전산 선박 설계

Part 1. 최적 설계 기법

2006.9.

서울대학교 조선해양공학과
이규열

Advanced
Ship
Design
Automation
Laboratory

목차

Ch1. 최적 설계 개요

1.1 최적 설계 문제의 일반적 정식화

1.2 최적 설계 문제의 구성 요소

1.3 최적화 기법의 분류

Advanced
Ship
Design
Automation
Laboratory

1.1 최적 설계 문제의 일반적 정식화

Minimize

$$f(\mathbf{x})$$

Objective Function

Subject to

Constraints

$$g_j(\mathbf{x}) \le 0,\ j = 1,\Lambda\ ,m$$

: **부등호 제약 조건**(Inequality Constraint)

$$h_k(\mathbf{x}) = 0, k = 1,\Lambda\ ,p$$

: **등호 제약 조건**(Equality Constraint)

$$\mathbf{x}_l \le \mathbf{x} \le \mathbf{x}_u$$

: **설계 변수 벡터에 대한 상·하한값 제약 조건**

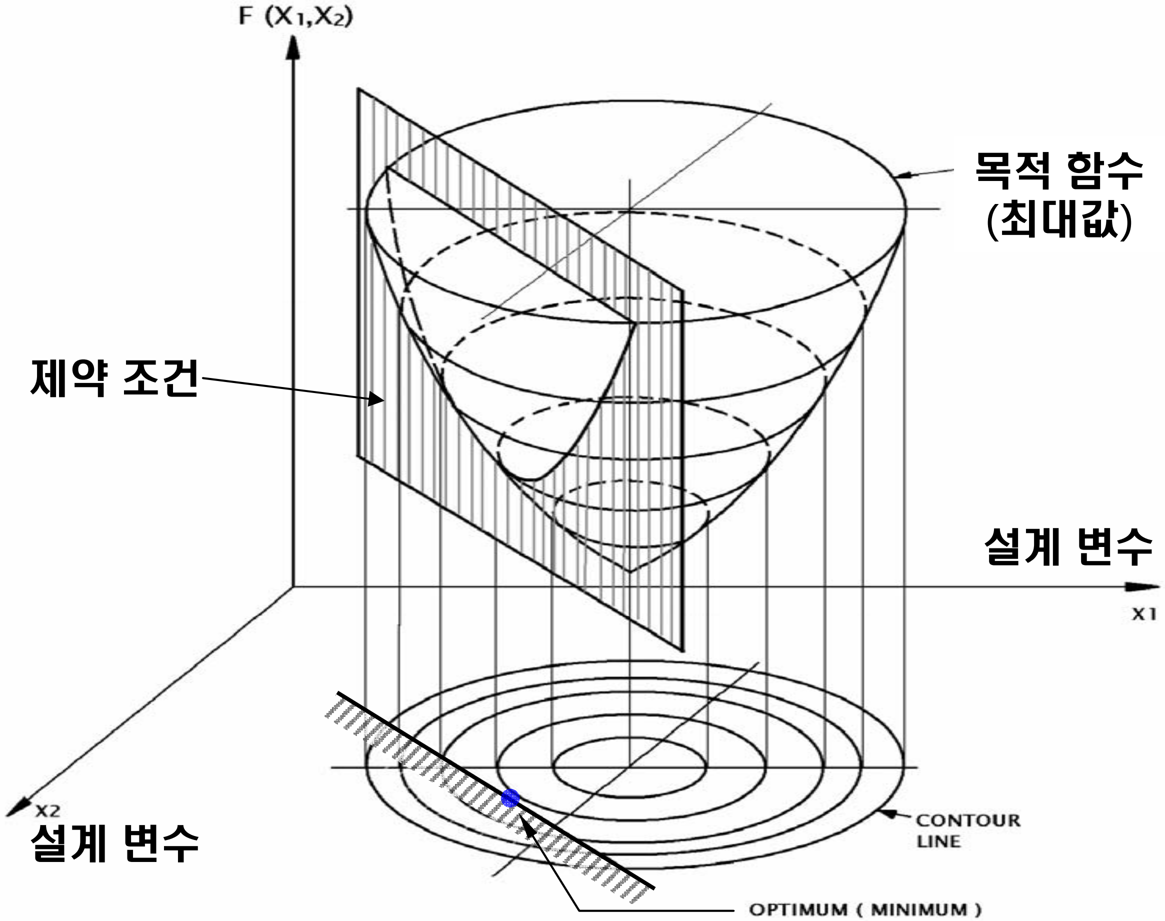
Where $\mathbf{x} = (x_1, x_2, \Lambda\ , x_n)$

1.2 최적 설계 문제의 구성 요소(1)

- **설계 변수(Design Variable)**
  - 설계하고자 하는 치수, 위치 등을 나타내는 변수로서 자유 변수(Free Variable) 또는 독립 변수(Independent Variable)라고 함
  - 종속 변수(Dependent Variable)
    - 설계 변수의 결정 후 종속적으로 결정되어지는 변수
- **제약 조건(Constraint)**
  - 설계에서 기능상 요구되는 조건 또는 크기의 제한 등을 정의
  - 부등호 제약 조건(Inequality Constraint), 등호 제약 조건(Equality Constraint)

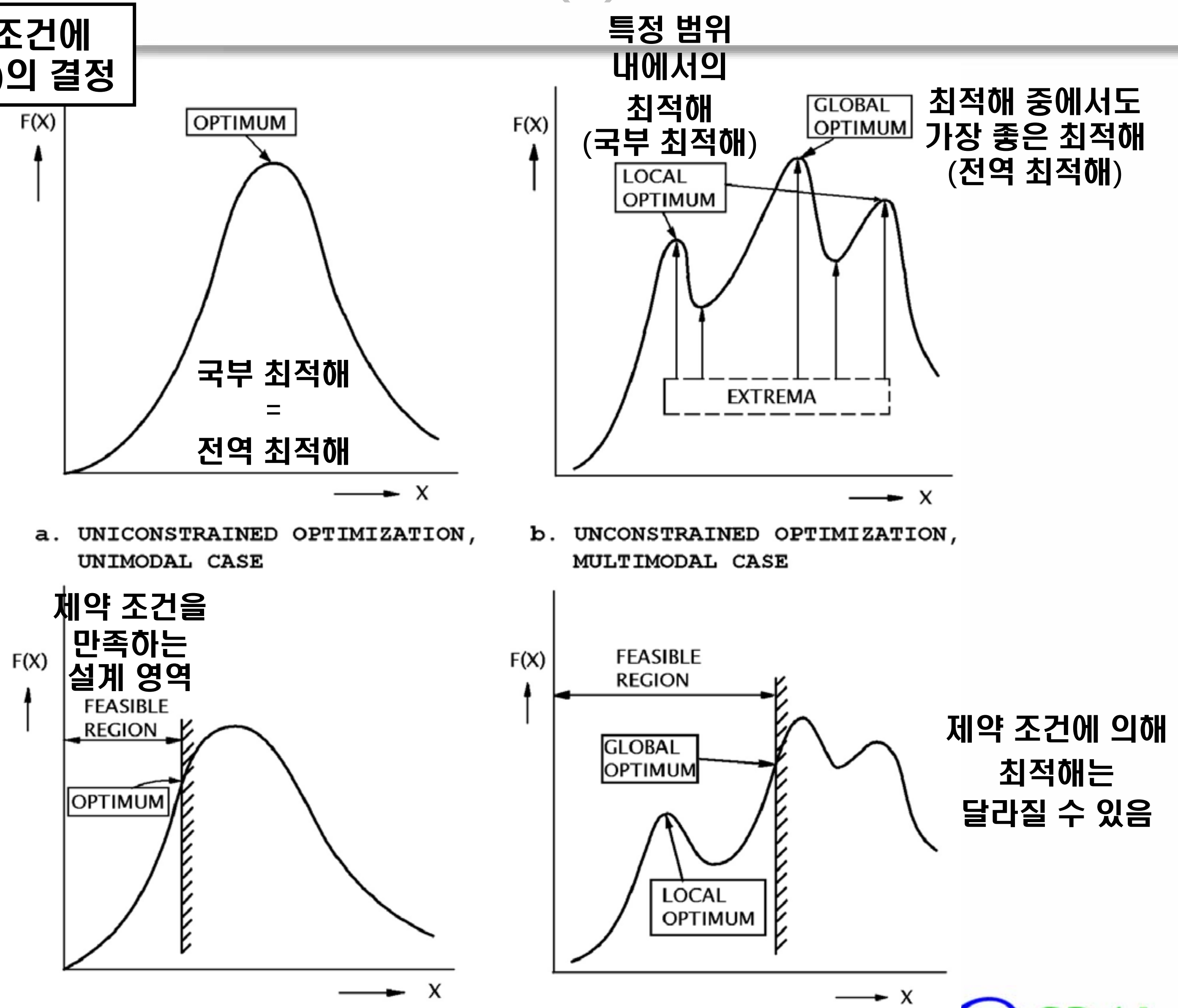
1.2 최적 설계 문제의 구성 요소(2)

- **목적 함수**(Objective Function)
  - **최적**(Optimum)**을 나타내는 기준으로 비용, 무게 등과 같은 값을 비교하여 어느 설계 대안**(Design Alternative)**이 보다 나은지를 나타낼 수 있는 함수**



1.2 최적 설계 문제의 구성 요소(3)

목적 함수와 제약 조건에 따른 최적해(최대값)의 결정



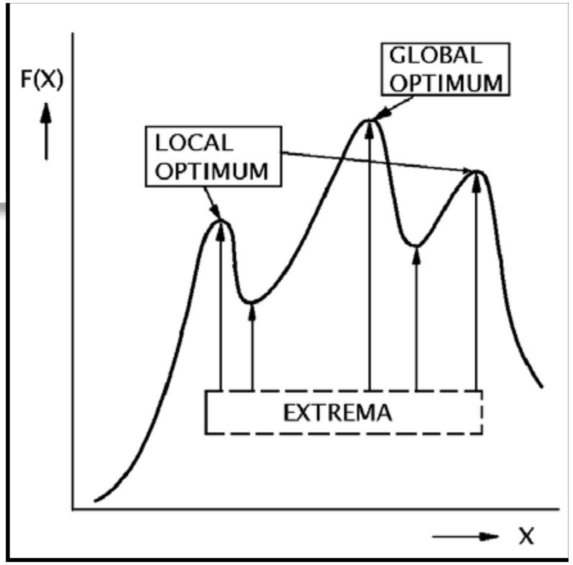
a. UNICONSTRAINED OPTIMIZATION, UNIMODAL CASE

b. UNCONSTRAINED OPTIMIZATION, MULTIMODAL CASE

c. CONSTRAINED OPTIMIZATION, UNIMODAL CASE

d. CONSTRAINED OPTIMIZATION, MULTIMODAL CASE

1.3 최적화 기법의 분류(1)



- **전역 최적화 기법(Global Optimization Methods)**
  - 장점
    - 다수의 국부 최적해(Local Optima)를 가진 대규모 문제에 적합
  - 단점
    - 최적해를 얻기 위해 많은 Iteration이 필요(긴 계산 시간 요구)
  - Genetic Algorithms(GA), Simulated Annealing, etc.

- **국부 최적화 기법(Local Optimization Methods)**
  - 장점
    - 최적해를 얻기 위해 상대적으로 적은 Iteration이 필요(짧은 계산 시간 요구)
  - 단점
    - 초기해(Starting Point)에 가까운 국부 최적해를 도출(거의 국부 최적해만 찾음)
  - Method of Feasible Directions(MFD), Sequential Quadratic Programming(SQP), Multi-Start Optimization Method, etc.

1.3 최적화 문제의 분류(2)

- 제약 조건의 유무
  - 비제약 최적화 문제(Unconstrained Optimization Problem)
    - 제약 조건이 없는 최적화 문제
    - 직접 탐사법(Hooke & Jeeves Method, Nelder & Mead's Simplex Method), Gradient 방법(Steepest Descent Method, etc.), etc.
  - 제약 최적화 문제(Constrained Optimization Problem)
    - 제약 조건이 있는 최적화 문제
    - Penalty Function Method, Sequential Linear Programming, Constrained Steepest Descent Method, Method of Feasible Directions(MFD), Sequential Quadratic Programming(SQP), etc.
- 목적 함수의 수
  - 단일 최적화 문제(Single-Objective Optimization Problem)
  - 다중 최적화 문제(Multi-Objective Optimization Problem)
    - Weighting Method, Constraint Method

Ch2. 비제약 최적화 기법

2.1 Gradient 방법

- Steepest Descent 방법
- 공액 경사도 방법(Conjugate Gradient 방법)
- Newton의 방법
- Davidon-Fletcher-Powell(DFP) 방법
- Broyden-Fletcher-Goldfarb-Shanno(BFGS) 방법

2.2 황금 분할에 의한 1차원 탐색 방법

2.3 직접 탐사법(Direct Search Method)

- Hooke & Jeeves의 직접 탐사법
- Nelder & Mead의 Simplex 방법

Advanced
Ship
Design
Automation
Laboratory

2.1 Gradient 방법

Advanced
Ship
Design
Automation
Laboratory

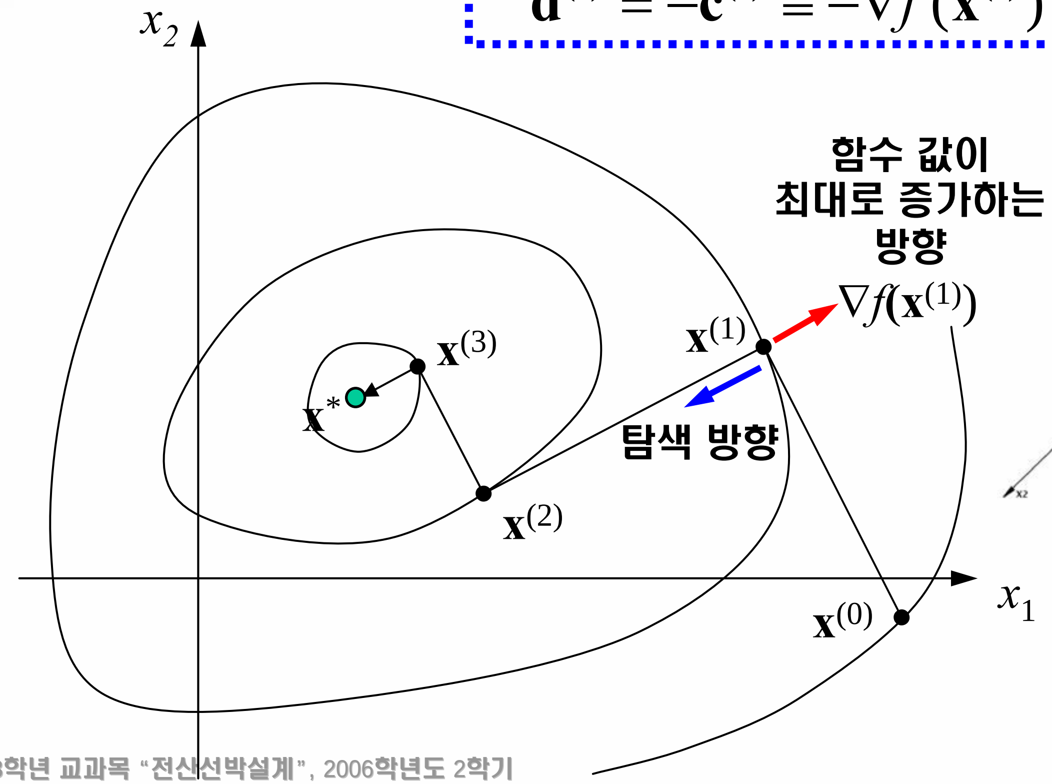
2.1 Gradient 방법
- Steepest Descent 방법(최속 강하법)

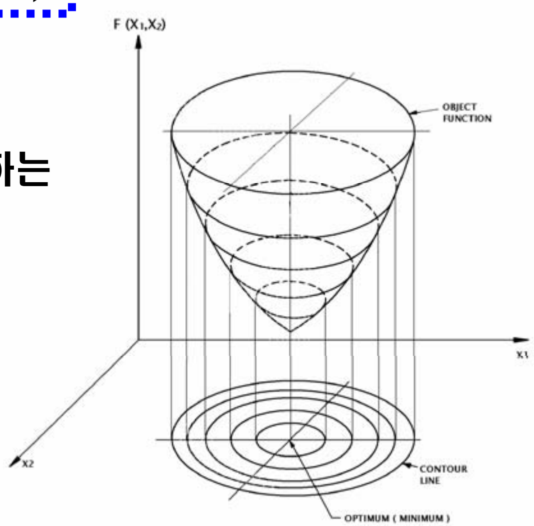
- **탐색 방향**(Search Direction)**을 목적 함수의** Gradient Vector**의 반대 방향으로 가정하고 순차적으로 최적해를 찾는 방법**
  - ➡ Gradient Vector($\nabla f(\mathbf{x})$): **함수 값이 최대로 증가하는 방향**

<u>목적 함수를 최소화 하는 문제일 경우</u>

탐색 방향

$$\mathbf{d}^{(1)} = -\mathbf{c}^{(1)} \equiv -\nabla f(\mathbf{x}^{(1)})$$



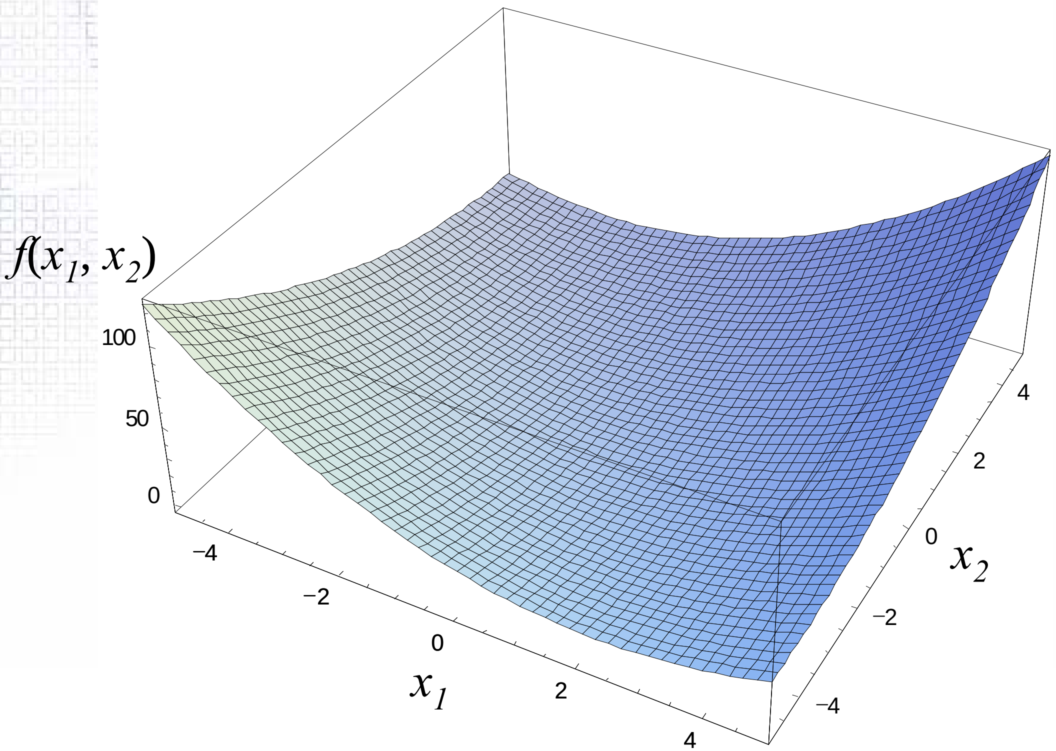


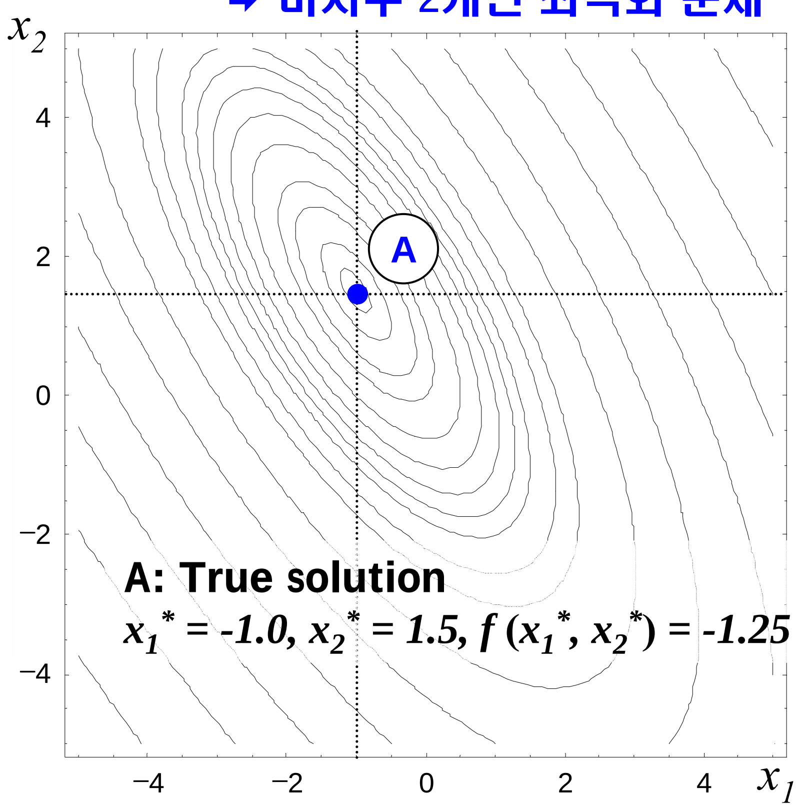
비제약 최적화 문제

- Steepest descent **방법(최속 강하법, 최대 경사법)을 이용하여** 2**변수 함수의 최소점을 구하시오. 단, 시작점 x**$^{(0)}$ = (0, 0), convergence tolerance ε = 0.001**이며, x**$^{(3)}$**까지 구하시오**.

$$\textit{Minimize}\quad f(x_1, x_2) = x_1 - x_2 + 2x_1^2 + 2x_1x_2 + x_2^2$$

➡ **미지수 2개인 최적화 문제**





비제약 최적화 문제
- Steepest descent 방법을 이용한 해법(1)

$$\textit{Minimize} \quad f(x_1, x_2) = x_1 - x_2 + 2x_1^2 + 2x_1x_2 + x_2^2$$

$$\nabla f(\mathbf{x}) = \nabla f(x_1, x_2) = \begin{pmatrix} 1 + 4x_1 + 2x_2 \\ -1 + 2x_1 + 2x_2 \end{pmatrix}$$

- **단계** 1 - $\mathbf{x}^{(1)}$구하기

$$\nabla f(\mathbf{x}^{(0)}) = \nabla f\begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 + 4x_1 + 2x_2 \\ -1 + 2x_1 + 2x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

$$\mathbf{x}^{(1)} = \mathbf{x}^{(0)} - \alpha^{(0)} \nabla f(\mathbf{x}^{(0)})$$

$$= \begin{pmatrix} 0 \\ 0 \end{pmatrix} - \alpha \begin{pmatrix} 1 \\ -1 \end{pmatrix} = \begin{pmatrix} -\alpha \\ \alpha \end{pmatrix}$$

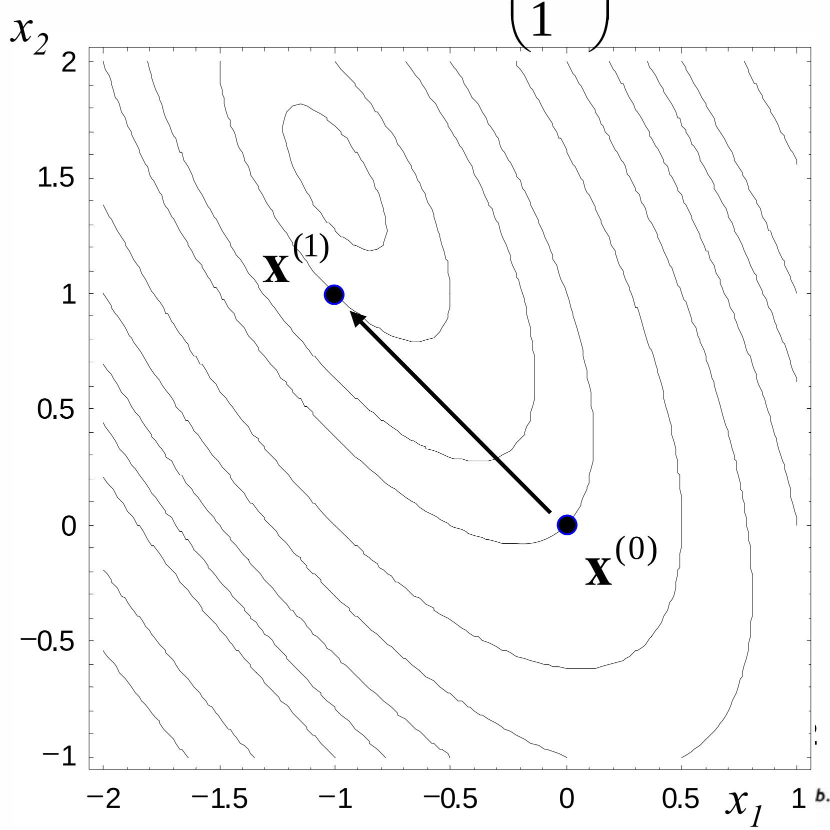
편의상 $\alpha^{(0)}$을 α로 대체함

$$f(\mathbf{x}^{(1)}) = \alpha^2 - 2\alpha$$

함수 $f(\mathbf{x}^{(1)})$가 최소값을 가질 조건

$$\frac{df(\mathbf{x}^{(1)})}{d\alpha} = 0 \text{으로부터 } \alpha = 1.0$$

$$\therefore \mathbf{x}^{(1)} = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$$



비제약 최적화 문제
- Steepest descent 방법을 이용한 해법(2)

- **단계 2 -** $\mathbf{x}^{(2)}$구하기

$$\nabla f(\mathbf{x}^{(1)}) = \nabla f\begin{pmatrix}-1\\1\end{pmatrix} = \begin{pmatrix}1+4x_1+2x_2\\-1+2x_1+2x_2\end{pmatrix} = \begin{pmatrix}-1\\-1\end{pmatrix}$$

$$\mathbf{x}^{(2)} = \mathbf{x}^{(1)} - \alpha^{(1)}\nabla f(\mathbf{x}^{(1)})$$

$$= \begin{pmatrix}-1\\1\end{pmatrix} - \alpha\begin{pmatrix}-1\\-1\end{pmatrix} = \begin{pmatrix}-1+\alpha\\1+\alpha\end{pmatrix}$$

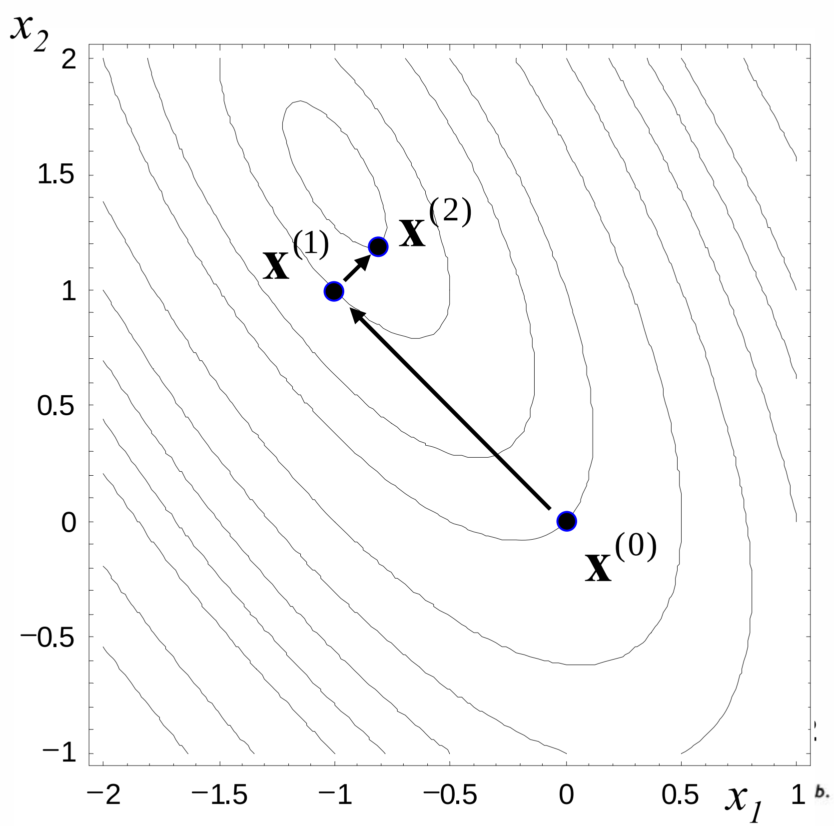
편의상 $\alpha^{(1)}$을 α로 대체함

$$f(\mathbf{x}^{(2)}) = 5\alpha^2 - 2\alpha - 1$$

함수 $f(\mathbf{x}^{(2)})$가 최소값을 가질 조건

$\dfrac{df(\mathbf{x}^{(2)})}{d\alpha} = 0$으로부터 $\alpha = 0.2$

$$\therefore \mathbf{x}^{(2)} = \begin{pmatrix}-0.8\\1.2\end{pmatrix}$$



비제약 최적화 문제
- Steepest descent 방법을 이용한 해법(3)

- **단계** 3 - $\mathbf{x}^{(3)}$구하기

$$\nabla f(\mathbf{x}^{(1)}) = \nabla f\begin{pmatrix} -0.8 \\ 1.2 \end{pmatrix} = \begin{pmatrix} 1+4x_1+2x_2 \\ -1+2x_1+2x_2 \end{pmatrix} = \begin{pmatrix} 0.2 \\ -0.2 \end{pmatrix}$$

$$\mathbf{x}^{(3)} = \mathbf{x}^{(2)} - \alpha^{(2)}\nabla f(\mathbf{x}^{(2)})$$

$$= \begin{pmatrix} -0.8 \\ 1.2 \end{pmatrix} - \alpha\begin{pmatrix} 0.2 \\ -0.2 \end{pmatrix} = \begin{pmatrix} -0.8-0.2\alpha \\ 1.2+0.2\alpha \end{pmatrix}$$

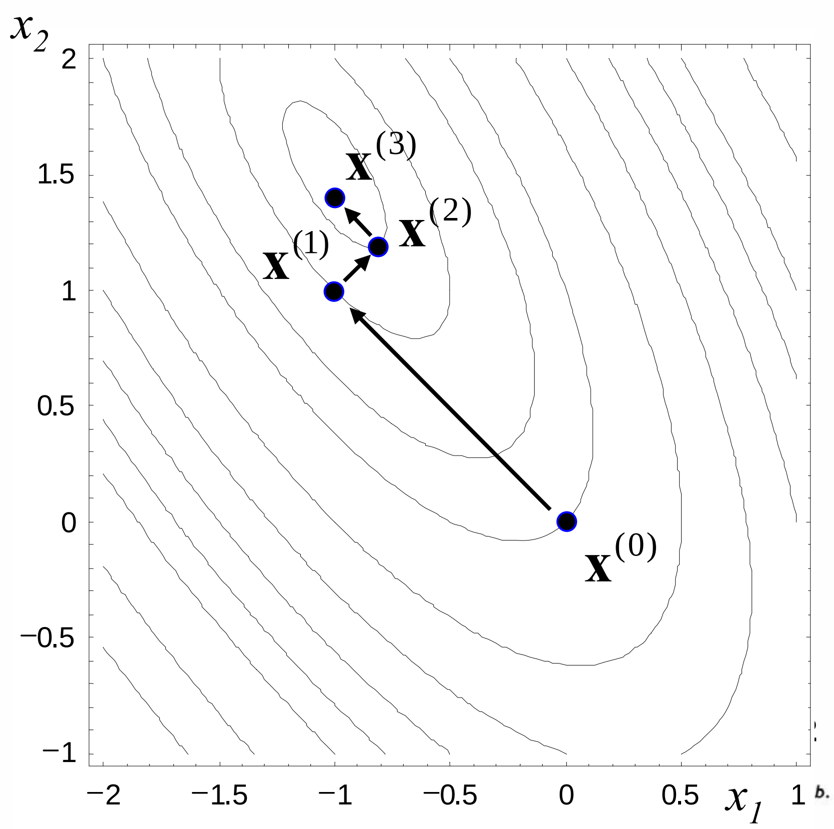
편의상 $\alpha^{(2)}$을 α로 대체함

$$f(\mathbf{x}^{(3)}) = 0.04\alpha^2 - 0.08\alpha - 1.2$$

함수 $f(\mathbf{x}^{(3)})$가 최소값을 가질 조건

$\dfrac{df(\mathbf{x}^{(3)})}{d\alpha} = 0$으로부터 $\alpha = 1.0$

$$\therefore \mathbf{x}^{(3)} = \begin{pmatrix} -1 \\ 1.4 \end{pmatrix}$$

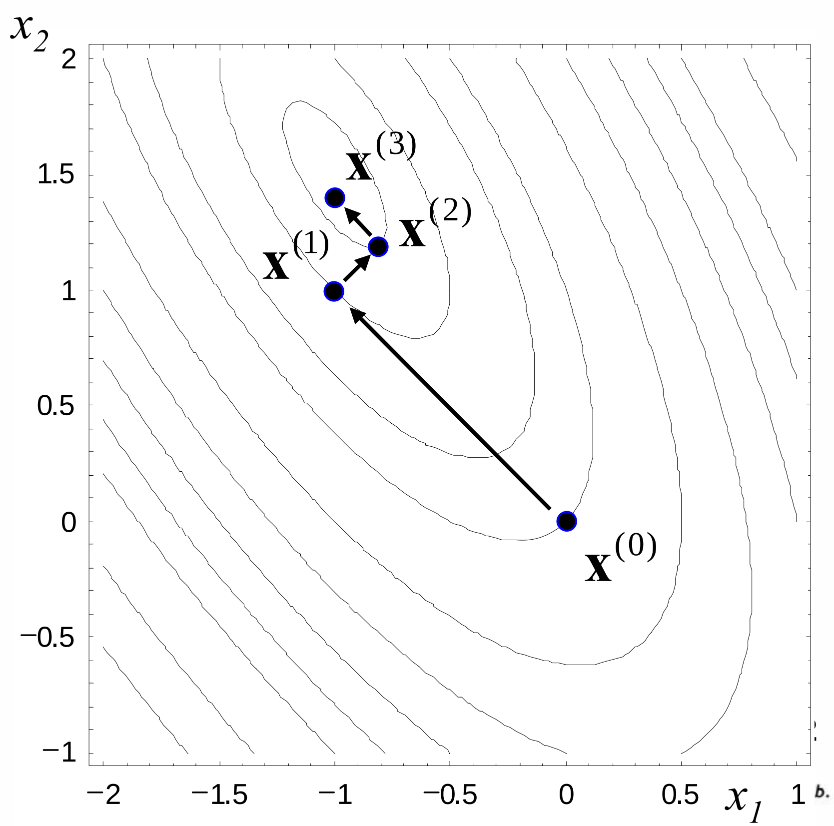


비제약 최적화 문제
- Steepest descent 방법을 이용한 해법(4)

- **단계 4** - 최적해 구하기

  이와 같은 과정을 반복하여 $\left|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\right| \le \varepsilon$일 경우

  중지하며 그때의 $\mathbf{x}^{(k+1)}$이 최적해가 된다.



