대형선형계획문제의 사전처리†

Preprocessing in Large Scale Linear Programming Problems†

성명기*, 박순달*

* 서울대학교 산업공학과

Abstract

Generally MPS, standardized by IBM, is the input type of large scale linear programming problems, and there may be unnecessary variables or constraints. These can be discarded by preprocessing. As the size of a problem is reduced by preprocessing, the running time is reduced. And more, the infeasibility of a problem may be detected before using solution methods.

When the preprocessing implemented by this paper is used in NETLIB problems, it removes nunecessary variables and constraints by 21%, 15%, respectively.

The use of preprocessing gives in the average 21% reduction in running time by applying the interior point method.

Preprocessing can detect 10 out of 30 infeasible NETLIB problems.

1. 서론

선형계획문제의 해법으로는 정점해를 찾아가면서 최적해를 구하는 단체법(simplex method)과 가능해 영역의 내부에서 해를 이동하여 최적해를 구하는 내부점기법(interior point method) 등이 있다[1][6]. 이들 방법을 프로그램으로 구현해서 문제를 풀려고 하면, 우선 문제를 입력받아야 하는데, 대형 문제의 경우에는 입력할 자료가 많고, 또 행렬 A의 대부분의 요소가 0이기 때문에 특정한 형태로 문제를 입력하게 된다. 그 중에서 IBM에서 표준으로 정한 MPS형태가 널리 사용되고 있다[1]. 그런데, 문제가 크기 때문에 중복적이거나 비가능인 제약식이 생길 수 있고, 또 변수의 상한과 하한에 오류가 생길 수 있다. 또 행렬 A의 비영요소만을 입력하다 보면 빈 행 또는 열이 생길 수 있게 된다. 그래서 선형계획 프로그램에서는 문제를 입력받은 후, 해법 수행 전에 필요 없는 제약식이나 변수를 제거해야 할 필요가 있는데, 이러한 일을 사전처리(preprocessing)라 한다[3][4][5].

사전처리로 얻을 수 있는 이익으로는, 해법을 수행하기 전에 문제의 비가능성을 판정할 수 있고, 중복제약식의 제거 및 변수 값의 고정 등으로 문제 크기를 작게 만들어 보다 빨리 최적해에 도달할 수 있다는 것 등이다[4].

본 논문에서는 내부점기법 프로그램으로 NETLIB의 문제를 풀어 사전처리의 효과를 알아보고자 한다.

일반적인 선형계획문제는 변수의 상한과 하한이 있는 다음과 같은 일반한계문제이다.

$$
\begin{aligned}
Min \quad & \boldsymbol{c}^T \boldsymbol{x} \\
s.t. \quad & \mathrm{A}\boldsymbol{x} \le \boldsymbol{b} \\
& \boldsymbol{l} \le \boldsymbol{x} \le \boldsymbol{u}
\end{aligned}
$$

여기서 A는 m×n 행렬이고, $\boldsymbol{x}$, $\boldsymbol{l}$, $\boldsymbol{w}$, 그리고 $\boldsymbol{c}$는 n차 벡터, $\boldsymbol{b}$는 m차 벡터이다.

내부점기법은 해법 수행을 위해서 일반한계문제를 하한이 0인 단순상한문제로 변환하여 문제를 풀기 때문에, 다음과 같은 방법으로 문제를 변환하였다[1][4].

변수 x_j가 $l_j \le x_j \le u_j$의 범위에 있을 경우에는 $x_j - l_j$를 x_j'으로 두면, $0 \le x_j' \le u_j - l_j$가 되어 단순상한문제로 변환할 수 있다. 이 때, 우변상수 $\boldsymbol{b}$의 값은 $\boldsymbol{b}' = \boldsymbol{b} - l_j A_{\cdot j}$와 같이 수정한다.

하한이 $-\infty$인 변수는 새로운 변수 x_j'을 $-x_j + u_j$로 두고 상한을 ∞로 두면 되고, 자유변수의 경우는 $x_j = x_j^+ + x_j^-$로 나누어 x_j^+, x_j^-의 상한을 ∞로 하면 된다. 한편, 이와 같은 변환 에는, 최적해를 구한 다음 원래 변수의 값으로 환원해 주는 과정이 필요하게 된다.

