한국경영과학회/대한산업공학회 2003 춘계공동학술대회
2003년 5월 16일-17일 한동대학교(포항)

클릭을 이용한 근사최소 부족수 순서화†

도승용[1], 박찬규[2], 이상욱[1], 박순달[1]

[1]서울대학교 산업공학과 / [2]한국전산원 정보기술 감리부

An Approximate Minimum Deficiency Ordering using Cliques

Seungyong Doh[1], Chan-Kyoo Park[2], Sangwook Lee[1], Soondal Park[1]

ABSTRACT

For fast Cholesky factorization, it is most important to reduce the number of non-zero elements by ordering methods. Minimum deficiency ordering produces less non-zero elements. However, since it is very slow. the minimum degree algorithm is widely used. To improve the computation time, Rothberg's AMF uses an approximate deficiency instead of computing the deficiency. In this paper we present simple efficient methods to obtain a good approximate deficiency using information related to cliques. Experimental results show that our proposed method produces better ordering quality than that of AMF.

1. 서론

선형계획법문제를 내부점 방법(interior-point method)으로 풀 때 대칭양정치(symmetric positive definite) 행렬의 선형방정식을 푸는 과정이 필요하다[11]. 일반적으로 대형의 선형계획법 문제들은 희소한 특성을 가지는데, 내부점 선형계획법에서 나타나는 대칭양정치 행렬도 역시 희소행렬(sparse matrix)이다[2]. 내부점 선형계획법에서는 전체 해법 소요 시간의 많은 부분이 선형방정식을 푸는 데에 소요되어, 효율적으로 선형방정식을 풀어내는 것이 해법의 성능을 크게 좌우한다.

선형방정식을 푸는 데에는 주로 촐레스키 분해(Choleksy factorization)를 사용한다. 희소행렬에 대하여 촐레스키 분해를 수행할 때에는 촐레스키 분해행렬의 비영요소(non-zero element)수를 줄이기 위하여 순서화 방법을 적용한다. 최적의 순서화를 찾는 것은 NP-Complete인 것으로 알려져 있다[20]. 따라서, 순서화에서는 최소 차수 순서화(minimum degree ordering)[7,9], 최소 부족수 순서화(minimum deficiency ordering or minimum local fill ordering)[7] 등의 발견적 방법이 사용되는데, 빠른 수행시간으로 인해 최소 차수 순서화가 널리 사용되고 있다[4]. 최소 부족수 순서화는 대체로 최소 차수 순서화에 비해서 추가요소의 개수를 더 줄여준다. 그럼에도 불구하고 순서화에 소요되는 시간이 최소 차수 순서화에 비해 매우 많이 걸

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

리는 문제점을 가지고 있다. Vanderbei[19]의 연구에 의하면 내부점 방법의 수행 회수가 많을 경우에 최소 차수 순서화보다 최소 부족수 순서화가 전체 수행 시간 면에서 유리할 수 있다. Lustig[11] 등은 예측자-수정자 방법과 같이 수행 회수가 적은 내부점 방법이 등장함으로 말미암아 최소 부족수 순서화보다는 최소 차수 순서화가 일반적인 경우에 더 효율적이라고 주장하였지만, Mészáros[5]는 최소 부족수 순서화의 개념을 이용하여 최소 차수 순서화보다 비영요소를 줄임으로써 내부점 방법의 속도를 더 개선할 수 있음을 보였다.

최소 부족수 순서화 방법은 매 회마다 최소의 부족수를 가지는 점을 찾아서 삭제하는 방법으로 Tinney & Walker[18]에 의해 제안된 순서화 방법 가운데 방법 3의 대칭행렬에 해당하는 방법이다. 최소 부족수 순서화 방법도 역시 최소 차수 순서화와 같이 최적 순서화 방법은 아니고, 발견적 기법이다.

최소 부족수 순서화라는 이름은 Rose[16])에 의해서 사용되었다. Duff 등[5]은 현 단계에서 추가되는 비영요소의 개수를 가장 작게 하는 점을 찾아서 삭제하는 방법이라는 의미에서 국부 최소 추가요소 순서화(local minimum fill-in order)라고 이름하였다.