Gradient 방법
- 공액 경사도 방법(Conjugate Gradient 방법)(1)

- 탐색 방향을 수정하여 최속 강하법보다 수렴율을 향상시킨 방법. 현재의 최속 강하 방향에 직전에 사용된 탐색 방향을 척도화 시켜 더한 것을 탐색 방향으로 사용함
  - 단계 1 : 초기 설계점 $\mathbf{x}^{(0)}$를 추정한다. 반복 횟수 번호를 $k=0$으로 둔다. 또한 수렴 매개 변수 ε 을 선정하고 최적 탐색 방향을 계산한다.

$$\mathbf{d}^{(0)} = -\mathbf{c}^{(0)} \equiv -\nabla f(\mathbf{x}^{(0)})$$

그리고, $\left\|\mathbf{c}^{(0)}\right\| < \varepsilon$ 식을 만족하면 반복 과정을 마치고 만족하지 않으면 단계 4로 간다(공액 경사법과 최속 강하법의 단계 1은 동일함).

  - 단계 2 : 목적 함수의 경사도를 계산한다.

$$\mathbf{c}^{(k)} = \nabla f(\mathbf{x}^{(k)})$$

여기서, 만일 $\left\|\mathbf{c}^{(k)}\right\| < \varepsilon$ 이면 멈춘다. 그렇지 않으면 계속한다.

Gradient 방법
- 공액 경사도 방법(Conjugate Gradient 방법)(2)

- 단계 3 : 다음과 같이 새로운 탐색 방향을 계산한다.

$$\boxed{\mathbf{d}^{(k)} = -\mathbf{c}^{(k)} + \beta_k \mathbf{d}^{(k-1)}}$$

$$\beta_k = \left(\left\| \mathbf{c}^{(k)} \right\| / \left\| \mathbf{c}^{(k-1)} \right\| \right)^2$$

- 단계 4 : $f(\mathbf{x}^{(k)} + \alpha \mathbf{d}^{(k)})$ 를 최소화 하는 $\alpha = \alpha_k$를 계산한다.
- 단계 5 : 현재의 설계점을 다음과 같이 변경한다.

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{d}^{(k)}$$

$k = k + 1$ 로 두고 단계 2로 간다.

비제약 최적화 문제
- Conjugate gradient 방법을 이용한 해법(1)

$$\textbf{\textit{Minimize}}\quad f(x_1,x_2)=x_1-x_2+2x_1^2+2x_1x_2+x_2^2$$

$$\nabla f(\mathbf{x})=\nabla f(x_1,x_2)=\begin{pmatrix}1+4x_1+2x_2\\-1+2x_1+2x_2\end{pmatrix}$$

- **단계** 1 - $\mathbf{x}^{(1)}$구하기

$$\nabla f(\mathbf{x}^{(0)})=\nabla f\begin{pmatrix}0\\0\end{pmatrix}=\begin{pmatrix}1+4x_1+2x_2\\-1+2x_1+2x_2\end{pmatrix}=\begin{pmatrix}1\\-1\end{pmatrix}$$

$$\mathbf{x}^{(1)}=\mathbf{x}^{(0)}-\alpha^{(0)}\nabla f(\mathbf{x}^{(0)})$$

$$=\begin{pmatrix}0\\0\end{pmatrix}-\alpha\begin{pmatrix}1\\-1\end{pmatrix}=\begin{pmatrix}-\alpha\\\alpha\end{pmatrix}$$

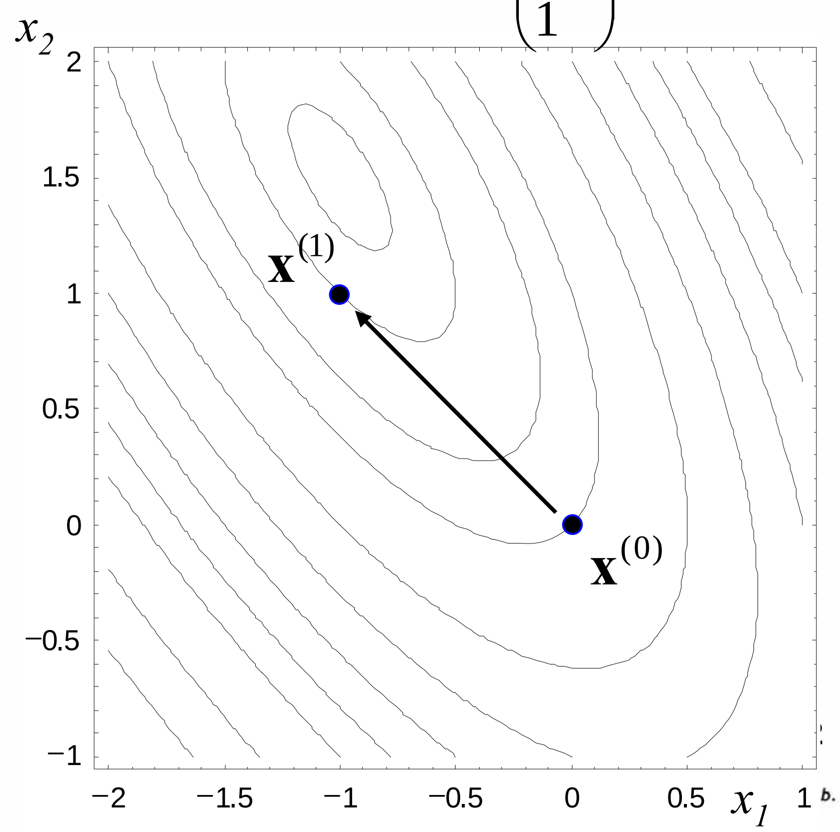
편의상 $\alpha^{(0)}$을 α로 대체함

$$f(\mathbf{x}^{(1)})=\alpha^2-2\alpha$$

함수 $f(\mathbf{x}^{(1)})$가 최소값을 가질 조건

$$\frac{df(\mathbf{x}^{(1)})}{d\alpha}=0\text{으로부터 }\alpha=1.0$$

$$\therefore\ \mathbf{x}^{(1)}=\begin{pmatrix}-1\\1\end{pmatrix}$$



비제약 최적화 문제
- Conjugate gradient 방법을 이용한 해법(2)

- **단계** 2 - $\mathbf{x}^{(2)}$구하기

$\mathbf{x}^{(1)}$에서 $\nabla f(\mathbf{x}^{(1)})$를 구하고

"*Conjugate direction*"을 구한다.

$$\nabla f(\mathbf{x}^{(1)}) = \nabla f\begin{pmatrix} -1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1+4x_1+2x_2 \\ -1+2x_1+2x_2 \end{pmatrix} = \begin{pmatrix} -1 \\ -1 \end{pmatrix}$$

$$\mathbf{x}^{(2)} = \mathbf{x}^{(1)} - \alpha^{(1)}[-\nabla f(\mathbf{x}^{(1)}) + \frac{\left|\nabla f(\mathbf{x}^{(1)})\right|^2}{\left|\nabla f(\mathbf{x}^{(0)})\right|^2} \cdot \mathbf{d}^{(0)}]$$

로 가정하고 $f(\mathbf{x}^{(2)})$이 최소가 되는 $\alpha^{(1)}$를 구한다.

여기서, $\mathbf{d}^{(0)} = -\nabla f(\mathbf{x}^{(0)})$이다.

비제약 최적화 문제
- Conjugate gradient 방법을 이용한 해법(3)

$$\mathbf{x}^{(2)} = \begin{pmatrix} -1 \\ 1 \end{pmatrix} + \alpha\left\{ -\begin{pmatrix} -1 \\ -1 \end{pmatrix} + \frac{2}{2}(-1)\begin{pmatrix} 1 \\ -1 \end{pmatrix} \right\}$$

$$= \begin{pmatrix} -1 \\ 1+2\alpha \end{pmatrix}$$

편의상 $\alpha^{(1)}$을 α로 대체함

$$f(\mathbf{x}^{(2)}) = 4\alpha^2 - 2\alpha - 1$$

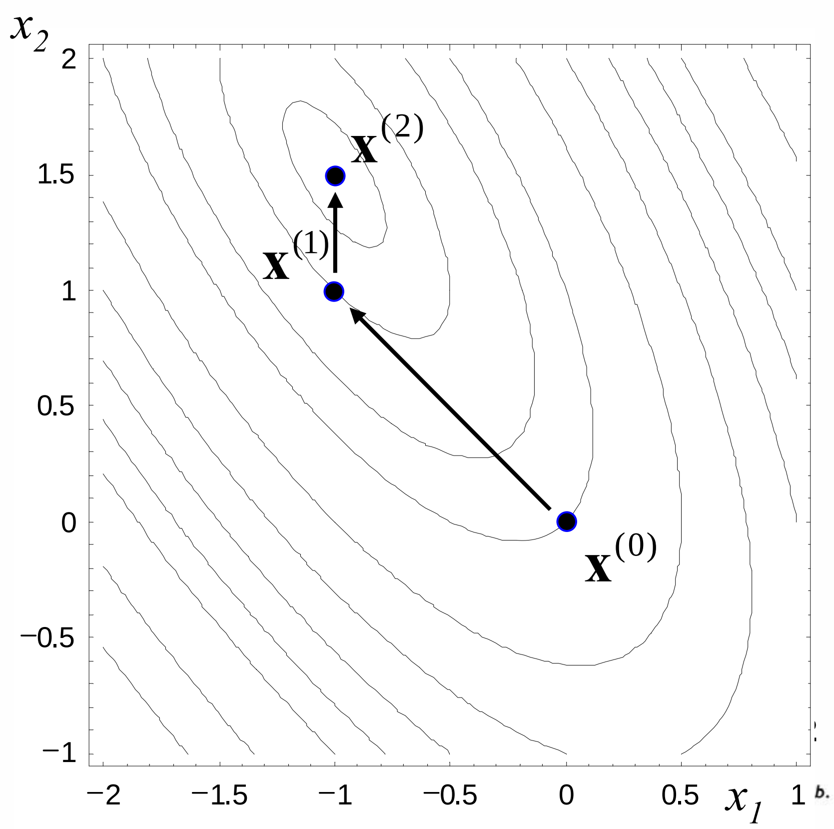
함수 $f(\mathbf{x}^{(2)})$가 최소값을 가질 조건

$$\frac{df(\mathbf{x}^{(2)})}{d\alpha} = 0$$으로부터 $\alpha = 0.25$

$$\therefore \mathbf{x}^{(2)} = \begin{pmatrix} -1 \\ 1.5 \end{pmatrix}$$

$\left\|\mathbf{c}^{(2)}\right\| = 0 \le \varepsilon$이므로

$\mathbf{x}^{(2)}$가 최적점이 된다.



Gradient 방법
- Newton의 방법

- 헷세 행렬을 사용하여 탐색 방법을 개선한 방법
  - 단계 1 : 시작점 $\mathbf{x}^{(0)}$을 추정한다. 반복 회수 번호를 $k=0$ 으로 둔다. 그리고 종료 기준으로 허용치 ε 을 선정한다.
  - 단계 2 : $i=1$ 에서 n 까지 $c_i^{(k)} = \partial f(\mathbf{x}^{(k)}) / \partial x_i$ 를 계산한다.
  - 단계 3 : 헷세 행렬을 계산한다.

$$H(\mathbf{x}^{(k)}) = \left[\frac{\partial^2 f}{\partial x_i \partial x_j} \right], \qquad i=1,\Lambda\ ,n;\ j=1,\Lambda\ ,n$$

  - 단계 4 : 탐색 방향을 계산한다.

$$\Delta \mathbf{x}^{(k)} = \boxed{\mathbf{d}^{(k)} = -\mathbf{H}^{-1}\mathbf{c}^{(k)}}$$

$f(\mathbf{x}) = f(\mathbf{x}^*) + \mathbf{c}^T\mathbf{d} + \frac{1}{2}\mathbf{d}^T\mathbf{H}(\mathbf{x}^*)\,\mathbf{d}$ 일 때

최소값을 가질 조건

$f'(\mathbf{x}) = \mathbf{c} + \mathbf{H}(\mathbf{x}^*)\,\mathbf{d} = 0$

$\Rightarrow \mathbf{H}(\mathbf{x}^*)\,\mathbf{d} = -\mathbf{c} \Rightarrow \mathbf{d} = -\mathbf{H}(\mathbf{x}^*)^{-1}\mathbf{c}$

  - 단계 5 : 설계점을 $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{d}^{(k)}$ 로 수정한다. 이때 α_k 는 $f(\mathbf{x}^{(k)} + \alpha \mathbf{d}^{(k)})$ 를 최소화하도록 계산된다. α 를 계산하기 위해서는 어떠한 1차원 탐색법도 사용할 수 있다. 초기 이동 거리 추정으로 $\alpha=1$ 로 가정한다.
  - 단계 6 : $k=k+1$ 로 두고 단계 2를 수행한다.

[참고] 2변수 함수의 테일러 전개(Taylor Series Expansion)(1)

- 2변수 함수 $f(x_1, x_2)$에 대한 점 (x_1^*, x_2^*)에서의 테일러 전개식

$$f(x_1,x_2) = f(x_1^*,x_2^*) + \frac{\partial f}{\partial x_1}(x_1 - x_1^*) + \frac{\partial f}{\partial x_2}(x_2 - x_2^*)$$
$$+\frac{1}{2}\left(\frac{\partial^2 f}{\partial x_1^2}(x_1 - x_1^*)^2 + 2\frac{\partial^2 f}{\partial x_1 \partial x_2}(x_1 - x_1^*)(x_2 - x_2^*) + \frac{\partial^2 f}{\partial x_2^2}(x_2 - x_2^*)^2\right) + R$$

⬇ 이를 다시 표현하면,

$$f(\mathbf{x}) = f(\mathbf{x}^*) + \nabla f(\mathbf{x}^*)^T(\mathbf{x} - \mathbf{x}^*) + \frac{1}{2}(\mathbf{x} - \mathbf{x}^*)^T \mathbf{H}(\mathbf{x}^*)(\mathbf{x} - \mathbf{x}^*) + R$$

$$\left(\mathbf{x} = (x_1, x_2)^T,\ \mathbf{x}^* = (x_1^*, x_2^*)^T,\ \mathbf{H} \in M_{2\times 2}\right)$$

2x2 Matrix의 원소

$$\nabla f(\mathbf{x}^*)^T(\mathbf{x} - \mathbf{x}^*) = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{bmatrix}^T \begin{bmatrix} x_1 - x_1^* \\ x_2 - x_2^* \end{bmatrix}$$

$$\frac{1}{2}(\mathbf{x} - \mathbf{x}^*)^T \mathbf{H}(\mathbf{x}^*)(\mathbf{x} - \mathbf{x}^*) = \frac{1}{2}\left[\frac{\partial^2 f}{\partial x_1^2}(x_1 - x_1^*) + \frac{\partial^2 f}{\partial x_2 \partial x_1}(x_2 - x_2^*) \quad \frac{\partial^2 f}{\partial x_1 \partial x_2}(x_1 - x_1^*) + \frac{\partial^2 f}{\partial x_2^2}(x_2 - x_2^*)\right]\begin{bmatrix} x_1 - x_1^* \\ x_2 - x_2^* \end{bmatrix}$$

$$= \frac{1}{2}\begin{bmatrix} x_1 - x_1^* & x_2 - x_2^* \end{bmatrix} \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} \end{bmatrix} \begin{bmatrix} x_1 - x_1^* \\ x_2 - x_2^* \end{bmatrix}$$

[참고] 2변수 함수의 테일러 전개)(2)

- 2변수 함수 $f(x_1, x_2)$에 대한 점 (x_1^*, x_2^*)에서의 테일러 전개식(3차 이상의 고차항은 무시할 경우)

$$f(x_1,x_2) = f(x_1^*,x_2^*) + \frac{\partial f}{\partial x_1}(x_1 - x_1^*) + \frac{\partial f}{\partial x_2}(x_2 - x_2^*)$$

$$+\frac{1}{2}\left(\frac{\partial^2 f}{\partial x_1^2}(x_1 - x_1^*)^2 + 2\frac{\partial^2 f}{\partial x_1 \partial x_2}(x_1 - x_1^*)(x_2 - x_2^*) + \frac{\partial^2 f}{\partial x_2^2}(x_2 - x_2^*)^2\right)$$

$\mathbf{c} = \nabla f(\mathbf{x}^*),\ \mathbf{d} = (\mathbf{x} - \mathbf{x}^*)$라 가정

$$\Rightarrow f(\mathbf{x}) = f(\mathbf{x}^*) + \nabla f(\mathbf{x}^*)^T(\mathbf{x} - \mathbf{x}^*) + \frac{1}{2}(\mathbf{x} - \mathbf{x}^*)^T \mathbf{H}(\mathbf{x}^*)(\mathbf{x} - \mathbf{x}^*) = f(\mathbf{x}^*) + \mathbf{c}^T\mathbf{d} + \frac{1}{2}\mathbf{d}^T\mathbf{H}(\mathbf{x}^*)\,\mathbf{d}$$

$$\frac{\partial f(x_1,x_2)}{\partial x_1} = \frac{\partial f}{\partial x_1} + \frac{\partial^2 f}{\partial x_1^2}(x_1 - x_1^*) + \frac{\partial^2 f}{\partial x_1 \partial x_2}(x_2 - x_2^*),\quad \frac{\partial f(x_1,x_2)}{\partial x_2} = \frac{\partial f}{\partial x_2} + \frac{\partial^2 f}{\partial x_1 \partial x_2}(x_1 - x_1^*) + \frac{\partial^2 f}{\partial x_2^2}(x_2 - x_2^*)$$

$$\Rightarrow f'(\mathbf{x}) = \begin{bmatrix} \dfrac{\partial f(x_1,x_2)}{\partial x_1} \\ \dfrac{\partial f(x_1,x_2)}{\partial x_1} \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f}{\partial x_1} + \dfrac{\partial^2 f}{\partial x_1^2}(x_1 - x_1^*) + \dfrac{\partial^2 f}{\partial x_1 \partial x_2}(x_2 - x_2^*) \\ \dfrac{\partial f}{\partial x_2} + \dfrac{\partial^2 f}{\partial x_1 \partial x_2}(x_1 - x_1^*) + \dfrac{\partial^2 f}{\partial x_2^2}(x_2 - x_2^*) \end{bmatrix}$$

$$= \begin{bmatrix} \dfrac{\partial f}{\partial x_1} \\ \dfrac{\partial f}{\partial x_2} \end{bmatrix} + \begin{bmatrix} \dfrac{\partial^2 f}{\partial x_1^2}(x_1 - x_1^*) + \dfrac{\partial^2 f}{\partial x_1 \partial x_2}(x_2 - x_2^*) \\ \dfrac{\partial^2 f}{\partial x_1 \partial x_2}(x_1 - x_1^*) + \dfrac{\partial^2 f}{\partial x_2^2}(x_2 - x_2^*) \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f}{\partial x_1} \\ \dfrac{\partial f}{\partial x_2} \end{bmatrix} + \begin{bmatrix} \dfrac{\partial^2 f}{\partial x_1^2} & \dfrac{\partial^2 f}{\partial x_1 \partial x_2} \\ \dfrac{\partial^2 f}{\partial x_1 \partial x_2} & \dfrac{\partial^2 f}{\partial x_1^2} \end{bmatrix} \begin{bmatrix} (x_1 - x_1^*) \\ (x_2 - x_2^*) \end{bmatrix}$$

$$= \nabla f(\mathbf{x}^*) + \mathbf{H}(\mathbf{x}^*)(\mathbf{x} - \mathbf{x}^*) = \mathbf{c} + \mathbf{H}(\mathbf{x}^*)\,\mathbf{d} \longleftarrow \mathbf{c} = \nabla f(\mathbf{x}^*),\ \mathbf{d} = (\mathbf{x} - \mathbf{x}^*)\text{라 가정}$$

비제약 최적화 문제
- Newton 방법을 이용한 해법(1)

$$Minimize\ \ f(x_1,x_2)=x_1-x_2+2x_1^2+2x_1x_2+x_2^2$$

$$\nabla f(\mathbf{x})=\nabla f(x_1,x_2)=\begin{pmatrix} f_{x_1} \\ f_{x_2} \end{pmatrix}=\begin{pmatrix} 1+4x_1+2x_2 \\ -1+2x_1+2x_2 \end{pmatrix}$$

$$H(\mathbf{x})=\begin{pmatrix} \dfrac{\partial^2 f}{\partial x_1^2} & \dfrac{\partial^2 f}{\partial x_1 \partial x_2} \\ \dfrac{\partial^2 f}{\partial x_1 \partial x_2} & \dfrac{\partial^2 f}{\partial x_2^2} \end{pmatrix}=\begin{pmatrix} f_{x_1x_1} & f_{x_1x_2} \\ f_{x_1x_2} & f_{x_2x_2} \end{pmatrix}=\begin{pmatrix} 4 & 2 \\ 2 & 2 \end{pmatrix}$$

- **단계** 1 - $\mathbf{x}^{(1)}$구하기

이 방법은 출발점 $\mathbf{x}^{(k)}$에서 $\Delta\mathbf{x}^{(k)}$만큼 증가한점$\mathbf{x}^{(k+1)}=\mathbf{x}^{(k)}+\Delta\mathbf{x}^{(k)}$에서 함수 f가 최소값을갖는다고 가정한다.

여기서, $\Delta\mathbf{x}^{(0)}=\mathbf{x}^{(1)}-\mathbf{x}^{(0)}$

즉, $\Delta x_1^{(0)}=x_1^{(1)}-x_1^{(0)}$, $\Delta x_2^{(1)}=x_2^{(2)}-x_2^{(1)}$이다.

비제약 최적화 문제
- Newton 방법을 이용한 해법(2)

$$\begin{pmatrix} f_{x_1x_1} & f_{x_1x_2} \\ f_{x_1x_2} & f_{x_2x_2} \end{pmatrix}\begin{pmatrix} \Delta x_1^{(0)} \\ \Delta x_2^{(0)} \end{pmatrix} = -\begin{pmatrix} f_{x_1} \\ f_{x_2} \end{pmatrix}$$ 이고

$$f_{x_1} = 1 + 4x_1 + 2x_2$$

$$f_{x_2} = -1 + 2x_1 + 2x_2$$

$$f_{x_1x_1} = 4$$

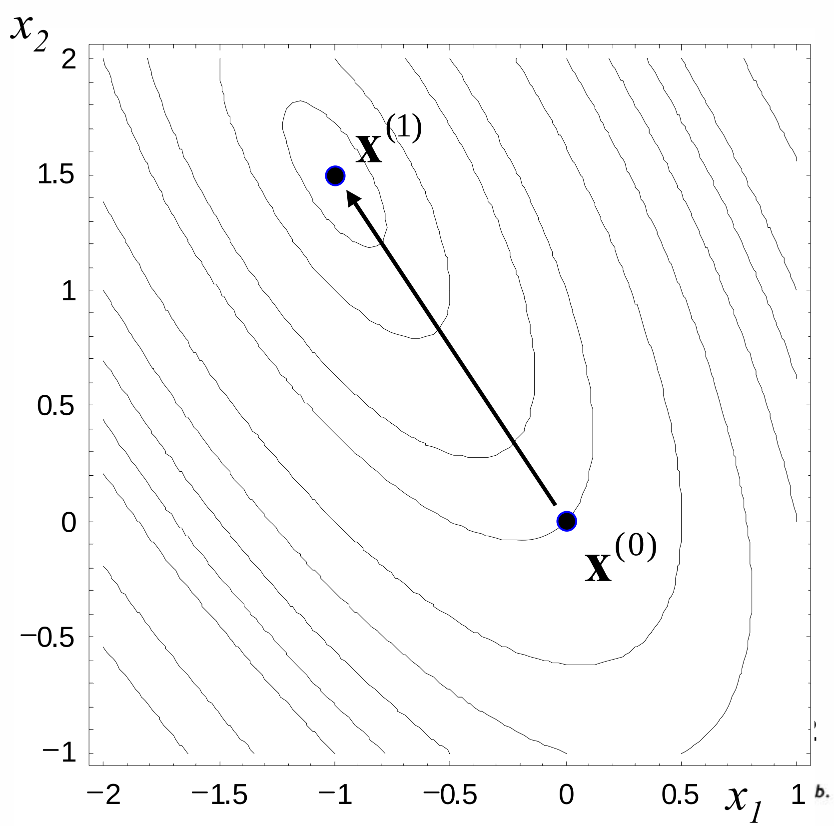
$$f_{x_2x_2} = 2$$

$f_{x_1x_2} = f_{x_2x_1} = 2$이므로

$$\begin{pmatrix} 4 & 2 \\ 2 & 2 \end{pmatrix}\begin{pmatrix} \Delta x_1^{(0)} \\ \Delta x_2^{(0)} \end{pmatrix} = -\begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

에서 $\Delta x_1^{(0)} = -1,\ \Delta x_2^{(0)} = 1.5$이다.

$$\therefore \mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \Delta\mathbf{x}^{(0)} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} + \begin{pmatrix} -1 \\ 1.5 \end{pmatrix} = \begin{pmatrix} -1 \\ 1.5 \end{pmatrix}$$



비제약 최적화 문제
- Newton 방법을 이용한 해법(3)

- **단계 2 - $\mathbf{x}^{(2)}$구하기**

단계 1과 같은 과정을 거치면

$$\begin{pmatrix} f_{x_1x_1} & f_{x_1x_2} \\ f_{x_1x_2} & f_{x_2x_2} \end{pmatrix}\begin{pmatrix} \Delta x_1^{(1)} \\ \Delta x_2^{(1)} \end{pmatrix} = -\begin{pmatrix} f_{x_1} \\ f_{x_2} \end{pmatrix}$$

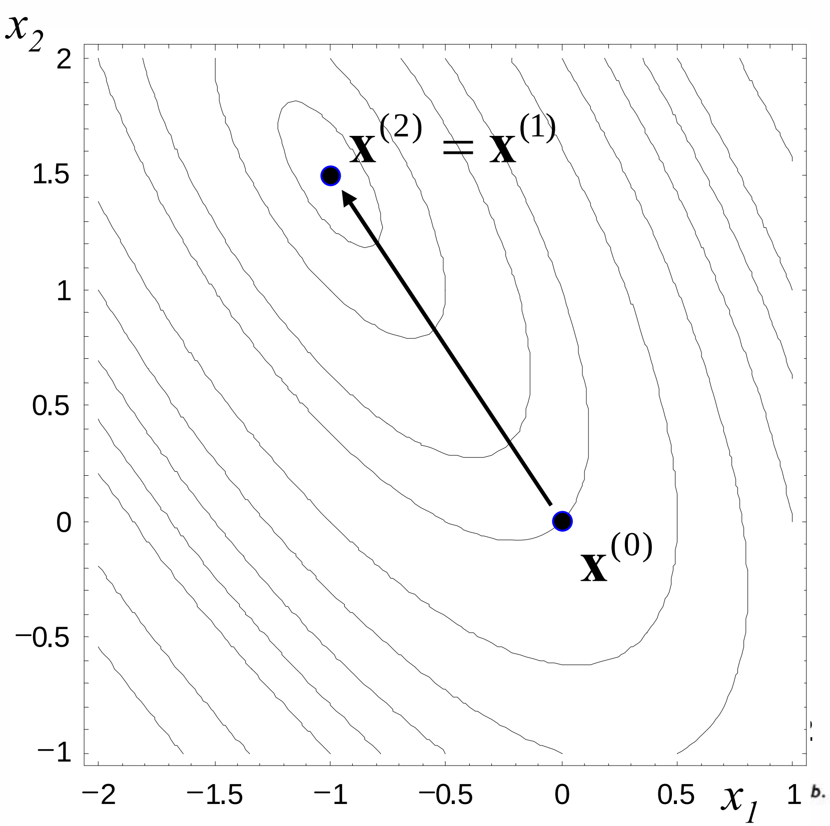
$$\begin{pmatrix} 4 & 2 \\ 2 & 2 \end{pmatrix}\begin{pmatrix} \Delta x_1^{(1)} \\ \Delta x_2^{(1)} \end{pmatrix} = -\begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

에서 $\Delta x_1^{(1)} = 0,\ \Delta x_2^{(1)} = 0$이다.

$$\therefore \mathbf{x}^{(2)} = \mathbf{x}^{(1)} + \Delta\mathbf{x}^{(1)} = \begin{pmatrix} -1 \\ 1.5 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} -1 \\ 1.5 \end{pmatrix}$$

따라서 $\left|\mathbf{x}^{(2)} - \mathbf{x}^{(1)}\right| \le \varepsilon = 0.001$이므로

$\mathbf{x}^{(2)}$이 최적해가 된다.



