원쌍대 내부점기법에서 초기해 선정과 중심화 힘을 이용한 개선 방향의 수정

Finding an Initial Solution and Modifying Search Direction by the Centering Force in the Primal-Dual Interior Point Mehod

성명기*, 박순달*

* 서울대학교 산업공학과

Abstract

This paper deals with finding an initial solution and modifying search direction by the centrering force in the predictor-corrector method which is a variant of the primal-dual barrier method. These methods were tested with NETLIB problems.

Initial solutions which are located close to the center of the feasible set lower the number of iterations, as they enlarge the step length. Three heuristic methods to find such initial solution are suggested. The new methods reduce the average number of iterations by 52% to at most, compared with the old method assigning 1 to initial values,

Solutions can move closer to the central path fast by enlarging the centering force in early steps. It enlarges the step length, so reduces the number of iterations. The more effective this method is the closer the initial solution is to the boundary of the feasible set.

1. 서론

1984년 N. Karmarkar가 사영법(projective method)을 발표하면서부터 본격적으로 발전하기 시작한 내부점기법(interior point method)은, 해집합을 나타내는 단체의 정점을 찾아가면서 최적해를 구하는 단체법(simplex method)과 달리 해집합의 내부 영역에서 최적해를 구하는 방법으로 단체법에 비해 대형 선형계획문제에 유리하고 이론적 계산 복잡도도 우수하다는 장점이 있다[2][3]. 실제로 여러 대형 문제를 수행해 본 연구 결과에서도 문제의 규모가 커져 갈수록 내부점기법의 수행 성능이 우수하게 나타나고 있다[8][10].

내부점기법으로 선형계획문제를 풀 때 개선 방향(search direction)을 구하는데 소요되는 시간이 전체 계산 시간의 70~90%를 차지하는데, 특히 다음과 같은 선형방정식을 풀 때 가장 많은 시간이 걸린다[5].

$$(A\Theta A^T)\mathbf{y} = \mathbf{w}$$

위의 방정식을 풀 때 $A\Theta A^T$의 역행렬을 구하는 것보다는 $A\Theta A^T$가 대칭양정치(symmetric positive definite) 행렬이라는 성질을 이용하여

$$A\Theta A^T = LL^T$$

와 같이 촐레스키분해(Cholesky decomposition)하여 전방치환(forward substitution), 후방치환(backward substitution)으로 푸는 방법이 효율적이어서 많이 사용되고 있다[4].

효율적인 내부점 선형계획 프로그램을 개발하기 위해서는 이 연산 시간을 줄일 필요가 있는데 이에는 크게 다음의 두 가지 방법이 있다.

- 촐레스키분해를 빠르게 하여 한 회에서의 계산 시간을 줄이는 방법
- 반복수(number of iterations)를 줄여서 촐레스키 분해의 횟수를 줄이는 방법

첫 번째 방법에 대한 연구로는 촐레스키 팩터 L의 비영요소의 수를 줄이기 위해 $A\Theta A^T$의 열을 재배치하는 순서화 방법(ordering method)과[1], 컴퓨터 메모리에서의 간접 어드레싱(indirect addressing)을 줄이기 위해서 삭제나무(elimination tree)를 이용한 수퍼노드(supernode) 방법[9], 밴드(band)의 폭을 줄여서 직접 어드레싱(direct addressing)과 병렬 처리를 이용하는 멀티프론털 방법(multi-frontal method) 등이 있다[4].

두 번째 방법에 관한 연구는 좋은 초기해(initial solution)와 개선 방향(search direction)을 구하는 방법 등이 있다[6][7]. 본 논문은 두 번째 방법, 즉 반복수를 줄이는 방법에 대해 다루고 있다.

초기해가 가능해 영역의 중심에 있으면 개선폭이 커지므로 반복수를 줄일 수 있다. 본 논문에서는 이러한 초기해를 구하는 휴리스틱(heuristic) 방법을 제시하였다.