최소차수 순서화 방법은 삭제점의 이웃점들에 대해서만 차수수정이 필요하다. 하지만 부족수를 계산하기 위해서는 삭제점의 이웃점 뿐만 아니라 이웃점들의 이웃점들에 대해서도 검색을 해야한다. 이로 인해 부족수를 계산하는데 많은 시간이 소요된다[. 최소부족수 순서화 방법이 최소차수 순서화 방법에 비해 추가되는 비영요소의 수가 작음에도 불구하고 수행시간의 비효율성으로 인해 많은 관심을 끌지 못했다. 최소부족수 순서화 방법의 수행시간을 향상시키기 위해 구별불능점, 외부차수, 복수삭제 등 최소차수 순서화 방법에 이용되었던 여러가지 효율화 기법들이 적용되었다. 하지만 수행시간의 향상에 있어서 큰 효과를 얻지 못했다.

NG[15]는 최소차수 순서화 방법에서와 같이 삭제점의 이웃점들에 대해서만 부분클릭(partial clique)을 이용하여 부족수를 계산하는 수정된 최소부족수 순서화(modified minimum deficiency ordering) 방법을 제안하였다. Rothberg[17]는 Amestoy 등의 근사최소차수 순서화(approximate minimum degree ordering : AMD)[6] 방법에서 생성된 가장 최근의 클릭을 이용한 근사최소부족수(approximate minimum local fill : AMF)순서화 방법을 제안하였다. AMF에서는 부족수를 계산하는 대신 근사부족수를 사용한다. 본 연구에서는 AMF의 근사부족수보다 좀 더 강화된 상한을 제공하는 근사부족수를 계산하는 방법을 제안한다. 그리고 내부점 방법에서 나타나는 행렬 $M=AA^T$에 대응되는 그래프 $G(M)$의 클릭을 이용하여 초기의 근사부족수를 설정하는 방법을 제안한다. 실험결과 본 연구에서 제시한 방법은 기존의 AMF에 비해 비영요소의 수를 5%정도 감소시킬 수 있었다.

본 논문의 구성은 다음과 같다. 2장에서는 최소부족수 순서화의 개념과 효율화 기법들에 대해 다룬다. 3장에서는 근사최소부족수 순서화 방법과 클릭을 이용하여 근사부족수의 상한을 강화하는 방법에 대해 다룬다. 4장에서는 본 연구에서 제안한 방법에 대한 실험결과이다. 그리고 5장은 본 논문의 결론이다.

2. 최소부족수 순서화

2.1 최소부족수 순서화의 개념

행렬 $M=AA^T$을 대칭양정치 행렬이라고 할 때 행렬 M의 비영요소의 구조는 무방향그래프 $G(M)=(N,E)$로 표시할 수 있다. $G(M)$의 점의 집합 N는 행렬 M의 행(열)에 대응되고 호의 집합 E는 행렬 M의 비영요소에 대응된다. 즉, 호 (i,j)는 $m_{ij}(\neq 0)$에 대응된다. 먼저 다음의 용어를 정의하자.

정의 1. 인접점 집합, 차수, 부족수

그래프 $G(N,E)$에서,

점 i의 인접점 집합은 $Adj(i)=\{j|(i,j)\in E\}$이다.

점 i의 차수는 $\deg(i)=|Adj(i)|$이다.

점 i의 부족수는

$\mathrm{def}(i) = |\{(j,k) \mid j,k \in Adj(i),\ j \neq k,\ j \notin Adj(k)\}|$이다.

대칭행렬 M의 분해과정은 일련의 삭제그래프로 $G^k = (N^k, E^k)$로 나타내어질 수 있다. G^0는 행렬 M에 대한 삭제그래프이다. 삭제그래프 G^k는 G^{k-1}로부터 변형되어진다. x_k를 G^{k-1}에서의 삭제점이라고 하자. 그러면 삭제그래프 G^k는 다음의 두 단계의 과정을 거쳐서 G^{k-1}로부터 변형되어진다.

단계 1 : 삭제그래프 G^{k-1}에서 점 x_k와 연결되어진 모든 점 v를 제거한다.

단계 2 : G^{k-1}에서 점 x_k의 인접점들이 모두 연결되도록 호를 추가한다.

