1 \\ \Delta \boldsymbol{y}_2 \end{pmatrix} = \begin{pmatrix} \bar{\boldsymbol{b}}_1 \\ \bar{\boldsymbol{b}}_2 \end{pmatrix} \quad (3)$$

식 (3)을 촐레스키 분해를 사용하여 한꺼번에 풀게 되면 다음 그림과 같이 D부분이 밀집행렬이 되어 계산속도를 느리게 만드는 원인이 된다. 그림에서 왼쪽 행렬은 A행렬의 비영요소 구조이고, 오른쪽 행렬은 촐레스키 요소의 비영요소 구조이다.



방정식 (3)을 푸는 데에는 $\Delta \boldsymbol{y}_1$을 먼저 소거하느냐 $\Delta \boldsymbol{y}_2$를 먼저 소거하느냐에 따라서 다음과 같은 두 가지 방법이 가능하다. 첫째, $\Delta \boldsymbol{y}_2$를 먼저 소거하게 되면 다음과 같은 식에 따라서 개선 방향을 구한다.

$$(B - CD^{-1}C^T)\Delta \boldsymbol{y}_1 = \bar{\boldsymbol{b}}_1 - CD^{-1}\bar{\boldsymbol{b}}_2 \quad (4)$$

$$\Delta \boldsymbol{y}_2 = D^{-1}\bar{\boldsymbol{b}}_2 - D^{-1}C^T\Delta \boldsymbol{y}_1 \quad (5)$$

둘째, $\Delta \boldsymbol{y}_1$를 먼저 소거하게 되면 다음과 같은 식에 따라서 개선 방향을 구하게 된다.

$$(D - C^TB^{-1}C)\Delta \boldsymbol{y}_2 = \bar{\boldsymbol{b}}_2 - C^TB^{-1}\bar{\boldsymbol{b}}_1 \quad (6)$$

$$\Delta \boldsymbol{y}_1 = B^{-1}\bar{\boldsymbol{b}}_1 - B^{-1}C\Delta \boldsymbol{y}_2 \quad (7)$$

여기서 $B - CD^{-1}C^T$는 $km \times km$ 행렬이고, $D - C^TB^{-1}C$는 $n \times n$ 행렬이다. 상품의 개수 k가 많은 경우, 즉 $km > n$인 경우에는 두 번째 식이 유리하다. 또한, 용량한계제약식이 모든 호에 대해서 주어지지 않고 일부 호에 대해서만 주어질 경우에 $D - C^TB^{-1}C$는 용량한계제약식 개수만큼의 크기로 줄어들게 된다. 이러한 이유로 여러 연구자들이 두 번째 경우에 대한 효율적인 계산법을 연구해 오고 있다.

식 (4)는 공액경사법에 의하여 해결하며, D^{-1}는 D가 대각행렬이기 때문에 간단히 해결할 수 있다. B는 $A_N\Theta^{(i)}A_N^T$ 블록들로 이루어진 행렬이기 때문에 최소비용문제에 내부점 방법을 적용할 때에 사용하는 선조절자를 활용할 수 있다. 최소비용문제는 $k=1$인 다수상품 흐름문제로 볼 수 있으므로, 최소비용 문제에서 사용하는 극대연결목(maximum spanning tree)이 B에서는 극대연결목들의 숲(forest)으로 나타난다. $\Theta^{(i)}$를 가중치로 사용하여 구한 극대연결목을 $A_N^{(i)}$라고 하면 이들이 형성하는 숲은 다음과 같다.

$$B_T = \begin{pmatrix} A_T^{(1)}\Theta^{(1)}A_T^{(1)\,T} & & \\ & \ddots & \\ & & A_T^{(k)}\Theta^{(k)}A_T^{(k)\,T} \end{pmatrix} \quad (8)$$

이제 식 (4)를 공액경사법으로 풀 때에 B_T를 선조절자로 사용할 수 있다. 한편, 식 (6)은 공액경사법에 의하여 해결하며, B^{-1}의 처리는 촐레스키 분해를 수행하여 처리한다. Castro (Castro, 1998)는 선조절자가 $D - C^TB^{-1}C$에 보다 더 가까운 값을 가지도록 하기 위하여 다음과 같은 기법을 제안하였다. 먼저 $S = D - C^TB^{-1}C$라고 하자. 스펙트럼 반경 $\rho(D^{-1}C^TB^{-1}C) < 1$임이 알려져 있기 때문에 S^{-1}는 다음을 만족한다.

