
공학 수치해석
Numerical Analysis for Engineers

박은천
단국대학교 건축대학

LECTURE NOTE
NOTE-2012-ver-20121113
13 November 2012

EUNCHURN PARK
(eunchurn@kmctech.co.kr)
Structrual Analysis & Dynamics Laboratory
Department of Architectural Engineering
Dankook University School of Architecture

Department of R&D Development
Korea Maintenance Co., LTD.
KM Tower 12F
130-5, Guro-5-dong, Gurogu, Seoul (152-842)
PHN: +82-2-830-7071 (236)
SEL: +82-10-4499-6420
FAX: +82-2-830-5256

공학 수치해석

Numerical Analysis for Engineers

박은천
단국대학교 건축대학

Dept. of Architectural Engineering, Dankook Univ. School of Architecture, NOTE-2012-ver-20121113

13 November 2012

Table of Contents

I 최적화		1.3 선형계획모형 해법	6
(Optimization)	2	1.4 패키지를 사용한 최적화	
		(MATLAB Optimization Toolbox	
		Functions)	9
1 구속 최적화		1.4.1 Linear programming	
(Constrained Optimization)	2	lingprog()	9
1.1 등식 구속조건			
Equality constraints	2		
1.1.1 Lagrangian Methods	2		
1.2 등식과 부등식 구속조건 (Equality and		II	11
Inequality Constraints)	4	Appendices	11

Part I

최적화

(Optimization)

1 구속 최적화

(Constrained Optimization)

목적함수와 구속조건이 선형인 문제를 이장에서 논의 한다. 이 경우 주어진 함수의 선형성을 이용하는 특별한 방법들이 존재한다. 선형프로그래밍(linear programming)법이라고 불리는 알고리즘은 수천 개의 변수와 구속조건을 갖는 대형 문제를 매우 효율적으로 계산한다. 이 방법은 공학과 경영 분야에 걸쳐 많은 문제들에 적용된다. 이후 비선형 구속 최적화의 일반적인 문제들을 간단히 다룬 후에 MATLAB 라이브러리의 사용법에 대해 개략적으로 설명한다. 선형프로그래밍(LP)은 제한된 자원 같은 구속조건을 갖는 경우, 이윤을 최대화하거나 비용을 최소화하여 원하는 목적을 만족시키는 최적화 문제를 다룬다. 선형이라는 용어는 목적함수나 구속조건을 표현하는 수학적 함수들이 모두 선형(linear)임을 나타낸다. 프로그래밍이라는 용어는 "컴퓨터 프로그래밍"을 의미하기보다는 "스케줄"을 정하거나 과제목록을 정하는 의미를 함축하고 있다.

1.1 등식 구속조건

Equality constraints

1.1.1 Lagrangian Methods

일반적인 최적화 문제를 가정하자.

$$\begin{aligned} P : & \underset{x}{\text{maximize}} && f(x) \\ & \text{subject to} && g(x) = b, x \in X, \end{aligned}$$

여기서, $x \in \mathbb{R}^n$, $b \in \mathbb{R}^m$ (n 개의 변수(variables)와 m 개의 구속조건(constraints)). 라그랑지안(Lagrangian)은

$$L(x, \lambda) = f(x) + \lambda^T (b - g(x)) \quad (1.1)$$

여기서, $\lambda \in \mathbb{R}^m$ (각 구속조건의 한개의 요소). λ 의 각 요소를 라그랑지안 승수(Lagrangian multiplier)라고 한다.

다음 일반적인 문제를 가정하자.

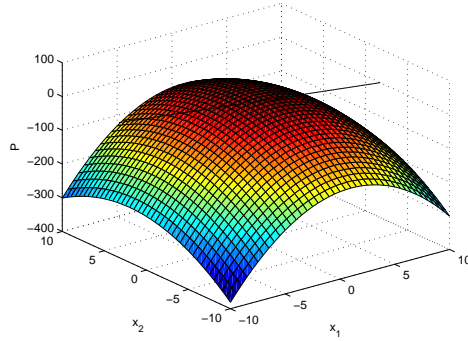
$$\begin{aligned} & \underset{x_1, x_2}{\text{maximize}} && 5 - (x_1 - 2)^2 - 2(x_2 - 1)^2 \\ & \text{subject to} && x_1 + 4x_2 = 3 \end{aligned}$$

만약 구속조건을 무시한다면, 우리는 이문제의 해를 쉽게 풀 수 있다. $x_1 = 2$, $x_2 = 1$ 그렇지만 구속조건에는 큰값으로 만족하지 않는다. 구속조건의 값을 크게 만드는데 있어 λ 를 도입하여 손실가중치를 정의하자. 위의 함수를 다음과 같이 쓸 수 있다.