원쌍대 내부점기법(primal-dual interior point method)은 개선되는 해들이 중심궤적(central path)을 따라 갈 때 가장 작은 반복수로 최적해를 구할 수 있다. 그런데 초기해가 가능해 영역의 경계에 가까이 있을 경우에는 중심궤적에서 멀리 떨어진 상태에서 해를 개선해 나가게 되므로 반복수가 많아지게 된다. 본 논문에서는 이러한 경우에 보다 빨리 해가 중심궤적에 가까이 가도록 하기 위해서, 개선 방향에서 중심화 힘(centering force)을 크게 하는 방법을 사용하여 반복수를 많이 줄일 수 있었다.

한편, 본 논문에서 사용한 원쌍대 내부점기법은 현재까지 알려진 방법 중에서는 가장 효율적이라고 알려진 S. Mehrotra의 예측자-수정자법(predictor-corrector method)이다[10].

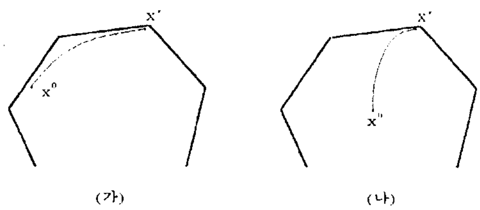
2. 초기해의 선정

반복수가 많은 문제들의 해법 진행 과정을 살펴보면 해의 개선 속도가 매우 느린 것을 알 수 있는데 이들 중 많은 경우가 초기해의 값이 최적해에 비해

상대적으로 작아서 개선폭이 작게 되는 것이 원인이다. 즉, 가능해 영역의 경계 부근에서 출발하기 때문에 최적해로 가는데 많은 반복수를 필요로 하게 된다. 이러한 문제는 초기해를 가능해 영역의 중심에 있도록 하면 해결할 수 있다.

내부점기법은 초기해가 가능해 영역의 중심에 있을 때 가장 효과적인데 가능해 영역의 중심을 구하는 것은 어렵기 때문에 보통 휴리스틱 방법을 사용해서 초기해를 구한다[11].

[그림2.1]은 초기해의 위치에 따른 최적해를 찾는 과정은 보여 주고 있다. 이 그림에서 보듯이 초기해가 가능해 영역의 경계에 있는 (가)보다 상대적으로 중심에 있는 (나)가 최적해를 빨리 찾을 수 있다.



[그림2.1] 초기해의 위치

원쌍대 내부점기법은 대수법을 사용하여 초기해를 구하는데, 일반한계 원문제와 쌍대문제에 대수법을 적용하면 다음의 (P), (D)와 같이 된다.

$$
(P)\quad
\begin{array}{llr}
\text{Max} & \mathbf{c}^T\mathbf{x} - Mx_{n+1} & \\
\text{s.t.} & A\mathbf{x} + \mathbf{b}^0 x_{n+1} & = \mathbf{b} \\
& (\mathbf{c}^0)^T\mathbf{x} \qquad + x_{n+2} & = M \\
& \mathbf{x} \qquad\qquad + \mathbf{v} & = \mathbf{u} \\
& \mathbf{x} \ge \mathbf{0},\ x_{n+1} \ge 0,\ x_{n+2} \ge 0,\ \mathbf{v} \ge \mathbf{0} &
\end{array}
$$

$$
(D)\quad
\begin{array}{llr}
\text{Min} & \mathbf{b}^T\mathbf{y} + My_{m+1} + \mathbf{u}^T\mathbf{w} & \\
\text{s.t.} & A^T\mathbf{y} + \mathbf{c}^0 y_{m+1} + \mathbf{w} - \mathbf{z} & = \mathbf{c} \\
& (\mathbf{b}^0)^T\mathbf{y} \qquad - z_{n+1} & = -M \\
& y_{m+1} \qquad - z_{n+2} & = 0 \\
& \mathbf{w} \ge \mathbf{0},\ \mathbf{z} \ge \mathbf{0},\ z_{n+1} \ge 0,\ z_{n+2} \ge 0 &
\end{array}
$$

단, $\mathbf{b}^0 = \mathbf{b} - A\mathbf{e}$, $\mathbf{c}^0 = \mathbf{c}$, M은 큰 수

이 문제에서 초기해로 정해야 하는 변수는 $\mathbf{x}$, $\mathbf{v}$, x_{n+1}, $\mathbf{y}$, y_{m+1}, $\mathbf{w}$, $\mathbf{z}$, z_{n+2} 등이다. M은 매우 큰 값인데 구현할 때 이 값을 무한대로 하였기 때문에 x_{n+2}, z_{n+1}의 값은 무한대가 되고 x_{n+1}, y_{m+1}의 값은 1로 주었다.