$$S^{-1} = \left[\sum_{i=1}^{\infty}(D^{-1}C^TB^{-1}C)^i\right]D^{-1} \quad (9)$$

식 (9)에 의해서 $\phi \geq 0$에 대해서 선조절자 M^{-1}는 다음과 같은 방법으로 구할 수 있다.

$$M^{-1} = [I + (D^{-1}C^TB^{-1}C) + (D^{-1}C^TB^{-1}C)^2 + \cdots + (D^{-1}C^TB^{-1}C)^{\phi}]D^{-1} \quad (10)$$

Castro의 실험결과에 따르면 ϕ가 영인 경우가 가장 수행속도가 빠르며 1인 경우가 좋은 경우도 간혹 있었다고 한다.

3. 구현 방법

3.1 개선방향

다음과 같은 선형계획법 문제를 생각해 보자.

$$(\mathrm{P}) \quad \begin{array}{ll} \min & \boldsymbol{c}^T\boldsymbol{x} \\ \text{s.t.} & A\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} + \boldsymbol{s} = \boldsymbol{u} \\ & \boldsymbol{x} \geq 0, \ \boldsymbol{s} \geq 0 \end{array}$$

$$(\mathrm{D}) \quad \begin{array}{ll} \max & \boldsymbol{b}^T\boldsymbol{y} + \boldsymbol{u}^T\boldsymbol{w} \\ \text{s.t.} & A^T\boldsymbol{y} + \boldsymbol{w} - \boldsymbol{z} = \boldsymbol{c} \\ & \boldsymbol{w} \geq 0, \ \boldsymbol{z} \geq 0 \end{array}$$

일반적인 비가능 내부점 방법은 다음과 같은 perturbed-KKT 조건을 풀어서 개선방향을 구한다. 단, $f = \boldsymbol{u} - \boldsymbol{x}$라고 둔다.

$$\begin{aligned} \boldsymbol{b}_{xz} &= \mu \boldsymbol{e}_n - XZ\boldsymbol{e}_n = 0 \\ \boldsymbol{b}_{fw} &= \mu \boldsymbol{e}_n - FW\boldsymbol{e}_n = 0 \\ \boldsymbol{b}_b &= \boldsymbol{b} - A\boldsymbol{x} = 0 \\ \boldsymbol{b}_c &= \boldsymbol{c} - (A^T\boldsymbol{y} + \boldsymbol{z} - \boldsymbol{w}) = 0 \end{aligned}$$

위 비선형방정식을 풀기 위해 Newton 방법을 적용하면, 다음과 같이 개선방향을 구할 수 있다.

$$\begin{aligned} (A\Theta A^T)\ \boldsymbol{d}_y &= \boldsymbol{b}_b + A\Theta \boldsymbol{r} \\ \boldsymbol{d}_x &= \Theta(A^T\boldsymbol{d}_y - \boldsymbol{r}) \\ \boldsymbol{d}_w &= F^{-1}(\boldsymbol{b}_{fw} + W\boldsymbol{d}_x) \\ \boldsymbol{d}_z &= \boldsymbol{b}_c + \boldsymbol{d}_w - A^T\boldsymbol{d}_y \end{aligned}$$

$$r = F^{-1}b_{fw} + b_c - X^{-1}b_{xz}$$
$$\Theta = FX(ZF + XW)^{-1}$$

위 방정식을 μ값을 줄여가면서 계속 풀어나가게 되는데, 다음과 같이 μ값을 설정한다.