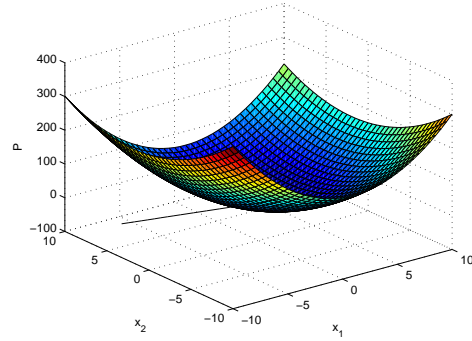
$$L(x_1, x_2, \lambda) = 5 - (x_1 - 2)^2 - 2(x_2 - 1)^2 + \lambda(3 - x_1 - 4x_2) \quad (1.2)$$

이러한 함수를 문제의 라그랑지안(Lagrangian of the problem)이라고 한다. 결국 식의 오른쪽에 구속조건의 값이 추가된 λ 값을 조정하는 문제로 귀결된다. λ 의 값에 따른 답을 찾아보자.

- $\lambda = 0$ 일 때, $(x_1, x_2) = (2, 1)$.
- $\lambda = 1$ 일 때, $(x_1, x_2) = (3/2, 0)$.
- $\lambda = 2/3$ 일 때, $(x_1, x_2) = (5/3, 1/3)$. (구속조건을 정확히 만족시킴)



(a) Concave problem



(b) Convex problem

Figure 1.1: Convex and concave problem

여러개의 구속조건을 갖는 최적화 방식을 일반화 시키면

$$\begin{aligned}
 P : & \underset{x}{\text{maximize}} && f(x) \\
 & \text{subject to} && g_1(x) = b_1 \\
 & && g_2(x) = b_1 \\
 & && \vdots \\
 & && g_m(x) = b_m
 \end{aligned}$$

라그랑지안(Lagrangian)을 사용한 해는 다음 식으로 찾을 수 있다.

$$L(x, \lambda) = f(x) + \sum_{i=1}^m \lambda_i (b_i - g_i(x)) \quad (1.3)$$

여기서, λ_i 는 i 번째 구속조건에 관련한 가중치(가격)으로 생각할 수 있다.

Theorem 1.1 (Lagrangian multiplier). 최적화 문제 P 를 만족시키는 x^* 에 대하여 x^* 와 λ^* 가 존재 하고, $x^* = (x_1^*, x_2^*, \dots, x_n^*)$ 를 함수 $f(x)$ 를 구속조건 $g_i(x) = b_i$, for $i = 1, 2, \dots, m$ 을 만족시키면서 최대화(maximizes) 시키거나 최소화(minimizes) 시키는 값을 가지는 벡터로 가정하면,

- 벡터 $\nabla g_1(x^*), \nabla g_2(x^*), \dots, \nabla g_m(x^*)$ 가 모두 선형적으로 독립이거나,
- $\nabla L(x^*, \lambda^*) = 0$ 을 만족시키는 벡터 $\lambda^* = (\lambda_1^*, \lambda_2^*, \dots, \lambda_m^*)$ 가 존재한다.

여기서 ∇ 연산자(Nabla-operator)는 벡터의 편미분 연산자로 다음 식처럼 쓰인다.

$$\nabla = \left(\frac{\partial}{\partial x_1}, \dots, \frac{\partial}{\partial x_n} \right) \quad (1.4)$$

즉, 위의 라그랑지안 정리는 다음식처럼 쓸 수 있다.

$$\frac{\partial L}{\partial x_1}(x^*, \lambda^*) = \frac{\partial L}{\partial x_2}(x^*, \lambda^*) = \dots = \frac{\partial L}{\partial x_n}(x^*, \lambda^*) = 0 \quad (1.5)$$

$$\frac{\partial L}{\partial \lambda_1}(x^*, \lambda^*) = \frac{\partial L}{\partial \lambda_2}(x^*, \lambda^*) = \dots = \frac{\partial L}{\partial \lambda_m}(x^*, \lambda^*) = 0 \quad (1.6)$$

