

## 내부점 방법에서 Augmented System의 출레스키 분해\*

도승용\*\* · 성명기\*\*\* · 박순달\*\*

### Cholesky Factorization of the Augmented System in Interior Point Methods for Linear Programming\*

Seungyong Doh\*\* · Myeongki Seong\*\*\* · Soondal Park\*\*

#### ■ Abstract ■

In the normal equations approach in which the ordering and factorization phases are separated, the factorization in the augmented system approach is computed dynamically. This means that in the augmented system the numerical factorization should be performed to obtain the non-zero structure of Cholesky factor  $L$ . This causes much time to set up the non-zero structure of Cholesky factor  $L$ . So, we present a method which can separate the ordering and numerical factorization in the augmented system. Experimental results show that the proposed method reduces the time for obtaining the non-zero structure of Cholesky factor  $L$ .

Keyword : Cholesky Factorization, Augmented System, Ordering

## 1. 개 요

선형계획법을 푸는 대표적인 해법으로는 단체법 (simplex method)과 내부점 방법(interior point method)이 있다. 내부점 방법은 Karmarkar가 사영

법(projection method)을 제안한 이후로 계속 발전 하여 사용하기에 편리한 아핀법(affine method)이나 장벽법(barrier method)이 개발되었으며 현재에 이르러서는 단체법에 견줄만한 선형계획법 해법으로 자리 잡았다. 특히, 이론적으로 다항복잡도(pol-

논문접수일 : 2002년 10월 30일 논문게재확정일 : 2003년 2월 27일

\* 본 연구는 한국과학재단 특정기초연구사업(과제번호 R01-2002-000-00168-0)의 지원을 받음.

\*\* 서울대학교 산업공학과

\*\*\* LG-CNS 컨설팅 부문 공공그룹

ynomial complexity)를 가지고 있으며 실제로도 대형문제에 대해서는 단체법에 비해 빠른 계산 속도를 보여 준다[1].

내부점 방법에서는 매 회마다 개선방향을 구하기 위해 연립선형방정식을 풀어야 한다. 대형 선형 계획법 문제를 내부점 방법으로 푸는 경우, 수행시간의 약 70%이상의 시간이 이 연립선형방정식을 푸는 데에 소요된다[4]. 그러므로 연립선형방정식을 얼마나 빨리 정확하게 푸느냐가 내부점 방법의 실질적인 효율성을 크게 좌우한다. 내부점 방법을 구현한 프로그램들은 연립선형방정식 풀기 위해 출레스키 분해(Cholesky factorization)의 변형인 대칭 분해  $LDL^T$ 를 사용한다[8]. 여기서  $D$ 는 대각행렬을 나타낸다.

내부점 방법에서 선형연립방정식을 풀기 위해 두 가지 접근법이 널리 사용되어지고 있다. 하나는 정규방정식(normal equation)을 이용하는 방법이고, 다른 하나는 Augmented System을 이용하는 방법이다[11].

정규방정식 방법에서는 선형연립방정식을 구성하는 행렬이 대칭양정치(symmetric positive definite)의 특성을 가지기 때문에 출레스키 분해  $LL^T$ 를 사용한다. Augmented System에서 나타나는 행렬은 대칭양정치 행렬이 아니므로 출레스키 분해를 직접 적용할 수 없고, 출레스키 분해의 변형인 대칭분해  $LDL^T$ 를 사용한다. 일반적으로 내부점 방법을 구현한 프로그램들은 수치적인 문제로 인해 출레스키 분해보다는 대칭 분해인  $LDL^T$ 를 사용한다. 따라서 본 논문에서는 출레스키 분해는  $LDL^T$ 를 의미하는 것으로 하겠다.

출레스키 분해시에 소요되는 연산은 출레스키 인자(Cholesky factor)  $L$ 의 비영요소(non-zero element) 수에 의해 결정되어진다[2, 3, 7, 9]. 즉 출레스키 분해의 성능은  $L$ 을 얼마나 희소하게 유지하는가에 달려있다.  $L$ 의 비영요소 수는 출레스키 분해시의 선회순서(pivot order)에 의해 영향을 받는다.

선회순서의 질이 좋으면  $L$ 의 비영요소의 수를 감소시킬 수 있지만, 그렇지 않으면  $L$ 의 비영요소의 수는 증가한다. 따라서 출레스키 분해는  $L$ 의 희소성을 유지하기 위해 선회순서를 결정하는 순서화(ordering)의 역할이 중요하다.

정규방정식 방법에서는 선형연립 시스템을 구성하는 행렬이 대칭양정치의 성질을 가진다. 이는 수치적 안정성에 대한 고려 없이 선회순서를 결정할 수 있다는 것을 의미한다. 따라서 순서화를 수행하는 단계와 수치적 분해(numerical factorization)를 수행하는 단계가 분리되어질 수 있다[9]. 이로 인해 빠른 시간 안에 선회순서를 결정 한 다음 상징적 분해(symbolic factorization)를 통해 출레스키 분해요소  $L$ 의 구조를 구성할 수 있는 장점이 있다. 정규방정식 방법에서는 선회순서를 결정하기 위한 방법으로 최소차수 순서화(minimum degree ordering)[7]나 최소 부족수 순서화(minimum deficiency ordering)[9]를 사용한다. 대부분의 내부점 프로그램에서는 수행속도가 빠른 최소차수 순서화를 이용한다[10].