$$\mu^{k+1} = \sigma((x^k)^T z^k + (w^k)^T s^k),\ 0 < \sigma < 1$$

여기서, σ의 설정은 해법의 수렴에 큰 영향을 미치게 된다. Mehrotra의 예측자-수정자 방법을 사용할 때에는 주어진 방정식을 비교적 정확하게 풀어나가므로 매회의 해는 중심경로 가까이에 머무르게 된다. 따라서, σ를 비교적 작게 설정해도 해법의 수렴성에는 별 문제가 발생하지 않는다. 촐레스키 분해를 이용하면 예측자-수정자 방법처럼 서로 다른 우변상수에 대해 선형방정식을 두 번 푸는 방법이 적은 비용으로 해결될 수 있지만 본 연구에서 다루고 있는 공액경사법을 사용하게 되면 두 배의 비용이 소요되므로 예측자-수정자 방법이 가지는 반복횟수의 감소가 별다른 효과가 없고 오히려 계산시간이 늘어나게 된다. 따라서 본 연구에서는 선형방정식을 한 번만 푸는 방법을 채택한다. 그러나 공액경사법으로 선형방정식을 풀고 예측자-수정자 방법과 같은 방법을 사용하지 않게 됨으로써 매회의 해가 중심경로에서 비교적 많이 벗어나 있는 경우가 많이 발생할 수 있다. 그러므로 σ를 작게 설정하게 되면 해법이 수렴하지 못할 수 있기 때문에 본 연구에서는 σ의 값을 0.1로 설정하였다.

3.2 선조절자 공액경사법의 적용을 통한 선형방정식의 풀이

식 (6)과 (7)을 사용하는 경우에 대해서 효율적인 구현 방법에 대하여 고려하자. 먼저 B^{-1}를 효율적으로 계산하는 방법에 대해서 다루고, 두 번째로 선조절자 계산 방법에 대해서 토의하겠다.

B를 구성하는 각 블록 $A_N\Theta_{(i)}A_N^T$는 $\Theta_{(i)}$만 다르고 계수행렬 A_N이 동일하므로 같은 비영요소 구조를 가진다. 즉, $A_N\Theta_{(1)}A_N^T$, $A_N\Theta_{(2)}A_N^T$, $\cdots$, $A_N\Theta_{(k)}A_N^T$의 비영요소 구조는 $A_NA_N^T$의 비영요소 구조와 같다. 따라서, B를 촐레스키 분해하기 위한 순서화 및 상징적 분해 과정은 B 전체에 대해서 수행할 필요가 없다. $A_NA_N^T$에 대해서 순서화와 상징적 분해를 수행한 다음 각 상품에 따른 블록에 대해서 동일한 자료구조를 사용하도록 하면 된다.

식 (6)과 식 (7)에는 B^{-1}가 적용되는 계산이 많이 수행되어야 한다. $B = LL^T$로 분해한다면 식 (6)과 식 (7)은 다음과 같이 정리할 수 있다.

$$\{D-(L^{-1}C)^T(L^{-1}C)\}\Delta y_2 = \bar{b}_2 - (L^{-1}C)^T(L^{-1}\bar{b}_1) \quad (11)$$

$$\Delta y_1 = L^{-T}\{(L^{-1}\bar{b}_1) - (L^{-1}C)\Delta y_2\} \quad (12)$$

따라서, 실제로 전방치환연산은 $L^{-1}C$, $L^{-1}\bar{b}_1$을 푸는 데에 필요하며, 식 (12)를 풀 때에 후방치환연산이 필요하다. $L^{-1}C$는 C가 $km \times n$ 행렬이므로 각 열에 대해서 한 번씩 전방치환연산을 수행하는 셈이기 때문에 n번 전방치환연산을 하는 것으로 이해할 수 있다. 전방치환연산은 모두 $n+1$회, 후방치환연산은 1회 수행된다. 계산절차는 다음과 같다.

단계 1. (촐레스키 분해) $B = LL^T$ (13)

단계 2. (전방치환) $\Gamma = L^{-1}C$, $\beta = L^{-1}\bar{b}_1$ (14)

단계 3. (공액경사법) $(D - \Gamma^T\Gamma)\Delta y_1 = \bar{b}_2 - \Gamma^T\beta$ (15)

단계 4. (후방치환) $\Delta y_1 = L^{-T}\{\beta - \Gamma\Delta y_2\}$ (16)

그런데, $\Theta^{(i)}$가 $(Z^{(i)})^{-1}X^{(i)}$이므로 내부점 방법이 최적해에 가까이 수렴하게 되면 상보성에 의해서 $n-m$개의 변수들에 대해서는 0에 가까운 값을 가지게 된다. 즉, A_N에서 m개의 열들만 남기고 나머지 열들은 제거되는 효과를 가져온다. 다수상품 흐름문제에서 $k=1$인 특수한 경우로 이해할 수 있는 최소비용문제를 푸는 경우에 이러한 성질을 이용하여 극대연결목 선조절자 또는 불완전 촐레스키 분해 선조절자를 사용하는 기법이 개발된 바 있다(Seol, Cho and Park, 1998). 즉, 식 (8)에서 $A_N\Theta^{(i)}A_N^T \approx A_T\Theta^{(i)}A_T^T$가 된다. A_T가 나무인 경우에는 $O(m)$에 행과 열의 순서만을 바꾸는 것으로 A_T를 하삼각행렬로 만들 수 있다. 결과적으로 $A_T\Theta^{(i)}A_T^T$를 $O(m)$에 촐레스키 분해할 수 있게 된다. 이러한 성질을 활용하면 해법의 하반기에는 B^{-1}를 매우 빠른 시간에 계산할 수 있다.