사전처리는 위와 같이 문제를 단순상한문제로 변환한 다음, '문제확인'의 단계를 거친 후, 다음의 과정으로 구성된다.

- 치환에 의한 처리
- 행에 대한 처리
- 열에 대한 처리

2. 문제확인

입력된 문제에 오류가 있는 지를 확인하는 과정이다.

비영요소가 하나도 없는 행을 제거하는데, 이 때 우변상수와 제약식 형태(≤, =, ≥)를 보고 비가능인지를 검사한다. 영역제약식일 경우에는 우변상수뿐만 아니라 좌변상수도 보아야 한다. 예를 들

† 이 논문은 1995년도 한국학술진흥재단의 공모과제 "내부점 선형계획법에서 사전처리와 순서화에 관한 연구" 연구비에 의하여 연구되었음.

어 '≤' 형태의 제약식에 비영요소가 하나도 없으면서 우변상수 값이 0보다 작으면 비가능이다.

각 변수의 상한과 하한을 확인하여 상한이 하한보다 작으면 비가능으로 판정하고 상한과 하한이 같으면 이 변수 값을 하한으로 고정시키고 이 변수에 해당되는 열을 제거한다.

3. 치환에 의한 처리

변수를 2개 가지면서 등식제약식인 제약식은 치환에 의해 제거할 수 있다. 즉, 다음과 같은 행이 있다고 하자.

$$a_{ij_1}x_{j_1} + a_{ij_2}x_{j_2} = b_i$$

x_{j1}은 다음과 같이 x_{j2}의 식으로 표현될 수 있다.

$$x_{j_1} = \frac{b_i}{a_{ij_1}} - \frac{a_{ij_2}}{a_{ij_1}} x_{j_2}$$

이 식을 x_{j1}이 있는 모든 제약식과 목적함수에 대입하게 되면 이 행을 제거할 수 있다.

이 때, 선회연산에 의해 비영요소가 추가[fill_in]되는데, 너무 많은 비영요소의 추가는 촐레스키 팩터의 비영요소 수를 크게 하므로 j_1이나 j_2 열이 밀집(dense)하면 치환을 하지 않게 하였다.

x_{j1}의 값은, 최적해를 구한 다음, 해를 복원할 때 x_{j2}의 값을 이용하여 구한다.

4. 행에 대한 처리

행에 대한 처리로는 동일부호행에 대한 처리와 행의 최대값과 최소값을 이용한 처리가 있다.

동일부호행을 이용한 처리

동일부호행이란 아래와 같은 식에서 행렬계수인 $a_{i\cdot}$ 의 부호가 같은 행을 의미한다.

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n$$

이와 같은 경우에 제약식의 형태와 우변상수의 값에 따라 그 행을 제거할 수 있는데 그 과정은 다음과 같다.

각 행의 비영요소의 갯수를 구한다. 이 때 빈 행이면 제거하는데 다음과 같은 상태이면 비가능이다.

- 제약식 형태가 '≤'이고 우변상수가 음수
- 제약식 형태가 '='이고 우변상수가 비영
- 제약식 형태가 '≥'이고 우변상수가 양수
- 영역제약식이고 좌변상수가 양수거나 우변상수가 음수

빈 행이 아니면 행렬계수의 부호를 조사해서 모두 양인 경우, 즉 동일부호행이면 다음과 같이 처리한다.

제약식의 형태가 '≤', '=' 또는 영역제약식이면서 우변상수가 0이면 이 제약식을 제거하고 해당되는 변수를 하한, 즉 0으로 고정시키고 각 열을 제거하며, 우변상수가 음수이면 비가능 문제가 된다. 그리고 제약식의 형태가 '≥' 이고 우변상수가 음수이면 중복제약식이므로 제거한다. 예를 들어 아래와 같은 제약식은 중복제약식이므로 제거할 수 있다.

$$2x_1 + 3x_2 + 5x_3 \geq -1$$
$$0 \leq x_1 \leq 2,\ 0 \leq x_2 \leq 7,\ 0 \leq x_3 \leq 4$$

행렬계수의 부호가 모두 음인 경우는 위와 비슷하게 처리한다.