단계 2에서 삭제되는 점의 인접점들이 서로 연결이 되지 않은 경우에 이 점들을 서로 연결하는 새로운 호가 추가되는 데 이 호들은 행렬의 관점에서 보았을 때 추가되는 비영요소이다.
삭제점 선택기준에 따라 여러 가지 순서화 방법이 있다. 최소차수 순서화 방법에서는 현재의 삭제그래프 G^k에서 최소의 차수를 가지는 점을 삭제점으로 선택한다. 최소부족수 순서화 방법에서는 최소의 부족수를 가진 점을 삭제점으로 선택한다. 다음은 최소부족수 순서화 알고리즘이다.

알고리즘 1. 최소 부족수 순서화

단계 1 : $G^0 = (N^0, E^0)$

단계 2 : 모든 $u \in N^0$에 대해 부족수를 계산한다.

단계 3 : $G^k = (N^k, E^k) = \varnothing$이면 종료하고 아니면 최소의 부족수를 가지는 점 x_k를 선택한다.

단계 4 : G^k를 G^{k+1}로 변형한다.

단계 5 : 모든 $u \in N^{k+1}$에 대해 부족수를 계산하고, $k \leftarrow k+1$로 하고 단계 3으로 간다.

점 u의 부족수는 점 u가 삭제될 때 추가되는 비영요소의 수를 의미한다. 최소부족수 알고리즘에서 수행시간이 가장 많이 소요되는 부분이 바로 단계 5의 부족수를 계산하는 부분이다. 점 u의 부족수를 계산하기 위해서는 점 u의 인접점 집합 $Adj_{G^k}(u)$와 $Adj_{G^k}(u)$의 인접점 집합을 탐색해야 한다.

2.2 최소부족수 순서화의 효율화 기법

부족수의 계산시 구별불능점과 외부차수를 이용하면 부족수를 효율적으로 계산할 수 있다. 먼저 구별불능에 대한 다음의 정의를 보자.

정의 2. 구별불능(indistinguishable)[10]

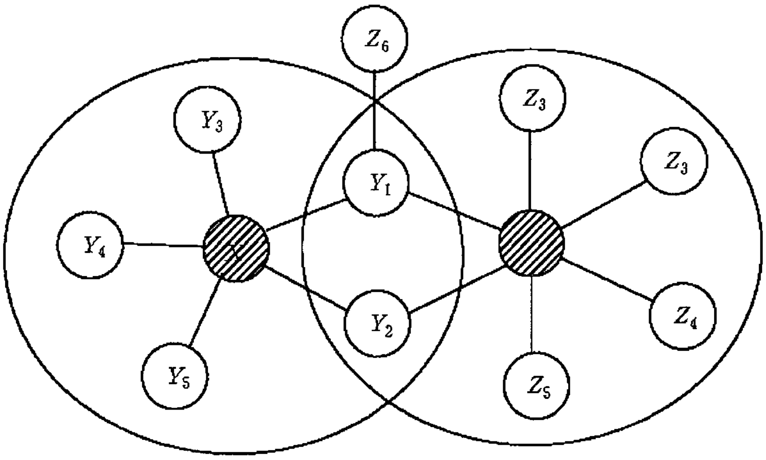
점 i와 j가 $Adj_{G^k}(j) \cup \{j\} = Adj_{G^k}(i) \cup \{i\}$이면 두 점은 구별불능이라고 한다.

점 i와 j가 구별불능이면 자신을 포함한 서로의 인접점 집합이 같다는 것이다. G^k에서 구별불능인 점들은 G^{k+1}에서 역시 구별불능점을 형성한다. 구별불능점들의 주요 특징은 서로 하나의 수퍼노드(supernode)로서 간주되어질 수 있다는 것이다. 이로 인해 여러 개의 점들을 하나의 수퍼노드로서 나타낼 수 있으므로 삭제그래프의 크기를 감소시킬 수 있다. 그리고 부족수의 계산시 구별불능점에 속하는 점들은 한 번만 계산을 해주면 된다. 왜냐하면 구별불능점들은 같은 부족수를 갖기 때문이다. 점 i와 점 j가 구별불능인 경우에는 두 개의 점을 대표하기 위해 수퍼노드 $I = \{i, j\}$로 나타내기로 한다.

정의 3 외부차수[6]

점 i의 외부차수 $\mathrm{edeg}(i) = \deg(i) - |I| + 1$이다.