아래의 세 가지 가정에 기초한 각각의 방법으로 초기해를 구해서 이들 방법의 효율을 비교해 보기로 한다.

방법 1

이 방법은 다음을 가정하고 초기해를 구하는 방법이다.

- ▶ 변수에 상한이 있을 경우에는 초기해를 상한의 절반으로 하는 것이 좋다.
- ▶ 상한이 없는 변수에 대해서는 상한이 있는 다른 변수에 비해서 너무 작거나 큰 값을 가지면 좋지 않다.

위와 같이 가정을 정한 이유는, 변수가 가질 수 있는 값의 범위는 0에서 상한까지인데 최적해가 어디에 있는지 모르기 때문에 초기값으로 중점을 두는 것이 평균적으로 가장 좋은 예측이 되고, 상한이 없을 경우에 상대적으로 너무 크거나 작은 값으로 초기해를 주면 개선폭을 작게 하여 해의 수렴을 느리게 할 수 있기 때문이다.

$\mathbf{x}$의 초기해는 다음과 같이 정하였다. 여기서 $\bar{u}$는 상한이 있는 변수들의 평균이다.

$$
\begin{aligned}
x_j^0 &= u_j \times 0.5 \quad, \text{ 상한이 있는 경우} \\
&= \bar{u} \quad\quad\quad\ , \text{ 상한이 없는 경우}
\end{aligned}
$$

모든 변수에 상한이 없는 경우에는 다음과 같이 일정한 값 K를 주기로 하였다.

$$x_j^0 = K,\ \text{모든 변수에 상한이 없는 경우}$$

쌍대문제의 변수인 $\mathbf{y}$, $\mathbf{w}$, $\mathbf{z}$의 초기해는 다음과 같이 정했다.

$\mathbf{y}$는 제약이 없기 때문에 초기해를 $\mathbf{0}$으로 정하는 것이 좋고, $\mathbf{w}$, $\mathbf{z}$는 상한이 없기 때문에 초기해로 일정한 값을 준다. 이 때 $\mathbf{y}^0$를 $\mathbf{0}$으로 두면 초기해를 정할 때 계산상의 이득도 생기게 된다. 즉, (D)에서 초기해를 구한 다음 아래 식에서 $\mathbf{c}^0$를 계산해야 하는데 $\mathbf{y}^0$가 0이면 이 계산이 간단해지기 때문이다.

$$A^T\mathbf{y} + \mathbf{c}^0 y_{m+1} + \mathbf{w} - \mathbf{z} = \mathbf{c}$$

$\mathbf{w}^0$, $\mathbf{z}^0$의 값으로는 $\mathbf{x}$에 상한이 있는 경우에는 $\bar{u} \times 0.1$로 하고 그렇지 않은 경우에는 K의 10분의 1로 하였다. 즉,

$$
\begin{aligned}
\mathbf{y}^0 \quad &= \mathbf{0} \\
\mathbf{w}^0,\ \mathbf{z}^0 &= \bar{u} \times 0.1,\ \ \mathbf{x}\text{에 상한이 있는 경우} \\
&= K \times 0.1,\ \ \mathbf{x}\text{에 상한이 없는 경우}
\end{aligned}
$$

$\mathbf{w}^0$의 값을 $\bar{u}$나 K로 하지 않고 이 값의 10분의 1로 한 이유는 쌍대목적함수 값이 너무 큰 값이 되지 않게 하기 위해서이다. 즉, 쌍대목적함수는

$$\text{Min}\ A^T\mathbf{y} + \mathbf{u}^T\mathbf{w}$$

와 같은데 보통 $\mathbf{u}$는 큰 값이므로 $\mathbf{w}^0$의 값을 $\mathbf{x}^0$와 같은 값을 주게 되면 쌍대목적함수 값이 너무 커져서 해의 수렴성이 좋아지지 않기 때문에 $\mathbf{x}^0$보다 작은 값을 주었다.