Augmented System 방법에서는 선형시스템을 구성하는 행렬이 대칭양정치 행렬이 아니다. 따라서 순서화 단계와 수치적 분해단계가 분리되어질 수 없다. 즉 출레스키 인자  $L$ 의 비영요소의 구조를 결정하기 위해 수치적 분해를 수행하면서 선회순서를 동시에 결정해야 한다. 이는 정규방정식에서와 같이 수행속도가 빠른 최소차수 순서화 방법을 Augmented System 방법에서는 사용할 수 없다. 이로 인해 대형의 선형계획법 문제를 Augmented System을 풀때 출레스키 인자  $L$ 을 구성하는데에 많은 시간이 소요된다. 이는 수치적 안정성과 밀집열 처리에 우수한 성능을 가진 Augmented System 방법의 성능을 저하시키는 결과를 가져온다.

Augmented System 접근법에서는 선형연립 시스템을 구성하는 행렬이 부정치(Indefinite)라는 성질을 가진다. 하지만 이 선형시스템은 항상 대각요

소를 선회요소(pivot element)로서 결정할 수가 있다[8]. 이러한 성질을 이용하면 Augmented System에서도 최소차수 순서화 방법과 같이 순서화 단계와 수치적 분해단계를 분리할 수 있다. 본 연구에서는 Augmented System 접근법에서 선회순서와 홀레스키 인자  $L$ 을 빠른 시간 안에 구성할 수 있는 수정된 삭제그래프를 이용한 최소차수 순서화 방법을 제시한다. 이 방법은 삭제그래프 상에서 가상의 노드를 도입하여 최소차수 순서화를 수행함으로써 선회순서를 결정하는 순서화 단계와 수치적 분해단계를 분리할 수 있다. 그리고 수치적 안정성에 대한 고려 없이 선회순서를 결정하기 때문에 상징적 분해를 통해 홀레스키 인자  $L$ 을 구성할 수 있다. 따라서 빠른 시간 안에 홀레스키 인자  $L$ 을 구성할 수 있는 장점이 있는 반면 수치적으로 불안정해 질 수 있다. 하지만 이는 홀레스키 분해과정시 해당열을 0으로 두거나 선회순서를 재조정함으로써 해결할 수 있다. 본 논문의 구성은 다음과 같다.

2장에서는 내부점 방법에서 선형계획법 문제를 Augmented System 접근법으로 푸는 방법을 설명하고, 3장에서는 수치적 분해를 통해 홀레스키 인자  $L$ 을 구성하는 방법을 기술한다. 그리고 4장에서는 가상의 노드를 첨가한 수정된 삭제 그래프를 이용한 최소차수 순서화 방법을 제시한다. 마지막으로 5장과 6장은 본 연구의 실험결과와 결론이다.

## 2. Augmented System을 이용한 내부점 방법

다음의 선형계획문제를 고려하자.

$$(P) \quad \begin{aligned} \max \quad & c^T x \\ \text{s.t.} \quad & Ax = b \\ & x + s = u \\ & x \geq 0, \quad s \geq 0 \end{aligned}$$

여기서  $c, x, s, u \in R^n$ 이고,  $A \in R^{m \times n}$

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

여기서  $y \in R^m$ 이고,  $z, w \in R^n$ 이다.

문제 (P), (D)에 장벽함수(barrier function)를 넣어서 라그랑지 함수(lagrange function)의 쿤-터커(Kuhn Tucker)조건을 구한다. 이 조건을 이용한 해의 개선방향은 다음 식 (1)을 풀어서 얻을 수 있다.

$$\begin{bmatrix} A & 0 & 0 & 0 & 0 \\ I & I & 0 & 0 & 0 \\ 0 & 0 & A^T & I & -I \\ Z & 0 & 0 & 0 & X \\ 0 & W & 0 & S & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta s \\ \Delta y \\ \Delta w \\ \Delta z \end{bmatrix} = \begin{bmatrix} \zeta_b \\ \zeta_u \\ \zeta_c \\ \mu e - XZe \\ \mu e - SWe \end{bmatrix} \quad (1)$$

$$\begin{aligned} \zeta_b &= b - Ax \\ \zeta_u &= u - x - s \\ \zeta_c &= c - A^T y - w + z \end{aligned}$$

단,  $\mu$ 는 장벽매개변수

위 식의  $\zeta_b, \zeta_u, \zeta_c$ 는 원제약식과 쌍대제약식이 만족되지 않는 정도를 나타내고, 해법을 수행하는 중간에 원·쌍대가능성은 만족하게 된다.

식 (1)의 해  $\Delta z, \Delta s, \Delta w$ 는 다음과 같다.

$$\begin{aligned} \Delta z &= X^{-1}(\mu e - XZe - Z\Delta x) \\ \Delta s &= \zeta_u - \Delta x \\ \Delta w &= S^{-1}(\mu e - SWe + W\Delta s) \\ &= S^{-1}(\mu e - SWe - W\zeta_u + W\Delta x) \end{aligned} \quad (2)$$

여기서  $e$ 는 모든 요소가 1인 벡터이다.