정의 3의 외부차수는 차수계산시 같은 수퍼노드 안에 속하는 점들을 배제한 것이다. 수퍼노드 안에 존재하는 점들은 모두 서로 연결되어져 있다. 즉 수퍼노드에 속하는 점들은 서로 클릭을 형성한다. 수퍼노드 I를 구성하는 점들의 수를 수퍼노드의 크기라고 한다. 수퍼노드가 아닌 점 i는 크기가 1



[그림 1] 삭제그래프

인 수퍼노드 $I-\{i\}$로 간주할 수 있다. 따라서 외부차수와 수퍼노드를 이용하면 효율적으로 부족수를 계산할 수 있다.

정리 1. 부족수의 계산[3,4]
그래프 $G(N,E)$에서 점 I의 부족수는

$$\mathrm{def}(I) - \binom{\mathrm{edeg}(I)}{2}$$

$$- \sum_{J \in Adj(I)} \binom{|J|}{2} - \sum_{J>K;\, J,K \in Adj(I)} |J| \times |K|$$

이다.

3. 클릭을 이용한 근사최소부족수 순서화

최소부족수 순서화 방법의 가장 큰 단점은 수행시간의 비효율성이다. 외부차수와 구별불능점등의 기법등이 적용되었지만 수행시간면에서 최소차수 순서화 방법에 비해 매우 느리다. Rothberg[17]는 AMD를 기반으로 하여 삭제과정속에서 생성된 클릭의 정보를 이용한 근사최소부족수 순서화 방법 AMF를 제안하였다.

3.1 근사부족수의 계산

삭제그래프 G^k에서 점 X가 삭제가 되면 $Adj_{G^k}(X)$는 클릭을 형성한다. 클릭에 속하는 모든 점들 사이에는 서로서로 연결하는 호가 존재한다. 점 $Y \in Adj_{G^k}(X)$의 부족수를 계산시에 $Adj_{G^k}(X)$를 통해 Y가 삭제되기 전에 이미 존재하는 호를 쉽게 파악할 수 가 있다. 이를 이용하여 AMF에서는 삭제점 선택기준으로 근사부족수 $d_{AMF}(Y)$를 다음과 같이 정의하였다.

정의 4.
X를 현재의 삭제점이라하고, $C-Adj(X)$를 가장 최근에 생성된 클릭이라고 하자. 그러면 $Y \in Adj(X)$에 대해,

$$d_{AMF}(Y) - \delta - c(c-1)$$

여기서 $\delta - \mathrm{edeg}(Y)(\mathrm{edeg}(Y)-1)$이고, $c-|C \setminus Y|$이다.

AMF에서는 삭제된 점을 선택할 때 $d_{AMF}()$의 값이 가장 작은 점을 선택한다.

[그림 1]을 보자. 빗금 친 점들은 삭제된 점들이다. X를 가장 최근에 삭제된 점이라고 하면 가장 최근에 생성된 클릭은 $C=\{Y_1,\cdots,Y_5\}$이다. C에 속한 점들의 $d_{AMF}()$의 값이 수정되어진다. Y_1에 대해

$$d_{AMF}(Y_1) = \text{edeg}(Y_1)(\text{edeg}(Y_1)-1) - |C\setminus Y_1|(|C\setminus Y_1|-1)$$

이다.

3.2 근사부족수의 강화

순서화과정에서 어떤 점이 삭제가 되면 삭제된 점의 이웃점들은 클릭을 형성하게 된다. 그러면 점 X의 이웃점들의 집합은 그래프 G^0에서의 X에서 이웃한 점들과 점 X가 속한 클릭들을 이용하여 구할 수 있다. C를 가장 최근에 생성된 클릭이라 하고, $C_1^X, C_2^X, \cdots, C_k^X$를 점 X를 포함하는 클릭이라고 하자. 그러면 점 $X\in C$의 인접점 집합은 다음과 같다.

$$Adj_{G^k}(X) = \{Y|(X,Y)\in E^0\}\cup C\cup C_1^X\cup\cdots\cup C_k^X$$

AMD를 이용하면 차수의 계산시 가장 최근에 생성된 클릭과 이전에 생성된 클릭과의 교집합을 알 수 있다. 즉 $C_i^X\setminus C$를 알 수 있다. 이를 이용하여 다음과 같이 강화된 근사부족수를 정의할 수 있다.