1.2 등식과 부등식 구속조건 (Equality and Inequality Constraints)

등식 구속조건과 부등식 구속조건이 포함된 함수의 최적화는 어떻게 두가지의 조건들을 사용할 수 있느냐에 달려있다. 일반화된 다음 식을 보자.

$$\begin{aligned} P : & \underset{x}{\text{maximize}} && f(x) \\ & \text{subject to} && g_1(x) = b_1 \\ & && \vdots \\ & && g_m(x) = b_m \\ & && h_1(x) \leq d_1 \\ & && \vdots \\ & && h_p(x) \leq d_p \end{aligned}$$

만약 이 조건이 아닌 \geq 조건의 경우 \leq 를 만들어주기 위해 양변에 -1 을 곱하여 위와 동일한 조건을 만들어준다. 또한 같은 방식으로 최소값 문제는 최대값 문제로 변형할 수 있다. 이 때 라그랑지안은

$$L(x, \lambda, \mu) = f(x) + \sum_{i=1}^m \lambda_i (b_i - g_i(x)) + \sum_{j=1}^p \mu_j (d_j - h_j(x)) \quad (1.7)$$

정리 1.1와 유사하게 다음과 같이 정리한다. $x^* = (x_1^*, x_2^*, \dots, x_n^*)$ 를 함수 $f(x)$ 를 구속조건 $g_i(x) = b_i$, for $i = 1, 2, \dots, m$ 과 $h_j(x) \leq d_j$, for $j = 1, 2, \dots, p$ 을 만족시키면서 최대화(maximizes)시키는 값을 가지는 벡터로 가정하면,

(i) 벡터 $\nabla g_1(x^*), \dots, \nabla g_m(x^*)$ 와 $\nabla h_1(x^*), \dots, \nabla h_p(x^*)$ 가 모두 선형적으로 독립이거나,

(ii) 아래의 식을 만족시키는 벡터 $\lambda^* = (\lambda_1^*, \dots, \lambda_m^*)$ 와 $\mu^* = (\mu_1^*, \dots, \mu_p^*)$ 가 존재한다.

$$\begin{aligned} \nabla f(x^*) - \sum_{i=1}^m \lambda_i^* \nabla g_i(x^*) - \sum_{j=1}^p \mu_j^* \nabla h_j(x^*) &= 0 \\ \mu_j^* (d_j - h_j(x^*)) &= 0 \text{ (Complementarity)} \\ \mu_j^* &\geq 0 \end{aligned}$$

일반적으로 부등식조건이 포함된 해를 구하기 위해서는 우리는 μ_j^* 가 0이 되어야하는 조건과 $d_j - h_j(x^*) = 0$ 이 되는 상보성(complementarity)조건을 검토해야한다. 이 조건에 만족하는 여러가지 경우의 수에 근거하여 우리는 적절한 하나의 해 혹은 후보해들을 선택해야한다. 만약 최적의 해가 존재한다면, 하나의 해 아니면 후보해들이 그 답이 될 것이다. 위의

조건들은 Kuhn-Tucker(혹은 Karush-Kuhn-Tucker)조건이라고 말한다. KKT조건에서 말하는 것은 최적의 해 x^* 는 부등식 조건에서 꼭 찬 조건이거나 그렇지 않은 경우의 수가 있다. 즉, 부등식 조건에 꼭 찬 조건이 아닌 경우는 무시할 수 있다(해당 경우엔 $\mu_j^* = 0$ 이 되므로). 그러나 부등식 조건에 꼭 찬 경우는 Lagrangian의 등식조건으로 생각할 수 있기 때문이다.

$$\mu_j^*(d_j - h_j(x^*)) = 0 \text{ (Complementarity)}$$

결국 이 식이 의미하는 것은 최적화 문제에서 부등식조건인 가중치 μ_j^* 를 0으로 잡거나, 구속조건을 꼭 채운 등식조건으로 인식하게 한다.