식 (2)를 식 (1)에 대입하여  $\Delta z, \Delta s, \Delta w$ 를 소거하면

$$\begin{bmatrix} -D^{-2} & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} r \\ h \end{bmatrix} \quad (3)$$

$$\begin{aligned}
 D^2 &= (X^{-1}Z + S^{-1}W)^{-1} \\
 \text{단, } \mathbf{r} &= \boldsymbol{\zeta}_c - X^{-1}(\mu \mathbf{e} - XZ\mathbf{e}) + \\
 &\quad S^{-1}(\mu \mathbf{e} - S\mathbf{W}\mathbf{e}) - S^{-1}W\boldsymbol{\zeta}_u \\
 \mathbf{h} &= \boldsymbol{\zeta}_b
 \end{aligned} \quad (4)$$

식 (3)을 Augmented System이라 하고, 식 (3)에서  $\Delta \mathbf{x}$ ,  $\Delta \mathbf{y}$ 를 구한 다음 (2)를 이용하여  $\Delta \mathbf{z}$ ,  $\Delta \mathbf{s}$ ,  $\Delta \mathbf{w}$ 를 구한다.

내부점 방법에서 가장 어려운 부분이 식 (3)을 푸는 것이다. 식 (3)을 푸는 방법으로 반복적인 방법(Iterative Method)과 식 (3)의 행렬에 수치적 분해를 적용하여 직접 푸는 방법이 있다. 대부분의 내부점 프로그램은 식 (5)와 같이 행렬  $M$ 에 대칭 분해를 이용한다.

$$M = \begin{bmatrix} -D^{-2} & A^T \\ A & 0 \end{bmatrix} = L \wedge L^T \quad (5)$$

여기서  $\wedge$ 은 대각행렬을 의미한다.

행렬  $M$ 의 선회순서에 따라 두 가지 방법으로 나누어진다. 첫 번째가 정규방정식 접근법이다. (3)에서  $\Delta \mathbf{x}$ 를 소거하면  $AD^2A^T\Delta \mathbf{y} = AD^2\mathbf{r} + \mathbf{h}$ 만 남는다.  $AD^2A^T$ 가 대칭양정치행렬이므로 최소차수 순서화와 상징적 분해를 수행하여 출레스키 인자  $L$ 의 구조를 구성한 다음, 출레스키 분해를 수행할 수 있다. 그러나 정규방정식 접근법은  $A$ 가 밀집열을 포함하면 출레스키 인자  $L$ 에 비영요소의 수가 매우 많이 증가한다. 이러한 단점을 보완하기 위해 밀집열 분할 방법, 쌍대문제 전환방법, Schur 상보법 등 여러 가지 방법이 적용되었다[4]. 하지만 만족할 만한 성과를 거두지 못하고 있다.

정규방정식 접근법의 대안으로 Augmented System 접근법을 사용할 수 있다. 이 방법은 초기의 선회연산을  $-D^{-2}$  부분에서 고려하지 않고, (5)의 행렬  $M$ 을 직접  $L \wedge L^T$ 로 분해한다. Augmented System 접근법은 선회순서의 선택이 정규방정식 방법에 비해 자유롭고, 수치적으로 안정적이라 관

점으로 내부점 프로그램에 이용되었다. 특히 밀집열 문제가 있는 문제에 대해서도 수치적 안정성을 가지고 풀 수 있다는 장점이 있다. 하지만 이 접근법은 출레스키 인자  $L$ 을 구성하기 위해 직접 수치적 분해를 수행하여야 한다. 이로 인해 대형의 선형계획법 문제에 있어 출레스키 인자  $L$ 을 구성하는데 많은 소요시간이 소요된다는 단점이 있다.

### 3. 수치적 분해를 통한 출레스키 인자 $L$ 의 구성

이 장에서는 수치적 분해를 통해 출레스키 인자  $L$ 을 구성하는 방법[8]을 기술한다. Augmented System 접근법은 매회 개선방향을 구하기 위해 대칭 부정치(symmetric indefinite)행렬을 분해해야 한다. 선회순서를 결정하는 순서화 단계와 수치적 분해를 분리해서 수행하는 정규방정식 접근법과 달리, Augmented System 접근법에서는 하삼각 행렬  $L$ 의 회소성과 수치적 안정성을 동시에 고려하여 선회순서를 결정한다. 수치적 안정성을 고려하면서 선회요소를 선택하므로 정규방정식 접근법에 비해서 수치적 안정성이 높고 선회순서의 선택도 자유롭다.

Augmented System 접근법에서는 행렬  $M$ 을 대칭분해 한다.  $A$ 가 완전계수행렬인 경우,  $M$ 의 차수는  $m+n$ 이므로  $m+n-1$  회의 소거연산이 필요하다.

$I^s$ :  $s$  단계까지 선회연산이 이루어지지 않은 행·열의 지수

$$M^s = \begin{bmatrix} D^s & (A^s)^T \\ A^s & P^s \end{bmatrix} : s \text{ 단계까지 분해되지 않}$$

은  $M$ 의 부분행렬

위의 기호를 사용하면  $M^0 = M$ ,  $A^0 = A$ ,  $D^0 = -D^{-2}$ ,  $P^0$ 은 영행렬이다.

그리고 식 (4)에 의해  $D_{ii}^0 < 0$ ,  $i = 1, \dots, n$ 이다.