정의 5

가장 최근에 생성된 클릭을 C라 하고, $C_1^X, \cdots, C_k^X$를 X를 포함한 클릭이라고 하자. 점 $X\in C$에 대해, $\overline{d}(X)$를 다음과 같이 정의한다.

$$\overline{d}(X) = \delta - c(c-1) - \alpha(\alpha-1) - 2\alpha\beta$$

여기서,

$c = |C\setminus X|$,

$\alpha = |C_m^X\setminus C|$, $m = argmax\{|C_i^X\setminus C|\,|\,C_i^X\in K\}$

$\beta = \{|C_m\cap C - \{X\}|$

$\overline{d}(X)$는 정리 2에 의해 $d_{AMF}(X)$에 비해 부족수에 대해 강화된 상한을 제공한다.

정리 2

$X\in C$에 대해 $\overline{d}(X)\le d_{AMF}(X)$이다.

증명)

$$\begin{aligned}\overline{d}(X) &= \delta - c(c-1) - \alpha(\alpha-1) - 2\alpha\beta\\ &= d_{AMF}(X) - \alpha(\alpha-1) - 2\alpha\beta\\ &\le d_{AMF(X)}(\because \alpha\ge 0, \beta\ge 0)\end{aligned}$$

[그림 1]에서 $\overline{d}(Y_1)$을 계산해보자. [그림 1]에서는 Y_1을 포함하는 클릭이 두 개가 있다. 하나는 가장 최근에 생성된 클릭 $C=\{Y_1, Y_2, Y_3, Y_4, Y_5\}$와 $C_1=\{Y_1, Y_2, Z_3, Z_4, Z_5\}$이다. 그러면 $\alpha = |\{Z_3, Z_4, Z_5\}|$이고, $\beta = |\{Y_2\}|$이다. AMF에서는 $\alpha(\alpha-1)+2\alpha\beta$항을 고려하지 않는다.

3.3 초기클릭을 이용한 근사부족수의 강화

내부점 방법에서는 다루는 행렬은 $M = AA^T$와 같은 형태이다. 근사최소부족수 순서화방법의 초기단계에는 삭제점이 없기 때문에 클릭을 쉽게 파악할 수 없다. 하지만 다음의 관찰은 행렬 A를 이용하여 M에 해당하는 그래프 $G(M)$에서 클릭을 쉽게 파악할 수 있게 해준다.

관찰 1[1].

주어진 행렬 A의 각 열의 비영요소에 해당하는 각 행의 비영요소는 행렬 $M=AA^T$에 해당되는 그래프에서 클릭을 형성한다.

(증명) $M=AA^T$에서 m_{ij}가 비영요소가 되는 경우는 $a_{ik}\ne 0$이고 $a_{jk}\ne 0$인 열 k가 존재하는 경우이다. 즉, 행렬 A의 임의의 열 k에서 비영요소가 행 $I=\{i_1, i_2, \cdots, i_l\}$인 위치가 존재한다면 I의 원소

로 이루어지는 모든 쌍 (p, q)에 대해여 $m_{pq} \neq 0$. 단 $p, q \in I$. 이것은 그래프의 관점에서 보면 I에 해당하는 점들은 모두 클릭을 형성한다는 것이다.

관찰 1로부터 행렬 A의 열로부터 행렬 M의 그래프 G^0의 클릭을 쉽게 파악할 수 있다. 따라서 초기단계에서 다음과 같이 $\overline{d}(i)$를 다음과 같이 설정할 수 있다.

모든 $i \in N^0$에 대해 $\overline{d}(i) = \delta - \gamma(\gamma - 1)$이다. 여기서 $\delta = \deg(i)(\deg(i) - 1)$,

$\gamma = \max\{|Nonz(A_{\cdot j}) - 1| : a_{ij} \neq 0,\ j = 1, \cdots, n\}$,

$Nonz(A_{\cdot j})$는 행렬 A에서 j번째 열의 비영요소의 수이다.

4. 구현 및 실험결과

본 연구에서 제안된 근사최소부족수 순서화 방법은 AMD기반하에 구현된다. δ값의 계산시 원래의 차수가 아닌 근사차수를 사용하여 계산된다. 그리고 AMD에서 삭제과정을 통해 생성되어지는 클릭들을 이용하여 $\overline{d}()$를 계산한다. <표 1>은 대형의 선형계획법 문제들에 대해 실험한 결과이다. 실험환경은 PentiumIII 600Mhz, RAM 128M이다.