Gradient 방법
- Davidon-Fletcher-Powell(DFP) 방법(1/2)

- 1차 미분만을 이용하여 $f(\mathbf{x})$의 역 헷세 행렬을 근사적으로 계산하는 방법
  - 단계 1 : 시작점 $\mathbf{x}^{(0)}$를 추정한다.
    목적 함수의 역 헷세 행렬을 근사적으로 추정하기 위해 대칭의 양정 행렬 $\mathbf{A}^{(0)}$을 선정한다. 처음 시작할 때는 $\mathbf{A}^{(0)} = \mathbf{I}$ 를 선택할 수 있다. 수렴 매개 변수 ε을 정하고, $k = 0$ 이라 두고 Gradient 벡터를 다음과 같이 계산한다.

$$\mathbf{d}^{(0)} = \mathbf{c}^{(0)} \equiv \nabla f(\mathbf{x}^{(0)})$$

  - 단계 2 : Gradient 벡터의 Norm $\left\|\mathbf{c}^{(k)}\right\|$를 계산한다.
    만일 $\left\|\mathbf{c}^{(k)}\right\| < \varepsilon$ 이면 반복 과정을 멈추고 그렇지 않으면 다음 단계로 계속 진행한다. 이 방법의 첫 단계는 최속 강하법과 같다. 즉, 첫 단계에서는 단계 4로 간다.
  - 단계 3 : 탐색 방향을 다음과 같이 계산한다.

$$\boxed{\mathbf{d}^{(k)} = -\mathbf{A}^{(k)}\mathbf{c}^{(k)}}$$

    즉, Newton 방법의 $\mathbf{H}^{-1}$을 근사적으로 $\mathbf{A}$로 대체한다.

Gradient 방법
- Davidon-Fletcher-Powell(DFP) 방법(2/2)

- 단계 4 : $f(\mathbf{x}^{(k)}+\alpha\mathbf{d}^{(k)})$를 최소화하는 이동 거리 $\alpha_k=\alpha$를 계산한다.
- 단계 5 : 설계점을 $\mathbf{x}^{(k+1)}=\mathbf{x}^{(k)}+\alpha_k\mathbf{d}^{(k)}$로 수정한다.
- 단계 6 : 목적 함수의 근사적 역 헷세 행렬 $\mathbf{A}^{(k)}$를 다음과 같이 보정한다.

$$\mathbf{A}^{(k+1)}=\mathbf{A}^{(k)}+\mathbf{B}^{(k)}+\mathbf{C}^{(k)} \quad ; \quad n\times n \text{ 행렬}$$

여기서, 보정 행렬 $\mathbf{B}^{(k)}$, $\mathbf{C}^{(k)}$는 다음과 같이 계산한다.

$$\mathbf{B}^{(k)}=\frac{\mathbf{s}^{(k)}(\mathbf{s}^{(k)})^T}{(\mathbf{s}^{(k)}\cdot\mathbf{y}^{(k)})} \quad ; \quad n\times n \text{ 행렬} \qquad \mathbf{C}^{(k)}=\frac{-\mathbf{z}^{(k)}(\mathbf{z}^{(k)})^T}{(\mathbf{y}^{(k)}\cdot\mathbf{z}^{(k)})} \quad ; \quad n\times n \text{ 행렬}$$

$$\mathbf{s}^{(k)}=\alpha_k\mathbf{d}^{(k)} \quad ; \quad n\times 1 \text{ 벡터}$$

$$\mathbf{y}^{(k)}=\mathbf{c}^{(k+1)}-\mathbf{c}^{(k)} \quad ; \quad n\times 1 \text{ 벡터}$$

$$\mathbf{c}^{(k+1)}=\nabla f(\mathbf{x}^{(k+1)}) \quad ; \quad n\times 1 \text{ 벡터}$$

$$\mathbf{z}^{(k)}=\mathbf{A}^{(k)}\mathbf{y}^{(k)} \quad ; \quad [n\times n \text{ 행렬}][n\times 1 \text{ 벡터}]=n\times 1 \text{ 행렬}$$

- 단계 7 : $k=k+1$로 두고 단계 2를 수행한다.

비제약 최적화 문제
- Davidon-Fletcher-Powell(DFP) 방법을 이용한 해법(1)

$$f(x_1, x_2) = x_1 - x_2 + 2x_1^2 + 2x_1x_2 + x_2^2$$

$$\nabla f(\mathbf{x}) = \nabla f(x_1, x_2) = \begin{pmatrix} 1 + 4x_1 + 2x_2 \\ -1 + 2x_1 + 2x_2 \end{pmatrix}$$

- **단계** 1 - $\mathbf{x}^{(1)}$구하기

$$\mathbf{x}^{(0)} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad \mathbf{A}^{(0)} = \mathbf{I}$$

$$\mathbf{c}^{(0)} = \begin{pmatrix} 1 + 4x_1 + 2x_2 \\ -1 + 2x_1 + 2x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

$$\left\| \mathbf{c}^{(0)} \right\| = \sqrt{1^2 + (-1)^2} > \varepsilon = 0.001$$

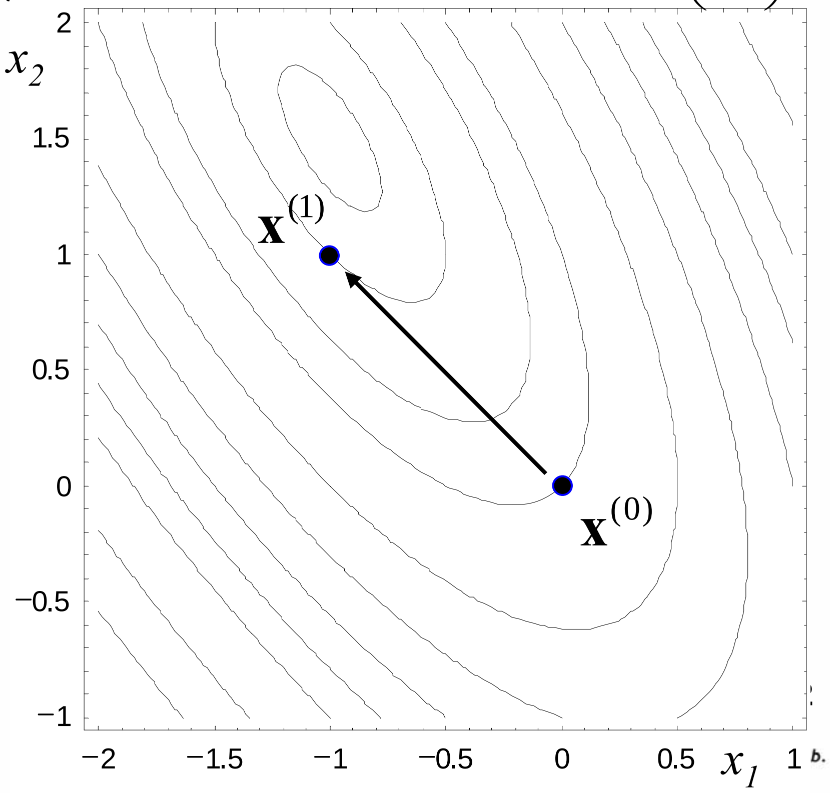
$$\mathbf{d}^{(0)} = -\mathbf{A}^{(0)}\mathbf{c}^{(0)} = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$$

$$\mathbf{x}^{(1)} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} + \alpha \begin{pmatrix} -1 \\ 1 \end{pmatrix} = \begin{pmatrix} -\alpha \\ \alpha \end{pmatrix}$$

$$f(\mathbf{x}^{(1)}) = \alpha^2 - 2\alpha$$

함수 $f(\mathbf{x}^{(1)})$가 최소값을 가질 조건

$$\frac{df(\mathbf{x}^{(1)})}{d\alpha} = 0\text{으로부터 } \alpha = 1 \quad \therefore \mathbf{x}^{(1)} = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$$



비제약 최적화 문제
- Davidon-Fletcher-Powell(DFP) 방법을 이용한 해법(2)

$$\mathbf{s}^{(0)} = \alpha\mathbf{d}^{(0)} = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$$

$$\mathbf{c}^{(1)} = \begin{pmatrix} -1 \\ -1 \end{pmatrix}$$

$$\mathbf{y}^{(0)} = \mathbf{c}^{(1)} - \mathbf{c}^{(0)} = \begin{pmatrix} -2 \\ 0 \end{pmatrix}$$

$$\mathbf{z}^{(0)} = \mathbf{A}^{(0)}\mathbf{y}^{(0)} = \begin{pmatrix} -2 \\ 0 \end{pmatrix}$$

$$\mathbf{s}^{(0)} \cdot \mathbf{y}^{(0)} = 2$$

$$\mathbf{y}^{(0)} \cdot \mathbf{z}^{(0)} = 4$$

$$\mathbf{s}^{(0)}\mathbf{s}^{(0)^T} = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$$

$$\mathbf{B}^{(0)} = \frac{\mathbf{s}^{(0)}\mathbf{s}^{(0)^T}}{\mathbf{s}^{(0)} \cdot \mathbf{y}^{(0)}} = \begin{pmatrix} 0.5 & -0.5 \\ -0.5 & 0.5 \end{pmatrix}$$

$$\mathbf{z}^{(0)}\mathbf{z}^{(0)^T} = \begin{pmatrix} 4 & 0 \\ 0 & 0 \end{pmatrix}$$

$$\mathbf{C}^{(0)} = \frac{-\mathbf{z}^{(0)}\mathbf{z}^{(0)^T}}{\mathbf{y}^{(0)} \cdot \mathbf{z}^{(0)}} = \begin{pmatrix} -1 & 0 \\ 0 & 0 \end{pmatrix}$$

$$\mathbf{A}^{(0)} = \mathbf{A}^{(0)} + \mathbf{B}^{(0)} + \mathbf{C}^{(0)} = \begin{pmatrix} 0.5 & -0.5 \\ -0.5 & 1.5 \end{pmatrix}$$

비제약 최적화 문제
- Davidon-Fletcher-Powell(DFP) 방법을 이용한 해법(3)

- **단계 2 - $\mathbf{x}^{(2)}$구하기**

$$\left\|\mathbf{c}^{(1)}\right\| = \sqrt{2} > \varepsilon$$

$$\mathbf{d}^{(1)} = -\mathbf{A}^{(1)}\mathbf{c}^{(1)} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$\mathbf{x}^{(2)} = \begin{pmatrix} -1 \\ 1 \end{pmatrix} + \alpha \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} -1 \\ 1+\alpha \end{pmatrix}$$

$$f(\mathbf{x}^{(2)}) = \alpha^2 - \alpha - 1$$

함수 $f(\mathbf{x}^{(2)})$가 최소값을 가질 조건

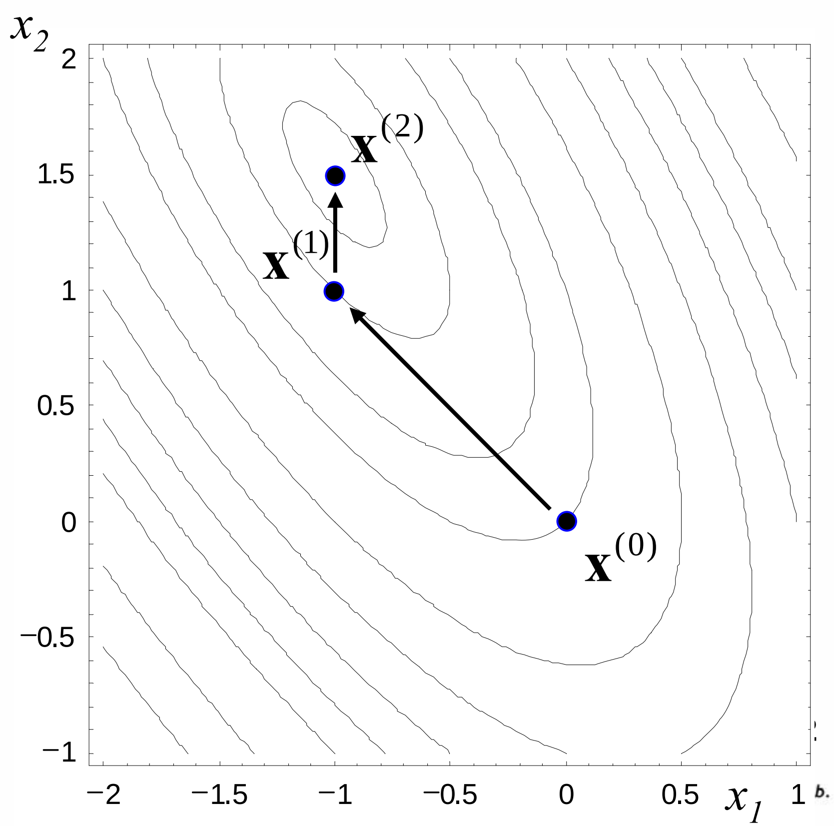
$\dfrac{df(\mathbf{x}^{(2)})}{d\alpha} = 0$으로부터 $\alpha = 0.5$

$$\therefore \mathbf{x}^{(2)} = \begin{pmatrix} -1 \\ 1.5 \end{pmatrix}$$

$$\mathbf{s}^{(1)} = \alpha \mathbf{d}^{(1)} = \begin{pmatrix} 0 \\ 0.5 \end{pmatrix} \qquad \mathbf{c}^{(2)} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$\left\|\mathbf{c}^{(2)}\right\| = 0 \le \varepsilon$이므로

$\mathbf{x}^{(2)}$가 최적점이 된다.



Gradient 방법
- Broyden-Fletcher-Goldfarb-Shanno(BFGS) 방법(1/2)

- DFP 방법은 매 반복 단계에서 역 헷세 행렬을 근사적으로 계산하는데 비해, BFGS 방법은 헷세 행렬을 수정하는 방법
  - 단계 1 : 시작점 $\mathbf{x}^{(0)}$을 추정한다.
    목적 함수의 헷세 행렬을 근사적으로 추정하기 위하여 대칭의 양정 행렬 $\mathbf{H}^{(0)}$를 선정한다. 처음 시작할 때는 $\mathbf{H}^{(0)} = \mathbf{I}$를 선택할 수 있다.
    수렴 매개 변수 ε을 정하고, $k = 0$이라 두고 Gradient 벡터를 계산한다.

$$\mathbf{c}^{(0)} = \nabla f(\mathbf{x}^{(0)})$$

  - 단계 2 : Gradient 벡터의 Norm $\left\|\mathbf{c}^{(k)}\right\|$를 계산한다.
    $\left\|\mathbf{c}^{(k)}\right\| < \varepsilon$이면 반복 과정을 멈추고 아니면 다음 단계를 수행한다.
  - 단계 3 : 탐색 방향 $\mathbf{d}^{(k)}$를 구하기 위해 다음의 선형 방정식을 푼다.

$$\boxed{\mathbf{H}^{(k)}\mathbf{d}^{(k)} = -\mathbf{c}^{(k)}}$$

    수학적으로는 Newton 방법의 $\mathbf{d}^{(k)} = -\mathbf{H}^{-1(k)}\mathbf{c}^{(k)}$와 같으나 실제로는 $\mathbf{H}$를 근사적으로 보정한다.

Gradient 방법
- Broyden-Fletcher-Goldfarb-Shanno(BFGS) 방법(2/2)

- 단계 4 : $f(\mathbf{x}^{(k)} + \alpha \mathbf{d}^{(k)})$를 최소화 하는 이동 거리 $\alpha_k = \alpha$ 를 계산한다.
- 단계 5 : 설계점을 $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{d}^{(k)}$ 로 수정한다.
- 단계 6 : 목적 함수의 근사적 헷세 행렬 $\mathbf{H}^{(k)}$를 다음과 같이 보정한다.

$$\mathbf{H}^{(k+1)} = \mathbf{H}^{(k)} + \mathbf{D}^{(k)} + \mathbf{E}^{(k)} \qquad ; \ n \times n \text{ 행렬}$$

여기서, 보정 행렬 $D^{(k)}$ 와 $E^{(k)}$는 다음과 같이 계산한다.

$$\mathbf{D}^{(k)} = \frac{\mathbf{y}^{(k)}(\mathbf{y}^{(k)})^T}{(\mathbf{y}^{(k)} \cdot \mathbf{s}^{(k)})} \qquad ; \text{ 설계점의 변화}$$

$$\mathbf{E}^{(k)} = \frac{\mathbf{c}^{(k)}(\mathbf{c}^{(k)})^T}{(\mathbf{c}^{(k)} \cdot \mathbf{d}^{(k)})} \qquad ; \text{ 경사도의 변화}$$

$$\mathbf{s}^{(k)} = \alpha_k \mathbf{d}^{(k)}$$

$$\mathbf{y}^{(k)} = \mathbf{c}^{(k+1)} - \mathbf{c}^{(k)}$$

$$\mathbf{c}^{(k+1)} = \nabla f(\mathbf{x}^{(k+1)})$$

- 단계 7 : $k = k + 1$ 로 두고 단계 2를 수행한다.

비제약 최적화 문제
- Broydon-Fletcher-Goldfarb-Shanno(BFGS) 방법을 이용한 해법(1)

$$f(x_1, x_2) = x_1 - x_2 + 2x_1^2 + 2x_1x_2 + x_2^2$$

$$\nabla f(\mathbf{x}) = \nabla f(x_1, x_2) = \begin{pmatrix} 1 + 4x_1 + 2x_2 \\ -1 + 2x_1 + 2x_2 \end{pmatrix}$$

- **단계** 1 - $\mathbf{x}^{(1)}$구하기

$$\mathbf{x}^{(0)} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \ \mathbf{H}^{(0)} = \mathbf{I}$$

$$\mathbf{c}^{(0)} = \begin{pmatrix} 1 + 4x_1 + 2x_2 \\ -1 + 2x_1 + 2x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

$$\left\| \mathbf{c}^{(0)} \right\| = \sqrt{1^2 + (-1)^2} > \varepsilon = 0.001$$

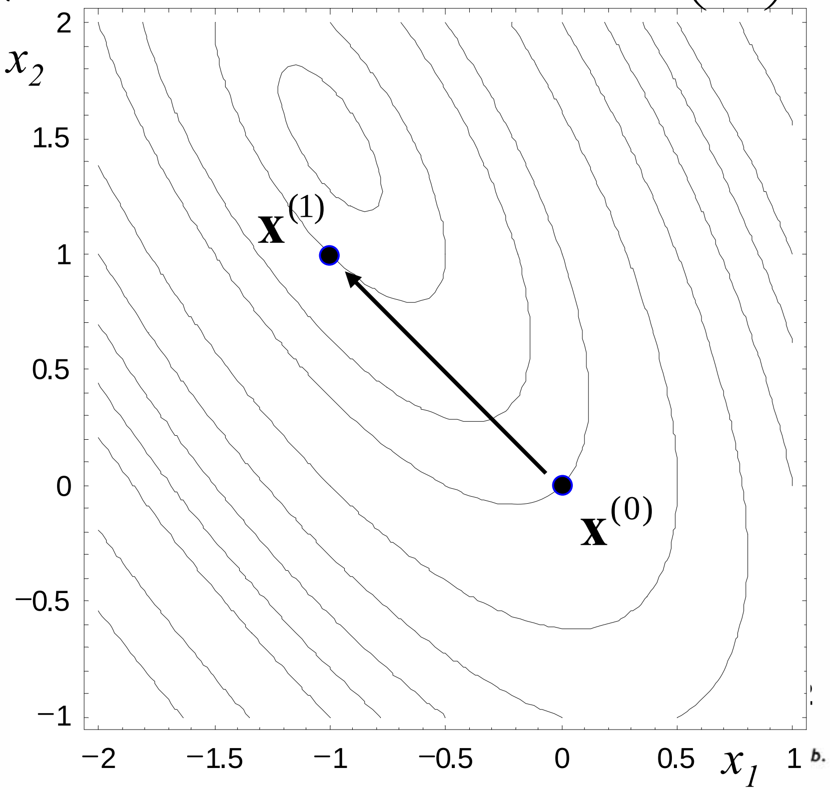
$$\mathbf{d}^{(0)} = -\mathbf{H}^{-1(0)}\mathbf{c}^{(0)} = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$$

$$\mathbf{x}^{(1)} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} + \alpha \begin{pmatrix} -1 \\ 1 \end{pmatrix} = \begin{pmatrix} -\alpha \\ \alpha \end{pmatrix}$$

$$f(\mathbf{x}^{(1)}) = \alpha^2 - 2\alpha$$

함수 $f(\mathbf{x}^{(1)})$가 최소값을 가질 조건

$\dfrac{df(\mathbf{x}^{(1)})}{d\alpha} = 0$으로부터 $\alpha = 1 \quad \therefore \mathbf{x}^{(1)} = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$



비제약 최적화 문제
- Broydon-Fletcher-Goldfarb-Shanno(BFGS) 방법을 이용한 해법(2)

$$\mathbf{s}^{(0)} = \alpha \mathbf{d}^{(0)} = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$$

$$\mathbf{c}^{(1)} = \begin{pmatrix} -1 \\ -1 \end{pmatrix} \qquad \nabla f(\mathbf{x}^{(1)}) = \mathbf{c}^{(1)}$$

$$\mathbf{y}^{(0)} = \mathbf{c}^{(1)} - \mathbf{c}^{(0)} = \begin{pmatrix} -2 \\ 0 \end{pmatrix}$$

$$\mathbf{y}^{(0)} \cdot \mathbf{s}^{(0)} = (-2,0)\cdot(-1,1) = 2$$

$$\mathbf{c}^{(0)} \cdot \mathbf{d}^{(0)} = (1,-1)\cdot(-1,1) = -2$$

$$\mathbf{y}^{(0)}\mathbf{y}^{(0)^T} = \begin{pmatrix} -2 \\ 0 \end{pmatrix}\begin{pmatrix} -2 & 0 \end{pmatrix} = \begin{pmatrix} 4 & 0 \\ 0 & 0 \end{pmatrix}$$

$$\mathbf{D}^{(0)} = \frac{\mathbf{y}^{(0)}\mathbf{y}^{(0)^T}}{\mathbf{y}^{(0)} \cdot \mathbf{s}^{(0)}} = \begin{pmatrix} 2 & 0 \\ 0 & 0 \end{pmatrix}$$

$$\mathbf{c}^{(0)}\mathbf{c}^{(0)^T} = \begin{pmatrix} 1 \\ -1 \end{pmatrix}\begin{pmatrix} 1 & -1 \end{pmatrix} = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$$

$$\mathbf{E}^{(0)} = \frac{\mathbf{c}^{(0)}\mathbf{c}^{(0)^T}}{\mathbf{c}^{(0)} \cdot \mathbf{d}^{(0)}} = \begin{pmatrix} -0.5 & 0.5 \\ 0.5 & -0.5 \end{pmatrix}$$

$$\mathbf{H}^{(1)} = \mathbf{H}^{(0)} + \mathbf{D}^{(0)} + \mathbf{E}^{(0)} = \begin{pmatrix} 2.5 & 0.5 \\ 0.5 & 0.5 \end{pmatrix}$$

비제약 최적화 문제
- Broydon-Fletcher-Goldfarb-Shanno(BFGS) 방법을 이용한 해법(3)

- **단계** 2 - $\mathbf{x}^{(2)}$구하기

$\left\|\mathbf{c}^{(1)}\right\| = \sqrt{2} > \varepsilon$

$\mathbf{H}^{(1)}\mathbf{d}^{(1)} = -\mathbf{c}^{(1)}$에서 $\mathbf{d}^{(1)} = \begin{pmatrix} 0 \\ 2 \end{pmatrix}$

$$\mathbf{x}^{(2)} = \begin{pmatrix} -1 \\ 1 \end{pmatrix} + \alpha \begin{pmatrix} 0 \\ 2 \end{pmatrix} = \begin{pmatrix} -1 \\ 1+2\alpha \end{pmatrix}$$

$$f(\mathbf{x}^{(2)}) = 4\alpha^2 - 2\alpha - 1$$

함수 $f(\mathbf{x}^{(2)})$가 최소값을 가질 조건

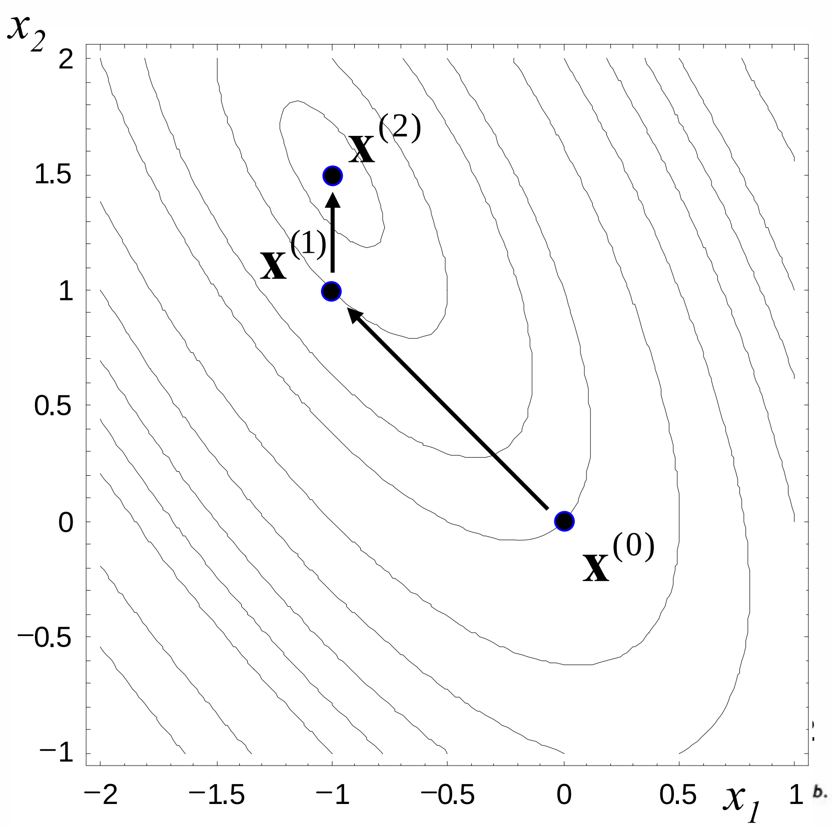
$\dfrac{df(\mathbf{x}^{(2)})}{d\alpha} = 0$으로부터 $\alpha = 0.25$

$$\therefore \mathbf{x}^{(2)} = \begin{pmatrix} -1 \\ 1.5 \end{pmatrix}$$

$\mathbf{s}^{(1)} = \alpha \mathbf{d}^{(1)} = \begin{pmatrix} 0 \\ 0.5 \end{pmatrix}$ $\quad \mathbf{c}^{(2)} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$

$\left\|\mathbf{c}^{(2)}\right\| = 0 \le \varepsilon$이므로

$\mathbf{x}^{(2)}$가 최적점이 된다.