행의 최대값과 최소값을 이용한 처리

각 제약식이 가질 수 있는 최대값과 최소값을 구하여 제약식의 형태와 우변상수 값을 비교하여, 제약식을 제거하거나 변수 값을 고정시키는데, 그 과정은 다음과 같다.

각 행의 비영요소의 수를 구한다. 빈 행이면 그 행을 제거하는데 이 때 비가능인지를 확인한다.

비영요소가 하나이면 그 행을 제거한 다음 아래와 같이 b'을 구하여

$$b' = \frac{b_i}{a_{ij}}$$

단, a_{ij} : 비영요소의 행렬계수,
b_i : 우변상수

· $x_j \leq b'$인 경우, 즉 제약식의 형태가 '≤'이고 $a_{ij}>0$ 또는 제약식의 형태가 '≥'이고 $a_{ij}<0$인 경우에는 $b'<0$이면 비가능, $b'=0$이면 변수를 하한으로 고정, $b'<u_j$이면 변수의 상한을 b'로 변경한다.

· $x_j=b'$인 경우, 즉 제약식의 형태가 '='인 경우에는 b'가 하한보다 작거나 상한보다 크면 비가능으로 판정하고 그렇지 않으면 변수 값을 b'로 고정하고 우변상수 값을 이에 따라 변경시킨다.

· $x_j \geq b'$인 경우에는 $b'>u_j$이면 비가능, $b'=u_j$이면 변수 값을 상한에 고정한 후 우변상수를 변경시키고, $b'>l_j(=0)$이면 하한을 b'만큼 증가시킨다.

· 영역제약식에서는 제약식의 영역과 변수의 상·하한을 비교하여 그 변수의 상·하한을 수정시켜준다.

행의 비영요소가 두 개 이상이면 그 행이 가질 수 있는 최대값과 최소값을 구한 다음 다음과 같이 처리한다.

제약식의 형태가 '≤'인 경우에 행의 최대값이 우변상수보다 작거나 같으면 중복제약식이므로 제거하고, 행의 최소값이 우변상수보다 크면 비가능이며 우변상수와 같으면 그 변수들을 행의 최소값이 되도록 고정하고 우변상수를 이에 따라 변경한다. 예를 들어 아래의 제약식의 최대값은 11인데 우변상수가 50이므로 중복제약식으로 판정하고 이를 제거한다.

$$2x_1 - 3x_2 + x_3 \leq 50$$
$$0 \leq x_1 \leq 2,\ 0 \leq x_2 \leq 7,\ 0 \leq x_3 \leq 4$$

제약식의 형태가 '=', '≥'인 경우와 영역제약식일 때에도 위와 비슷하게 처리한다.

5. 열에 대한 처리

각 열의 비영요소 수를 구한다. 이 때 빈 열이면 목적함수 계수의 부호에 따라 목적함수를 값을 크게 하도록 상한이나 하한으로 변수를 고정하고 그 열을 제거한다. 이 경우에는 빈 열이기 때문에 우변상수는 변경하지 않는다.

빈 열이 아니면 다음의 경우에는 할인가의 부호가 일정하므로 변수의 값을 상한이나 하한으로 고정시키고 그 열을 제거한다.

목적함수 계수가 양일 때 제약식의 형태가 '≤'인 행에서는 행렬계수가 양수, '='인 행에서는 행렬계수가 영, '≥'인 행에서는 행렬계수가 음수인 열에 해당하는 변수는 그 값이 상한으로 가더라도 제약조건에 아무런 영향을 주지 못하면서 목적함수 값을 증가시키므로 변수 값을 상한으로 고정하고 이 열을 제거한다. 이 때 우변상수 값도 이에 따라 변경한다.