<표 1>에서 m과 n은 각각 행렬 A의 행과

<표 1> AMF와의 비교

문제이름	m	n	$Nonz(A)$	AMF		제안된 방법	
				$Nonz(L)$	시간	$Nonz(L)$	시간
bnl2	2324	3489	13999	85231	0.05	83692	0.05
cycle	1903	2857	20720	93796	0.05	58379	0.05
d2q06c	2171	5167	32417	124959	0.07	101165	0.07
dfl001	6071	12230	35632	1470479	0.59	1479413	0.62
pilot	1441	3652	43167	181781	0.17	175862	0.18
pilot87	2030	4883	73152	431454	0.34	422630	0.34
cre-a	3516	4067	14987	33593	0.16	33535	0.20
cre-b	9648	72447	256095	940262	1.31	940775	1.53
cre-c	3068	3678	13244	31565	0.14	31260	0.17
cre-d	8926	69980	242646	935750	1.18	867973	1.39
ken-07	2426	3602	8404	13081	0.03	13032	0.03
ken-18	105127	154699	358171	2215374	7.20	2116446	9.35
pds-10	16558	48763	106436	1535107	1.51	1535801	1.67
pds-20	33874	105728	230200	6676288	5.22	6607511	5.81
ch	3700	5062	20873	134242	0.15	136967	0.16
co5	5774	7993	53661	182402	0.45	179406	0.54
co9	10789	14851	101578	497637	1.09	459134	1.38
cq5	5048	7530	47353	165328	0.40	161352	0.51
cq9	9278	13778	88897	452145	0.88	438844	1.12
mod2	34774	31728	165129	1798203	2.55	1776611	2.87
nl	7039	9718	41428	289542	0.39	288438	0.48
world	34506	32734	164470	1773170	2.54	1752896	2.82
bas1lp	5411	4461	582411	2356174	21.48	2171341	15.41
baxter	27441	15128	95971	7332343	2.96	5301296	3.02
dbic1	43200	183235	1038761	2417102	77.05	2404728	78.63
p10	10090	19000	117910	588340	0.86	566531	0.87
rlfddd	4050	57471	260577	810826	0.63	798485	0.51
rlfdual	8052	66918	273979	2650895	1.51	2564057	1.54
route	20894	23923	187686	3118127	1.45	2913325	1.38
testbig	17613	31223	61639	6767459	3.64	5502987	3.07
ulevimin	6590	44605	162206	353442	0.66	356619	0.75
sws	14310	12465	93015	134671	0.91	135133	1.27
lpl1	39951	125000	381259	2968279	3.87	2971619	3.63
seymourl	4944	1372	33549	457870	0.69	417804	0.85

열의 수를 나타낸다. 그리고 $Nonz(A)$와 $Nonz(L)$은 행렬 A와 하삼각 행렬 L의 비영요소의 수를 나타낸다.

본 연구에서 제안된 근사부족수와 행렬 $M-AA^T$의 클릭을 이용한 방법이 AMF에 비해 하삼각 행렬 L의 비영요소의 수는 5%정도 작고 수행시간은 8%정도 증가했다. 선형계획법 문제를 내부점 방법으로 푸는 경우 순서화는 초기단계에서 한 번만 수행된다. 이후에는 하삼각 행렬 L의 비영요소 구조를 계속해서 사용하여 수치적 분해를 수행한다. 따라서 약간의 수행시간의 증가가 있더라도 비영요소의 수를 줄이는 것이 더 중요하다고 할 수 있겠다.

5. 결론

최소부족수 순서화 방법은 추가되는 비영요소의 수가 작음에도 불구하고 수행시간의 비효율성으로 인해 실용적인 관점에서 관심을 끌지 못했다. 수행시간의 향상을 위해 여러 가지 효율화 기법들이 적용되었지만 큰 효과가 없었다. AMF는 클릭을 이용하여 근사부족수를 사용함으로서 부족수 계산에 소요되는 시간을 줄일 수 있었다. 본 연구에서는 근사부족수를 효율적으로 계산하는 방법들을 제안하였다. 첫째, 차수의 계산시 얻어지는 클릭들간의 정보를 이용하여 AMF에 비해 강화된 근사부족수를 계산하는 방법을 제안하였다. 둘째, 그래프 $G(M)$의 클릭들을 이용하여 초기의 근사부족수를 설정하는 방법을 제안하였다. 그래프 $G(M)$의 클릭들은 행렬 A의 열로부터 쉽게 파악될 수 있다. 실험결과 본 연구에서 제시한 방법은 AMF방법에 비해 하삼각행렬 L의 비영요소의 수를 약 5%정도 줄일 수 있었다.