예제 Complementarity

$$\begin{aligned} &\underset{x}{\text{maximize}} && x^3 - 3x \\ &\text{subject to} && x \leq 2 \end{aligned}$$

라그랑지안은

$$L = x^3 - 3x + \mu(2 - x)$$

풀면

$$\begin{aligned} \frac{\partial L}{\partial x} &= 3x^2 - 3 - \mu = 0 \\ x &\leq 2 \\ \mu(2 - x) &= 0 \\ \mu &\geq 0 \end{aligned}$$

2가지의 상보성 조건이 존재한다.

- 만약 $\mu = 0$ 의 경우에 $3x^2 - 3 = 0$ 이 되어 $x = \pm 1$ 이 되고 $f(1) = -2$, $f(-1) = 2$ 가 되기 때문에 해로 적합하다.
- 만약 $x = 2$ 라면 $\mu = 9$ 가 된다. 이때도 마찬가지로 $f(2) = 2$ 가 되기 때문에 해로 적합하다.

따라서 최대값을 갖는 경우는 2가지의 해를 가지게 된다. $x = -1$ 과 $x = 2$.

예제 Standard form

$$\begin{aligned} &\underset{x,y}{\text{minimize}} && (x-2)^2 + 2(y-1)^2 \\ &\text{subject to} && x + 4y \leq 3 \\ &&& x \geq y \end{aligned}$$

위의 식을 최대값을 갖는 형태와 표준형태로 변형하면,

$$\begin{aligned} &\underset{x,y}{\text{maximize}} && -(x-2)^2 - 2(y-1)^2 \\ &\text{subject to} && x + 4y \leq 3 \\ &&& -x + y \leq 0 \end{aligned}$$

라그랑지안은

$$L(x, y, \mu_1, \mu_2) = -(x-2)^2 - 2(y-1)^2 + \mu_1(3-x-4y) + \mu_2(0+x-y)$$

풀면 다음 조건이 된다.

$$\frac{\partial L}{\partial x} = -2(x-2) - \mu_1 + \mu_2 = 0$$

$$\frac{\partial L}{\partial y} = -4(y-1) - 4\mu_1 - \mu_2 = 0$$

$$\frac{\partial L}{\partial \mu_1} = \mu_1(3-x-4y) = 0$$

$$\frac{\partial L}{\partial \mu_2} = \mu_2(x-y) = 0$$

$$x+4y \leq 3$$

$$-x+y \leq 0$$

$$\mu_1, \mu_2 \geq 0$$

여기서 총 2개의 상보성 조건이 존재하기 때문에, 우리는 4가지의 경우를 체크해야한다.

- $\mu_1 = 0, \mu_2 = 0$ 의 경우 $x = 2, y = 1$ 이 되기 때문에 해로 부적합하다.
- $\mu_1 = 0, x - y = 0$ 의 경우 $x = 4/3, y = 4/3, \mu_2 = -4/3$ 이 되기 때문에 해로 부적합하다.
- $\mu_2 = 0, 3 - x - 4y = 0$ 의 경우 $x = 5/3, y = 1/3, \mu_1 = 2/3$ 이 되므로 해로 적합하다.
- $3 - x - 4y = 0, x - y = 0$ 의 경우 $x = 3/5, y = 3/5, \mu_1 = 22/25, \mu_2 = -48/25$ 가 되므로 해로 부적합하다.

따라서 최적해는 $x = 5/3, y = 1/3$ 임을 알 수 있다.

1.3 선형계획모형 해법

선형계획모형의 해를 구하는 해법으로 도해법(graphical solution)과 대수적 방법(algebraic method)이 있다. 도해법은 의사결정변수가 2개인 경우에 2차원 평면에서 적용될 수 있어 유용하지만, 3개이상인 경우 적용되는데 어려움이 발생하여 일반적인 해법이라고 보기 어렵다. 일반적으로 2개 이상의 변수에서 도해법이 유용한 경우는 고정된 조건에 의해 특성을 알아보려할 때 유용하다. 또한 최적해를 구하는 해법의 특성을 이해하는데에도 필요하다. 보통 변수가 2개를 초과하면 대수적인 방법으로 해를 구해야한다. 대표적인 대수적 방법으로는 Simplex법, 타원체해법(ellipsoid algorithm), 그리고 Karmarkar의 알고리즘이 있다. Simplex법은 1947년 George B. Danzig에 의하여, 타원체해법은 1979년 Khachian에 의하여, Karmarkar의 알고리즘은 1984년 Karmarkar에 의하여 개발되었으며 내부점해법(interior-point algorithm)으로도 불린다. 타원체해법은 현재 거의 사용되지 않고 있고, 규모가 큰(large scale) 선형계획모형에 있어서는 근사치를 제고하는 내부점해법(interior-point method)이 Simplex법보다 복잡성 측면에서 우수한 것으로 알려져 있으나 일반적인 규모에서는 Simplex법이 효용성이 인정되어 아직도 널리 사용되고 있다. MATLAB Optimization Toolbox 패키지의 linprog의 함수에서도 규모를 분리하여 내부점해법과 Simplex법을 모두 차용하고 있다.