예를 들어 아래와 같은 문제가 있을 때, x_3을 상한으로 보내면 각 제약 조건을 더 강화시키지 못하면서 목적함수 값을 증가시키므로 x3의 값을 상한으로 고정하고 그에 따라 우변상수를 변경한 다음 세 번째 열을 제거할 수 있다.

$$
\begin{array}{llllllll}
Max & -x_1 & + & 2x_2 & + & 2x_3 & & \\
s.t. & 2x_1 & + & 4x_2 & - & 8x_3 & \leq & 1 \\
 & -x_1 & - & x_2 & - & 2x_3 & \leq & 12 \\
 & 9x_1 & + & 3x_2 & - & x_3 & \leq & 5 \\
\end{array}
$$
$$0 \leq x_1 \leq 2,\ 0 \leq x_2 \leq 7,\ 0 \leq x_3 \leq 4$$

목적함수 계수가 음일 때는 위와 반대가 되므로 제약식의 형태가 '≤'인 행에서는 행렬계수가 음수, '='인 행에서는 행렬계수가 영, '≥'인 행에서는 행렬계수가 양수이면 변수 값을 하한으로 두고 이 열을 제거한다. 이 때는 하한이 0이기 때문에 우변상수는 변화가 없다.

5. 실험결과

사전처리의 효과를 실험하기 위해, 내부점기법을 구현한 프로그램인 LPABO[1]로 20개의 NETLIB 문제를 풀어보았다.

실험에 사용된 문제에 대한 자료는 [표 1]과 같다.

[표 1] NETLIB 문제 자료

문제 이름	행	열	비영요소
25fv47	822	1571	11127
maros	847	1443	10006
czprob	930	3523	14173
pilotja	941	1988	14706
pilotnov	976	2172	13129
scfxm3	991	1371	7846
truss	1001	8806	36642
sctap2	1091	1880	8124
woodw	1099	8405	37478
ship12l	1152	5427	21597
ship12s	1152	2763	10941
sierra	1228	2036	9252
ganges	1310	1681	7021
sctap3	1481	2480	10734
degen3	1504	1818	26230
cycle	1904	2857	21322
stocfor2	2158	2031	9492
80bau3b	2263	9799	29063
bnl2	2325	3489	16124
greenbeb	2393	5405	31499

1) 서울대학교 산업공학과 체계분석실에서 개발한 고속 내부점기법 프로그램이다.

실험 문제에 대해 사전처리를 수행했을 때, 제거되는 제약식과 변수의 갯수와 그 비율은 [표 2]와 같다.

[표 2] 사전처리로 제거되는 행과 열의 갯수

문제 이름	제거행 (비율)		제거열 (비율)	
25fv47	77	(0.09)	59	(0.04)
maros	228	(0.30)	443	(0.31)
czprob	437	(0.47)	950	(0.27)
pilotja	147	(0.16)	435	(0.22)
pilotnov	141	(0.14)	301	(0.14)
scfxm3	153	(0.15)	63	(0.05)
truss	0	(0.00)	0	(0.00)
sctap2	57	(0.05)	0	(0.00)
woodw	387	(0.35)	3050	(0.36)
ship12l	464	(0.40)	1203	(0.22)
ship12s	735	(0.64)	768	(0.28)
sierra	115	(0.09)	120	(0.06)
ganges	332	(0.25)	332	(0.20)
sctap3	72	(0.05)	0	(0.00)
degen3	47	(0.03)	47	(0.03)
cycle	549	(0.29)	501	(0.18)
stocfor2	187	(0.09)	175	(0.09)
80bau3b	241	(0.11)	604	(0.06)
bnl2	652	(0.28)	557	(0.16)
greenbeb	623	(0.26)	1462	(0.27)
평균		0.21		0.15

[표 2]에서 보는 바와 같이 사전처리를 하게 되면 제약식의 21%, 변수의 15%를 줄일 수 있다.

한편, 사전처리를 했을 때의 수행시간의 변화는 [표 3]과 같다. 이 때, 실험 기종은 SUN SPARC ULTRA 170이다.