매 선회연산에서  $M^s$ 가 대칭행렬을 유지하도록 대각요소 중에서 선회요소를 선택한다.  $s$  단계의 소거연산에서 선회요소로  $p$  ( $p \in I^s, m_{pp}^s \neq 0$ )를 선택했다면,  $I^s = I^s - \{p\}$ 로 수정하고  $M^{s+1}$ 의 요소를 다음과 같이 수정한다.

$$m_{ij}^{s+1} = \begin{cases} m_{ij}^s & \text{if } m_{pp}^s = 0 \\ m_{ij}^s - \frac{m_{pi}^s m_{pj}^s}{m_{pp}^s} & \text{if } m_{pp}^s \neq 0 \end{cases} \quad (6)$$

for  $(i, j) \in I^{s+1} \times I^{s+1}$

선회지수의 집합  $\{p\}^m$ 은 선회순서를 나타내고, 분해요소는 다음과 같이 구할 수 있다.

$$\wedge_{ii} = m_{p_i, p_i}^{p_i}, \quad \text{for } i = 1, \dots, m+n,$$

$$L_{ij} = \frac{m_{p_i, p_j}^{p_i}}{m_{p_i, p_i}^{p_i}}, \quad \text{for } i, j = 1, \dots, m+n,$$

$p_i \leq p_j$

$M$ 이 부정치행렬(indefinite matrix)이므로, 위의 과정이 실패할 수 있다. 예를 들면, 다음의 대칭 부정치행렬은 (2, 3, 1)순서로  $1 \times 1$  선회연산으로 분해할 수 있다.

$$\begin{bmatrix} 1 & 1 & 2 \\ 1 & 1 & 0 \\ 2 & 0 & 4 \end{bmatrix}$$

그러나 (1, 1)요소를 처음 선회요소로 선택하면 분해할 수 없게 된다. 다음의 [정리 1]은  $M$ 에 대해  $1 \times 1$  선회연산이 항상 가능함을 보여준다.

[정리 1][8]:  $s$ 번째 소거연산에서,  $D^s$ 는 음정치(negative definite)이고  $P^s$ 는 대칭양정치를 유지한다. 여기서  $0 \leq s < m+n$ .

$A$ 가 완전계수(full rank)이면 [정리 1]로부터 행렬  $M$ 은  $L \wedge L^T$ 로 분해되어질 수 있다. 여기서

$\wedge$ 은 대각행렬을 의미한다.

[중정리 1][8]:  $P_{ii}^s = 0$ 인것은  $s$ 번째 선회까지 선회연산과 관련된 행·열로 이뤄진  $A$ 의 부분행렬과  $A$ 의  $i$ 번째 행이 완전행계수가 아닐 경우이다.

Augmented System에서는 처음으로 출레스키 인자  $L$ 의 구조를 구성하기 위해  $D=I$ 로 두고 수치적 분해를 수행한다. 이 결과로 선회순서와 출레스키 인자  $L$ 의 비영요소 구조가 결정되어진다. 결정된 선회순서와 출레스키 인자  $L$ 의 구조는 수치적으로 불안정하지 않는 이상 다음 회에서도 계속 사용되어진다.

#### 4. 수정된 삭제 그래프를 이용한 순서화 방법

3장에서는 Augmented System 접근법에서 수치적 분해를 이용하여 출레스키 인자  $L$ 을 구성하는 방법에 대해 기술하였다. 이는 선회요소의 선택폭의 자유로움과 수치적 정확성의 장점에도 불구하고 출레스키 인자  $L$ 을 구성하는데 소요되는 시간이 정규방정식 접근법에 비해 많이 걸린다는 단점이 있다.

정규방정식 접근법에서는 대칭양정치 행렬의 특성을 이용하여 순서화와 수치적 분해를 분리할 수 있다. 따라서 최소차수 순서화를 적용한 다음 상징적 분해를 통해 하삼각 행렬  $L$ 의 비영요소 구조를 결정한다. 하지만 Augmented System 접근법에서는 희소성과 수치적 안정성을 고려하면서, 직접 수치적 분해를 통해 하삼각 행렬  $L$ 을 구성한다. 따라서 행렬의 분해과정 중에 추가되는 비영요소의 수가 많은 경우에는 시간이 많이 걸리게 된다. 여기서는 Augmented System에서 순서화 단계와 수치적 분해단계를 분리할 수 있는 수정된 삭제 그래프를 이용한 최소차수 순서화 방법과, 이로 인해 발생할 수 있는 수치적 불안정성을 처리하는 방법

에 대해 설명한다.

4.1 최소차수 순서화

여러 가지 순서화 방법들 중에서 대칭 양정치 행렬 시스템에는 최소차수 순서화 방법을 주로 사용한다. 그 이유는 최소차수 순서화 방법은 다른 순서화 방법에 비해서 수행속도가 빠르기 때문이다.  $m \times m$  대칭양정치 행렬  $M$ 에 관련된 그래프를  $G = (N, E)$ 라고 하자.  $N$ 은  $M$ 의 행(또는 열)에 해당하고  $E$ 는  $M$ 의 비영요소 위치를 나타낸다. 즉,  $M$ 의  $(i, j)$ 번째 요소  $M_{ij} \neq 0$ 이면 호  $(i, j)$ 가  $E$ 에 들어 있다. 그래프  $G$ 에서 점  $i$ 의 인접점 집합  $Adj_G(i)$ 와 점  $i$ 의 차수  $deg_G(i)$ 를 다음과 같이 정의한다.