문제 LP 문제 세우기

다음의 문제는 화학공학, 혹은 석유공학분야에서 발생하는 문제이다. 그러나 유한한 자원을 가지고 제품을 생산하는 모든 공학분야에도 적용할 수 있다.

가스를 정제하는 공장이 매주 일정한 양의 원료가스를 받는다고 가정한다. 원료가스는 난방용 가스로 사용되는 보통 및 우수 품질의 두 가지 등급으로 정제된다. 이러한 등급들의 가스는 매우 수요가 높으며(항상 잘 팔림), 회사에 각각 다른 이윤을 가져다 준다. 그렇지만 이것들의 생산은 시간과 공장 내의 저장이라는 구속조건을 갖는다. 예를 들면 한 번에 한 개 등급의 제품만이 생산되며, 시설물은 단지 주당 80시간만 사용된다. 또한 각각의 제품들에 대하여 공장 내 보관장소의 제약이 따른다. 이러한 모든 인자들을 아래의 표로 정리되어 있다.

Resource	Product		Resource Availability
	Regular	Premium	
Raw gas	$7m^3/\text{tonne}$	$11m^3/\text{tonne}$	$77m^3/\text{week}$
Production time	$10hr/\text{tonne}$	$8hr/\text{tonne}$	$80hr/\text{week}$
Storage	9tonnes	6tonnes	-
Profit	150/tonne	175/tonne	-

위의 선형프로그래밍문제는 교재에서는 Simplex법을 기초로 설명하고 있다. 일단 이 서술형 문제에서 선형프로그래밍식으로 작성해보자. 공장을 가동하는 기술자는 이윤을 극대화하기 위하여 각각 얼마만큼의 가스들을 생산할지를 결정하여야 한다. 주당 생산되는 보통 등급과 우수 등급의 제품의 양을 각각 x_1 과 x_2 라고 한다면, 주당 전체 이윤은 다음과 같이 계산된다.

$$\text{total profit per week} = 150x_1 + 175x_2$$

구속조건들은 비슷한 방법으로 구할 수 있다. 예를 들면 전체 원료가스의 사용량은 다음과 같이 계산된다.

$$\text{total gas used amount per week} = 7x_1 + 11x_2$$

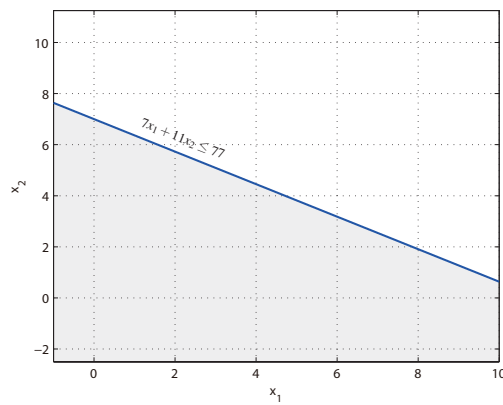
이 합계는 주당 가용한 양인 $77m^3/\text{week}$ 를 넘지 않아야한다. 따라서 구속조건은 다음 과 같이 표시된다.