이제 선조절자 계산 방법을 고려하자. 식 (10)에 의해서 선조절자 M^{-1}을 구하게 되면 어떻게 ϕ의 값을 결정할 것인가 하는 것이 문제가 된다. ϕ의 값이 커질수록 선조절자의 효과가 좋아져서 공액경사법의 반복수가 줄어들기는 하지만, 선조절자를 계산하는 데에 드는 시간이 매우 커져서 실질적으로 효과를 볼 수 없게 되기 때문이다. 결국 $\phi=0$으로 두게 되면 $M=D$를 사용하는 것과 마찬가지가 된다. $M=D$를 사용하는 방법 이외에 $M=D-diag(C^TB^{-1}C)$를 사용하는 방법을 고려할 수 있다.

공액경사법의 효과적인 활용을 위해서는 종료조건이 상당히 중요하다. 본 연구에서는 다음과 같은 종료조건을 채택하였다.

$$\frac{1 - b^T(b-r)}{\|b\|\,\|b-r\|} < \varepsilon,\ r = b - Ax$$

효과적인 ε의 값은 B에 대한 촐레스키분해 요소의 정확도, 개선방향, μ값의 설정에 따라 크게 달라지는데 본 연구에서는 10^{-4}을 사용하였다.

표 1. 사전처리의 효과

문제이름	사전처리 전			사전처리 후		
	행	열	비영요소	행	열	비영요소
PDS-02	2953	7535	16390	1393	5956	13277
PDS-06	9881	28655	62524	6833	25529	56331
PDS-10	16558	48763	106436	12536	44617	98312
PDS-20	33874	105728	230200	27186	98880	216831

3.3 사전처리

사전처리란 주어진 문제 내에 내포된 중복성을 파악하고 제거하여 문제의 크기를 줄이는 기법이다. 선형계획법에 대한 사전처리 기법은 현재까지 상당히 많은 연구가 있었다. 공백행, 공백열, 단일요소행, 단일요소열, 암시적 자유변수, 강박제약식, 치환 등에 의한 기법들이 개발되었는데, 일반적인 선형계획법 문제의 경우 20~40% 정도의 문제 크기 감소효과가 있어 그 효과는 상당히 크다고 할 수 있다. 다수상품 흐름문제의 경우도 선형계획법 사전처리를 적용하게 되면 25%에서 50%까지의 제약식, 변수 제거효과가 있다. 이는 사전처리에 대한 효과를 나타낸 다음 <표 1>을 보면 알 수 있다(Seong, Lim and Park, 1999). 실험대상으로 취한 문제들은 Netlib 문제(Gay, 1985)에 포함되어 있는 대표적인 다수상품문제인 PDS 문제들이다.

네트워크 제약식부분에만 영향이 미치는 사전처리를 적용하고 나면 네트워크 형태가 손상되지 않는다. 이는 다음 설명을 통해 알 수 있다.

네트워크 문제를 선형계획법으로 표현할 때에 행렬계수의 각 요소가 $-1,\ 0,\ 1$의 값만을 가지는 경우, 즉 $(A_N)_{ij} \in \{-1, 0, 1\}\ \forall i, j$ 인 경우에 다음의 성질을 만족한다.

성질 1.

선형계획 프로그램 LPABO와 LPAKO에서 사용하고 있는 사전처리(Seong, Lim and Park, 1999; Lim, Seong and Park, 1999)를 네트워크 문제에 적용했을 경우에 네트워크 제약식은 네트워크의 성질을 그대로 유지한다.

(증명)

가. 단순한 행 및 열의 제거(공백행 및 공백열, 단일요소행)
 마디 또는 호의 삭제이므로 네트워크의 성질이 유지된다.