[정의 1] 인접점 집합과 차수

$$Adj_G(i) = \{j \mid (i, j) \in E\}.$$

$$deg_G(i) = |Adj_G(i)|.$$

최소차수 순서화 방법은 최소차수인 점을 찾아 이를 삭제하고 변형된 그래프에서 각 점의 차수를 수정하여 다시 차수가 최소인 점을 찾아 삭제하는 과정을 반복하게 된다. 순서화 과정에서 점이 삭제되면 그래프도 같이 바뀌게 되는데 이러한 일련의 그래프를 삭제그래프(elimination graph)라 부른다.  $i$ 번째 점으로  $y$ 가 삭제될때 삭제그래프  $G = (N_i, E_i)$ 는 다음과 같이 구해진다.

$$N_i \leftarrow N_{i-1} - \{y\}$$

$$E_i \leftarrow ((E_{i-1} - \{(u, y) \mid (u, y) \in E_{i-1}\}) \cup \{(u, v) \notin E_{i-1} \mid (u, y) \in E_{i-1}, (v, y) \in E_{i-1}\})$$

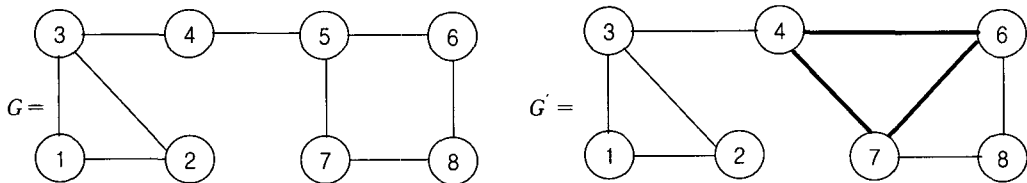
단,  $G_0 = (N_0, E_0)$ 는  $G = (N, E)$ 와 같다.

예를 들어, [그림 1]의 그래프  $G$ 에서 점 ⑤를 삭제하게 되면 삭제그래프  $G'$ 과 같이 된다.

4.2 수정된 삭제그래프를 이용한 순서화 방법

Augmented System 접근법에서는 최소차수 순서화를 사용할 수 없다. 왜냐하면 Augmented System을 구성하는 행렬이 대칭양정치 시스템이 아니기 때문이다. 선형계획법 문제의 Augmented System을 구성하는 행렬은 (5)의  $M$ 과 같다. 즉 처음  $n$ 개의 블록에는 대각요소가 비영(nonzero)이고 다음의  $m$ 개의 블록의 대각요소는 영이다. 대각요소의 값이 0인 블록으로 인해 정규방정식 접근법에서와 같이 최소차수 순서화를 적용할 수 없다. 왜냐하면 행렬  $M$ 에 대한 삭제그래프를 만들 수 없기 때문이다. 하지만 삭제그래프에서 가상의 점을 도입하면 최소차수 순서화를 적용할 수 있다. 예를 들어서 행렬  $M$ 의 비영요소의 구조가 [그림 2]와 같다고 하자.

그러면 위의 행렬에서 5, 6, 7, 8행에는 대각요소에 비영요소가 존재하지 않는다. 따라서 삭제그래프를 구성하지 못한다. 하지만 5, 6, 7, 8행의 대각요소에 비영요소가 있다고 가정을 하면 위의 최소



—는 원래 있던 호, —는 새로 추가된 호

[그림 1] 점의 삭제와 삭제그래프의 변형

차수 순서화 방법에서와 같이 삭제그래프를 구성할 수 있다. 위의 점 5, 6, 7, 8에 대한 차수는 위에서 언급한 정의에 의해서 쉽게 구해질 수 있다. 하지만 점 5, 6, 7, 8은 실제로는 대각요소의 값이 영이기 때문에 다음의 삭제점으로 선택되어질 수 없다. 즉 점 5, 6, 7, 8이 삭제점으로 선택되기 위해서는 5, 6, 7, 8행의 대각요소가 영에서 비영으로 바뀌어져야 한다.

$$M = \begin{pmatrix} * & & * & & * \\ & * & & * & * \\ & & * & * & * \\ * & & & * & * \\ & * & * & & * \\ * & * & & & \end{pmatrix}$$

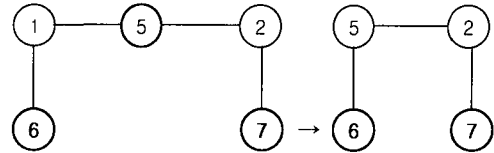
[그림 2] 행렬 M의 비영요소 구조

점 5, 6, 7, 8의 인접점 중에서 대각요소가 비영인 점이 삭제점으로 선택되어지면, 이 삭제점과 연결되어진 대각요소가 영인 점은 비영으로 변하게 된다. 그러면 이 점은 다음 회에서 삭제점으로 선택되어질 수 있는 후보가 된다. 하지만 차수가 아무리 작더라도 대각요소가 0인 점은 삭제점으로 선택되어질 수 없다. 대각요소가 영인 점과 비영인 점인 어떻게 구분하는가는 간단히 일차원 배열을 사용하여 해결할 수 있다. 즉 MARK[i]=0이면 이 점은 점 i의 대각요소가 영이라는 것이다. 따라서 이 점은 아무리 작은 차수를 가지고 있더라도 삭제점으로 선택되어질 수 없다. MARK[i]=1이면 점 i는 삭제점으로 선택되어질 수 있는 후보가 될 수 있다.

다음의 예제는 대각요소가 영인 점이 대각요소가 0가 아닌 점이 삭제됨에 따라 비영요소가 되어 삭제점 후보가 되는 과정을 나타내는 그림이다.

[그림 3]에서 점 1과 점 2는 대각요소가 비영인 점이고 5, 6, 7은 대각요소가 0인 점이다. 점 1이 삭제됨에 따라 5행과 6행에 추가요소가 생기게 되므로 점 5와 점 6은 대각요소가 0에서 비영요소로 바뀌게 된다. 따라서 다음 회에서는 삭제점으로 선

택되어질 수 있는 후보가 된다.