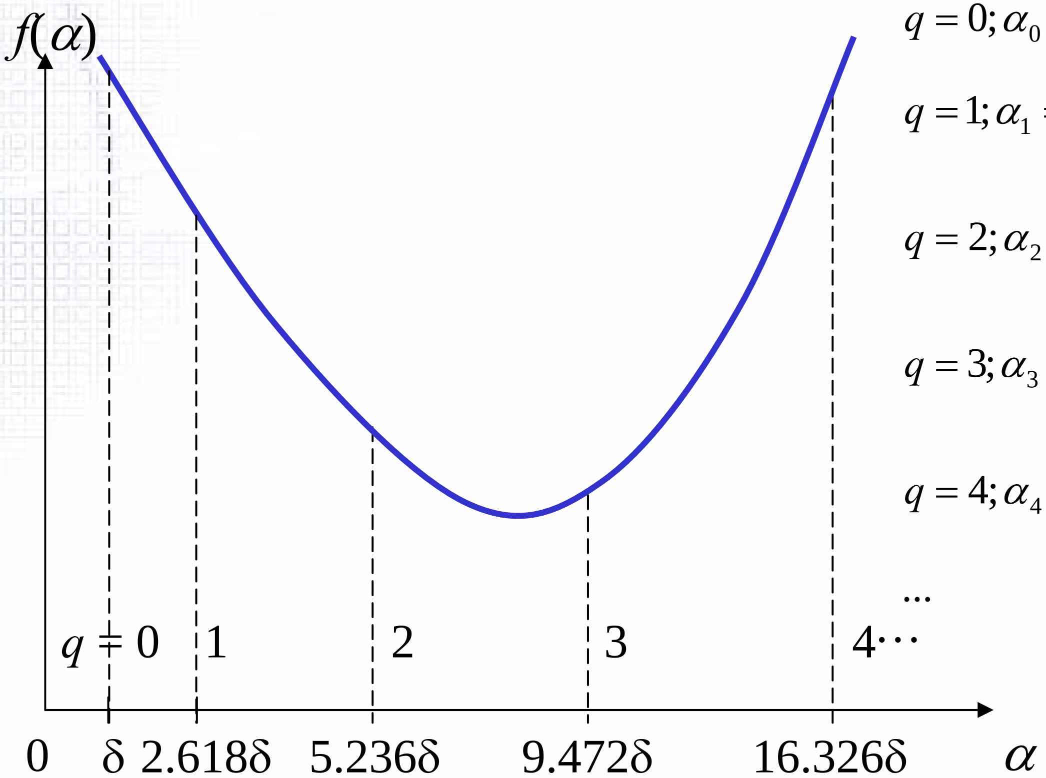


2.2 황금 분할에 의한 1차원 탐색 방법("황금 분할법")

Advanced
Ship
Design
Automation
Laboratory

황금 분할에 의한 1차원 탐색 방법(1)

- 최소값이 위치하는 구간 탐색
  - 황금 분할법은 최소값이 위치하는 초기 구간을 알고 시작해야 한다.
  - $\alpha = 0$과 $\alpha = \delta$에서의 함수값 $f(0)$과 $f(\delta)$를 계산하여 $f(\delta)$ 이 $f(0)$보다 작으면 이동량을 1.618δ의 증가치를 택한다.
    즉, 증분이 이전 증가의 1.618배이다.



$$q = 0; \alpha_0 = \delta$$

$$q = 1; \alpha_1 = \delta + 1.618\delta = 2.618\delta = \sum_{j=0}^{1} \delta(1.618)^j$$

$$q = 2; \alpha_2 = 2.618\delta + 1.618(1.618\delta) = 5.236\delta = \sum_{j=0}^{2} \delta(1.618)^j$$

$$q = 3; \alpha_3 = 5.236\delta + (1.618)^3\delta = 9.472\delta = \sum_{j=0}^{3} \delta(1.618)^j$$

$$q = 4; \alpha_4 = 9.472\delta + (1.618)^4\delta = 16.326\delta = \sum_{j=0}^{4} \delta(1.618)^j$$

...

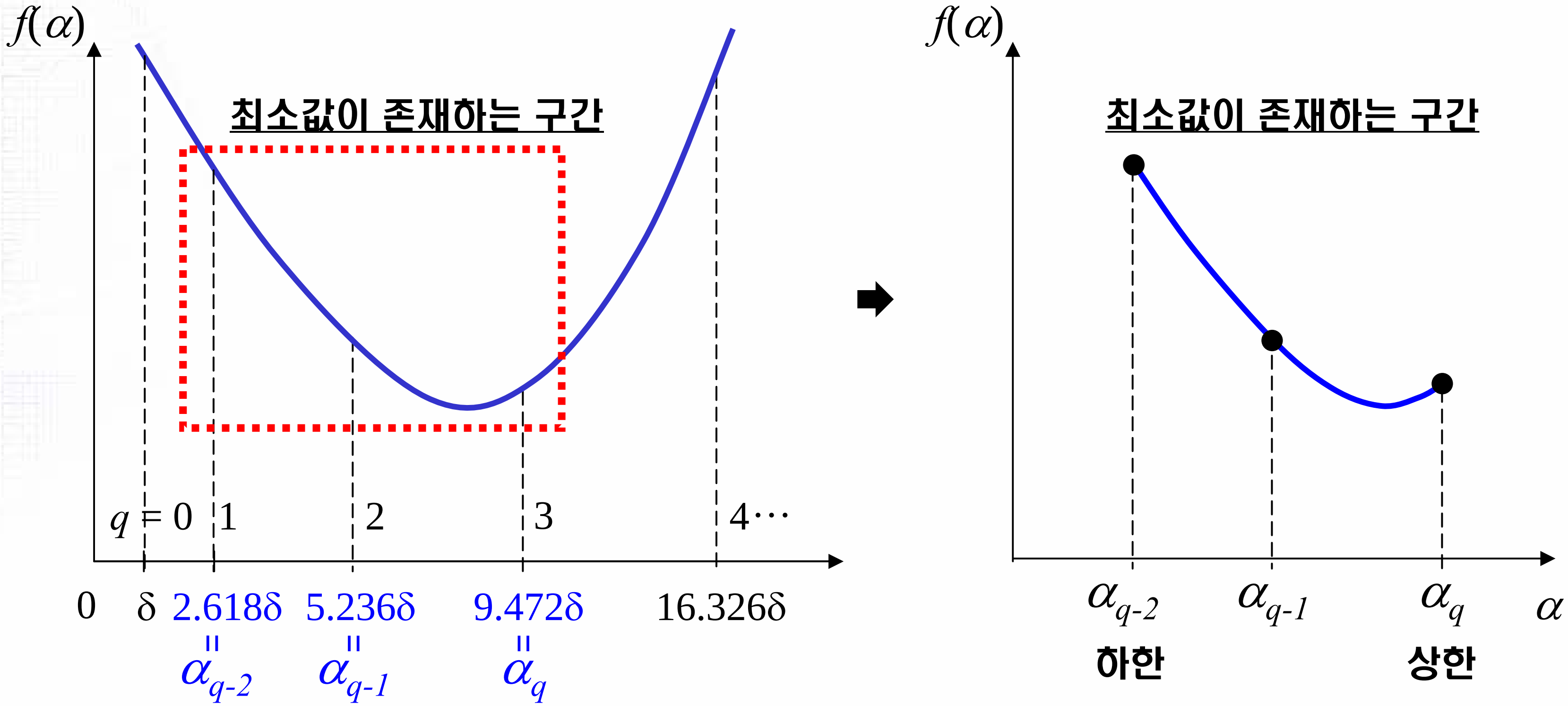
$$\boxed{\therefore \alpha_q = \sum_{j=0}^{q} \delta(1.618)^j, \; q = 0, 1, 2, \ldots}$$

황금 분할에 의한 1차원 탐색 방법(2)

- α_{q-1} 에서의 함수값이 α_{q-2} 와 α_q 에서의 값보다 작다면,

$$f(\alpha_{q-1}) < f(\alpha_{q-2}), \quad f(\alpha_{q-1}) < f(\alpha_q)$$

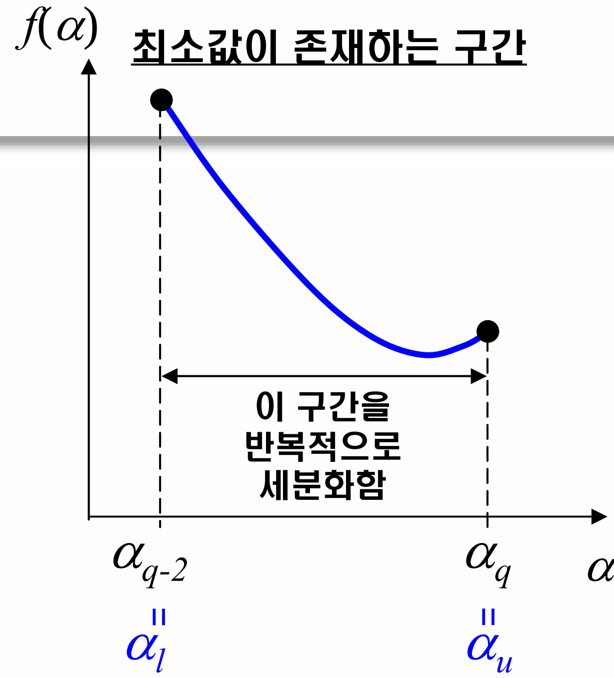
최소점은 두 구간, 즉 α_q 와 α_{q-2} 사이에 있다.



- 최소점이 존재하는 구간의 상한과 하한은

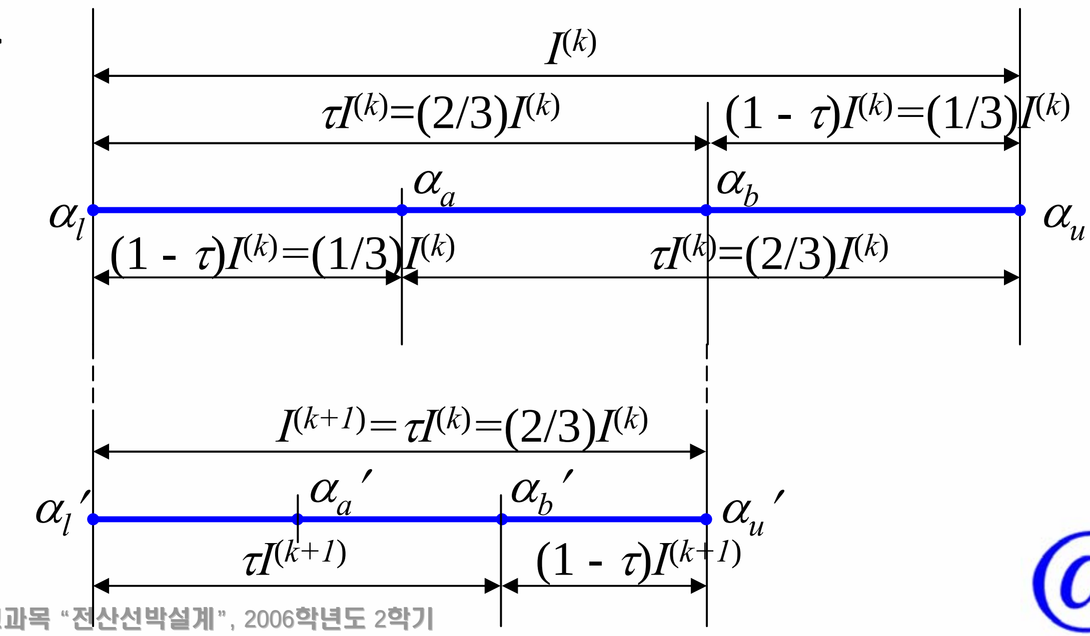
$$\underbrace{\alpha_u \equiv \alpha_q = \sum_{j=0}^{q} \delta(1.618)^j}_{\text{상한}}, \quad \underbrace{\alpha_l \equiv \alpha_{q-2} = \sum_{j=0}^{q-2} \delta(1.618)^j}_{\text{하한}}$$

황금 분할에 의한 1차원 탐색 방법(3)



- 최소값이 위치하는 구간의 세분화
  - 최소값이 위치하는 구간 $I^{(k)}$를 τ : 1-τ 로 내분하는 점을 양끝으로부터 각각 구한다.
  - 구한 점들에서의 함수값을 비교하여 구간을 세분화한다.
    - 아래 그림과 같이 양끝 점에서 대칭으로 위치한 두 점으로부터 같은 간격 $\tau I^{(k)}$만큼 떨어진 점 α_a 와 α_b 를 잡는다.
    - 그 점에서의 함수값을 계산하여 큰 쪽의 구간을 버리고, 남은 구간을 새로운 구간으로 정한다.

<τ = 2/3일 경우의 예>



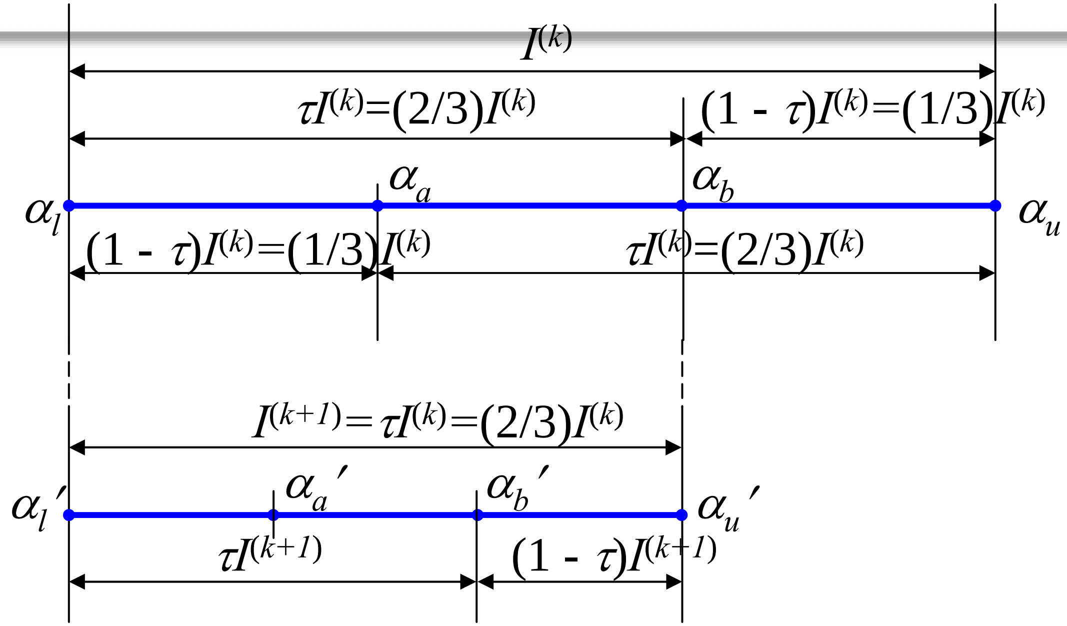
$f(\alpha_a) < f(\alpha_a)$라고 가정하면 최소값이 존재하는 구간은 다음과 같이 변경됨

황금 분할에 의한 1차원 탐색 방법(4)

<τ = 2/3일 경우의 예>

(a)

(b)



최소값이 존재하는 구간이 변경될 때마다
$f(\alpha_a'), f(\alpha_b')$에서의 함수값을 새로 구해야 함

<문제 제기>

이전 구간에서의 함수값을 활용할 수 없을까?

$I^{(k+1)}=\tau I^{(k)}$이고 만약 $\tau I^{(k+1)}=(1-\tau)I^{(k)}$이 되도록
τ를 정한다면 $\alpha_b'=\alpha_a$이므로
$f(\alpha_b')$는 새 구간에서 계산할 필요 없이
이전 구간에서의 $f(\alpha_a)$을 그대로
사용할 수 있음

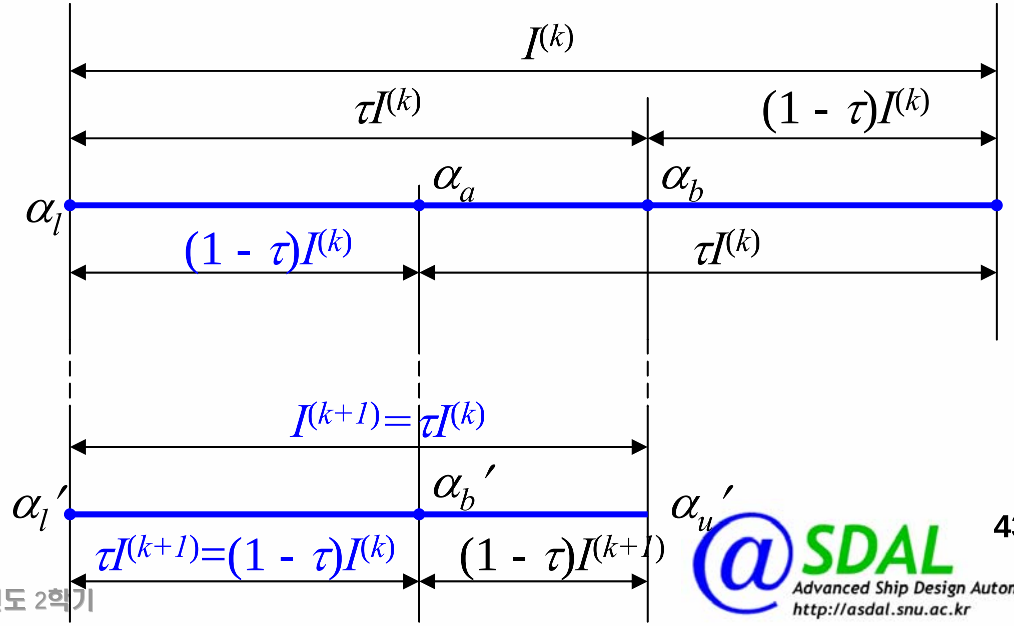
$$\left.\begin{aligned} I^{(k+1)} &= \tau I^{(k)} \\ \tau I^{(k+1)} &= (1-\tau)I^{(k)} \end{aligned}\right\}$$ 연립해서 품

$$\tau\cdot\tau I-(1-\tau)I=0,\ \ \tau^2+\tau-1=0$$

$$\tau=\frac{-1\pm\sqrt{5}}{2}=0.618,-1.618\Rightarrow 0.618$$

(a)

(b)



황금 분할에 의한 1차원 탐색 알고리즘(1)

- **단계** 1 : **최소점이 존재하는 구간 탐색**
  α **에서 미소의 이동량** δ**를 선정하고** $f(\alpha_{q-1}) < f(\alpha_{q-2}), f(\alpha_{q-1}) < f(\alpha_q)$ **을 만족하는** q**를 정하고**, **다음의 식으로 상한과 하한을 결정한다**.

$$\alpha_u \equiv \alpha_q = \sum_{j=0}^{q} \delta(1.618)^j, \alpha_l \equiv \alpha_{q-2} = \sum_{j=0}^{q-2} \delta(1.618)^j$$

  여기서, **간격** $I = \alpha_u - \alpha_l$ **을 정한다**.

- **단계** 2 : $f(\alpha_a)$ **와** $f(\alpha_b)$**를 계산한다**. **이 때**, $\alpha_a = \alpha_l + 0.382I$ **이고** $\alpha_b = \alpha_l + 0.618I$**이다**.

- **단계** 3 : $f(\alpha_a)$ **와** $f(\alpha_b)$**를 비교하여 그 결과에 따라 단계** 4, 5, 6**으로 간다**.

황금 분할에 의한 1차원 탐색 알고리즘(2)

- 단계 4 : $f(\alpha_a) < f(\alpha_b)$ 이면, 최적점 α^* 는 α_l과 α_b 사이에 있다. 세분화된 구간의 새로운 한계는 $\alpha_l' = \alpha_l$ 이고 $\alpha_u' = \alpha_b$ 이다. 또한 $\alpha_b' = \alpha_a$ 이다. $\alpha_a' = \alpha_l' + 0.382(\alpha_u' - \alpha_l')$ 에서 $f(\alpha_a')$ 를 계산하고 단계 7로 간다.

- 단계 5 : $f(\alpha_a) > f(\alpha_b)$ 이면, 최적점 α^* 는 α_a과 α_u 사이에 있다. 세분화된 구간의 새로운 한계는 $\alpha_l' = \alpha_a$ 이고 $\alpha_u' = \alpha_u$ 이다. 또한 $\alpha_a' = \alpha_b$ 이다. $\alpha_b' = \alpha_l' + 0.618(\alpha_u' - \alpha_l')$ 에서 $f(\alpha_b')$ 를 계산하고 단계 7로 간다.

- 단계 6 : $f(\alpha_a) = f(\alpha_b)$ 이면, $\alpha_l = \alpha_a, \alpha_u = \alpha_b$ 라 두고 단계 7로 간다.

- 단계 7 : 만일 새로운 세분화된 구간 $(I' = \alpha_u' - \alpha_l')$ 이 충분히 작아서 수렴 기준을 만족하면(즉, $I' < \varepsilon$), $\alpha^* = (\alpha_u' - \alpha_l')/2$ 라 두고 최적화 과정을 마친다. 그렇지 않으면, α_l', α_a', α_b', α_u'의 Prime(') 부호를 삭제하고 단계 3으로 돌아간다.

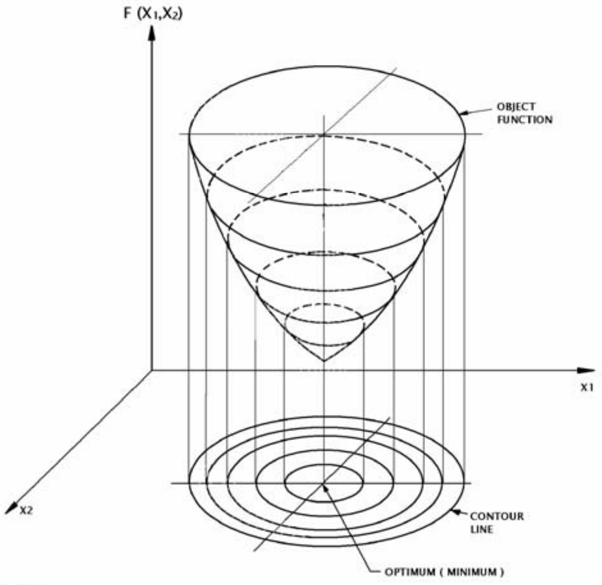
황금 분할법의 구현 예

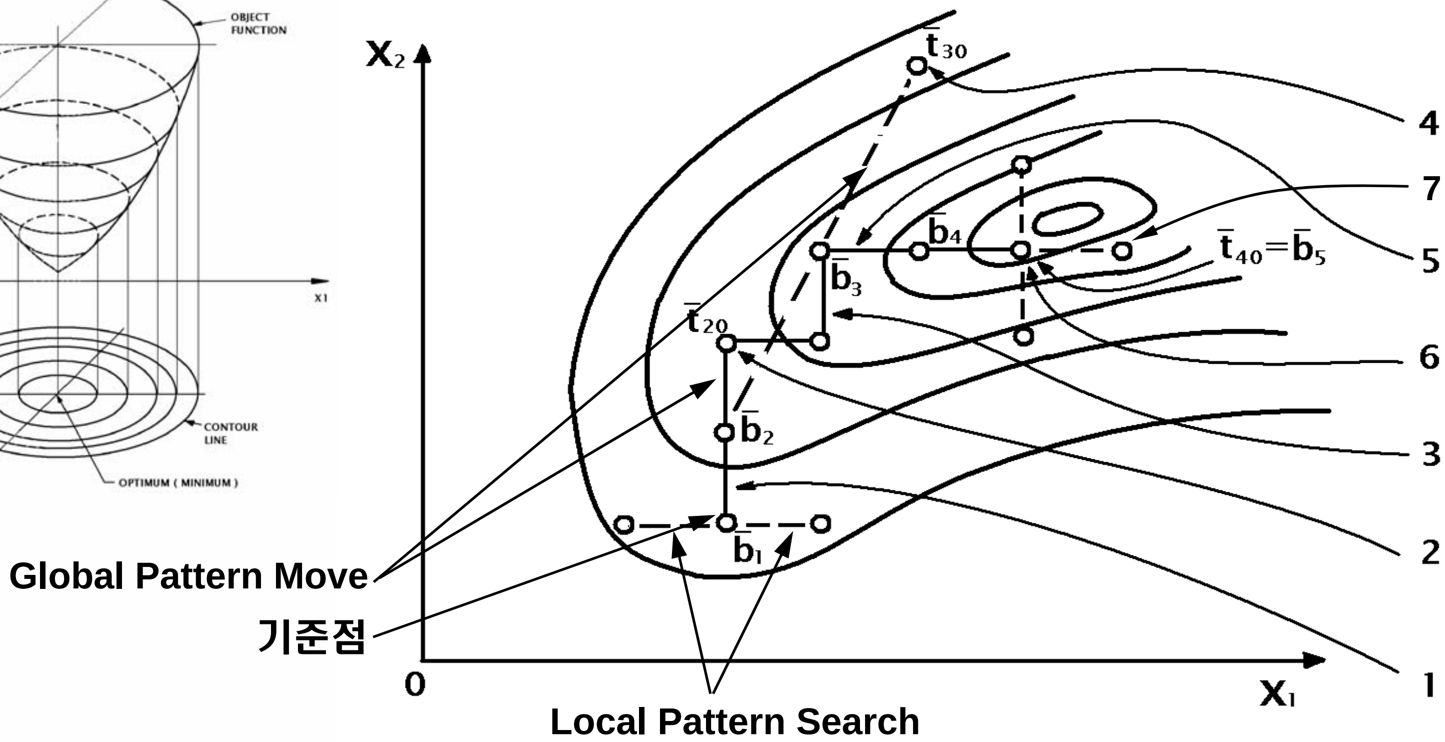
```
// [Input]            a : 최소값이 존재하는 영역의 하한, c : 최소값이 존재하는 영역의 상한
//                    b : a < b < c인 동시에 f(a) > f(b) and f(c) > f(b)인 점
// [Output]           f를 최소로 하는 점과 이 점에서의 목적 함수값
double GoldenSectionSearch(double a, double b, double c, double (*f)(double), double *xmin)
{
    double TOLERANCE = 1.0e-6;
    double f1, f2, x0, x1, x2, x3;
    x0 = a;           x3 = c;

    if (fabs(c - b) > fabs(b - a)) {
              x1 = b;      x2 = b + (1.0 - 0.618) * (c - b);     }
    else {    x2 = b;      x1 = b - (1.0 - 0.618) * (b - a);     }

    f1 = (*f)(x1);         f2 = (*f)(x2);

    while (fabs(x3 - x0) > TOLERANCE * (fabs(x1) + fabs(x2))) {
              if (f2 < f1) {          x0 = x1;     x1 = x2;     x2 = 0.618 * x1 + (1.0 - 0.618) * x3;
                                      f1 = f2;     f2 = (*f)(x2);                                     }
              else {       x3 = x2;   x2 = x1;     x1 = 0.618 * x2 + (1.0 - 0.618) * x0;
                           f2 = f1;   f1 = (*f)(x1);                                                  }
    }

    if (f1 < f2) {         *xmin = x1;             return f1;  }
    else {                 *xmin = x2;             return f2;  }
}
```

2.3 직접 탐사법 (Direct Search Method)

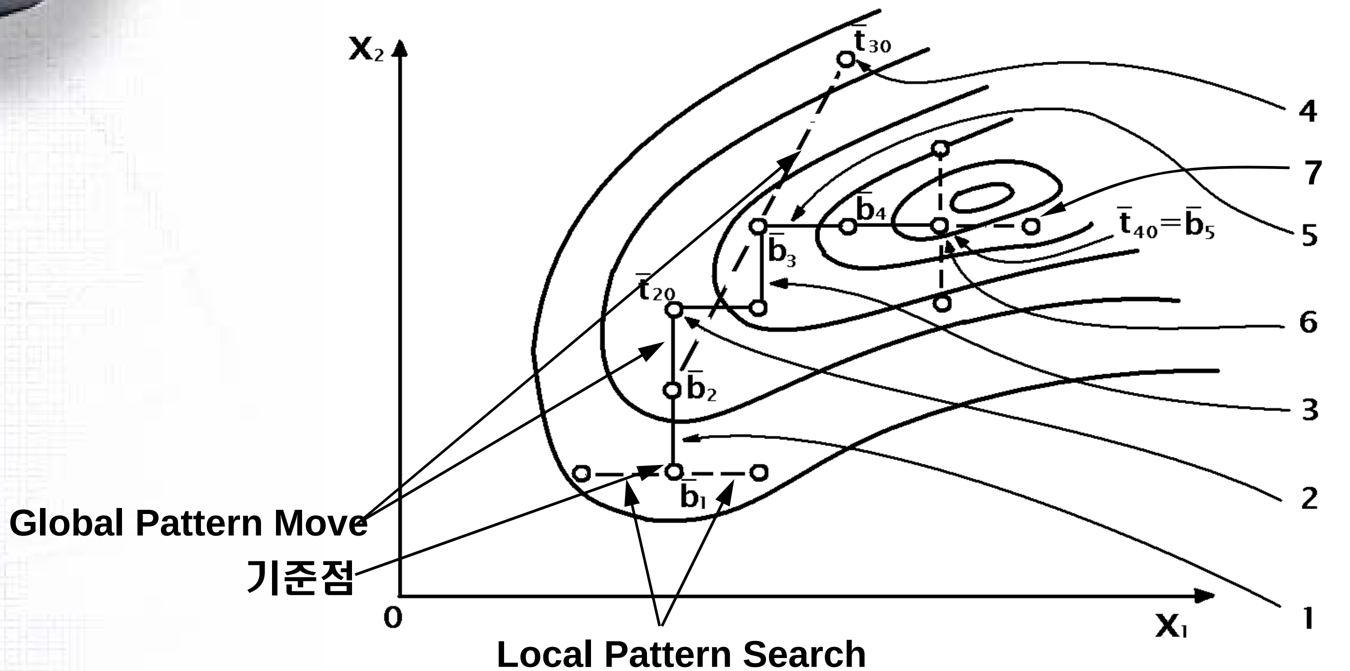
Advanced
Ship
Design
Automation
Laboratory

- **기준점에서의** Local Pattern Search**와 최적해 방향으로 가속화하는** Global Pattern Move**를 이용해 최적해를 찾는 방법**

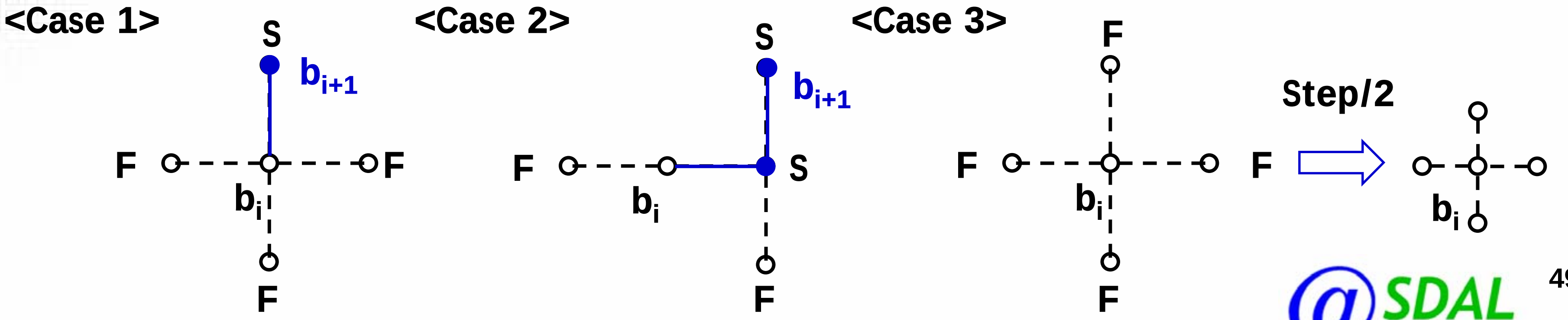




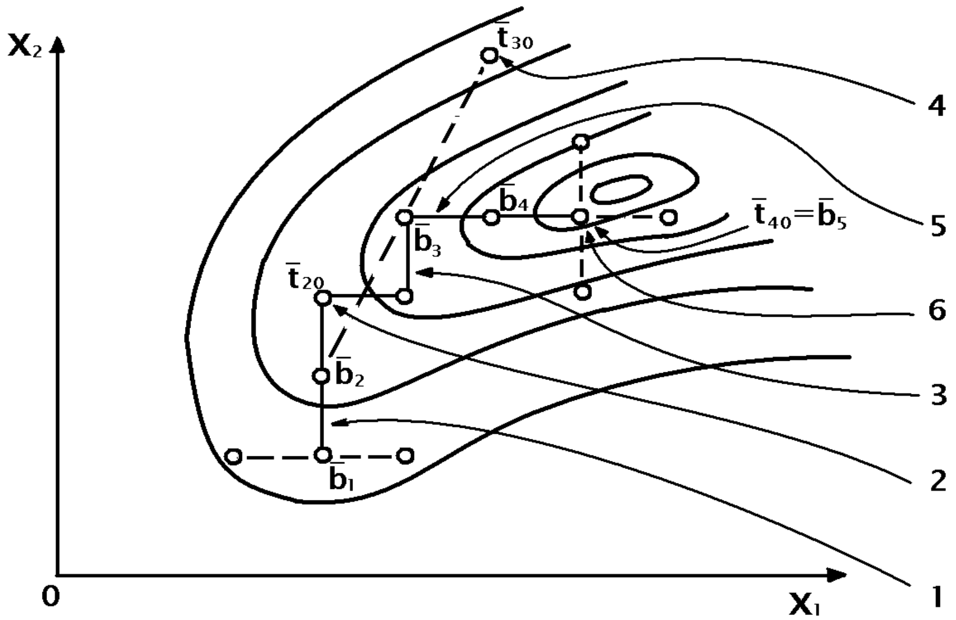
2.3 직접 탐사법
- Hooke & Jeeves의 직접 탐사법(2)