방법 2

▶ 초기 목적함수 값은 원문제(P)에서는 작게, 쌍대문제(D)는 크게 하는 것이 좋다.

위와 같이 가정하고 초기해를 구했는데 이 이유는, 초기해가 (P), (D)의 가능해가 아니더라도 최적값과 비교하여 원목적함수 값은 작고, 쌍대목적함수 값은 크다면 해를 개선하는데 유리할 것이기 때문이다.

목적함수 값이 원문제에서는 작게, 쌍대문제에서는 크게 되게 하기 위하여 초기해의 값을 목적함수 계수의 부호에 따라 다음과 같이 정했다.

$$
\begin{aligned}
x_j &= \text{Big}, & \text{if } c_j \le 0 \\
&= \text{Small}, & \text{if } c_j > 0 \\
y_i &= \text{양수}, & \text{if } b_i > 0 \\
&= \text{음수}, & \text{if } b_i \le 0
\end{aligned}
$$

x_j의 값은 상한이 있을 때에는 Big, Small을 각

각 uj×0.8, uj×0.2로 두고, 상한이 없을 경우에는 Big을 임의의 값으로 하고 Small은 Big×0.1로 했다.

y_i의 값은 Big으로 했고, w_j, z_j는 Big×0.1로 하였다.

방법 3

원비가능성과 쌍대비가능성이 작은 초기해를 구하게 되면 최적해에 가까이 있을 가능성이 많아서 적은 반복수로 최적해에 이를 수 있을 것이기 때문에 아래와 같은 가정으로 초기해를 정했다.

▶ 원가능성과 쌍대가능성을 가능한 만족하게 초기해를 잡는 것이 좋다.

초기해가 원가능성과 쌍대가능성을 최대한 유지하도록 초기해를 구하는 방법이다.

$\mathbf{x} = p \times \mathbf{e}$라고 가정하였다. 이러한 해가 원가능성이 최대가 되게 하기 위해서는

$$\min \ \| A\mathbf{x} - \mathbf{b} \|$$

가 되는 p를 구하면 된다.

위 문제에 라그랑지 완화 함수(Lagrangean relaxation funtion)를 사용하여 구한 해는 다음과 같다.

$$p^* = \frac{\mathbf{b}^T A \mathbf{e}}{\| A\mathbf{e} \|^2}$$

x는 상한을 넘을 수 없으므로 초기해는 다음과 같이 된다.

$$\begin{aligned} x_j^0 &= p^*, && \text{if } u_j > p^* \\ &= u_j \times 0.5, && \text{if } u_j \le p^* \end{aligned}$$

쌍대변수의 초기해는 $\mathbf{y}^0$를 0으로 두었고 **z**의 값을 조절하여 쌍대가능을 만족하게 하였다.

$\mathbf{y}^0$가 0이면 쌍대제약식은 다음과 같다.

$$\mathbf{w} - \mathbf{z} = \mathbf{c}$$

w와 **z**는 상한이 없으므로 **c**의 값에 따라 다음과 같이 값을 주었다.

$$\begin{aligned} z_j^0 &= p^*, \ w_j^0 = c_j + p^*, && \text{if } c_j > -p^* \\ w_j &= p^*, \ z_j^0 = p^* - c_j, && \text{if } c_j \le -p^* \end{aligned}$$

각 변수의 값이 너무 작으면 수렴성이 좋지 않기 때문에 p^*의 값이 P보다 작으면 p^*를 P로 하였다.

세 가지 방법에서 결정해야 할 상수 값은 각각 K, Big, P 등이다. NETLIB 문제 중 60개에 대해 이들의 값을 변화시키면서 평균 반복수를 구한 결과는 [표2.1]과 같다.

[표 2.1] 각 방법에서의 평균 반복수

각 방법의 상수 값	방법 1 (K)	방법 2 (Big)	방법 3 (P)
1	44.2	50.4	27.4
10	30.3	46.2	24.6
50	23.7	31.4	23.2
100	23.0	28.1	23.0
500	22.5	24.2	22.8
1000	23.0	23.4	23.7
10000	24.8	24.3	24.8

세 가지 방법에 대한 실험 결과를 분석해 보면 다음의 결과를 알 수 있다.