참고문헌

[1] 모정훈, 박순달, "최소차수의 자료구조개선과 효율화에 관한 연구", 경영과학, 12권, 2호, pp. 31-42, 1995. 8.

[2] 박순달, 김병규, 성명기, "내부점기법에 있어서 효율적인 순서화와 자료구조", 한국경영과학회지, 제21권 3호, pp. 63-74, 1996. 12.

[3] 설동렬, 박순달, "내부점 방법에서 최소 부족수 순서화의 실험적 고찰", 97 춘계 한국경영과학회/대한산업공학회 공동학술대회 논문집, 포항공과대학교, pp. 410-413, 1997.

[4] 설동렬, 박찬규, 박순달, "클릭저장구조에서 최소 부족수 순서화의 효율화", 대한산업공학회, 제24권, 제3호, pp. 407-416, 1998. 9

[5] Csaba Mészáros, "The inexact minimum local fill-in ordering algorithm", Technical Report WP 95-7, Computer and Automation Institute, Hungarian Academy of Sciences, Budapest, 1995.

[6] Amestoy, Patrick, Timothy A. Davis, and Iain S. Duff, "An aproximate minimum degree ordering algorithm", SIAM J. Matrix Anal. Appl., Vol. 17, pp. 886-905, 1996

[7] Duff, I., A. Erisman and J. Reid, Direct Methods for Sparse Matrices, Oxford University Press, New York, 1986.

[8] Gay, D. M., "Electronic mail distribution of linear programming test problems", Mathematical Programming Society Committee on Algorithms Newsletter, no. 13, pp. 10-12, 1985.

[9] George, Alan and Joseph. W. A. Liu, Computer Solution of Large Sparse Positive Definite Systems, Prentice-Hall, Englewood Cliffs, NJ., 1981.

[10] George, Alan and Joseph. W. A. Liu, "The evolution of the minimum degree ordering algorithm", SIAM Review, No. 31, pp. 1-19, 1989.

[11] Lustig, Irvin J., Roy E. Marsten and David F. Shanno, "The Interaction of Algorithms and Architectures for Interior Point Methods", Advances in Optimization and Parallel Computing, P. M. Pardalos(Ed.), Elsevier Science Publishers, pp. 190-205, 1992.

[12] Kou, L. T., L. J. Stockmeyer, C. K. Wong, "Covering edges by cliques with regard to keyword conflicts and intersection graphs", Comm. ACM, no. 21, pp. 135-138.

[13] Liu, Joseph W. A. "Modification of the minimum-degree algorithm by multiple elimination", ACM Transaction of Mathematical Software, vol. 11, no. 2, pp. 141-153, 1985.

[14] Markowitz, H. M., "The elimination form of the inverse and its application to linear programming", Management Science, no. 3, pp. 255-269, 1957.

[15] Ng, Esmond G. and Padma Raghavan, "Performance of gree요 ordering heuristics for sparse cholesky factorization",

[16] Rose, D. J., "A graph-theoretic study of the neumerical solution of sparse postitive definite systems of linear equations", in Graphs Theory and Computing, R. C. Read, ed., Academic Press, pp.183-217, 1972

[17] Rothberg, Edward and Stanley C. Eisenstat, "Node selection strategies for bottom-up sparse matrix ordering", SIAM J. Matrix Anal. and Appl. Vol. 19, No. 3. pp. 682-695, July 1998.

[18] Tinney, W.F., J. W. Walker, "Direct solutions of sparse network equations by optimally ordered triangular factorization", Proceedings of the IEEE 55, ppp. 1801-1809, 1967

[19] Vanderbei, Robert J., "A comparison between the minimum-local-fill and minimum-degree algorithm", Technical Report, Princeton University, Department of Civil Engineering and Operations Research, Princeton, NJ, 1990.

[20] Yannakakis, M., "Computing the minimum fill-in is NP-complete", SIAM Journal of Algebraic Discrete Methods , no. 2, pp. 77-79, 1981.