- **Local Pattern Search** (F: Fail, S: Success)



2.3 직접 탐사법
- Hooke & Jeeves의 직접 탐사법의 과정(1)



1. Initial Local Search : x_1 방향에 대해서는 목적 함수 값의 개선이 없음. 그런데 x_2 방향에 대해서는 개선이 있음. 따라서 $\bar{b}_2$가 새로운 기준점으로 선정됨

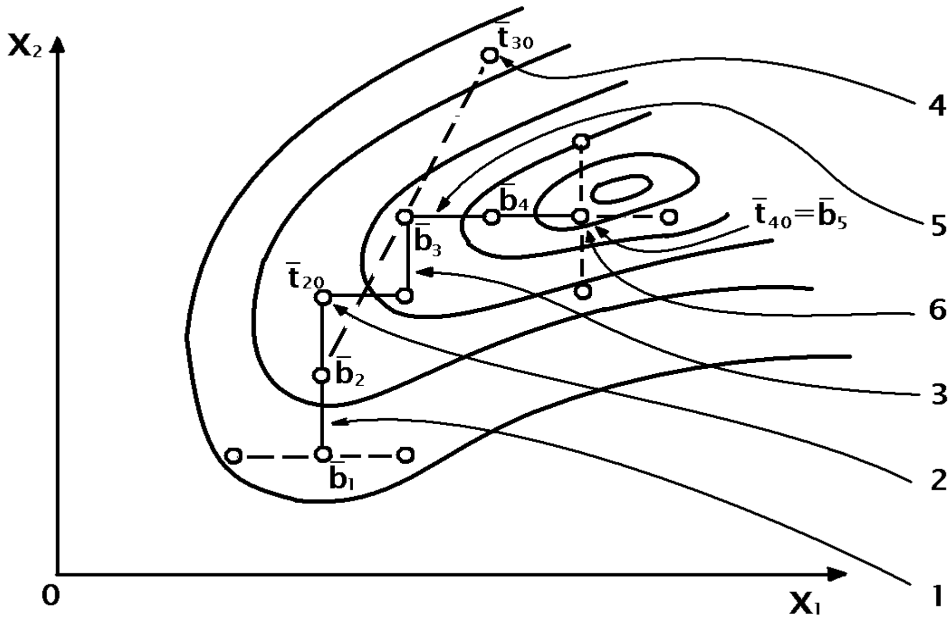
2. 직전의 2개의 기준점 $\bar{b}_1$, $\bar{b}_2$ 방향으로, $\bar{b}_2$에서 이 두 점과 같은 거리만큼 떨어져 있는 임시점 $\bar{t}_{20}$를 구함

3. $\bar{t}_{20}$점에서 함수값의 개선이 이루어졌으므로 이 점으로부터 Local Pattern Search를 수행하여 새로운 기준점 $\bar{b}_3$를 구함

4. 직전의 2개의 기준점 $\bar{b}_2$, $\bar{b}_3$ 방향으로, $\bar{b}_3$에서 이 두 점과 같은 거리만큼 떨어져 있는 임시점 $\bar{t}_{30}$를 구함

2.3 직접 탐사법

- Hooke & Jeeves의 직접 탐사법의 과정(2)



5. $\bar{t}_{30}$에서 함수 값의 개선이 없음. 따라서 기준점 $\bar{b}_3$ 로 돌아가서 Local Pattern Search를 수행함. 탐색 결과 $\bar{b}_4$를 찾음

6. 직전의 2개의 기준점 $\bar{b}_3, \bar{b}_4$ 방향으로, $\bar{b}_4$에서 이 두 점과 같은 거리만큼 떨어져 있는 임시점 $\bar{t}_{40}$을 구함. 이 점에서 함수 값이 개선되므로 Local Pattern Search를 수행하여 $\bar{b}_5(=\bar{t}_{40})$ 를 구함

7. 직전의 2개의 기준점 $\bar{b}_4, \bar{b}_5$ 방향으로, $\bar{b}_5$ 에서 이 두 점과 같은 거리만큼 떨어져 있는 임시점 $\bar{t}_{50}$을 구함. 이 점에서 함수 값의 개선이 없음. 따라서 기준점 $\bar{b}_5$로 돌아가서 Local Pattern Search를 수행함. 그런데 Local Pattern Search 결과 구해진 $\bar{b}_6$이 이전의 기준점 $\bar{b}_5$와 동일하므로 모든 Step Size 반으로 줄인 후 위의 과정을 반복함

2.3 직접 탐사법
- Hooke & Jeeves의 직접 탐사법의 알고리즘 요약(1)

1) Local Pattern Search

1. 기준점(Base Point) $\bar{\mathbf{b}}_1$ 에서의 함수 f 의 값을 계산한다.

2. 점 $\bar{b}_1 \pm \bar{\delta}_1$ 에서의 함수 f 의 값을 계산한다. 여기서 $\bar{\boldsymbol{\delta}}_1$ 은 Input Step Size이며 $\bar{\boldsymbol{\delta}}_1 = [\delta_1, 0, \ldots\ldots, 0]^T$ 이다. 함수 값의 개선이 있는 점을 $\bar{\mathbf{t}}_{11}$ 이라 놓고 이로부터 계속 탐사를 진행한다.

3. 점 $\bar{\mathbf{t}}_{11} \pm \bar{\boldsymbol{\delta}}_2$ 에서의 함수 f 의 값을 계산한다. 여기서 $\bar{\boldsymbol{\delta}}_2$ 는 역시 Input Step Size이며 $\bar{\boldsymbol{\delta}}_2 = [0, \delta_2, \ldots\ldots, 0]^T$ 이다. 그리고 함수 값의 개선이 있는 점을 $\bar{\mathbf{t}}_{12}$ 이라 놓는다.

4. 나머지 좌표 축에 대해 위와 같은 Local Pattern Search 과정을 수행하고 새로운 기준점(New Base Point)을 설정한다. 즉, Local Pattern Search 과정이 끝나면 새로운 기준점이 선정된다. 선정된 새로운 기준점 $\bar{\mathbf{b}}_2$ 는 $\bar{\mathbf{b}}_2 = \bar{\mathbf{t}}_{1n}$ 이다.

5. 새로운 기준점과 그 직전 기준점의 방향으로 Global Pattern Move를 수행한다.

2.3 직접 탐사법
- Hooke & Jeeves의 직접 탐사법의 알고리즘 요약(2)

2) Global Pattern Move

1. Local Pattern Search에 의해 얻어진 2개의 기준점을 연결하는 방향을 따라 제일 마지막 기준점에서 이 두 기준점 사이의 거리만큼 떨어져 있는 점을 임시 기준점(Temporary Point)로 설정하고("Global Pattern Move"), 임시점에서의 함수 f의 값을 계산한다. Global Pattern Move에 의해 선정된 임시 기준점은 다음과 같다.

$$\bar{\mathbf{t}}_{n+1,0} = \bar{\mathbf{b}}_n + 2(\bar{\mathbf{b}}_{n+1} - \bar{\mathbf{b}}_n) = 2\bar{\mathbf{b}}_{n+1} - \bar{\mathbf{b}}_n$$

2. 만일 이 임시점에서 함수 f의 값이 개선된다면 이 점에서부터 다시 Local Pattern Search를 수행한다. 만일 함수 f의 값이 개선되지 않는다면 그 직전의 기준점으로 돌아가 Local Pattern Search를 수행한다.

3) Closing Conditions

1. 만일 Local Pattern Search를 수행한 후에도 $\bar{\mathbf{b}}_{n+1} = \bar{\mathbf{b}}_n$이 되어 어떠한 개선이 이루어지지 않는다면 모든 δ_i를 $\delta_i / 2$로 바꾸어 위 과정을 반복한다.

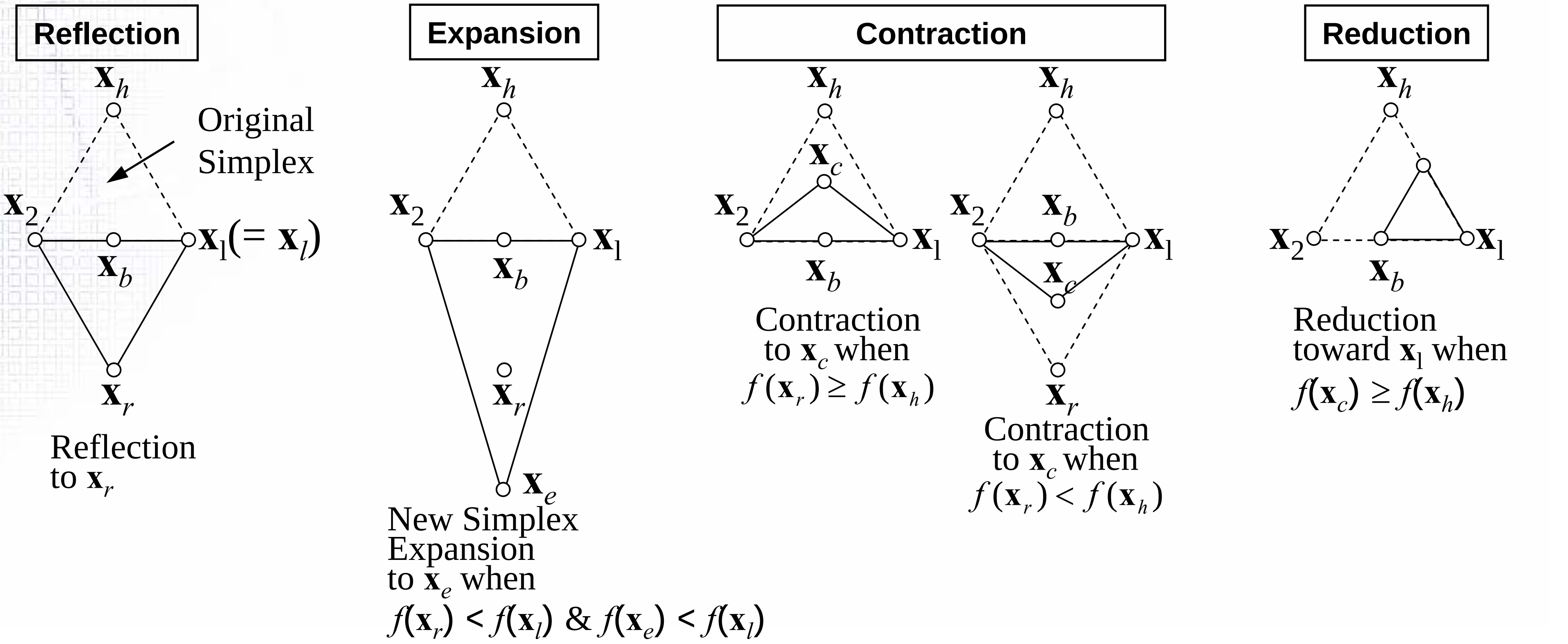
2. 만일 모든 δ_i가 ε_i보다 작으면 탐사 과정을 마친다.

Hooke & Jeeves의 직접 탐사법(Local Pattern Search)의 구현 예

```
// [Input]     del : 설계점에 대한 이동폭, x : 현재의 설계점, fmin_pre : 이전 설계점에서의 함수값, variables_no : 설계 변수의 수
// [Output]    개선된 설계점과 이 점에서의 목적 함수값
double LocalPatternSearch(double* del, double* x, double fmin_pre, int variables_no)
{
	double* temp_x = new double[MAX_DV_NUM];
	double fmin, temp_function_value;
	int i;
	fmin = fmin_pre;
	for (i = 0; i < variables_no; i++) temp_x[i] = x[i];
	for (i = 0; i < variables_no; I++) {
		temp_x[i] = x[i] + del[i];
		temp_function_value = f(temp_x, variables_no);

		if (temp_function_value < fmin) {
				fmin = temp_function_value;
		} else {
				del[i] = 0.0 - del[i];
				temp_x[i] = x[i] + del[i];
				temp_function_value = f(temp_x, variables_no);
				if (temp_function_value < fmin) {
						fmin = temp_function_value;
				}
				else		temp_x[i] = x[i];
		}
	}

	for (i = 0; i < variables_no; i++) {
		x[i] = temp_x[i];
	}

	delete[] temp_x;
	return fmin;
}
```

2.3 직접 탐사법
- Nelder & Mead's Simplex 방법

- n개의 설계 변수를 가진 n차원 문제에서 (n+1)개의 모서리를 가진 기하학적 형상(Simplex)의 모양과 위치를 반사(Reflection), 확장(Expansion), 수축(Contraction) 및 감소(Reduction)의 4가지 형태로 계속 변화시키면서 최적점을 찾는 방법



$\mathbf{x}_h$: Simplex point having the largest objective function value

$\mathbf{x}_b$: Center point between $\mathbf{x}_1$ and $\mathbf{x}_2$

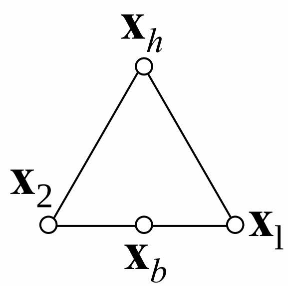
2.3 직접 탐사법
- Nelder & Mead의 Simplex 방법의 알고리즘

- **단계 1 : Simplex 생성 및 목적 함수 값 계산**
  - Simplex의 (n+1)개의 점에서의 목적 함수 f 의 값을 계산한다.

- **단계 2 : 최대 및 최소값 설정**
  - 현재의 Simplex에서 $f(\mathbf{x})$의 값을 최대 및 최소로 하는 점을 각각 $\mathbf{x}_h$, $\mathbf{x}_l$이라 둔다.

- **단계 3 : 중심 계산**
  - $\mathbf{x}_h$를 제외한 모든 $\mathbf{x}_i$에 대한 중심($\mathbf{x}_b$)을 다음과 같이 구하고 이 점에서의 함수 값을 계산한다.

$$\mathbf{x}_b = \frac{1}{n}\sum_{i=1}^{n+1}\mathbf{x}_i \text{(단, } \mathbf{x}_h \text{는 제외)}$$



$$\mathbf{x}_b = \frac{\mathbf{x}_1 + \mathbf{x}_2}{2}$$

2.3 직접 탐사법(Direct Search Method)
- Nelder & Mead의 Simplex 방법의 알고리즘의 요약(1)

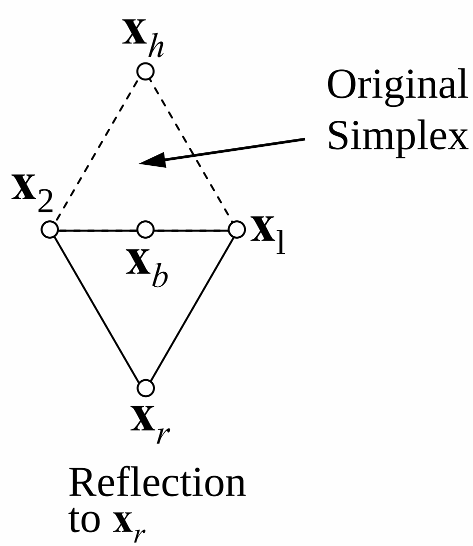
- 단계 4 : 중지 조건

$$\left\{\frac{1}{n+1}\sum_{i=1}^{n+1}[f(\mathbf{x}_i)-f(\mathbf{x}_b)]^2\right\}^{1/2} \le \varepsilon$$

  - 만일 중지 조건을 만족하면 탐사 과정을 중지하고 $\mathbf{x}_l$을 최적 값으로 선택한다. 그렇지 않으면 다음 과정으로 간다.

- 단계 5 : 반사(Reflection)
  - $\mathbf{x}_h$를 $\mathbf{x}_b$에 대해 반사한 점 $\mathbf{x}_r$을 다음과 같이 구한다.

$$\mathbf{x}_r = 2\mathbf{x}_b - \mathbf{x}_h$$



이 점에서의 함수 값 $f(\mathbf{x}_r)$을 계산하고,
다음 조건에 따라 Simplex를 변경시킨다.

2.3 직접 탐사법
- Nelder & Mead의 Simplex 방법의 알고리즘의 요약(2)

- 단계 6 : 확장(Expansion)
  - 단계 6-1 : $f(\mathbf{x}_r) < f(\mathbf{x}_l)$일 때

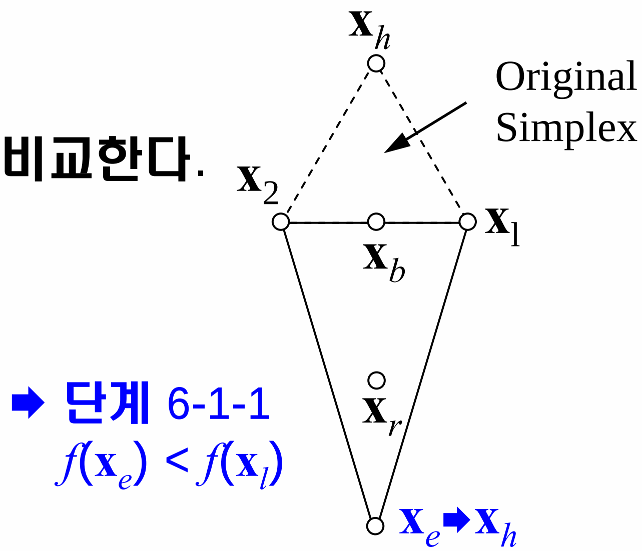
    $\mathbf{x}_b$를 $\mathbf{x}_r$에 대해 반사시켜 확장한 점 $\mathbf{x}_e$를 구한다.

    $$\mathbf{x}_e = 2\mathbf{x}_r - \mathbf{x}_b$$

    이 점에서 함수 값 $f(\mathbf{x}_e)$를 계산하고 이를 $f(\mathbf{x}_l)$과 비교한다.

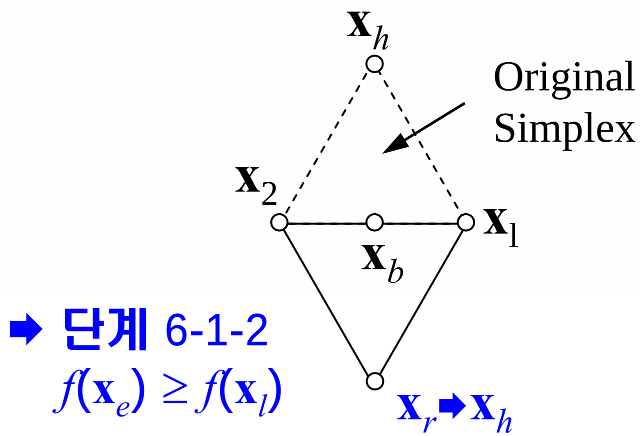
    - 단계 6-1-1 : $f(\mathbf{x}_e) < f(\mathbf{x}_l)$일 때

      $\mathbf{x}_h$를 $\mathbf{x}_e$(확장점)로 대체하고 단계 2로 간다.



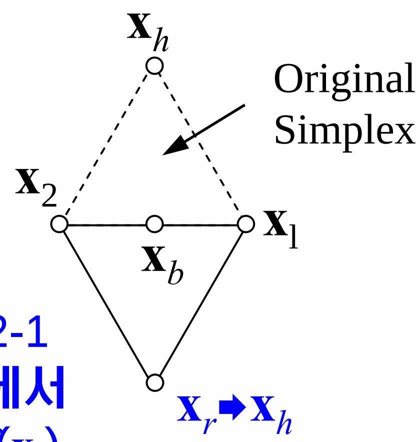
    - 단계 6-1-2 : $f(\mathbf{x}_e) \ge f(\mathbf{x}_l)$일 때

      $\mathbf{x}_h$를 $\mathbf{x}_r$(반사점)로 대체하고 단계 2로 간다.



2.3 직접 탐사법
- Nelder & Mead의 Simplex 방법의 알고리즘의 요약(3)

- 단계 6 : 확장(Expansion)
  - 단계 6-2 : $f(\mathbf{x}_r) \geq f(\mathbf{x}_l)$일 때
    - 단계 6-2-1 : $\mathbf{x}_h$를 제외한 어떤 $\mathbf{x}_i$에서 $f(\mathbf{x}_r) < f(\mathbf{x}_i)$일 때 $\mathbf{x}_h$를 $\mathbf{x}_r$(반사점)로 대체하고 단계 2로 간다.



➡ 단계 6-2-1
어떤 $\mathbf{x}_i$에서
$f(\mathbf{x}_r) < f(\mathbf{x}_i)$

    - 단계 6-2-2 : $\mathbf{x}_h$를 제외한 어떤 $\mathbf{x}_i$에서도 $f(\mathbf{x}_r) \geq f(\mathbf{x}_i)$일 때 다음 단계로 간다.

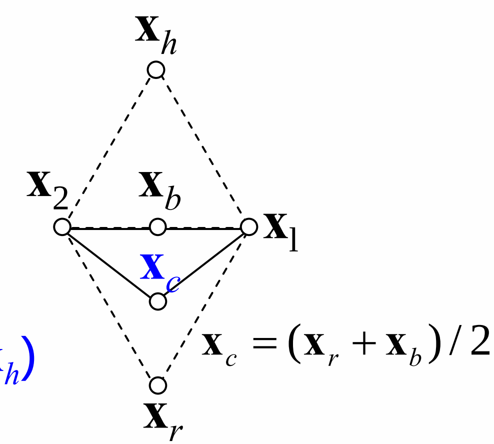
2.3 직접 탐사법
- Nelder & Mead의 Simplex 방법의 알고리즘의 요약(4)

- 단계 7 : 수축(Contraction)
  - 단계 7-1 : $f(\mathbf{x}_r) < f(\mathbf{x}_h)$일 때

    수축점을 다음과 같이 구하고 이 점에서의 함수 값을 계산한다.

    $$\mathbf{x}_c = (\mathbf{x}_r + \mathbf{x}_b)/2$$

➡ 단계 7-1
$f(\mathbf{x}_r) < f(\mathbf{x}_h)$

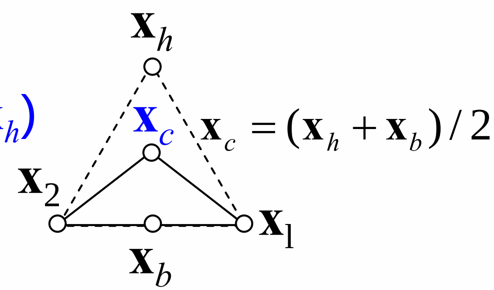


  - 단계 7-2 : $f(\mathbf{x}_r) \geq f(\mathbf{x}_h)$일 때

    수축점을 다음과 같이 구하고 이 점에서의 함수 값을 계산한다.

    $$\mathbf{x}_c = (\mathbf{x}_h + \mathbf{x}_b)/2$$

➡ 단계 7-2
$f(\mathbf{x}_r) \geq f(\mathbf{x}_h)$



  - 그리고 나서 $f(\mathbf{x}_c) < f(\mathbf{x}_h)$일 때

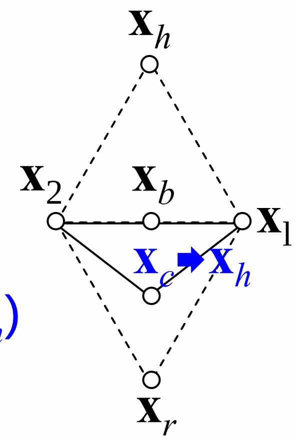
    $\mathbf{x}_h$를 $\mathbf{x}_c$(수축점)로 대체하고 단계 2로 간다.

2.3 직접 탐사법
- Nelder & Mead의 Simplex 방법의 알고리즘의 요약(5)

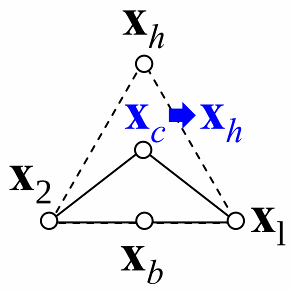
- **단계 8 : 감소(Reduction)**
  - **단계 8-1 : $f(\mathbf{x}_c) < f(\mathbf{x}_h)$일 때**

    $\mathbf{x}_h$를 $\mathbf{x}_c$(수축점)로 대체하고 단계 2로 간다.

➡ 단계 8-1
$f(\mathbf{x}_c) < f(\mathbf{x}_h)$



또는

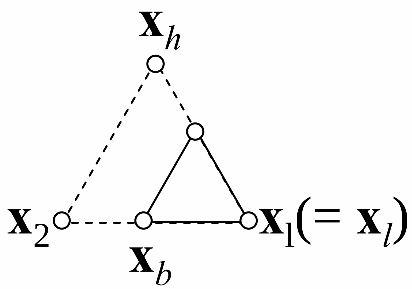


  - **단계 8-2 : $f(\mathbf{x}_c) \geq f(\mathbf{x}_h)$일 때**

    Simplex의 점들을 $\mathbf{x}_l$로 향하여 감소시킨 후 단계 2로 간다. 이 때 Simplex의 감소된 점들은 다음과 같이 구한다.

$$\mathbf{x}_i = (\mathbf{x}_i + \mathbf{x}_l)/2$$

➡ 단계 8-2
$f(\mathbf{x}_c) \geq f(\mathbf{x}_h)$



Reduction toward $\mathbf{x}_1$

Nelder & Mead의 Simplex 방법의 구현 예

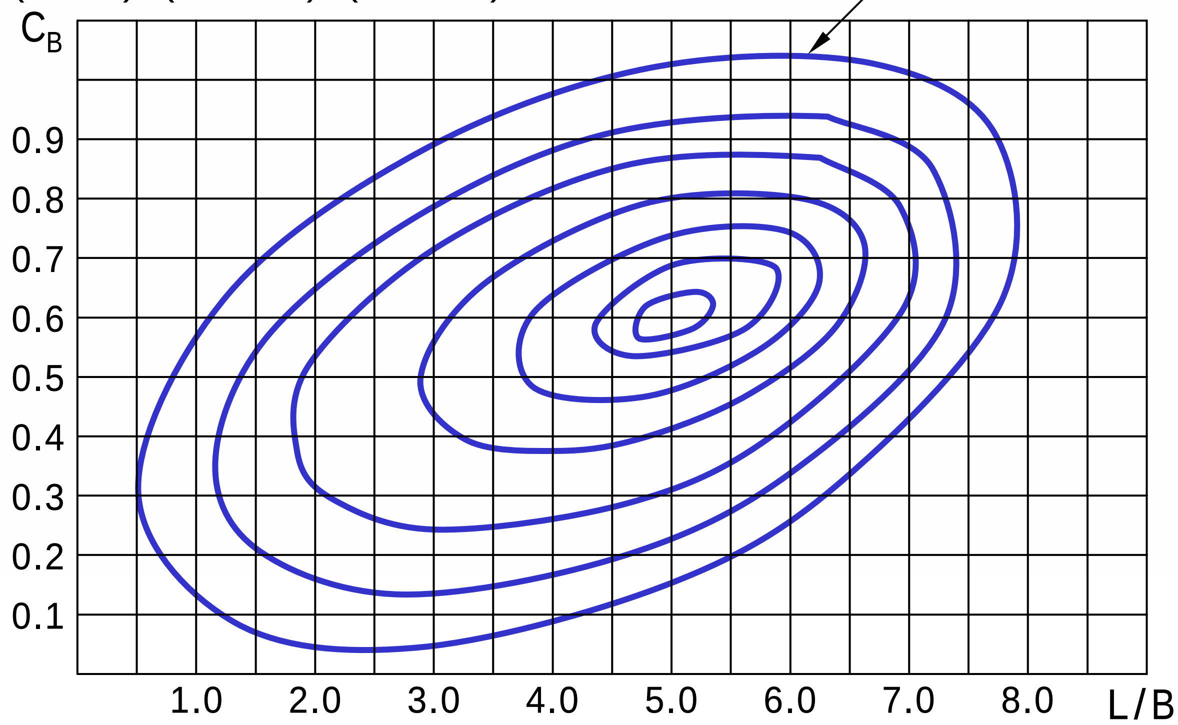
```
void MoveSimplex(double **x, double* y, int variables_dimension, double (*f)(double *xx))
{
	double TOLERANCE = 1.0e-6;
	int i, i_high, i_low, inhi, j;
	int Simplex_Dimension = variables_dimension + 1;
	double Simplex_Tolerance, temp_y1, temp_y2;			double* xsum = new double[variables_dimension];
	while (1) {
		i_low=1;		i_high = y[1] > y[2] ? (inhi = 2, 1) : (inhi = 1, 2);
		for (i = 1; i <= Simplex_Dimension; i++) {
			if (y[i] <= y[i_low]) i_low = i;
			if (y[i] > y[i_high]) {				inhi = i_high;	i_high = i;
			} else if (y[i] > y[inhi] && i != i_high)		inhi = i;	}
		Simplex_Tolerance = 2.0 * fabs(y[i_high] - y[i_low]) / (fabs(y[i_high]) + fabs(y[i_low]));
		if (Simplex_Tolerance < TOLERANCE) {			double temp;	temp = y[1];
			y[1] = y[i_low];	y[i_low] = temp;
			for (i = 1; i <= variables_dimension; i++) {		temp = x[1][i];
				x[1][i] = x[i_low][i];		x[i_low][i] = temp;		}
			break;
		}
		// Reflection
		temp_y1 = Simplex(x, y, xsum, variables_dimension, f, i_high, -alpha);
		if (temp_y1 <= y[i_low])
			// Expansion
			temp_y1 = Simplex(x, y, xsum, variables_dimension, f, i_high, gamma);
		else if (temp_y1 >= y[inhi]) {
			temp_y2 = y[i_high];		temp_y1 = Simplex(x, y, xsum, variables_dimension, f, i_high, beta);
			if (temp_y1 >= temp_y2) {
				for (i = 1; i <= Simplex_Dimension; i++) {
					if (i != i_low) {
						for (j = 1; j <= variables_dimension; j++)
							x[i][j] = xsum[j] = 0.5 * (x[i][j] + x[i_low][j]);
						y[i] = (*f)(xsum);
					}
				}
			}
		}
	}
	delete[] xsum;
}
```

비제약 최적화 문제

- 어느 선박의 상대적 건조비를 L/B와 C_B의 함수로 다음 그림과 같이 표시할 수 있다고 하면 건조비가 최소가 되는 L/B와 C_B의 값을 Hooke & Jeeves의 탐사법과 Nelder & Mead의 Simplex 방법을 이용하여 최적점을 구하고 그 과정을 그림에 각각 표시하시오.
  - Hooke & Jeeves의 탐사법
    - 출발점: L/B = 1, C_B = 0.1
    - 출발점의 step size: Δ(L/B) = 0.5, $\Delta(C_B)$ = 0.1
  - Nelder & Mead의 Simplex 방법
    - 출발 모서리점: (L/B, CB) = (1, 0.1), (1.5, 0.1), (1.5, 0.2)
    - 중지 기준: 0.01

목적 함수의 contour line(f = const.)