[그림 3] 삭제점 후보의 변화

다음은 수정된 삭제 그래프를 이용한 최소차수 순서화 방법이다. 여기서 G는 행렬 M에 관련된 수정된 삭제그래프이고, |N| = n이라고 하자.

[수정된 삭제 그래프를 이용한 최소차수 순서화]

- 1:  $G_0 \leftarrow G = (N, E)$  ;
- 2: for  $i := 1$  to  $n$
- 3:    $deg(i) \leftarrow |Adj(i)|$  ;
- 4:   if ( $M_{ii}$  is equal to 0)  $mark[i] \leftarrow 0$  ;
- 5:   else  $mark[i] \leftarrow 1$  ;
- 6: end of for
- 7:  $S \leftarrow \emptyset$  ;
- 8: While ( $N - S \neq \emptyset$ )
- 9:    $mindeg \leftarrow \min\{deg(j) ;$   
     $j \in N - S \text{ and } mark(j) = 1\}$  ;
- 10:    $x_0 \leftarrow argmin\{deg(j) ;$   
     $j \in N - S \text{ and } mark(j) = 1\}$  ;
- 11:    $S \leftarrow S \cup \{x_0\}$  ;
- 12:    $T \leftarrow Adj_{G_{i-1}}(x_0)$  ;
- 13:   construct  $G_i$  from  $G_{i-1}$  ;
- 14:   for each  $u \in T$
- 15:     calculate the degree of u  $deg(u)$
- 16:     if ( $mark(u)$  is equal to 0)
- 17:        $mark[u] = 1$
- 18:   end of for
- 19: end of While

4번째 줄과 5번째 줄의 mark 배열은 삭제점으로 선택되어진 점의 대각요소가 비영요소인지를 확

인하는 것이다. 그리고 13~17번째 줄은 삭제점이 지워질 때, 삭제그래프가 갱신되어지고 삭제점의 이웃점들에 대해 차수를 수정하는 부분이다. 삭제점의 이웃점 대각요소가 영인 경우에는 비영요소로 변하기 때문에 해당되는 *mark* 배열의 값은 0에서 1로 바꾸어 준다.

수정된 최소차수 순서화 방법의 단점은 수치적 안정성을 보장할 수 없다는 것이다. 이는 순서화 단계와 수치적 분해단계가 분리되어 선회순서의 결정시 수치적 안정성을 고려하지 않기 때문이다. 따라서 출레스키 분해 과정시 하삼각 행렬  $L$ 의 대각요소 값에 0에 가까운 값이 나올 수 있다. 이러한 문제가 발생할 경우에는, 대각요소의 값과 그 대각요소에 해당되는 하삼각 행렬의 열에 해당되는 모든 요소를 0으로 둔다[5]. 이는 이 열이 종속 열이라는 것을 의미한다. 즉  $L_{ij}=0$ . 하지만 대각요소의 값이 0에 가까운 요소가 많은 경우에는 이러한 요소들의 선회순서를 맨 마지막으로 재조정하고 다시 상징적 분해를 통해 하삼각 행렬  $L$ 을 재구성할 수 있다.

## 5. 구현 및 실험결과

수정된 삭제그래프를 이용한 최소차수 순서화 방법은 기존의 순서화 방법에서 쉽게 구현되어질 수 있다. 기존의 최소차수 순서화 방법과 다른 점은 단지 대각요소가 영이나 비영이나를 구분하는 일차원 배열만 더 필요하다. 그리고 최소차수 순서화에서는 삭제점을 선택할 때 최소차수를 가진 점을 선택하였다. 하지만 Augmented System에서는 반드시 최소차수를 가진 점이 다음의 삭제점으로 선택되어지는 것이 아니다. 최소차수를 가지는 점의 대각요소가 영인 경우에는 대각요소가 비영인 점들 중에서 최소의 차수를 가지는 점이 선택되어진다. 이 방법은 수치적인 안정성을 고려하지 않음으로서 출레스키 분해과정시 불안정해 질 수 있다. 즉 출레스키 분해과정시 대각요소의 값이 0의 값에 가까워지는데 본 실험에서는  $10^{-16}$  이하인 경우에

는 해당열  $L_{ij}=0$ 으로 두었다.

<표 1>은 수정된 삭제그래프를 이용한 최소차수 순서화를 구현하여 실험을 한 결과이다. 대상문제는 NETLIB의 가능해 문제에서 LARGE 계열의 데이터이다. CPU는 알파 워크스테이션 533Mhz이며, RAM은 128M이다.