- 세 가지 방법 모두 상수 값이 어느 범위에 있을 때 비슷한 평균 반복수를 가진다.
- 평균 반복수가 가장 좋은 경우만을 비교하면 방법 1, 3, 2의 순서이다.
- 상수 값이 많이 변하더라도 비교적 일정한 평균 반복수를 가지는 것은 방법 3이다.

방법 3을 방법 1과 비교해 보면 평균 반복수가 가장 작은 경우에 그 값은 비슷한데 편차가 작기 때문에 방법 3이 우월함을 알 수 있다.

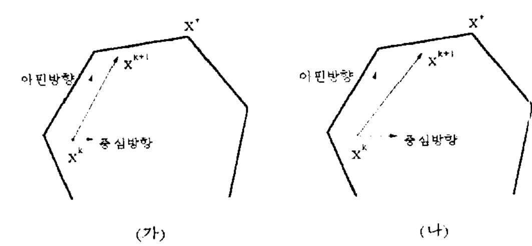
3. 중심화 힘의 변화

장벽법의 개선 방향은 목적함수 값을 개선시키는 아핀 방향과 해를 가능해 영역의 경계에서 떨어지게 하는 중심 방향으로 나뉘어진다[6]. 개선 방향에 중심 방향을 고려한 이유는 해를 경계에서 멀어지게 하여 수치안정성(numerical stability)을 높이고 개선폭을 크게 하여 반복수를 줄이기 위함이다[6]. 이 때 중심화 힘의 크기는 장벽매개변수 μ로 조절된다. 즉, μ를 크게 하면 중심화 힘이 커지게 된다[2].

예측자-수정자법도 원쌍대장벽법과 비슷한 방법으로 개선 방향을 구하기 때문에 개선 방향은 아핀 방향과 중심 방향으로 나뉘어진다.

내부점기법에서는 매회 생성되는 해가 중심궤적에 있을 경우에 개선폭이 가장 크게 되어 최소의 반복수로 최적해를 구할 수 있다[6]. 그래서 초기해가 중심궤적과 멀리 떨어져 있을 경우에는 반복수가 많아지게 된다.

[그림3.1]은 k회의 해가 경계에 붙어 있는 경우에 (나)와 같이 중심화 힘을 더 크게 해 주면 해의 개선폭이 커지게 되는 것을 보여 주고 있다.



[그림3.1] 중심화 힘과 개선 방향

본 논문에서는 초기해가 중심궤적과 멀리 떨어져 있을 경우에 반복수가 많아지므로 해법의 초기 단계에서 중심화 힘을 원래보다 크게 하여 매회 생성되는 해가 중심궤적에 가까이 가도록 하여 반복수를 줄이고자 한다.

앞에서 설명한 바와 같이 예측자-수정자법을 구현할 때는 초기해를 바로 구하지 못하기 때문에 대수법을 사용해서 초기해를 구한다. 이 때 인공변수를 사용하게 되는데 이를 이용하여 중심화 힘을 변화시킬 수 있다.

원래의 예측자-수정자법에서 구한 k회의 μ, 즉 μ^k를

$$\mu^k = \mu^k + \alpha^k$$

단, α^k는 최적해에 가까울수록 감소한다.

와 같이 수정하여 중심화 힘을 크게 하여 k+1회의 해가 중심궤적에 가까이 가도록 한다면 보다 적은 반복수로 최적해를 구할 수 있을 것이다. 이 때 k회의 해가 최적해에 충분히 가까울 때 α^k가 상대적으로 크다면 해가 수렴하지 않을 수도 있기 때문에 α^k는 0으로 수렴해야 한다.

대수법에서는 인공변수를 사용하게 되는데 이 값은 매회 단조 감소하여 최적해에서 0이 되는 성질이 있으므로 인공변수의 값을 α^k로 사용할 수 있다. 즉, 현재의 해가 최적해에 가까이 있는 정도를 알려주는 지시자(indicator)로 원문제의 인공변수 x_{n+1}을 사용해서 μ를 다음과 같이 둔다.

$$\mu^k = \mu^k (1 + c \times x_{n+1})$$

이와 같이 하였을 때 원래보다 반복수가 최대로 줄어드는 상수 c가 존재한다는 것을 실험으로 보이고자 한다.