미지수 2개인 최적화 문제



비제약 최적화 문제
- Hooke & Jeeves의 직접 탐사법을 이용한 해법(1)

$x_1 = L/B,\ x_2 = C_B$ **라고 놓고 각 탐사 단계를 설명하면 다음과 같다.**

- 단계 1 : 국부 탐사 1

$b_0 = (7,\ 0.2),\ \Delta x_1 = 0.5,\ \Delta x_2 = 0.1,$

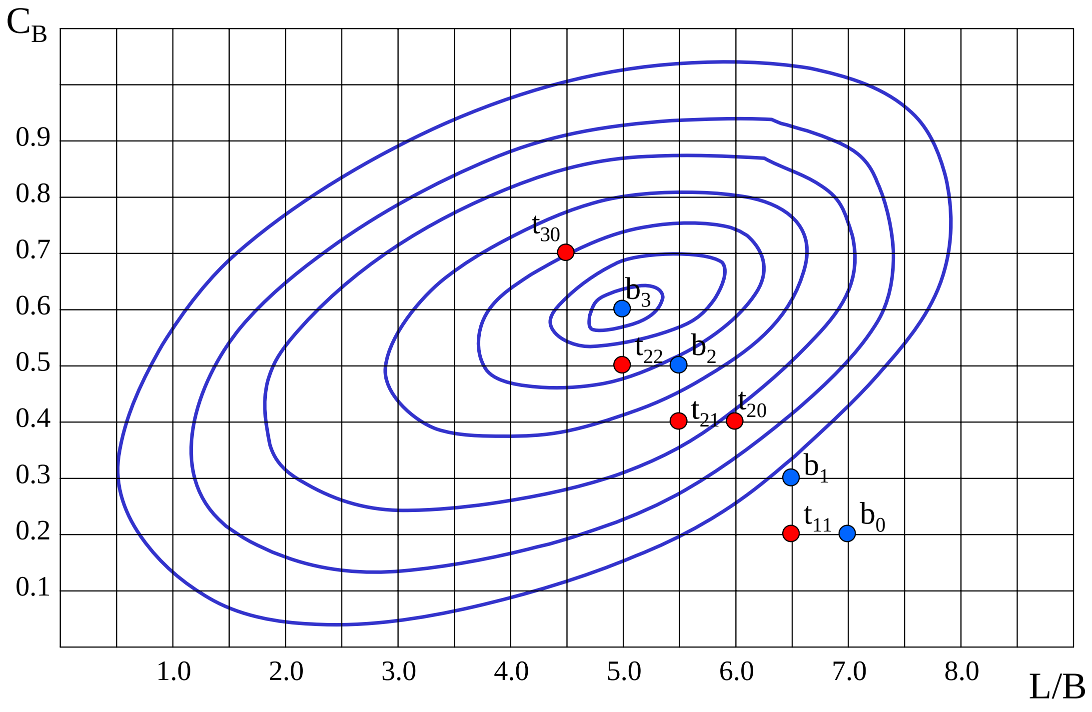
$t_{10} = b_0$

t_{10}에서 $-x_1$방향 $\rightarrow t_{11} = (6.5,\ 0.2)$

t_{11}에서 $+x_2$방향 $\rightarrow t_{12} = (6.5,\ 0.3)$

국부 탐사가 성공(함수값이 감소)하였으므로 t_{12}를 새로운 기준점(Base Point)으로 선정한다.

$$b_1 = t_{12}$$



- 단계 2 : 전체 탐사 1

b_0와 b_1에서 $t_{20} = (6,\ 0.4)$

임시점 t_{20}에서의 함수값이 개선되었으므로 이 점에서부터 다시 국부탐사를 한다.

비제약 최적화 문제
- Hooke & Jeeves의 직접 탐사법을 이용한 해법(2)

- 단계 3 : 국부 탐사 2

t_{20}에서 $-x_1$방향 $\rightarrow t_{21} = (5.5,\ 0.4)$

t_{21}에서 $+x_2$방향 $\rightarrow t_{22} = (5.5,\ 0.5)$

$$b_2 = t_{22}$$

- 단계 4 : 전체 탐사 2

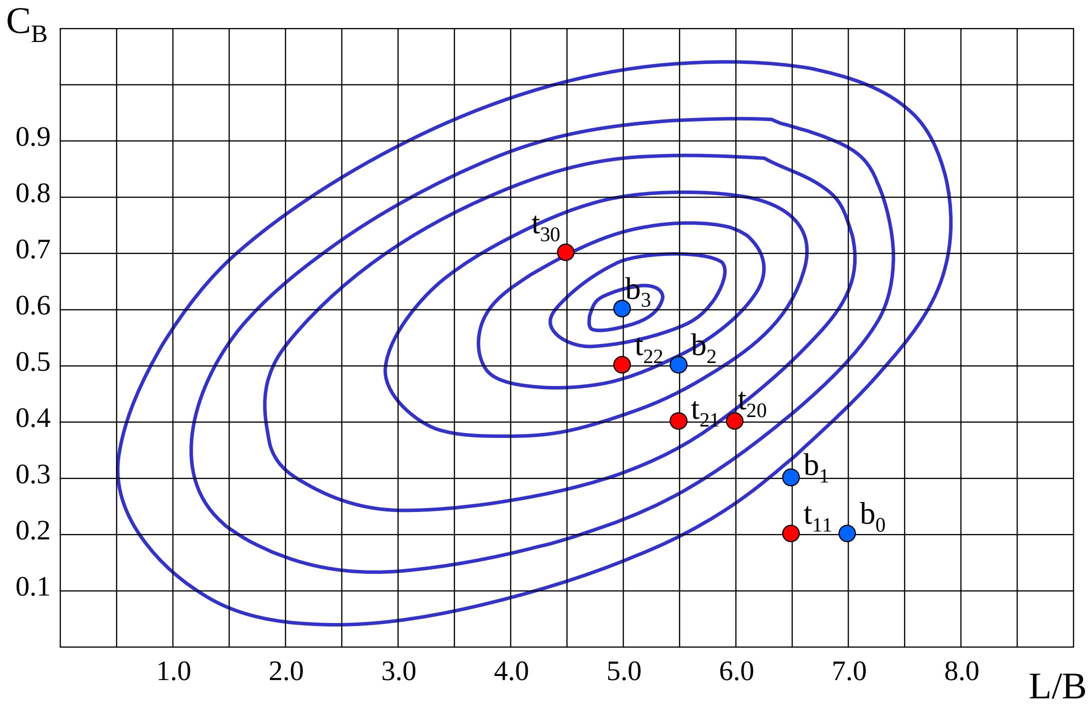
b_1와 b_2에서 $t_{30} = (4.5,\ 0.7)$

- 단계 5 : 국부 탐사 3

t_{30}에서 x_1방향 $\rightarrow t_{31} = (5,\ 0.7)$

t_{31}에서 $-x_2$방향 $\rightarrow t_{32} = (5,\ 0.6)$

$$b_3 = t_{32}$$



비제약 최적화 문제
- Hooke & Jeeves의 직접 탐사법을 이용한 해법(3)

• 단계 6 : 전체 탐사 3

b_2와 b_3에서 $t_{40} = (4.5,\ 0.7)$에서는 목적 함수값의 감소가 없으므로 $t_{40} = b_3$

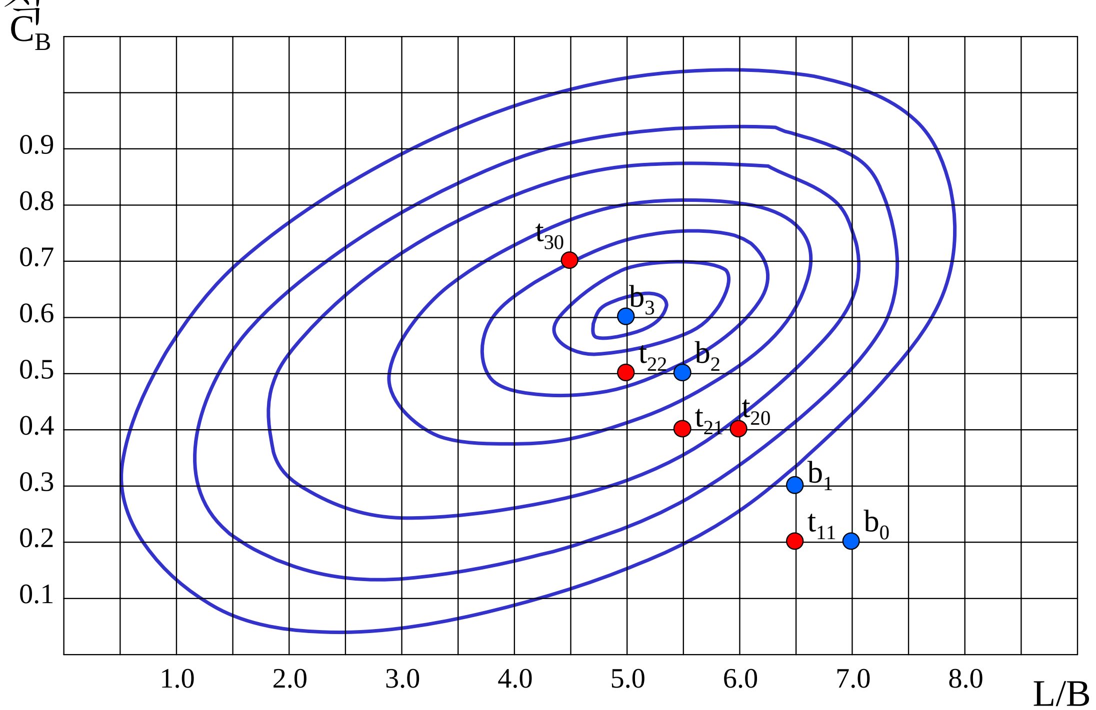
• 단계 7 : 국부 탐사 4

t_{40}에서는 x_1방향으로도, x_2방향으로도 목적함수의 감소가 없으므로 $t_{42} = t_{41} = t_{40}$

• 단계 8 : 전체 탐사 4

$b_4 = b_3 \rightarrow \Delta x_1 = 0.25,\ \Delta x_2 = 0.05,$

$t_{50} = b_4$



• 단계 9 : 탐사종료

$(x_1, x_2) = (L/B, C_B) = (5.0, 0.6)$까지의 탐사 이후로는 전체탐사와 국부탐사 모두 목적 함수의 감소가 없으므로 탐사가 종료

상대적 건조비가 최소가 되는 최적점은 $L/B = 5.0,\ C_B = 0.6$

비제약 최적화 문제
- Nelder & Mead의 Simplex 방법을 이용한 해법(1)

$x_1 = L/B,\ x_2 = C_B$ **라고 놓고 각 탐사 단계를 설명하면 다음과 같다.**

- 단계 1

삼각형 1 : $x_1,\ x_2,\ x_3$

- 단계 2

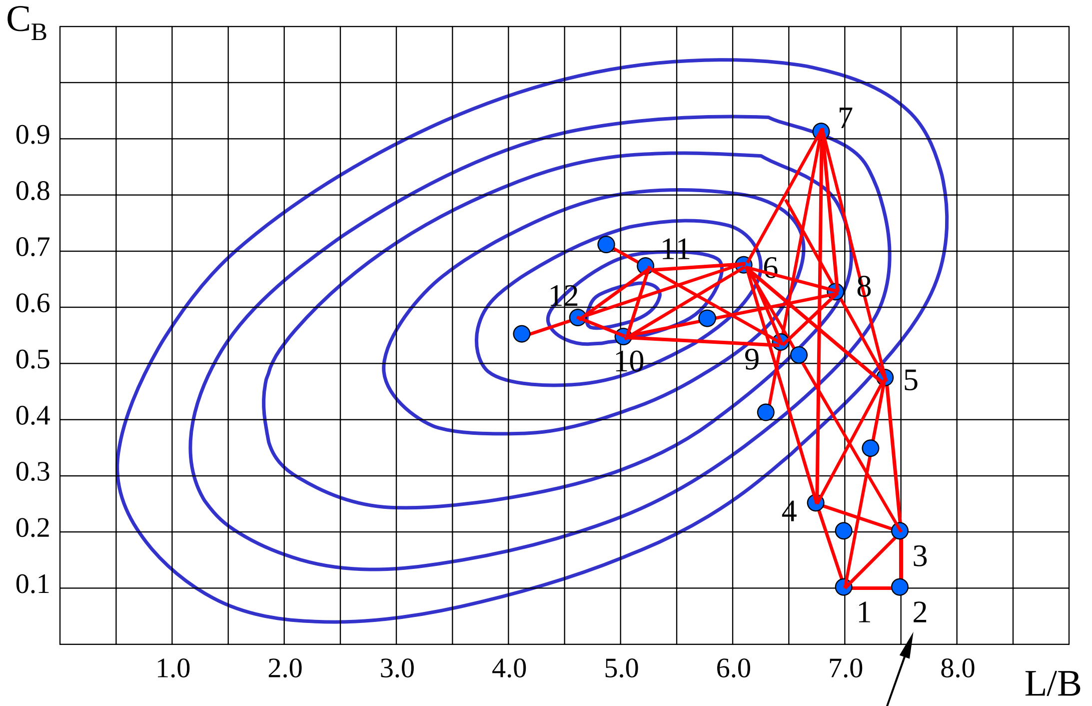
Expansion Point x_4가 $x_h = x_2$를 대체

$\rightarrow$ 삼각형 2 : $x_1,\ x_3,\ x_4$

- 단계 3

Expansion Point x_5가 $x_h = x_1$를 대체

$\rightarrow$ 삼각형 3 : $x_3,\ x_4,\ x_5$



숫자는 x_i의 인덱스 i를 나타냄

- 단계 4

Expansion Point x_6가 $x_h = x_3$를 대체 $\rightarrow$ 삼각형 4 : $x_4,\ x_5,\ x_6$

- 단계 5

Reflection Point x_7가 $x_h = x_4$를 대체 $\rightarrow$ 삼각형 5 : $x_5,\ x_6,\ x_7$

비제약 최적화 문제
- Nelder & Mead의 Simplex 방법을 이용한 해법(2)

● 단계 6

삼각형 5의 내부로 Contraction된

Point x_8이 $x_h = x_5$를 대체

$\rightarrow$ 삼각형 6 : x_6, x_7, x_8

● 단계 7

삼각형 6의 내부로 Contraction된

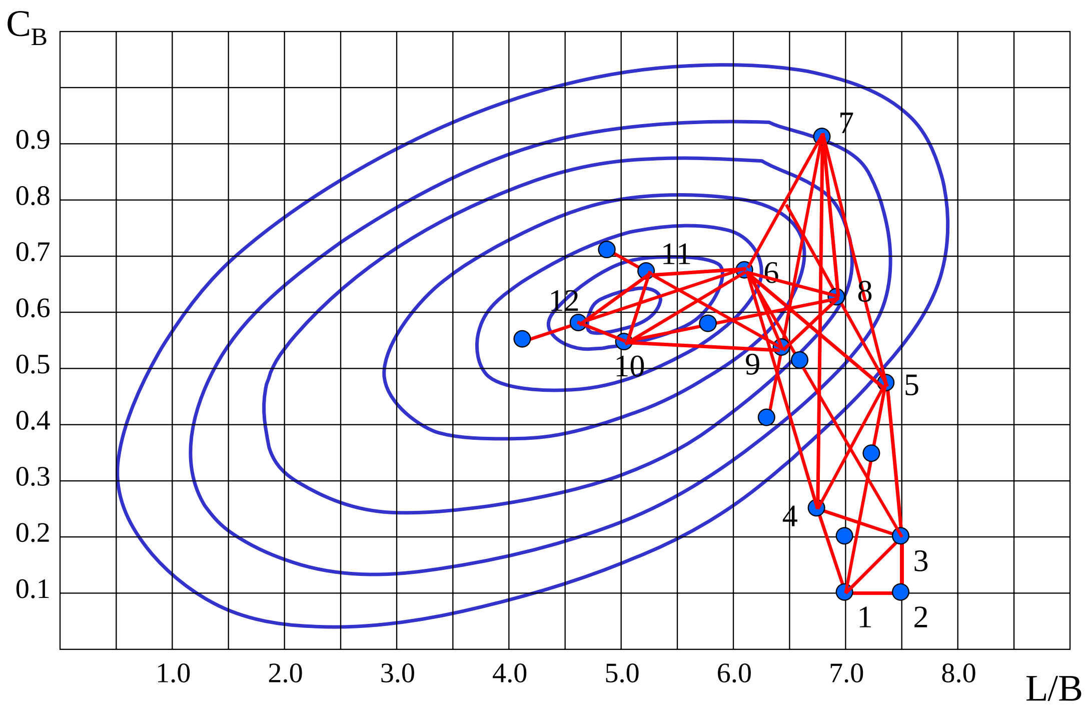
Point x_9이 $x_h = x_7$를 대체

$\rightarrow$ 삼각형 7 : x_6, x_8, x_9

● 단계 8

Expansion Point x_{10}이 $x_h = x_8$를 대체

$\rightarrow$ 삼각형 8 : x_6, x_9, x_{10}



● 단계 9

삼각형 8의 외부로 Contraction된 Point x_{11}이 $x_h = x_9$를 대체

$\rightarrow$ 삼각형 9 : x_6, x_{10}, x_{11}

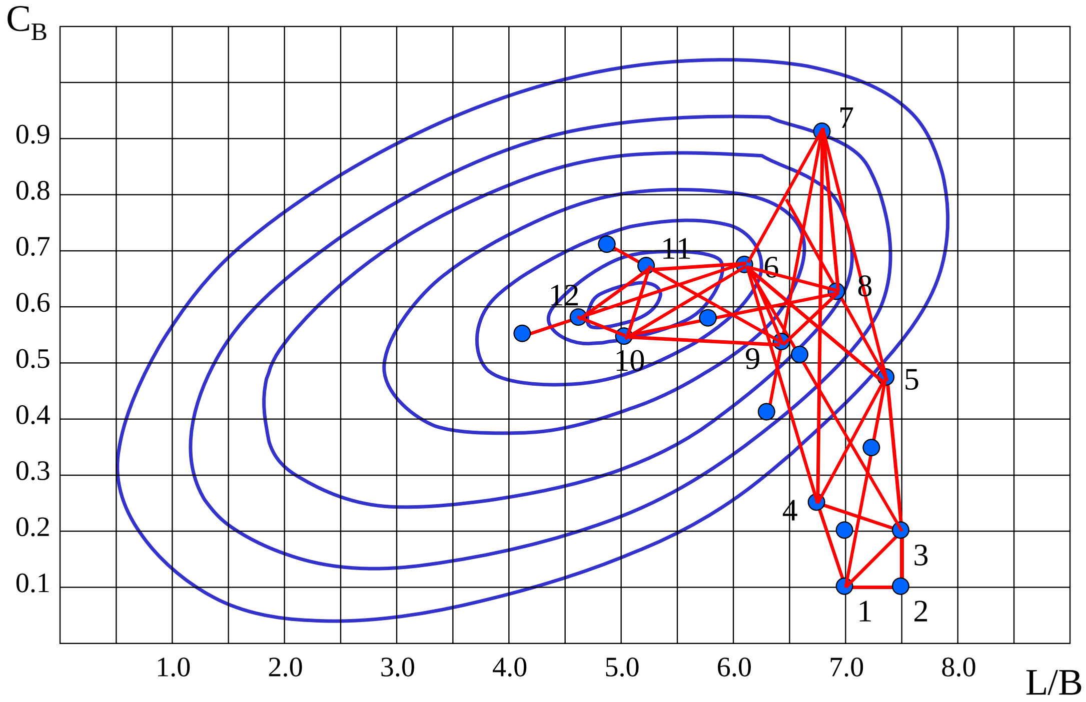
비제약 최적화 문제
- Nelder & Mead의 Simplex 방법을 이용한 해법(3)

- 단계 10

삼각형 9의 외부로 Contraction된

Point x_{12}이 $x_h = x_6$를 대체

$\rightarrow$ 삼각형 10 : x_{10}, x_{11}, x_{12}



$x_1(7,\ 0.1)$ $x_2(7.5,\ 0.1)$

$x_3(7.5,\ 0.2)$ $x_4(6.75,\ 0.25)$

$x_5(7.375,\ 0.475)$ $x_6(6.1875,\ 0.6875)$

$x_7(6.8125,\ 0.9125)$ $x_8(6.9375,\ 0.6375)$

$x_9(6.4375,\ 0.5375)$ $x_{10}(5.0625,\ 0.5625)$

$x_{11}(5.21875,\ 0.66875)$ $x_{12}(4.6171875,\ 0.5796875)$

삼각형 10의 완성으로서 탐사를 종료하였으나, 앞에서의 HOOKE & Jeeves의 탐사법을 이용한 결과로 얻은 최적 점으로 삼각형이 점차로 접근하게 됨을 확인할 수 있다.

Ch3. 선형 계획법 (Linear Programming)

Advanced
Ship
Design
Automation
Laboratory

3.1 선형 계획 문제(Linear Programming Problem)

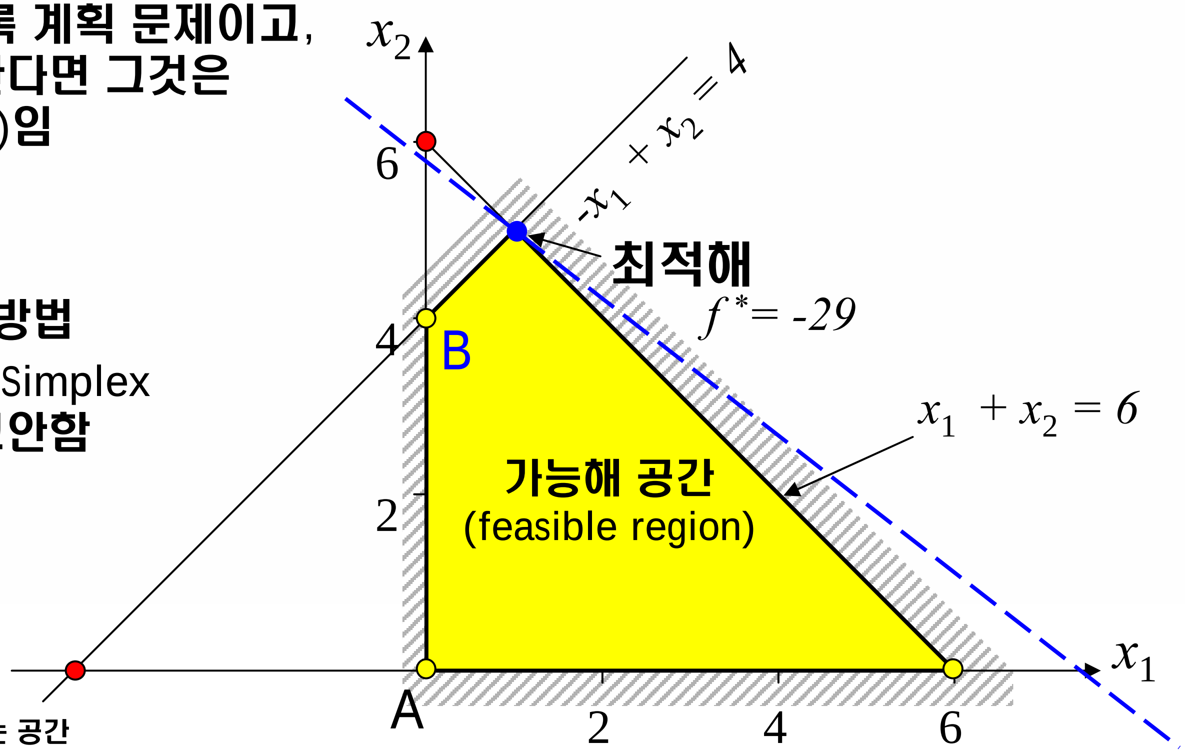
- **선형 계획 문제**
  - 목적 함수와 제약 조건이 설계 변수에 대하여 모두 선형임
  - 모든 함수들이 선형이기 때문에 등호 제약 조건이나 부등호 제약 조건에 의해 정의된 가능해 공간(feasible region)은 볼록(convex) 집합임
  - 따라서 선형 계획 문제는 볼록 계획 문제이고, 만일 하나의 최적해가 존재한다면 그것은 전역 최적해(global optimum)임

목적 함수

$$\textit{Minimize} \quad f = -4x_1 - 5x_2$$

제약 조건

$$\begin{aligned} \textit{Subject to} \quad & -x_1 + x_2 \le 4 \\ & x_1 + x_2 \le 6 \\ & x_1, x_2 \ge 0 \end{aligned}$$



- **선형 계획법**
  - 선형 계획 문제를 풀기 위한 방법
  - 1947년 George B. Dantzig가 Simplex 방법이라는 선형 계획법을 고안함

* 가능해 공간: 모든 제약 조건을 만족하며 최적해가 존재하는 공간

선형 계획 문제의 특징

- 목적 함수와 제약 조건들이 변수의 선형 관계를 표현함
  - 1개의 목적 함수와 1개 이상의 제약 조건으로 구성
  - 목적 함수는 최대화 혹은 최소화의 형태임

- 각 제약 조건들은 등식(=; equality constraint) 혹은 부등식(≥, ≤; inequality constraint)으로 표현됨

- 모든 선형 계획 문제의 변수들은 음수가 아니라고 가정함
  - 음수인 경우는 적절한 변형을 통해 양수화 시킴
    - 예, x = -y (x는 음수, y는 양수)
  - 부호에 제약이 없는 경우(양수 또는 음수를 가질 경우)는 적절한 변형을 통해 양수화 시킴
    - 예, x = y – z(x는 양수 또는 음수의 값을 가짐, y와 z는 양수)

목적 함수

$$\textit{Minimize} \quad f = -4x_1 - 5x_2$$

제약 조건

$$\textit{Subject to} \quad -x_1 + x_2 \le 4$$

$$x_1 + x_2 \le 6$$

$$x_1, x_2 \ge 0$$

선형 계획 문제의 예
- 2개의 설계 변수와 부등호("≤") 제약 조건을 가진 문제

목적 함수

$$Maximize \quad z = 4x_1 + 5x_2$$

제약 조건

$$\begin{aligned} Subject\ to \quad & -x_1 + x_2 \le 4 \\ & x_1 + x_2 \le 6 \\ & x_1, x_2 \ge 0 \end{aligned}$$

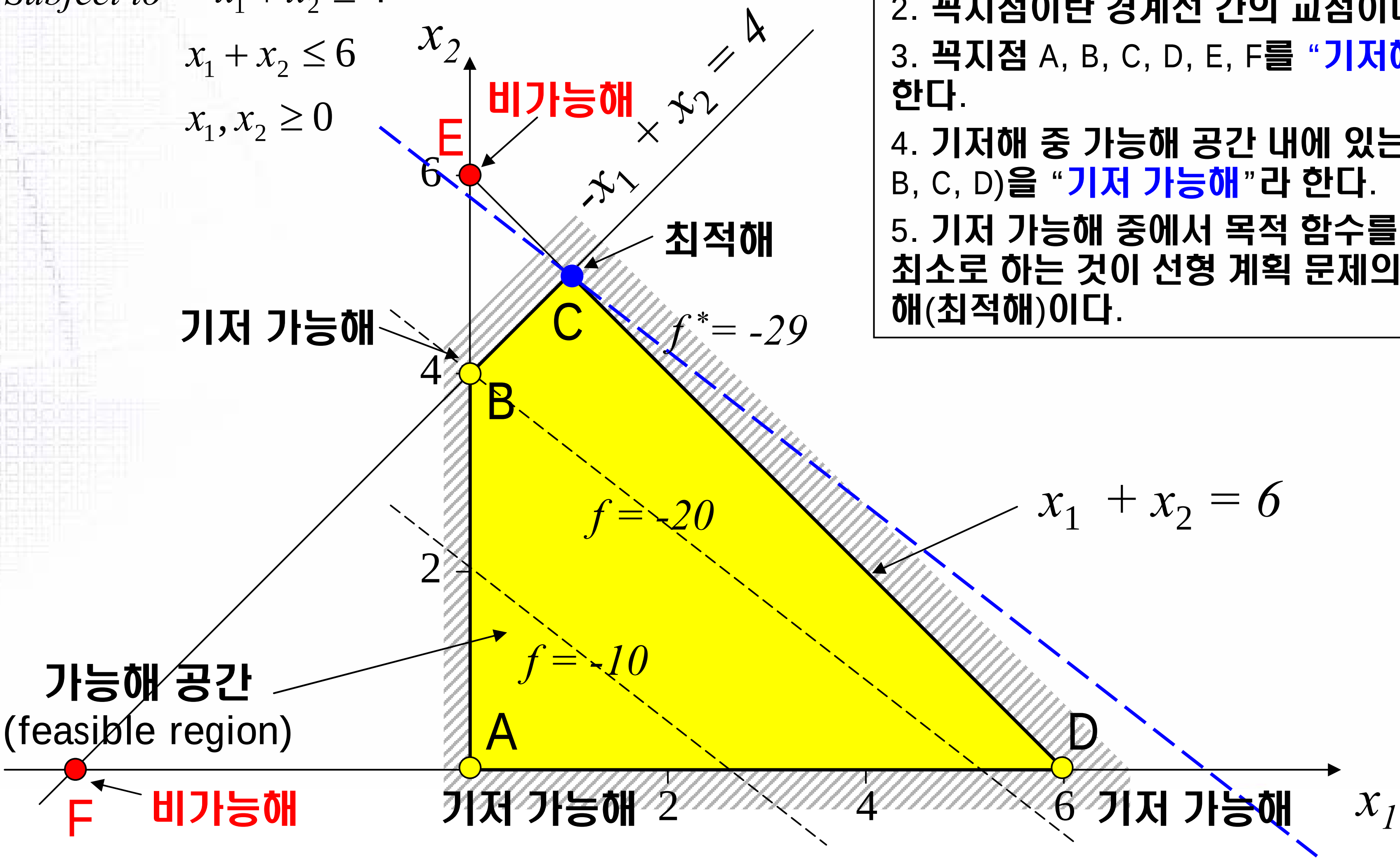
목적 함수의 최대화 문제를 최소화 문제로 변환

$$Minimize \quad f = -4x_1 - 5x_2$$

$$\begin{aligned} Subject\ to \quad & -x_1 + x_2 \le 4 \\ & x_1 + x_2 \le 6 \\ & x_1, x_2 \ge 0 \end{aligned}$$

3.2 선형 계획 문제의 기하학적 해법

$$\begin{aligned}
&\textit{Minimize} \quad f = -4x_1 - 5x_2 \\
&\textit{Subject to} \quad -x_1 + x_2 \le 4 \\
&\qquad x_1 + x_2 \le 6 \\
&\qquad x_1, x_2 \ge 0
\end{aligned}$$



1. 선형 계획 문제의 해는 꼭지점 상에 있다.
2. 꼭지점이란 경계선 간의 교점이다.
3. 꼭지점 A, B, C, D, E, F를 "기저해"라 한다.
4. 기저해 중 가능해 공간 내에 있는 것(A, B, C, D)을 "기저 가능해"라 한다.
5. 기저 가능해 중에서 목적 함수를 최소로 하는 것이 선형 계획 문제의 해(최적해)이다.