<표 1>의 실험결과를 보면 수치적 분해를 통해 하삼각 행렬  $L$ 을 구성하는 시간이 수정된 삭제그래프를 이용한 최소차수 순서화 방법에 비해 많은 시간이 소요됨을 알 수 있다. Augmented System의 행렬  $M$ 에서 비영요소가 추가되는 부분은 0에 해당되는  $(n+1) \sim (m+n)$ 에 해당되는 블록이다. 수치적 분해를 이용하는 방법은 비영요소가 추가되는블록들에 대해 가우스 소거과정이 필요하다. 하지만 수정된 삭제그래프를 이용한 최소차수 순서화 방법에서는 선회순서를 결정할 때, 수치적 안정성을 고려하지 않아도 되므로 가우스 소거연산이 필요 없다. 단지 정규방정식의 최소차수 순서화 방법과 같이 행렬  $M$ 의 수정된 삭제그래프를 변경하고 차수를 수정하는 연산이 필요하다. 따라서 하삼각 행렬  $L$ 에 추가되는 비영요소의 수가 많은 경우에는 수정된 삭제그래프를 이용한 최소차수 순서화 방법이 효율적이다. <표 1>의 실험결과를 보면 특히 maros-r7, pilot87, df1001등의 대형 문제들에 대해서는 수치적 분해를 이용하는 방법보다 수정된 최소차수 순서화 방법을 이용하여  $L$ 을 구성하는 하는 것이 매우 효과적임을 알 수 있다. 이 문제들은 모두 하삼각 행렬  $L$ 에서 출레스키 분해과정시 추가되는 비영요소의 수가 400,000개 이상이다. df1001인 경우는 하삼각 행렬  $L$ 에서의 추가되는 비영요소의 수가 약 1,000,000개나 된다.

하지만 fit1p, fit2p, fit1d, fit2d와 같은 문제들에 대해서는 수정된 삭제그래프를 이용한 최소차수 순서화 방법이 비효율적이다. 특히 fit2d와 같은 문제에 대해서는 수정된 삭제그래프를 이용한 최소차수 순서화 방법이 수치적 분해를 이용하여  $L$ 을 형성하는 방법에 비해 약 2.5배나 시간이 많이 걸림을



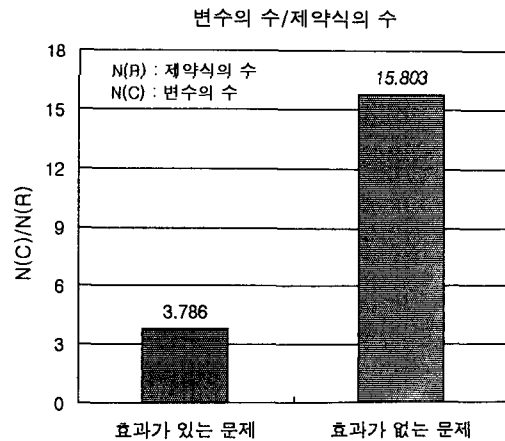
〈표 1〉 수정된 최소차수 순서화 방법의 적용

문제이름	수치적 분해를 통한 $L$ 의 형성			수정된 최소차수 순서화를 통한 $L$ 의 형성		
	(1)	(2)	(3)	(1)	(2)	(3)
25fv47	0.46	1.74	2.37	0.08	1.71	1.97
80bau3b	1.34	6.00	7.87	0.38	6.08	6.98
bni2	1.35	4.92	6.59	0.16	4.47	4.97
cycle	0.55	2.43	3.32	0.10	2.46	2.91
czprob	0.05	0.66	0.90	0.07	0.65	0.91
d2q06c	7.39	22.82	30.77	0.67	20.53	21.76
d6cube	3.55	7.00	11.00	5.15	7.21	12.83
degen3	5.77	9.54	15.74	0.60	9.31	10.28
fit1d	0.23	0.49	0.84	0.41	0.51	1.04
fit1p	0.11	0.40	0.74	0.15	0.43	0.82
fit2d	19.15	8.42	28.87	49.64	8.48	59.32
fit2p	1.60	4.76	7.05	3.53	4.83	9.14
greenbea	0.67	3.72	4.96	0.19	3.53	4.27
greenbeb	0.61	3.14	4.31	0.18	3.18	3.95
maros-r7	132.46	78.25	212.04	2.51	93.17	96.92
maros	0.13	0.57	0.85	0.03	0.57	0.75
nesm	0.30	1.68	2.20	0.07	1.74	2.04
pilot	21.58	34.08	56.18	0.86	30.28	31.60
pilot87	113.92	148.08	262.75	3.05	137.89	141.63
pilotja	0.88	3.32	4.43	0.14	3.31	3.67
pilotnov	1.00	2.15	3.37	0.14	1.92	2.29
scsd8	0.05	0.31	0.50	0.03	0.33	0.50
sctap3	0.30	0.79	1.24	0.09	0.82	1.06
ship08l	0.07	0.60	0.89	0.06	0.60	0.88
ship12l	0.09	0.86	1.23	0.06	0.87	1.20
ship12s	0.04	0.33	0.53	0.02	0.34	0.51
truss	0.66	3.45	4.51	0.18	3.75	4.38
woodlp	1.52	3.47	5.66	1.52	3.31	5.48
woodw	0.58	3.36	4.45	0.33	3.29	4.16
dfi001	851.85	864.73	717.95	16.27	802.87	820.44

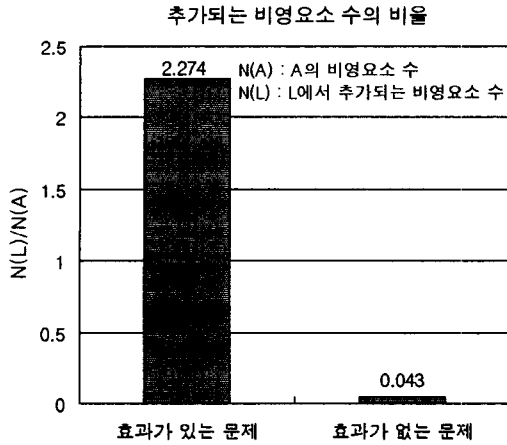
주) (1) : 하삼각 행렬  $L$ 을 구성하는 시간, (2) : 문제의 풀이시간(사전처리, 순서화 시간 제외) (3) : 전체풀이 시간

알 수 있다. 이와 같은 문제들은 제약식의 수에 비해 변수의 수가 매우 많다. 그리고 출레스키 분해 과정시 추가되는 비영요소의 수가  $A$ 가 가지고 있는 비영요소의 수에 비해 매우 작다. 이와 같은 문제들에 대해서는 추가되는 비영요소 수에 대한 수치적 분해과정을 고려하지 않음으로서 얻는 효과에 비해 수정된 삭제그래프를 유지 및 변형하는 시간이 더 많이 소요된다. 따라서 수정된 삭제그래프를 이용하는 방법은 위와 같은 문제들에 대해서는 비효율적이다.