$$7x_1 + 11x_2 \leq 77$$

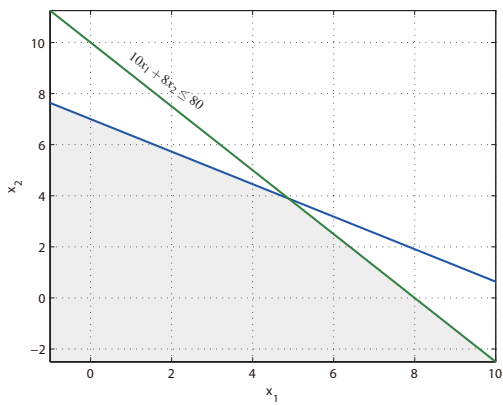
남은 구속조건들도 같은 방법으로 구해지며, 이를 종합하면 선형프로그래밍식은 다음과 같다.

maximize	$Z = 150x_1 + 175x_2$	이윤의 극대화
subject to	$7x_1 + 11x_2 \leq 77$	재료의 구속조건
	$10x_1 + 8x_2 \leq 80$	시간의 구속조건
	$x_1 \leq 9$	”보통등급”의 보관 구속조건
	$x_2 \leq 6$	”우수등급”의 보관 구속조건
	$x_1, x_2 \geq 0$	양수의 구속조건

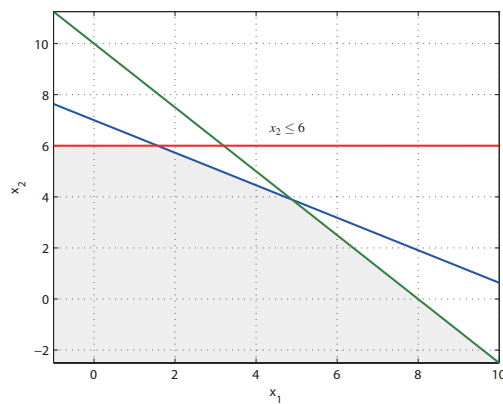
도해법으로 해를 구하면



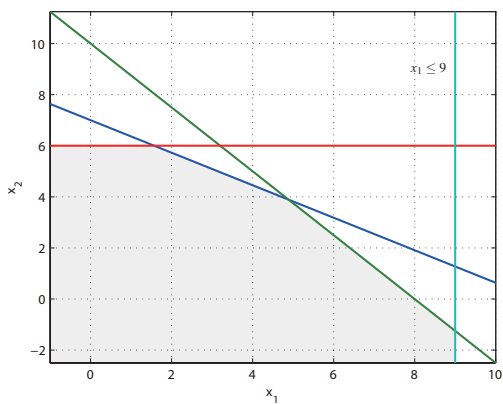
(a) $7x_1 + 11x_2 \leq 77$



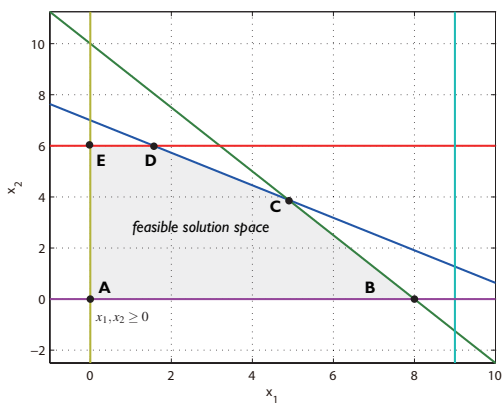
(b) $10x_1 + 8x_2 \leq 80$



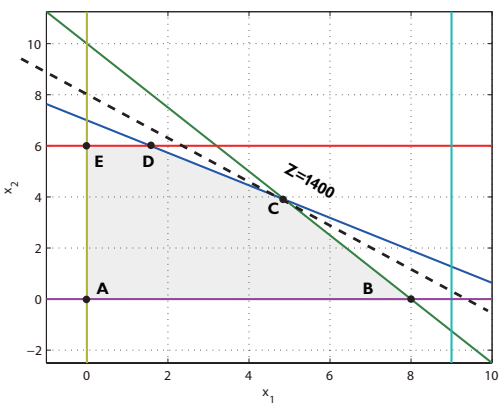
(c) $x_2 \leq 6$



(d) $x_1 \leq 9$



(e) $x_1, x_2 \geq 0$



(f) $Z = 150x_1 + 175x_2$

Figure 1.2: Graphical solution

1.4 패키지를 사용한 최적화

(MATLAB Optimization Toolbox Functions)

다음 테이블은 MATLAB에서 사용할 수 있는 최적화 함수들을 나타낸다. 이 함수들은 최적화 툴박스에서 사용되어지지만 근을 구하거나, 최소화, 그리고 여러개의 목적함수를 둔 최적화 그리고 회귀분석에 이용될 수 있다.