선형 계획 문제의 해결을 위한 부등호("≤") 제약 조건의 변환 방법

Minimize $f = -4x_1 - 5x_2$

Subject to $\boxed{-x_1 + x_2 \le 4}$

$x_1 + x_2 \le 6$

$x_1, x_2 \ge 0$

"≤" 형태의 부등호 제약 조건: 완화 변수(slack variable)의 도입

$$-x_1 + x_2 \le 4 \quad \Rightarrow \quad -x_1 + x_2 + \underline{x_3} = 4$$

완화 변수(0보다 크거나 같음)

선형 계획 문제를 풀 때 "≤" 형태의 부등호 제약 조건을 등호 제약 조건으로 변환함("표준형")

선형 계획 문제의 해법(1)

"≤" 형태의 제약 조건을 등호 제약 조건으로 변환하기 위해 도입된 완화 변수(slack variable)

$$\begin{aligned} &\textit{Minimize} && f = -4x_1 - 5x_2 \\ &\textit{Subject to} && -x_1 + x_2 \le 4 \\ & && x_1 + x_2 \le 6 \\ & && x_1, x_2 \ge 0 \end{aligned}$$

부등호 제약 조건을 등호 제약 조건으로 변환

$$\begin{aligned} &\textit{Minimize} && f = -4x_1 - 5x_2 \\ &\textit{Subject to} && -x_1 + x_2 + x_3 = 4 \\ & && x_1 + x_2 + x_4 = 6 \\ & && x_1, x_2, x_3, x_4 \ge 0 \end{aligned}$$

원래의 문제를 등호 제약 조건으로 표현("표준형")(단, 우변은 0보다 작지 않다고 가정함)

$$\left.\begin{aligned} -x_1 + x_2 + x_3 \quad &= 4 \\ x_1 + x_2 \quad + x_4 &= 6 \end{aligned}\right\}$$

무수히 많은 해가 존재(미지수 4개, 식 2개)하는 부정 방정식

➡ **4개의 변수 중 2개의 변수를 가정하면 해를 구할 수 있다.**

➡ **선형 계획 문제의 해법인 "Simplex 방법"에서는 2개의 변수를 0으로 가정하여 해를 구한다.
이때, 0으로 가정하는 변수를 "비기저 변수", 이로부터 구해지는 변수를 "기저 변수"라고 한다.**

선형 계획 문제의 해법(2)

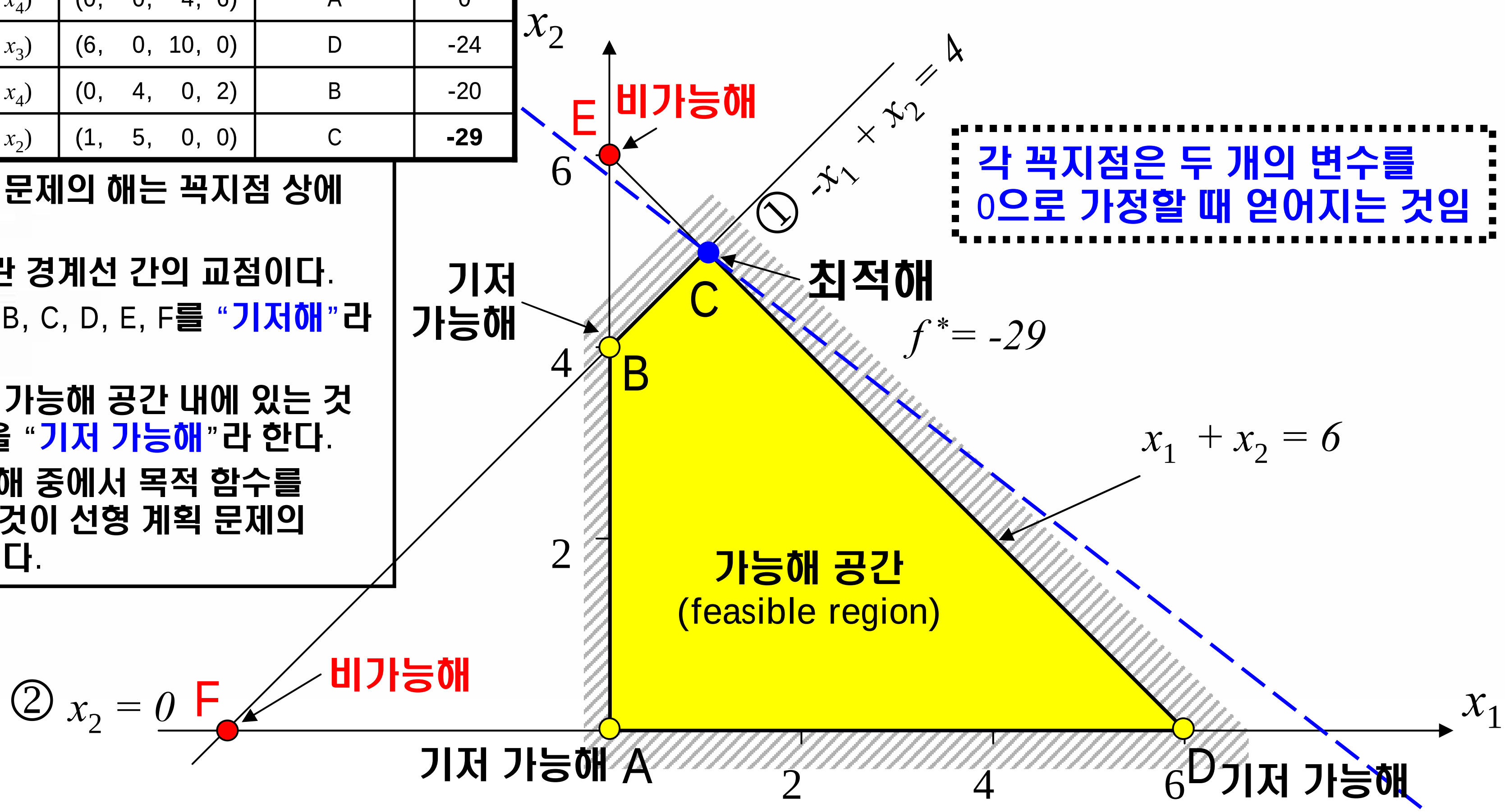
$$\text{Minimize} \quad f = -4x_1 - 5x_2$$
$$\text{Subject to} \quad -x_1 + x_2 + x_3 = 4$$
$$x_1 + x_2 + x_4 = 6$$
$$x_1, x_2, x_3, x_4 \geq 0$$

1. 4개의 변수 중 2개의 변수를 가정하면 해를 구할 수 있다.
2. 선형 계획 문제의 해법인 "Simplex 방법"에서는 2개의 변수를 0으로 가정하여 해를 구한다.
   이때, 0으로 가정하는 변수를 "비기저 변수", 이로부터 구해지는 변수를 "기저 변수"라고 한다.

비기저 변수 (0으로 가정)	기저 변수	해 (x_1, x_2, x_3, x_4)	해("꼭지점")의 위치	목적 함수
(x_2, x_3)	(x_1, x_4)	(-4, 0, 0, 10)	F	16
(x_1, x_4)	(x_2, x_3)	(0, 6, -2 0)	E	-30
(x_1, x_2)	(x_3, x_4)	(0, 0, 4, 6)	A	0
(x_2, x_4)	(x_1, x_3)	(6, 0, 10, 0)	D	-24
(x_1, x_3)	(x_2, x_4)	(0, 4, 0, 2)	B	-20
(x_3, x_4)	(x_1, x_2)	(1, 5, 0, 0)	C	**-29**

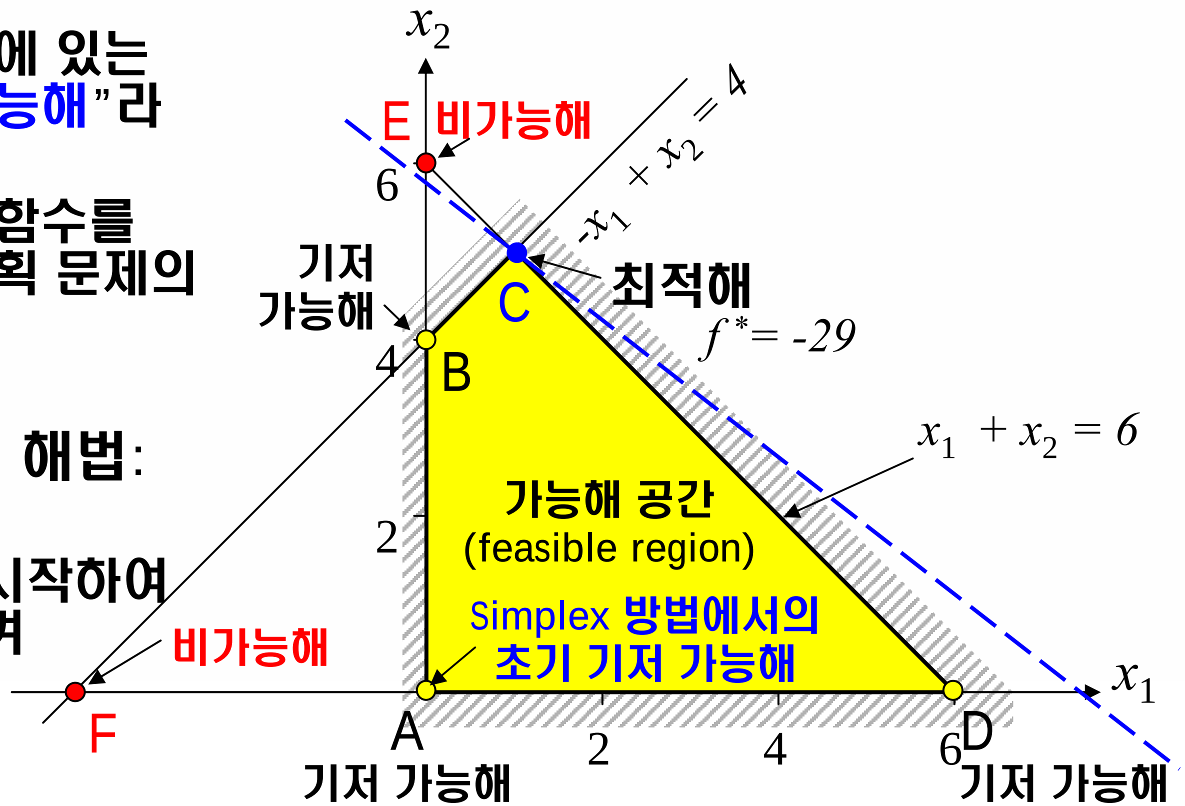
1. 선형 계획 문제의 해는 꼭지점 상에 있다.
2. 꼭지점이란 경계선 간의 교점이다.
3. 꼭지점 A, B, C, D, E, F를 "기저해"라 한다.
4. 기저해 중 가능해 공간 내에 있는 것 (A, B, C, D)을 "기저 가능해"라 한다.
5. 기저 가능해 중에서 목적 함수를 최소로 하는 것이 선형 계획 문제의 해(최적해)이다.

F 점의 경우, 제약 조건 $-x_1 + x_2 = 4$(①)과 $x_2 = 0$(②)의 교점이다. 이 교점은 $-x_1 + x_2 + x_3 = 4$에서 $x_3 = 0$으로 둔 것이고, 또한 $x_2 = 0$인 점이다



선형 계획 문제의 해법 요약

- 선형 계획법
  - 선형 계획 문제의 해는 꼭지점 상에 있다.
  - 꼭지점이란 경계선 간의 교점이다.
  - 꼭지점 A, B, C, D, E, F를 "**기저해**"라 한다.
  - 기저해 중 가능해 공간 내에 있는 것(A, B, C, D)을 "**기저 가능해**"라 한다.
  - 기저 가능해 중에서 목적 함수를 최소로 하는 것이 선형 계획 문제의 해(최적해)이다.
- 선형 계획 문제의 일반적인 해법: "Simplex 방법"
  - 초기 기저 가능해로부터 시작하여 목적 함수를 점차 개선시켜 최적해를 구하는 방법



3.3 Simplex 방법을 이용한 선형 계획 문제의 해법(1)

(1) 부등호 제약 조건을 등호 제약 조건으로 변형

$$\begin{aligned} \textit{Minimize} \quad & f = -4x_1 - 5x_2 \\ \textit{Subject to} \quad & -x_1 + x_2 \le 4 \\ & x_1 + x_2 \le 6 \\ & x_1, x_2 \ge 0 \end{aligned}$$

(2) 초기 기저 가능해의 결정

시작할 때는 문제의 원래 변수(x_1, x_2)를 "비기저 변수"로 가정하면(x_1=x_2=0), 완화 변수(x_3, x_4)의 값을 바로 구할 수 있다("기저 변수").

	기저 변수	비기저 변수		기저 변수			
1행:	x_3	$-x_1$	$+x_2$	$+x_3$		$=4$	$4/1 = 4$
2행:	x_4	x_1	$+x_2$		$+x_4$	$=6$	$6/1 = 6$
3행:		$-4x_1$	$-5x_2$			$=f-0$	

$$x_1, x_2, x_3, x_4 \ge 0$$

1행이 선택됨

2열이 선택됨
(최소값 = -5)

(3) 목적 함수 계수가 최소인 열을 선택
(목적 함수를 가장 빨리 감소시킬 수 있으므로)

(4) 제약 조건식 중 선택된 열의 계수가 양수이며 최소의 비율을 갖는 행을 선택

비율 = (각 행의 우변의 값) /
(각 행의 선택된 열에서의 해당 변수의 계수)

<참고> 만약 최소의 비율을 갖는 행을 선택하지 않는다면?

(5) 개선된 기저 가능해의 결정

선택된 열에 해당하는 비기저 변수(예, x_2)를 기저 변수로,
선택된 행에 해당하는 기저 변수(예, x_3)를
비기저 변수(x_3=0)로 변경함

(6) 선택된 행, 열(예, 1행 2열)의 변수(x_2)를 중심으로 Pivot을 실시

1행을 정리하면, $x_2 = 4 + x_1 - x_3$

이를 2, 3행에 대입하면

$$x_1 + (4 + x_1 - x_3) + x_4 = 6$$
$$\Rightarrow 2x_1 - x_3 + x_4 = 2$$
$$-4x_1 - 5(4 + x_1 - x_3) = f$$
$$\Rightarrow -9x_1 + 5x_3 = f + 20$$

➡ 위 결과는 실제로 1행 2열의 변수(x_2)를 중심으로 Pivot을 하는 것과 동일함

3.3 Simplex 방법을 이용한 선형 계획 문제의 해법(2)

Pivot 이전 식:

$$\begin{aligned} &\text{1행: } -x_1 + x_2 + \underbrace{x_3}_{\text{기저 변수}} = 4 \\ &\text{2행: } x_1 + x_2 + x_4 = 6 \\ &\text{3행: } -4x_1 - 5x_2 = f - 0 \end{aligned}$$

(6) 선택된 행, 열(예, 1행 2열)의 변수를 중심으로 Pivot을 실시

비기저 변수

기저 변수		
1행:	x_2	$-x_1 + x_2 + x_3 = 4$
2행:	x_4	$2x_1 - x_3 + x_4 = 2$
3행:		$-9x_1 + 5x_3 = f + 20$

$x_1, x_2, x_3, x_4 \geq 0$

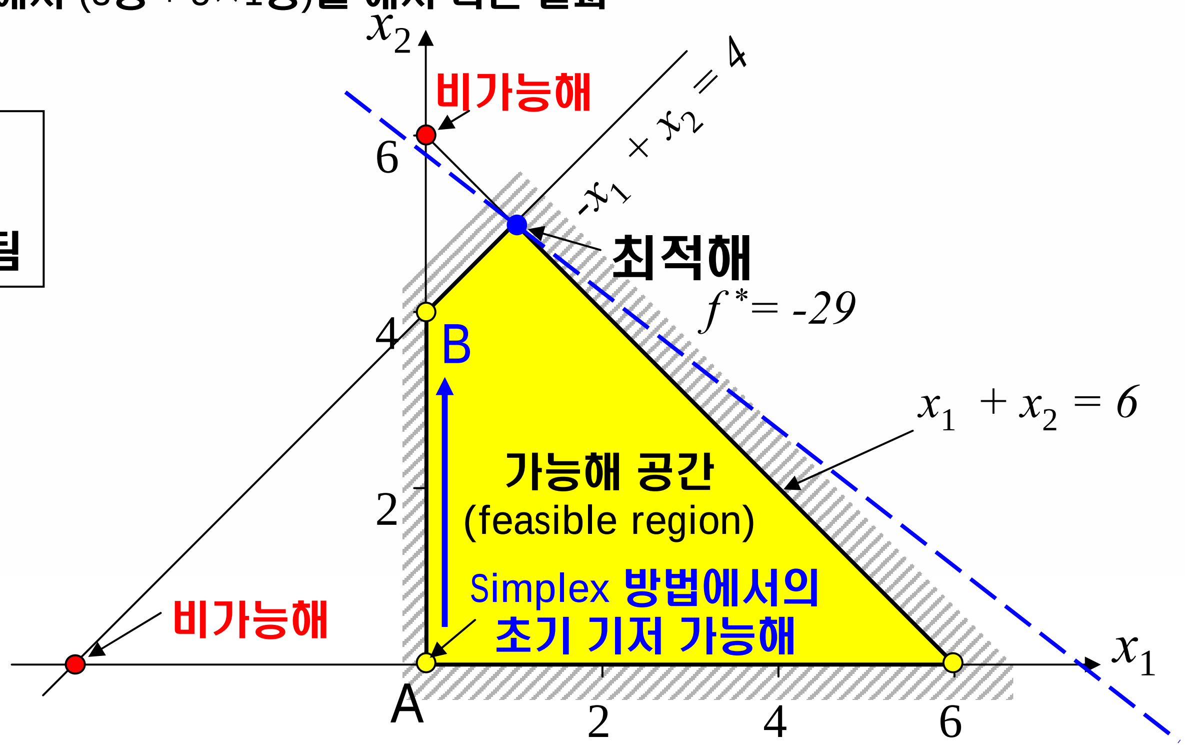
단계 (2)에서 (2행 - 1행)을 해서 나온 결과

단계 (2)에서 (3행 + 5×1행)을 해서 나온 결과

이전 단계의 비기저 변수 x2가 기저 변수로, 이전 단계의 기저 변수 x3가 비기저 변수로 변경됨

Pivot: 선택된 행, 열의 변수의 계수만 1이고 선택된 열의 나머지 변수의 계수는 모두 0이 되도록 각 행들끼리 연산을 하는 것

➡ **새로운 해** B(x_1, x_2, x_3, x_4) = (0, 4, 0, 2)**와**
개선된 목적 함수 값 -20을 얻게 됨



3.3 Simplex 방법을 이용한 선형 계획 문제의 해법(3)

(7) 목적 함수 계수가 최소인 열을 선택

비기저 변수

기저 변수		
1행:	x_2	$-x_1 + x_2 + x_3 \quad = 4$
2행:	x_4	$2x_1 \quad - x_3 + x_4 = 2$
3행:		$-9x_1 \quad + 5x_3 \quad = f + 20$

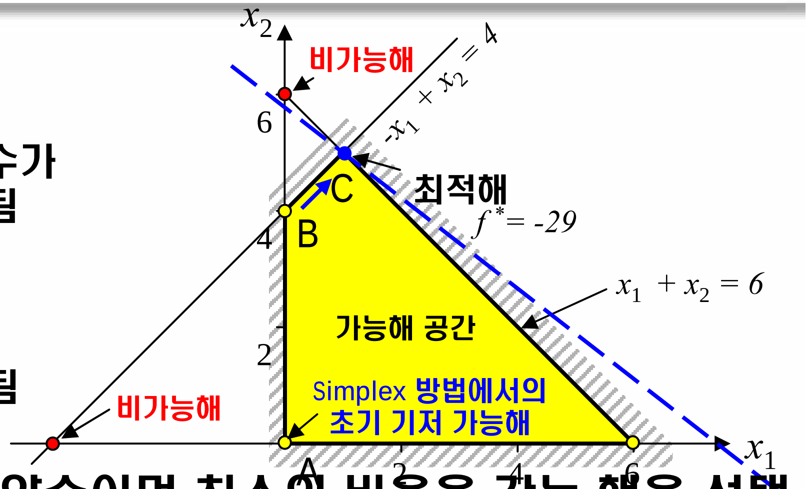
$x_1, x_2, x_3, x_4 \geq 0$

선택된 열의 계수가 음수면 선택 안됨

2/2 = 1

2행이 선택됨

1열이 선택됨 (최소값 = -9)



(8) 제약 조건식 중 선택된 열의 계수가 양수이며 최소의 비율을 갖는 행을 선택

(9) 개선된 기저 가능해의 결정(x_1이 기저 변수로, x_4가 비기저 변수로 변경됨)

(10) 선택된 행, 열(예, 2행 1열)의 변수를 중심으로 Pivot을 실시

비기저 변수

기저 변수		
1행:	x_2	$x_2 + 0.5x_3 + 0.5x_4 \quad = 5$
2행:	x_1	$x_1 \quad - 0.5x_3 + 0.5x_4 \quad = 1$
3행:		$+0.5x_3 + 4.5x_4 \quad = f + 29$

$x_1, x_2, x_3, x_4 \geq 0$

단계 (5)에서 (1행 + 0.5×2행)을 해서 나온 결과

단계 (5)에서 (0.5×2행)을 해서 나온 결과

단계 (5)에서 (3행 + 4.5×2행)을 해서 나온 결과

➡ 최적해 C(x_1, x_2, x_3, x_4)
= (1, 5, 0, 0)와
목적 함수 값 -29를 얻게 됨

(11) 목적 함수의 모든 계수가
0 이상이므로 현재의 해가 최적해임

[참고] Simplex 방법에서 선택된 열이 양수이며 최소의 비율을 갖는 행을 선택하는 이유

	기저 변수	비기저 변수 / 기저 변수		
1행:	x_3	$-x_1 + x_2 + x_3 = 4$	4/1 = 4	← 최소의 비율을 갖는 행(1행)
2행:	x_4	$x_1 + x_2 + x_4 = 6$	6/1 = 6	
3행:		$-4x_1 - 5x_2 = f - 0$		

$$x_1, x_2, x_3, x_4 \ge 0$$

2열이 선택됨
(최소값 = -5)

위의 1, 2행을 다시 정리하면,

$$-x_1 + x_3 = 4 - x_2$$
$$x_1 + x_4 = 6 - x_2$$

1) x_2 = 4일 때(1행을 선택했을 때)

$$x_1 = x_3 = 0, x_4 = 2$$

2) x_2 = 6일 때(2행을 선택했을 때)

$$x_1 = x_4 = 0, x_3 = -2$$ ➡ 음이 아니라는 조건을 위배함

Simplex 단체표를 이용한 선형 계획 문제의 해법

비기저 변수 (0으로 두는 변수) 기저 변수

기저 변수

1행: x_3 $-x_1 + x_2 + x_3 = 4$ ← $4/1 = 4$

2행: x_4 $x_1 + x_2 + x_4 = 6$ ← $6/1 = 6$

3행: $-4x_1 - 5x_2 = f - 0$

기저 변수

		x1	x2	x3	x4	bi	bi/ai
1행:	x3	-1	1	1	0	4	4
2행:	x4	1	1	0	1	6	6
3행:	Obj.	-4	-5	0	0	f-0	-

1행 2열 중심으로 Pivot을 실시

새로운 2행 = (2행 - 1행)
새로운 3행 = (3행 + 5×1행)

비기저 변수

기저 변수

1행: x_2 $-x_1 + x_2 + x_3 = 4$ ← $4/-1 = -4$ (음수면 선택 안됨)

2행: x_4 $2x_1 - x_3 + x_4 = 2$ ← $2/2 = 1$

3행: $-9x_1 + 5x_3 = f + 20$

기저 변수

		x1	x2	x3	x4	bi	bi/ai
1행:	X2	-1	1	1	0	4	-4
2행:	x4	2	0	-1	1	2	1
3행:	Obj.	-9	0	5	0	f+20	-

2행 1열 중심으로 Pivot을 실시

새로운 1행 = (1행 + 0.5×2행)
새로운 2행 = (0.5×2행)
새로운 3행 = (3행 + 4.5×2행)

비기저 변수

기저 변수

1행: x_2 $x_2 + 0.5x_3 + 0.5x_4 = 5$

2행: x_1 $x_1 - 0.5x_3 + 0.5x_4 = 1$

3행: $+0.5x_3 + 4.5x_4 = f + 29$

기저 변수

		x1	x2	x3	x4	bi	bi/ai
1행:	x2	0	1	0.5	0.5	5	-
2행:	x1	1	0	-0.5	0.5	1	-
3행:	Obj.	0	0	0.5	4.5	f+29	-

목적 함수의 모든 계수가 0 이상이므로
현재의 해가 최적해임(x_1=1, x_2=5, x_3=x_4=0, f=-29)

선형 계획 문제의 예
- 2개의 설계 변수와 부등호("≥") 제약 조건을 가진 문제

$$\text{Maximize} \quad z = y_1 + 2y_2$$

$$\text{Subject to} \quad 3y_1 + 2y_2 \le 12$$

$$2y_1 + 3y_2 \ge 6$$

$$y_1 \ge 0$$

y_2 는 부호 제한 없음

목적 함수의 최대화 문제를 최소화 문제로 변환

$$\text{Minimize} \quad F = -y_1 - 2y_2$$

$$\text{Subject to} \quad 3y_1 + 2y_2 \le 12$$

$$2y_1 + 3y_2 \ge 6$$

$$y_1 \ge 0$$

y_2 는 부호 제한 없음

부호 제한이 없는 변수를
값이 음이 아닌 변수로 변환
($y_2 = y_2^+ - y_2^-$)
$x_1 = y_1, x_2 = y_2^+, x_3 = y_2^-$라고 가정하면

$$\text{Minimize} \quad f = -x_1 - 2x_2 + 2x_3$$

$$\text{Subject to} \quad 3x_1 + 2x_2 - 2x_3 \le 12$$

$$2x_1 + 3x_2 - 3x_3 \ge 6$$

$$x_1, x_2, x_3 \ge 0$$

Simplex 방법을 이용하기 위한 부등호("≥") 제약 조건의 변환 방법(1)

Minimize $f = -x_1 - 2x_2 + 2x_3$

Subject to $3x_1 + 2x_2 - 2x_3 \le 12$

$$\boxed{2x_1 + 3x_2 - 3x_3 \ge 6}$$

$$x_1, x_2, x_3 \ge 0$$

"≥" 형태의 부등호 제약 조건:

잉여 변수(surplus variable) 및 인위 변수(artificial variable)의 도입

$$2x_1 + 3x_2 - 3x_3 \ge 6 \quad \Rightarrow \quad \mathbf{2x_1 + 3x_2 - 3x_3} - \underbrace{\boldsymbol{x_4}}_{\text{잉여 변수 (0보다 크거나 같음)}} + \underbrace{x_5}_{\text{인위 변수(0보다 크거나 같음)}} = 6$$

"인위 변수 도입 이유"

Simplex 방법을 시작할 때 문제의 원래 변수(x_1, x_2, x_3)를 "비기저 변수"로 가정하면(x_1=x_2=x_3=0), $-x_4$ = 6이 된다.

➡ 수학적으로 정합성이 없으므로 x_5를 인위적으로 추가하여 수학적인 정합성을 유지한다.
그런데 x_5는 인위적으로 추가한 것이므로 최적해가 존재하면 이 변수의 값은 반드시 0이 되어야 한다.