[그림 4]는 제약식의 수와 변수의 수의 비율들에 대한 기하평균을 나타내고 [그림 5]는  $A$  행렬이 가



[그림 4] 변수의 수/제약식의 수



지고 있는 비영요소 수와 하삼각 행렬  $L$ 에 추가되어지는 비영요소 수의 비율들에 대한 기하평균을 나타낸다.

## 6. 결 론

내부점 방법은 매회 개선방향을 구하기 위해 선형연립방정식을 풀어야 한다. 이 선형연립방정식을 푸는 방법에는 두 가지 접근법이 있다. 하나는 정규방정식을 푸는 방법이고 다른 하나는 Augmented System을 직접 대칭 분해하는 것이다. 정규방정식 방법은 선형시스템을 구성하는 행렬이 대칭 양정치라는 특성을 이용한 순서화를 통해 선회순서를 결정한다. 그리고 난 후 상징적 분해를 통해 하삼각 행렬의 비영요소 구조를 결정한다. 그 이후에는 이 하삼각 행렬의 비영요소 구조를 유지한 채 수치적 분해만 수행하면 된다. 즉 순서화와 수치적 분해가 분리될 수 있다. 이는 속도가 빠른 최소차수 순서화와 상징적 분해를 통해 하삼각 행렬의 비영요소 구조를 매우 빠르게 구성할 수가 있다. 하지만 Augmented System을 구성하는 행렬은 대칭 부정치 행렬이다. 따라서 순서화와 수치적 분해를 분리할 수 없다. 즉 하삼각 행렬의 비영요소 구조를 결정하기 위해서는 희소성과 수치적 안정성을 고려하여 직접 수치적 분해를 해야한다. 이로 인해

대형 선형계획법 문제를 Augmented System 방법으로 풀 때, 출레스키 인자  $L$ 의 비영요소 구조를 결정하기 위해 소요되는 시간이 정규방정식 접근법에 비해 증가하게 된다.

수정된 삭제 그래프의 개념을 이용하면 대칭 양정치 행렬에 적용되는 최소차수 순서화를 사용할 수 있는 장점이 있다. 단지 순서화 과정 중에서 대각요소가 비영요소인지 아닌지 구분하는 정보만 있으면 기존의 최소차수 순서화 방법을 쉽게 적용할 수 있다. 그리고 수정된 삭제그래프를 이용한 최소차수 순서화 방법으로 발생하는 수치적 불안정성 문제는 출레스키 인자의 해당 열을 0으로 두거나, 선회순서를 뒤로 변경함으로써 처리할 수 있다. 실험결과 수정된 최소차수 순서화 방법을 적용함으로써 출레스키 인자  $L$ 의 형성을 위한 시간과 이로 인한 전체적인 수행시간의 감소효과가 있었다.

## 참 고 문 헌

- [1] 박순달, 김우제, 설동렬, 박찬규, 성명기, 임성목, 「고등선형계획법」, 교우사, 2001.
- [2] 박찬규, 박순달, “하한을 이용한 효율적인 최소차수 순서화”, 「한국경영과학회지」, 제23권, 제4호(1998), pp.21-31.
- [3] 모정훈, “내부점 선형계획법에서의 순서화 방법과 자료구조에 관한 연구”, 「서울대학교 공학석사 학위논문」, 1995.
- [4] 설동렬, “내부점 방법에서의 출레스키 분해”, 「서울대학교 공학박사 학위논문」, 1999.
- [5] 설동렬, 성명기, 안재근, 박순달, “내부점 방법에서 출레스키 분해의 수치적 안정성”, 「대한산업공학회지」, 제25권, 제3호(1999), pp.290-297.
- [6] Andersen E.D., J. Gondzio, C. Mészáros, and X. Xu, “Implementation of interior point methods for large scale linear programming,” In : Terlaky T, editor. *Interior*

- point methods in mathematical programming*, Dordrecht : Kluwer Academic Publishers, 1996.
- [7] Duff I., A. Erisman and J. Reid, *Direct Methods for Sparse Matrices*, Oxford University Press, New York, 1986.
- [8] Mészáros C., "The Efficient Implementation of Interior Point Methods for Linear Programming and their Applications," *Dissertation*, Eötvös Loránd University of Sciences, Hungary, 1996.
- [9] George, A. and Joseph. W.H. Liu, *Computer Solution of Large Scale Positive Definite Systems*, Prentice-Hall, 1981.
- [10] George, A. and Joseph. W.H. Liu, "The evolution of the minimum degree ordering algorithm," *SIAM Review*, Vol.31, No. 1(1989), pp.1-19.
- [11] Maros I. and Mészáros C., "The role of the augmented system in interior point methods," *European Journal of Operations Research*, Vol.107(1998), pp.720-736.