초기해가 가능해 영역의 경계에 있을 때와 상대적으로 가능해 영역의 중심에 있을 경우에 이 방법의 효과를 알아보기 위해서 다음 두 가지 초기해에 대해서 실험하였다.

방법 1

초기해가 가능해 영역의 경계에 붙어 있는 경우로 다음과 같이 초기해를 주었다.

$$\begin{aligned} x_j^0 &= 1, \quad \text{if } u_j > 1 \\ &= u_j \times 0.5, \quad \text{if } u_j \le 1 \\ \mathbf{y}^0 &= \mathbf{0} \\ \mathbf{w}^0 &= \mathbf{e} \\ \mathbf{z}^0 &= \mathbf{e} \end{aligned}$$

방법 2

초기해가 상대적으로 가능해 영역의 중심에 있는 경우로 '2. 초기해의 변화'에서 '방법 1'중 K가 500인 경우이다.

두 가지 초기해에 대해 중심화 힘의 크기를 조절하는 c의 값을 변화시키면서 구한 평균 반복수는 [표 3.1]과 같다.

[표3.1] 중심화 힘의 변화에 따른 평균 반복수

c의 값	0	1	2	3	4
방법 1	47.0	39.6	34.5	31.2	29.1
방법 2	22.5	22.2	22.3	22.3	22.4
c의 값	5	10	15	20	50
방법 1	27.5	25.5	25.6	26.1	31.5
방법 2	22.5	23.0	23.2	24.2	25.6

[표3.1]을 보면 중심화 힘의 변화 방법은 초기해가 좋지 않은 방법 1에서는 효과가 크나 상대적으로 좋은 초기해인 방법 2에서는 효과가 작음을 알 수 있다.

4. 결론

초기해가 가능한 가능해 영역의 중심에 있도록 하는 것이 반복수를 줄일 수 있어 좋지만 이를 찾기 어렵기 때문에 세가지 휴리스틱 방법을 제시했는데 비가능성을 최소화하도록 초기해를 구하는 방법이 가장 좋은 결과를 내는 것으로 나타났다.

초기해가 중심궤적과 멀이 떨어져 있을 경우에 개선되는 해가 중심궤적에 가까이 가도록 중심화 힘을 원래보다 크게 해 주면 반복수를 줄일 수 있는데 초기해가 가능해 영역의 경계에 가까이 있을수록 효과가 컸다.

5. 참고 문헌

[1] 모정훈, "내부점 선형계획법에서의 순서화방법과 자료구조에 관한 연구", 서울대학교, 석사학위논문(1995)

[2] 박순달, 「선형계획법」, 제3판, 민영사, 1992

[3] 진희채, "선형계획법을 위한 아핀법과 장벽법의 통합화 및 알고리듬의 효율화", 서울대학교, 박사학위논문(1995)

[4] George, A., J. W. H. Liu, *Computer Solution of Large Sparse Positive Definite Systems*, Prentice-Hall, Inc., 1981

[5] Gondzio, J., "Multiple centrality corrections in a primal-dual method for linear programming", Technical Report 1994.20 (1995), Section of Management Studies, University of Geneva, Switzerland

[6] Gonzaga, C. C., "Path-following methods for linear programming", *SIAM Review* Vol. 34, No. 2(1992), pp.167-224

[7] Hertog, D. den, C. Roos, "A survey of search directions in interior point methods for linear programming", *Mathematical Programming* 52(1991), pp.481-509

[8] Karmarkar, N. K., K. G. Ramakrishnan, "Computational results of an interior point algorithm for large scale linear programming", *Mathematical Programming* 52(1991), pp.555-586

[9] Liu, J. W. H., "The role of elimination trees in sparse factorization", *SIAM J. Math. Anal. Appl.* vol 11, No 1(1990) , pp.134-172

[10] 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)

[11] Mcshane, K. A., C. L. Monma, D. Shanno, "An implementation of a primal-dual interior point method for linear programming", *ORSA Journal on Computing*, Vol. 1, No. 2(1989), pp. 70-83