[표 3] 사전처리 전·후의 수행시간 (단위 : 초)

문제 이름	사전처리 전	사전처리 후	비율
25fv47	29.65	28.66	0.97
maros	21.46	9.68	0.45
czprob	17.56	11.91	0.68
pilotja	84.18	75.32	0.89
pilotnov	70.28	65.63	0.93
scfxm3	7.61	6.78	0.89
truss	37.42	37.16	0.99
sctap2	6.57	5.82	0.88
woodw	162.85	68.61	0.42
ship12l	14.05	9.69	0.69
ship12s	6.06	3.66	0.60
sierra	7.79	7.97	1.02
ganges	13.86	12.58	0.91
sctap3	9.96	8.83	0.89
degen3	191.95	203.61	1.06
cycle	227.28	53.56	0.24
stocfor2	11.88	10.86	0.91
80bau3b	65.96	62.09	0.94
bnl2	146.48	139.89	0.96
greenbeb	129.03	61.95	0.48
평균 비율	·	·	0.79

비율 = (사전처리 후)/(사전처리 전)

각 문제에서 전체 수행시간 중 사전처리 하는데 걸리는 시간은 약 1% 정도이다. 그러므로 전체 수행시간의 1%를 차지하는 사전처리를 수행함으로써 전체 수행시간을 21% 줄일 수 있다.

한편, 비가능문제를 해법을 수행하지 않고 사전처리를 통해 알 수도 있다. 이를 실험하기 위해, 비가능 NETLIB 문제 30개에 대해서 실험한 결과는 [표 4]와 같다.

[표 4] 비가능 NETLIB문제에 대한 실험 결과

문제 이름	비가능성 판정 여부	문제 이름	비가능성 판정 여부
bgdbd1	○	greenbea	○
bgetam	X	itest2	○
bgindy	X	itest6	○
bgprtr	X	klein1	X
box1	X	klein2	X
ceria3d	X	klein3	X
chemcom	X	mondou2	○
cplex1	X	pang	X
cplex2	X	pilot4i	○
ex72a	X	qual	X
ex73a	X	reactor	○
forest6	X	refinery	X
galenet	○	vol1	X
gosh	X	vol2	X
gran	○	woodinfe	○

[표 3]에서 보는 바와 같이, 사전처리를 통해 해법을 수행하지 않고 비가능성을 판정할 수도 있기 때문에 수행시간을 많이 줄일 수 있다.

6. 결론

대형선형계획문제에 적용할 수 있는 사전처리 프로시저를 개발하였다. NETLIB에 있는 20개의 대형 문제에 대해 사전처리를 수행하여, 제약식의 21%, 변수의 15%를 제거할 수 있었다. 이 때, 사전처리를 한 후, 내부점기법 프로그램으로 문제를 수행한 결과 21%의 수행시간 감소의 효과가 있었는데, 사전처리에 소요되는 시간은 전체 수행시간의 1% 정도이다.

한편, 비가능 NETLIB문제 30개 중에서 10개를 사전처리로 비가능을 판정할 수 있었다.

7. 참고문헌

[1] 박순달, 「선형계획법」, 제3판, 민음사, 1992

[2] 진희채, "선형계획법을 위한 아핀법과 장벽법의 통합화 및 알고리듬의 효율화", 서울대학교 산업공학과, 박사학위논문, 1995

[3] Brearley, A. L., G. Mitran, H. P. Williams, *Analysis of Mathematical Progamming Problems Prior to Applying the Simplex Algorithm*, Mathematical Programming 8, 1975

[4] Lustig, I. J., R. E. Marsten, D. F. Shanno, *Interior Point Methods for Linear Programming: Computational State of the Art*, ORSA Journal on Computing, Vol. 6, No. 1, 1994

[5] Savelsbergh, M. W. P., *Preprocessing and Probing Techniques for Mixed Integer Programming Problems*, ORSA Journal on Computing, Vol. 6, No. 4, 1994

[6] Fang, Shu-Cherng, Sarat Puthenpura, *Linear Optimization and Extensions: Theory and Algorithms*, Prentice-Hall, Inc., 1993