나. 변수의 한계 강화
 호의 유통상한이 변화하는 것이므로 네트워크의 성질은 유지된다.

다. 치환에 의한 중복제약식 및 암시적 자유변수의 제거
 마디의 병합 또는 삭제가 일어나는 것이기 때문에 네트워크의 성질이 유지된다.

한편, 부가제약식부분에 비영요소의 도입이 발생하게 되는 사전처리를 수행하게 되면 D부분이 대각행렬형태가 아니므로 선조절자로서의 편의성이 떨어지게 되고, 공액경사법의 반복횟수가 증가하게 된다. 따라서, 본 연구에서는 부가제약식부분에 비영요소의 도입이 발생되는 치환연산은 수행하지 않도록 하였다.

3.4 네트워크 제약식의 추출

선형계획법의 형태로 입력이 되는 다수상품 흐름문제의 경우, 네트워크 제약식부분과 부가제약식부분을 자동적으로 분리할 수 있는 장치가 필요하다. CPLEX의 경우에도 주어진 선형계획법 문제를 Netopt를 사용하여 풀 때에는 네트워크 제약식을 추출하는 과정을 거친게 된다.

본 연구에서 개발한 네트워크 제약식 추출 방법은 다음과 같다.

단계 1. 초기해
 주어진 제약식 순서에서 전방 탐색을 통해 최대 네트워크 행을 구한다.

단계 2. Addition
 현재까지의 네트워크 제약식에 네트워크 특성을 그대로 유지해 주는 나머지 제약식 중 하나를 포함시킨다.

단계 3. Exchange
 네트워크 제약식의 특성을 유지하면서 현재까지의 네트워크 제약식들 중 하나와 나머지 제약식 중 하나를 교환한다.

단계 4. 주어진 반복횟수를 넘지 않았으면 단계 2로 간다.

3.5 부가제약식 형태의 활용

대부분의 다수상품 흐름문제에서 부가제약식부분이 k개의 행렬 I로 이루어지지는 않고 그 보다 작은 수의 행렬 I로 이루어 진다. 즉, 각 호에 부과되는 유통량 합의 상한이 주어지지 않는 호가 존재한다는 것이다. 이러한 구조를 활용하면 계산량을 보다 줄일 수 있다.

다음과 같이 호의 유통상한이 존재하는 경우와 존재하지 않는 경우를 기준으로 행렬 A_N을 분할하자.

$$A_N = [A_N^C \mid A_N^U]$$

단, $A_N^C \in R^{m \times p}$, $A_N^U \in R^{m \times (n-p)}$이다. 즉, 행렬 A는 다음과 같이 된다.

$$A = \begin{array}{|c|c|c|c|c|c|c|}
\hline
A_N^U & & & A_N^C & & & \\
\hline
 & \ddots & & & \ddots & & \\
\hline
 & & A_N^U & & & A_N^C & \\
\hline
 & & & I & \cdots & I & I \\
\hline
\end{array}$$

행렬 A_N을 위와 같이 분할하였을 경우에 $A\Theta A^T$는 다음과 같이 된다.

$$A\Theta A^T=\left[\begin{array}{c|c|c|c} A_N^C\Theta_{(1)}^C(A_N^C)^T + A_N^U\Theta_{(1)}^U(A_N^U)^T & & & A_N^C\Theta_{(1)}^C \\ \hline & \ddots & & \vdots \\ \hline & & A_N^C\Theta_{(k)}^C(A_N^C)^T + A_N^U\Theta_{(k)}^U(A_N^U)^T & A_N^C\Theta_{(k)}^C \\ \hline \Theta_{(1)}^C(A_N^C)^T & \cdots & \Theta_{(k)}^C(A_N^C)^T & \Theta_{mc}+\sum_{i=1}^{k}\Theta_{(i)}^C \end{array}\right]$$

$$=\left[\begin{array}{c|c|c|c} A_N\Theta_{(1)}A_N^T & & & A_N^C\Theta_{(1)}^C \\ \hline & \ddots & & \vdots \\ \hline & & A_N\Theta_{(k)}A_N^T & A_N^C\Theta_{(k)}^C \\ \hline \Theta_{(1)}^C(A_N^C)^T & \cdots & \Theta_{(k)}^C(A_N^C)^T & \Theta_{mc}+\sum_{i=1}^{k}\Theta_{(i)}^C \end{array}\right]$$

$$=\left[\begin{array}{c|c} B & C \\ \hline C^T & D \end{array}\right]$$

위의 행렬형태와 2절에서 설명한 형렬형태를 서로 비교해 보면, C부분에 대한 계산량이 줄어듦을 알 수 있다. 즉, 공액경사법을 적용할 때, $A\Theta A^T$와 어떤 벡터와의 곱셈연산에 대한 계산량이 줄어들게 되어 공액경사법의 계산시간이 감소한다.