Type	Formulation	Solver
Scalar minimization	$\min_x f(x)$ subject to $l < x < u, (x \text{ is scalar})$	fminbnd()
Unconstrained minimization	$\min_x f(x)$ subject to (none)	fminunc(),fminsearch()
Linear programming	$\min_x \mathbf{f}^T \mathbf{x}$ subject to $\mathbf{Ax} \leq \mathbf{b}, \mathbf{A}_{eq}\mathbf{x} = \mathbf{b}_{eq}, l \leq x \leq u$	linprog()
Quadratic programming	$\min_x \frac{1}{2} \mathbf{x}^T \mathbf{H} \mathbf{x} + \mathbf{c}^T \mathbf{x}$ subject to $\mathbf{Ax} \leq \mathbf{b}, \mathbf{A}_{eq}\mathbf{x} = \mathbf{b}_{eq}, l \leq x \leq u$	quadprog()
Constrained minimization	$\min_x f(\mathbf{x})$ subject to $c(x) \leq 0, c_{eq}(x) = 0,$ $\mathbf{Ax} \leq \mathbf{b}, \mathbf{A}_{eq}\mathbf{x} = \mathbf{b}_{eq}, l \leq x \leq u$	fmincon()

1.4.1 Linear programming `linprog()`

MATLAB함수의 `linprog()`는 선형계획법의 최적해를 구해준다. `linprog()`함수가 사용하는 알고리즘은 옵션에 선언해 주지 않는 이상 Large scale linear programming과 Medium scale linear programming 으로 자동선택되어 사용되어진다. 디폴트 옵션으로 사용되어지는 Large scale LP는 Mehrotra's predictor-corrector 알고리즘¹의 변형인 primal-dual interior-point method를 사용한 LIPSOL²법을 사용한다. Medium scale LP는 1947년 George Dantzig³가 개발한 Simplex법(Simplex method)를 사용한다.

$$\begin{aligned}
& \underset{\mathbf{x}}{\text{minimize}} && \mathbf{f}^T \\
& \text{subject to} && \mathbf{A} \cdot \mathbf{x} \leq \mathbf{b} \\
& && \mathbf{A}_{eq} \cdot \mathbf{x} = \mathbf{b}_{eq} \\
& && \mathbf{L}_b \leq \mathbf{x} \leq \mathbf{U}_b
\end{aligned}$$

위의 수식은 MATLAB에서 사용할 수 있는 형태로 표현하기 위해, 행렬형태를 사용한다.

`[x,fval,exitflag,output,lambda] = linprog(f,A,b,Aeq,beq,lb,ub,x0,options)` 앞서 언급한대로 MATLAB프로그램을 수행하면, Large-scale optimization 방식으로 선형프로그래밍의 최적해를 구한다.

¹Mehrotra, S., "On the Implementation of a Primal-Dual Interior Point Method," SIAM Journal on Optimization, Vol. 2, pp 575-601, 1992.

²Zhang, Y., "Solving Large-Scale Linear Programs by Interior-Point Methods Under the MATLAB Environment," Department of Mathematics and Statistics, University of Maryland, Baltimore County, Baltimore, MD, Technical Report TR96-01, July, 1995.

³George B. Dantzig, "Programming of Interdependent Activities: II Mathematical Model" Econometrica Vol. 17, No. 3/4, pp. 200-211, 1949.

f	Linear objective function vector f
Aineq	Matrix for linear inequality constraints
bineq	Vector for linear inequality constraints
Aeq	Matrix for linear equality constraints
beq	Vector for linear equality constraints
lb	Vector of lower boundsubVector of upper bounds
x0	Initial point for x, active set algorithm only
solver	'linprog'
options	Options structure created with optimset()

Table 1: Input Arguments

```

1 clear all; clc; close all;

3 f=[-150 -175];
  A=[7 11;
5    10 8];
  b=[77 80];
7 ub=[9 6];
  lb=[0 0];
9

[x,fval,exitflag,output,lambda] = linprog(f,A,b,[],[],lb,ub);

```

Code [MATLAB] 1: 가스정제공장의 최적해

Part II

Appendices