4. 실험결과

본 연구에서 실험대상으로 삼은 문제는 Netlib(Gay, 1985) 문제에 포함되어 있는, 대표적인 다수상품문제인 PDS 계열 문제이다. PDS 시리즈는 11개의 상품을 가지는 경우의 문제들이다. 한편, 내부점 방법으로 선형계획법 문제를 푸는 프로그램으로는 LPABO 5.5f(Park, 1999)를 선택하였는데, 이 프로그램은 비가능 원쌍대 내부점 방법을 사용하는 해법 프로그램이다. 그리고 MC는 식 (6)과 (7)을 사용하여 다중상품 흐름문제의 특성을 활용한 해법 프로그램으로 본 연구를 통해 개발되었다. 선조절자로는 $M=D$를 사용하였으며, 공액경사법의 효과를 보다 정확하게 파악하기 위해서 사전처리는 적용하지 않았다. <표 2>는 LPABO 5.5f와 MC를 이용하여 각각 PDS 문제를 풀었을 때의 CPU 시간 및 해법의 반복횟수를 정리한 것이다.

일반적인 내부점 방법을 적용하는 때보다 계산속도가 향상되었으며, 문제의 크기가 커질수록 효과적임을 알 수 있다. 많은 추가요소로 인해서 D부분은 밀집행렬에 가깝게 되기 때문에 직접 촐레스키 분해를 수행하는 LPABO 5.5f에 비해 행렬을 분할하여 풀어낸 MC 방법이 효과적이었다. 반복횟수를 살펴보면 LPABO 5.5f가 MC에 비해 2~3배 우월한데 이는 예측자-수정자 방법을 MC에는 구현할 수 없었고, 공액경사법의 정밀도가 촐레스키분해에 비해 많이 떨어진다는 점, μ값을 크게 줄여갈 수 없었다는 점 때문이다.

5. 결 론

본 연구에서는 대형의 다수상품 흐름문제를 풀기 위한 내부점 방법의 개발에 필요한 여러 가지 사항들을 설명하였다. 다수상품 흐름문제의 제약식형태에 의해 $A\Theta A^T$의 촐레스키분해 요소의 밀집도가 너무 높기 때문에, 밀집도가 낮은 부분과 밀집도가 높은 부분을 분리하여 처리하는 방법이 효과적이다. 밀집도가 낮은 부분을 처리할 때에는 촐레스키분해를 사용하고 밀집도가 높은 부분은 효과적인 선조절자를 사용한 공액경사법을 활용한다. Schur complement 행렬의 역행렬의 무한급수 표현을 활용한 선조절자의 개발은 Castro에 의해 수행되었는데 본 연구에서는 그 효과를 검증하였다. 또한, 공액경사법을 적용할 때 고려될 수 있는 여러가지 효율화 방안을 제시하였다. 즉, μ값을 줄여가는 방법, 부가제약식의 형태에 대한 활용, 효과적인 사전처리의 적용방법과 더불어 사용자 편의성을 고려한 네트워크 제약식의 추출방법을 제시하였다.

표 2. LPABO 5.5f와 MC의 수행도 비교

문제	문제 크기				CPU 시간 (초)		반복횟수	
	마디	호	변수	제약식	LPABO 5.5f	MC	LPABO 5.5f	MC
pds-06	835	2,827	33,924	12,012	133.35	63.81	30	67
pds-10	1,399	4,792	57,504	20,181	825.21	200.12	34	82
pds-0	2,857	10,858	130,296	42,285	-	1,279.12	-	154

(실험환경 : ALPHA, Linux, - : 풀지 못함)

본 연구의 실험결과에 의하면 밀집도에 따른 행렬의 분리와 선조절자 공액경사법의 적용이 일반적인 내부점 방법을 적용하였을 경우에 비해 수행속도가 40~50 % 정도 향상됨을 알 수 있다. 그러나 반복횟수는 2~3배 정도 증가하였는데 이는 예측자-수정자방법을 적용하지 않았다는 점과 촐레스키분해에 비해 공액경사법의 정밀도가 많이 떨어진다는 점을 그 이유로 들 수 있다. 또한, 정밀도의 문제에 의해 μ값을 줄여나가는 폭을 크게 할 수 없었다는 점도 있다.

공액경사법의 특성상 최적해에 매우 근접한 해를 구해낼 수는 없었는데, crossover기법을 통해 이를 극복할 수 있다. 따라서, 다수상품문제를 위한 효과적인 crossover기법의 개발이 연구과제가 될 수 있다. 또한, 본 연구에서 설명된 단순한 선조절자이외에 다양한 선조절자의 개발이 필요한다.

참고문헌

Ahuja, R. K., Magnanti, T. L. and Orlin, J. B. (1993), *Network Flows*, Prentice Hall, Englewood Cliffs, New Jersey.

Castro, J. (1998), A Specialized Interior Point Algorithm for Multicommodity Network Flows, Technical Report.

Choi, I. C. and Goldfarb, D. (1993), Exploiting special structure in a primal-dual path-following algorithm, *Mathematical Programming*, **58**, 33-52.

Duff. I., Erisman, A. and Reid, J. (1986), *Direct Methods for Sparse Matrices*, Oxford University Press, New York.

Gay, D. M. (1985), Electronic mail distributioin of linear programming test problems, *Mathematical Programming Society Committee on Algorithms Newsletter*, **13**, 10-12.

Golub, G. H. and Van Loan, C. F. (1989), *Matrix Computations*, 2nd ed., The Johns Hopkins University Press, Baltimore.

Grigoriadis, M. D. and Khachiyan, L. G. (1995), An exponential-function reduction method for block-angular convex programs, *Networks*, **26**, 59-68.

Hurd, J. K. and Murphy, F. H. (1992), Exploiting Special Structure in Primal Dual Interior Point Methods, *ORSA Journal on Computing*, **4**(1), 38-44.

Kelly, C. T. (1995), *Iterative Methods for Linear and Nonlinear Equations*, Society for Industrial and Applied Mathematics.

Lim, S. M., Seong, M. K. and Park, S. D. (1999), An Implementation of Preprocessing for the Simplex Method, *Journal of the Korean Institute of Industrial Engineers*, **25**(2), 217-225.

Lustig, I. J., Marsten, R. E. and Shanno, D. F. (1994), Interior point methods for linear programming: computational state of the art, *ORSA Journal on Computing*, **6**(1), 1-15.

McBride, R. D. (1998), Progress Made in Solving the Multicommodity Flow Problem, *SIAM Journal on Optimization*, **8**(4), 947-955.

Mehrotra, S. and Wang, J. S. (1995), Conjugate Gradient Based Implementation of Interior Point Methods for Network Flow Problems, Technical Report, Dept of Industrial Eng. and Management science, Northwestern Univ. Evanston, IL.

Park, S. D. (1999), LPABO ver 5.5f User's Manual, URL *http://orlab.snu.ac.kr/Software.html#LPABO*.

Resende, M. and Pardalos, P. M. (1995), Interior point algorithms for network flow problems, *Advances in linear and integer programming*, J. E. Beasley, ed., Oxford University Press.

Resende, M. and Veiga, G. (1993), An Efficient Implementation of a Network Interior Point Method, in Network Flows and Matching: First DIMACS Implementation Challenge, D. S. Johnson and C. C. McGeoch, eds., *DIMACS Series in Discrete Mathematics and Theoretical Computer Science* 12, American Mathematical Society, 299-348.

Seol, T. R., Cho, E. Y. and Park, S. D. (1998), Interior Point Methods for Network Problems(An Efficient Conjugate Gradient Method for Interior Point Methods), *Journal of the Military Operations Research Society of Korea*, **24**(1), 146-156.

Seong, M. K., Lim, S. M. and Park, S. D. (1999), An Implementation of Preprocessing for Interior Point Methods for Linear Programming, *Journal of Korean Operations Research and Management Science Society*, **24**(1), 1-12.

Todd, M. J. (1988), Exploiting Special Structure in Karmarkar's Linear Programming Algorithm, *Mathematical Programming*, **41**, 97-113.