Simplex 단체표를 이용한 선형 계획 문제의 해법
- "≥" 형태의 제약 조건을 가진 문제에 대한 Simplex 방법(1)

①

Maximize $z = y_1 + 2y_2$

Subject to $3y_1 + 2y_2 \le 12$

$2y_1 + 3y_2 \ge 6$

$y_1 \ge 0$

y_2 는 부호 제한 없음

1. 최소화 문제로 변환함
2. y_2 는 부호 제한이 없으므로
   $y_2 = y_2^+ - y_2^-$ 로 분해함
3. $x_1 = y_1, x_2 = y_2^+, x_3 = y_2^-$
   라고 가정함
4. 부등호 제약 조건을 등호 제약
   조건으로 만듬
   (완화, 잉여 변수의 도입)

②

완화 변수

Minimize $f = -x_1 - 2x_2 + 2x_3$

Subject to $3x_1 + 2x_2 - 2x_3 + x_4 = 12$

$2x_1 + 3x_2 - 3x_3 - x_5 = 6$

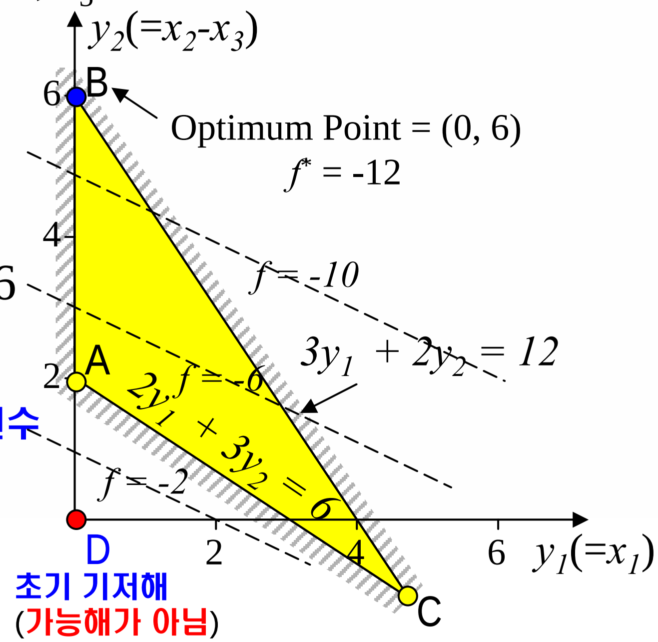
$x_i \ge 0; i = 1\ to\ 5$

잉여 변수

원래 변수 x_1, x_2, x_3 를 모두 0이라 두고(비기저 변수)

기저 변수 x_4, x_5 를 구하면

$x_4 = 12, x_5 = -6$ ➡ 음이 아니라는 조건을 위배함



"≥" 형태의 등호 제약 조건에
인위 변수 x_6 를 추가적으로
도입함

③

완화 변수

Minimize $f = -x_1 - 2x_2 + 2x_3$

Subject to $3x_1 + 2x_2 - 2x_3 + x_4 = 12$

$2x_1 + 3x_2 - 3x_3 - x_5 + x_6 = 6$

$x_i \ge 0; i = 1\ to\ 6$

잉여 변수 인위 변수

원래 변수 x_1, x_2, x_3 과 잉여 변수 x_5를

모두 0이라 두고(비기저 변수)

기저 변수 x_4, x_6 을 구하면

$x_4 = 12, x_6 = 6$ ➡ 초기 기저해(가능해가 아님)

그런데 x_6은 인위적으로 추가한 변수이므로

이 값을 0으로 만들어야 함

Simplex 단체표를 이용한 선형 계획 문제의 해법
- “≥” 형태의 제약 조건을 가진 문제에 대한 Simplex 방법(2)

3

완화 변수

Minimize $f = -x_1 - 2x_2 + 2x_3$

Subject to $3x_1 + 2x_2 - 2x_3 + x_4 = 12$

$2x_1 + 3x_2 - 3x_3 - x_5 + x_6 = 6$

$x_i \geq 0; i = 1\ to\ 6$

잉여 변수　인위 변수

인위 변수의 합으로 표현되는 인위 목적 함수(w=x_6)를 정의함

4

$$3x_1 + 2x_2 - 2x_3 + x_4 = 12$$

$$2x_1 + 3x_2 - 3x_3 - x_5 + x_6 = 6$$

$$-x_1 - 2x_2 + 2x_3 = f$$

$$\boxed{-2x_1 - 3x_2 + 3x_3 + x_5 = w - 6}$$

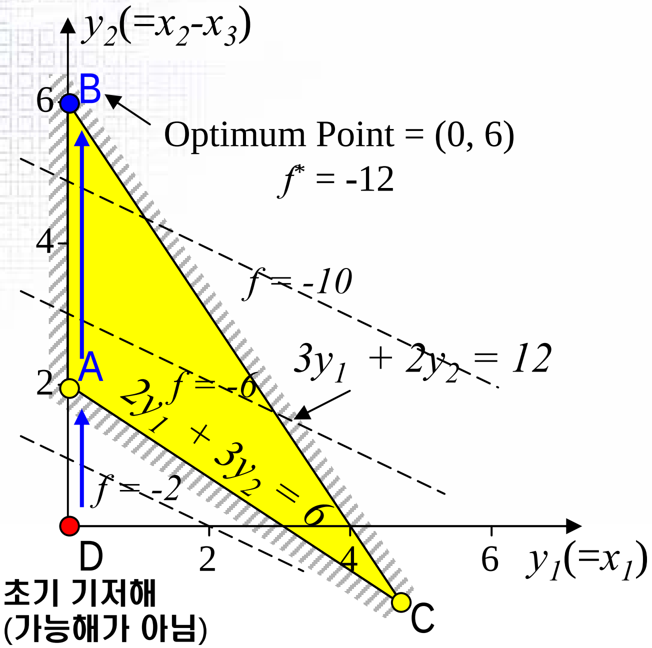
인위 목적 함수 식

$2x_1 + 3x_2 - 3x_3 - x_5 + x_6 = 6$ 식에서 x_6=w라 두고 정리한 것임

5 초기 기저 가능해(인위 목적 함수 w=x_6을 최소화(“w=0”) 하는 해)를 구함 (Simplex 방법의 Phase 1)

인위 변수는 인위적으로 추가한 것이므로 최적해가 존재하면 이 변수의 값은 0이어야 함

6 목적 함수 f 를 최소화 하는 해를 구함 (Simplex 방법의 Phase 2)

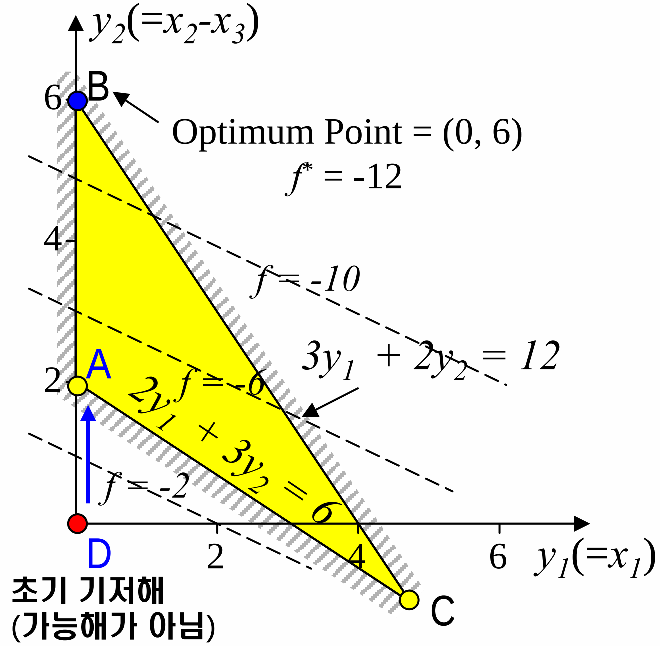


초기 기저해 (가능해가 아님)

Simplex 단체표를 이용한 선형 계획 문제의 해법
- "≥" 형태의 제약 조건을 가진 문제에 대한 Simplex 방법(3)

④

$$3x_1 + 2x_2 - 2x_3 + x_4 = 12$$
$$2x_1 + 3x_2 - 3x_3 - x_5 + x_6 = 6$$
$$-x_1 - 2x_2 + 2x_3 = f$$
$$-2x_1 - 3x_2 + 3x_3 + x_5 = w - 6$$



초기에는 원래 변수($x_1, \ldots, x_3$), 잉여 변수(x_5)를 0으로 가정하고("비기저 변수"), 완화 변수(x_4)와 인위 변수(x_6)를 기저 변수로 가정하여 풀기 시작함("초기 기저해로부터 시작")

	x1	x2	x3	x4	x5	x6	bi	bi/ai
x4	3	2	-2	1	0	0	12	-
x6	2	3	-3	0	-1	1	6	-
Obj.	-1	-2	2	0	0	0	*f*-0	-
A. Obj.	-2	-3	3	0	1	0	w-6	-

⑤ Phase 1: 인위 목적 함수를 기준으로 Pivot을 수행함(w = 0이 될 때까지 수행)

	x1	x2	x3	x4	x5	x6	bi	bi/ai
x4	3	2	-2	1	0	0	12	6
x6	2	**3**	-3	0	-1	1	6	**2**
Obj.	-1	**-2**	2	0	0	0	*f*-0	-
A. Obj.	-2	-3	3	0	1	0	w-6	-

➡

	x1	x2	x3	x4	x5	x6	bi	bi/ai
x4	5/3	0	0	1	2/3	-2/3	8	-
x2	2/3	1	-1	0	-1/3	1/3	2	-
Obj.	1/3	0	0	0	-2/3	2/3	*f*+4	-
A. Obj.	0	0	0	0	0	1	**w-0**	-

인위 변수는 인위적으로 추가한 것이므로 최적해가 존재하면 이 변수의 값은 0이어야 함

새로운 1행 = 1행 - (2/3)×2행
새로운 2행 = (1/3)×2행
새로운 2행 = 3행 - (2/3)×2행
새로운 4행 = 4행 + 2행

인위 목적 함수가 0이므로
Phase 1이 완료되었음
점 A(x_1=x_3=x_5=x_6=0, x_2=2, x_4=8)

Simplex 단체표를 이용한 선형 계획 문제의 해법
- "≥" 형태의 제약 조건을 가진 문제에 대한 Simplex 방법(4)

⑤ Phase 1: 인위 목적 함수를 기준으로 Pivot을 수행함(w = 0이 될 때까지 수행)

	x1	x2	x3	x4	x5	x6	bi	bi/ai
x4	3	2	-2	1	0	0	12	6
x6	2	**3**	-3	0	-1	1	6	**2**
Obj.	-1	**-2**	2	0	0	0	f-0	-
A. Obj.	-2	-3	3	0	1	0	w-6	-

➡

	x1	x2	x3	x4	x5	x6	bi	bi/ai
x4	5/3	0	0	1	2/3	-2/3	8	-
x2	2/3	1	-1	0	-1/3	1/3	2	-
Obj.	1/3	0	0	0	-2/3	2/3	f+4	-
A. Obj.	0	0	0	0	0	1	**w-0**	-

⑥ Phase 2: 목적 함수 f를 기준으로 Pivot을 수행함(목적 함수의 모든 계수가 음이 아닐 때까지 수행)

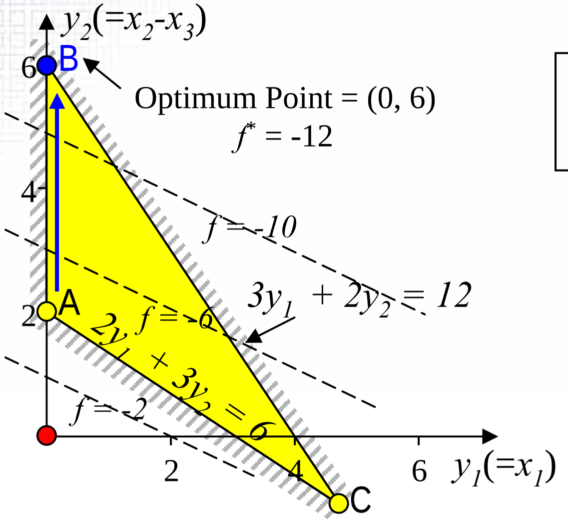
	x1	x2	x3	x4	x5	x6	bi	bi/ai
x4	5/3	0	0	1	**2/3**	-2/3	8	**12**
x2	2/3	1	-1	0	-1/3	1/3	2	-6
Obj.	1/3	0	0	0	**-2/3**	2/3	f+4	-

➡

	x1	x2	x3	x4	x5	x6	bi	bi/ai
x5	5/2	0	0	3/2	1	-1	**12**	-
x2	3/2	1	-1	1/2	0	0	**6**	-
Obj.	2	0	0	1	0	0	*f+12*	-

새로운 1행 = 1행 × (2/3)
새로운 2행 = 2행 + (1/2)×1행
새로운 3행 = 3행 + 1행

목적 함수의 모든 계수가 0 이상이므로
현재의 해가 최적해임
($x_1=x_3=x_4=0, x_2=6, x_5=12, f=-12$)



인위 목적 함수의 구성 방법Simplex 방법을 이용하기 위한 등호("=") 제약 조건의 변환 방법

Minimize $f = -x_1 - 2x_2 + 2x_3$

Subject to $3x_1 + 2x_2 - 2x_3 \le 12$

$$2x_1 + 3x_2 - 3x_3 \ge 6$$

$$\boxed{x_1 + x_2 + x_3 = 6}$$

$$x_1, x_2, x_3 \ge 0$$

"=" 형태의 등호 제약 조건: 인위 변수(artificial variable)의 도입

$$x_1 + x_2 + x_3 = 6 \quad \Rightarrow \quad x_1 + x_2 + x_3 + \underline{x_4} = 6$$

인위 변수(0보다 크거나 같음)

"인위 변수 도입 이유"

Simplex 방법을 시작할 때 문제의 원래 변수(x_1, x_2, x_3)를 "비기저 변수"로 가정하면(x_1=x_2=x_3=0), 이 식이 성립하지 않는다.

➡ 수학적으로 정합성이 없으므로 x_4를 인위적으로 추가하여 수학적인 정합성을 유지한다.
그런데 x_4는 인위적으로 추가한 것이므로 최적해가 존재하면 이 변수의 값은 반드시 0이 되어야 한다.

인위 목적 함수의 정의 방법

1

Minimize $f = -x_1 - 2x_2 + 2x_3$

Subject to $3x_1 + 2x_2 - 2x_3 \le 12$

$2x_1 + 3x_2 - 3x_3 \ge 6$

$x_1 + x_2 + x_3 = 6$

$x_1, x_2, x_3 \ge 0$

부등호 제약 조건을 등호 제약 조건으로 변환 →

2

Minimize $f = -x_1 - 2x_2 + 2x_3$

Subject to $3x_1 + 2x_2 - 2x_3 + x_4 = 12$

$2x_1 + 3x_2 - 3x_3 - x_5 + x_6 = 6$

$x_1 + x_2 + x_3 + x_7 = 6$

$x_i \ge 0;\ i = 1 \text{ to } 7$

$x_6 - 6 = -2x_1 - 3x_2 + 3x_3 + x_5$

$x_7 - 6 = -x_1 - x_2 - x_3$

$w(= x_6 + x_7) - 12 = -3x_1 - 4x_2 + 2x_3 + x_5$

인위 변수의 합으로 표현되는 인위 목적 함수($w=x_6+x_7$)를 정의함

3

$3x_1 + 2x_2 - 2x_3 + x_4 = 12$

$2x_1 + 3x_2 - 3x_3 - x_5 + x_6 = 6$

$x_1 + x_2 + x_3 + x_7 = 6$

$-x_1 - 2x_2 + 2x_3 = f$

$-3x_1 - 4x_2 + 2x_3 + x_5 = w - 12$

초기 기저 가능해(인위 목적 함수 $w=x_6+x_7$을 최소화 ("w=0"; $x_6=x_7=0$) 하는 해)를 구함

<참고> 인위 변수 각각에 대해 인위 목적 함수를 정의하는 경우

$3x_1 + 2x_2 - 2x_3 + x_4 = 12$

$2x_1 + 3x_2 - 3x_3 - x_5 + x_6 = 6$

$x_1 + x_2 + x_3 + x_7 = 6$

$-x_1 - 2x_2 + 2x_3 = f$

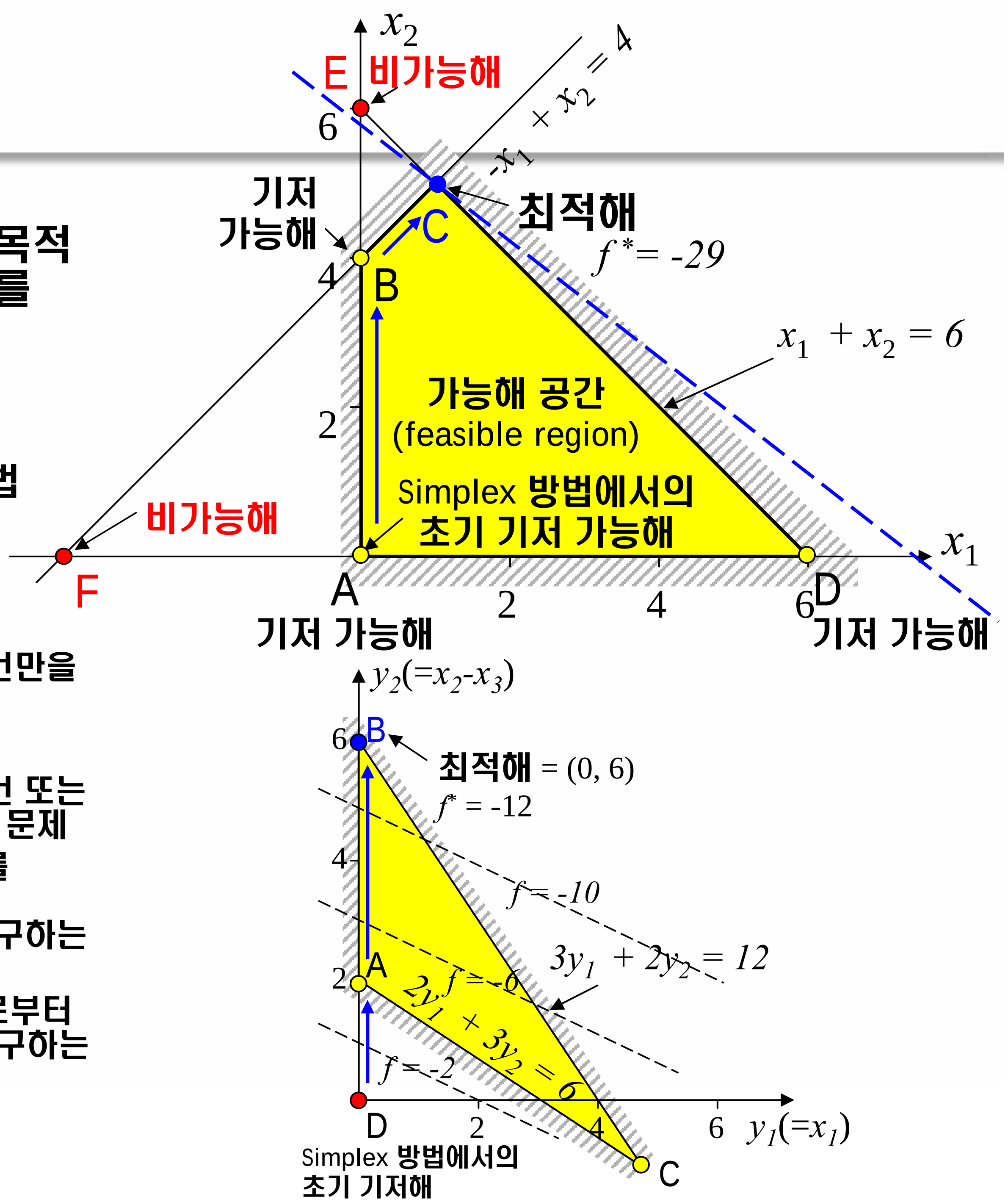
$$\left.\begin{aligned} -2x_1 - 3x_2 + 3x_3 + x_5 &= w_1 - 6 \\ -x_1 - x_2 - x_3 &= w_2 - 6 \end{aligned}\right\}$$

➡ **인위 목적 함수 w_1을 최소화(x_6=0) 하고, 인위 목적 함수 w_2를 최소화(x_7=0) 하는 과정을 각각 수행함**

인위 변수는 0보다 크거나 같으므로, 인위 목적 함수를 최소화 한다는 것은 인위 변수가 모두 0의 값을 가져 인위 목적 함수의 값이 0인 것을 의미함 따라서, 인위 변수의 합으로 인위 목적 함수를 정의하는 것이 편리함

Simplex 방법의 요약

- 기저 가능해로부터 시작하여 목적 함수를 점차 개선시켜 최적해를 구하는 방법
- 1차 연립 방정식의 이론을 바탕으로 함
  - 행렬 연산(가우스-조단 소거법 등)을 이용함
- Simplex 방법의 종류
  - One-phase Simplex 방법
    - "≤" 형태의 부등호 제약 조건만을 가진 문제
  - Two-phase Simplex 방법
    - "≥" 형태의 부등호 제약 조건 또는 등호 제약 조건("=")을 가진 문제
    - Phase 1: 초기 기저 가능해를 선정하기 위해 인위 목적 함수(w)를 0으로 하는 해를 구하는 단계
    - Phase 2: 초기 기저 가능해로부터 해의 개선을 통해 최적해를 구하는 단계



Simplex 방법의 알고리즘의 요약

- 단계 1: 초기 기저 가능해를 선정한다.
  - "<=" 형태의 부등호 제약 조건: 주어진 문제의 원래의 변수를 비기저 변수(0으로 가정)로, 완화 변수를 기저 변수로 가정하여 초기 기저 가능해를 결정함
  - ">=" 형태의 부등로 제약 조건 또는 등호 제약 조건("="): Two-phase Simplex 방법을 이용하며, Phase 1에서 인위 목적 함수 w를 0으로 하는 초기 기저 가능해를 선정함
- 단계 2: 목적 함수를 비기저 변수로만 표현한다.
- 단계 3: 비기저 변수에 대한 목적 함수의 계수가 모두 음이 아닌지를 확인한다. 만약 그렇다면 현재의 해가 최적해이며 그렇지 않다면 다음 단계로 간다.
- 단계 4: Pivot 열과 행을 결정한다. 이때 결정된 Pivot 열에 해당하는 비기저 변수가 새로운 기저 변수로 되고, Pivot 행에 해당하는 기저 변수가 비기저 변수로 된다.
- 단계 5: 가우스-조단 소거법을 이용하여 Pivot을 수행한다.
- 단계 6: 비기저 및 기저 변수의 값을 구한다. 그리고 단계 3으로 간다.

3.4 선형 계획 문제의 예 1
- 선형 계획법을 이용한 최적 화물 수송 문제

어느 선박은 항구 A에서 출발하여 항구 B, C, D를 거쳐 항구 E를 운항한다. 이 선박의 최대 화물 적재 능력은 50,000ton이며 각 항구에서는 다음과 같은 화물을 적재할 수 있다고 하면 화물 종류별로 얼마의 화물을 수송하면 최대의 운임을 얻을 수 있는지 계산하시오.

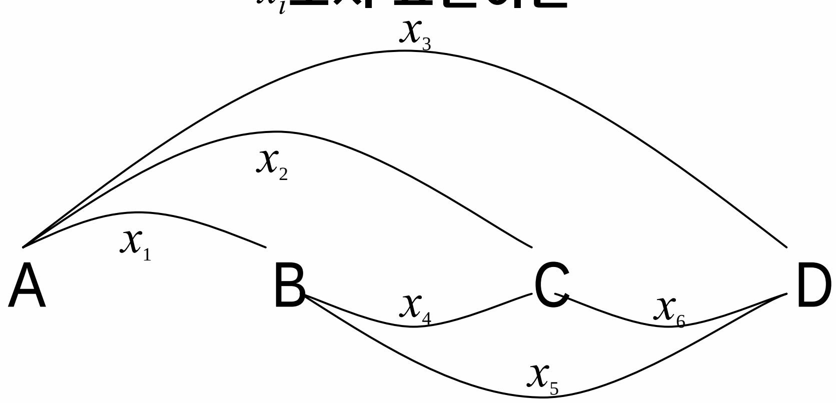
화물 종류	출발 항구	목적 항구	적재할 수 있는 화물량(1,000ton)	운임 요율($/ton)
1	A	B	100	5
2	A	C	40	10
3	A	D	25	20
4	B	C	50	8
5	B	D	100	12
6	C	D	50	6

3.4 선형 계획 문제의 예 1
- 풀이 과정(1)

어느 선박은 항구 A에서 출발하여 항구 B, C를 거쳐 항구 D로 간다. 이 선박의 최대 화물 적재 능력은 50,000ton이며 각 항구에서는 다음과 같은 화물을 적재할 수 있다고 하면 화물 종류별로 얼마의 화물을 수송하면 최대의 운임을 얻을 수 있는지 계산하시오.

화물 종류	출발 항구	목적 항구	적재할 수 있는 화물량(1,000ton)	운임 요율 ($/ton)
1	A	B	100	5
2	A	C	40	10
3	A	D	25	20
4	B	C	50	8
5	B	D	100	12
6	C	D	50	6

화물 종류 i의 적재 톤수를 1,000ton 단위로 x_i로서 표현하면



설계 변수: $x_1, x_2 x_3, x_4, x_5, x_6$

목적 함수: 운임 최대화

$$\textbf{\textit{Maximize}}\ Z = 5x_1 + 10x_2 + 20x_3 + 8x_4 + 12x_5 + 6x_6$$

➡ $f = -Z$라고 가정하면 최소화 문제로 변환시킬 수 있음

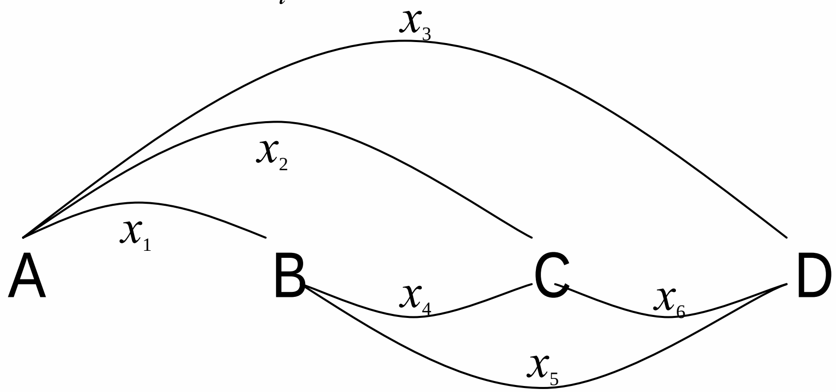
$$\textbf{\textit{Minimize}}\ \ f = -5x_1 - 10x_2 - 20x_3 - 8x_4 - 12x_5 - 6x_6$$

3.4 선형 계획 문제의 예 1 - 풀이 과정(2)

어느 선박은 항구 A에서 출발하여 항구 B, C를 거쳐 항구 D로 간다. 이 선박의 최대 화물 적재 능력은 50,000ton이며 각 항구에서는 다음과 같은 화물을 적재할 수 있다고 하면 화물 종류별로 얼마의 화물을 수송하면 최대의 운임을 얻을 수 있는지 계산하시오.

화물 종류	출발 항구	목적 항구	적재할 수 있는 화물량(1,000ton)	운임 요율 ($/ton)
1	A	B	100	5
2	A	C	40	10
3	A	D	25	20
4	B	C	50	8
5	B	D	100	12
6	C	D	50	6

화물 종류 i의 적재 톤수를 1,000ton 단위로 x_i로서 표현하면



제약 조건:

선박의 최대 화물 적재 능력:

$$A \Rightarrow B : x_1 + x_2 + x_3 \le 50 \quad B \Rightarrow C : x_2 + x_3 + x_4 + x_5 \le 50$$

$$C \Rightarrow D : x_3 + x_5 + x_6 \le 50$$

화물 종류별 최대 적재 가능량:

$$0 \le x_2 \le 40,\ 0 \le x_3 \le 25,\ 0 \le x_4 \le 50,\ 0 \le x_6 \le 50$$

x_1, x_5의 최대 적재 가능량은 50,000톤보다 크므로 x_1, x_5에 대한 상한 제약 조건은 없어도 됨

x_4, x_6 역시 최대 적재 가능량은 50,000톤이므로 상한 제약 조건은 없어도 됨

3.4 선형 계획 문제의 예 1 - 풀이 과정(3)

Find $x_1, x_2, x_3, x_4, x_5, x_6$

Minimize $f = -5x_1 - 10x_2 - 20x_3 - 8x_4 - 12x_5 - 6x_6$

Subject to

$$\left.\begin{aligned} & x_1 + x_2 + x_3 \le 50 \\ & x_2 + x_3 + x_4 + x_5 \le 50 \\ & x_3 + x_5 + x_6 \le 50 \end{aligned}\right\}$$: 선박의 최대 화물 적재 능력에 관한 제약 조건

$$\left.\begin{aligned} & 0 \le x_2 \le 40,\ 0 \le x_3 \le 25, \\ & 0 \le x_4 \le 50,\ 0 \le x_6 \le 50 \end{aligned}\right\}$$: 화물 종류별 최대 적재 가능량에 관한 제약 조건

➡ **미지수 6개, 부등호 제약 조건 7개인 최적화 문제**

3.4 선형 계획 문제의 예 1
- 풀이 과정(4)

① 표준형으로의 변환 → ② Simplex 방법을 이용한 해결 →

제약 조건

$$x_1 + x_2 + x_3 \le 50$$
$$x_2 + x_3 + x_4 + x_5 \le 50$$
$$x_3 + x_5 + x_6 \le 50$$
$$0 \le x_2 \le 40,\ 0 \le x_3 \le 25,$$
$$0 \le x_4 \le 50,\ 0 \le x_6 \le 50$$

목적 함수

$$f = -5x_1 - 10x_2 - 20x_3 - 8x_4 - 12x_5 - 6x_6$$

$$x_1 + x_2 + x_3 + x_7 = 50$$
$$x_2 + x_3 + x_4 + x_5 + x_8 = 50$$
$$x_3 + x_5 + x_6 + x_9 = 50$$
$$x_2 + x_{10} = 40,\ x_3 + x_{11} = 25,$$
$$x_4 + x_{12} = 50,\ x_6 + x_{13} = 50$$

Where, $x_7, x_8, x_9, x_{10}, x_{11}, x_{12} x_{13}$: 완화 변수[1]

$$f = -5x_1 - 10x_2 - 20x_3 - 8x_4 - 12x_5 - 6x_6$$

→ ③

Simplex 방법 수행

초기 기저 가능해로부터 해의 개선을 통해 최적해를 구함

1: 완화 변수(slack variable) - "≤" 형태의 제약 조건을 등호 제약 조건으로 변환하기 위해 도입한 변수

3.4 선형 계획 문제의 예 1
- 풀이 과정(5)

비율 = (각 행의 우변의 값) /
(각 행의 선택된 열에서의 해당 변수의 계수)

1

	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13	bi	bi/ai
x7	1	1	1	0	0	0	1	0	0	0	0	0	0	50	50
x8	0	1	1	1	1	0	0	1	0	0	0	0	0	50	50
x9	0	0	1	0	1	1	0	0	1	0	0	0	0	50	50
x10	0	1	0	0	0	0	0	0	0	1	0	0	0	40	-
x11	0	0	**1**	0	0	0	0	0	0	0	1	0	0	25	**25**
x12	0	0	0	1	0	0	0	0	0	0	0	1	0	50	-
x13	0	0	0	0	0	1	0	0	0	0	0	0	1	50	-
Obj.	-5	-10	**-20**	-8	-12	-6	0	0	0	0	0	0	0	f+0	-

(2) 선택된 열의 계수가 양수이며
최소의 비율
을 갖는 행을 선택

(1) 목적 함수 계수가 최소인 열을 선택

(3) 선택된 행과 열의 변수를 중심으로 Pivot을 실시

2

	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13	bi	bi/ai
x7	1	1	0	0	0	0	1	0	0	0	-1	0	0	25	-
x8	0	1	0	1	**1**	0	0	1	0	0	-1	0	0	25	**25**
x9	0	0	0	0	1	1	0	0	1	0	-1	0	0	25	25
x10	0	1	0	0	0	0	0	0	0	1	0	0	0	40	-
x3	0	0	1	0	0	0	0	0	0	0	1	0	0	25	-
x12	0	0	0	1	0	0	0	0	0	0	0	1	0	50	-
x13	0	0	0	0	0	1	0	0	0	0	0	0	1	50	-
Obj.	-5	-10	0	-8	**-12**	-6	0	0	0	0	20	0	0	f+500	-

3.4 선형 계획 문제의 예 1
- 풀이 과정(6)

3

	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13	bi	bi/ai
x7	1	1	0	0	0	0	1	0	0	0	-1	0	0	25	-
x5	0	1	0	1	1	0	0	1	0	0	-1	0	0	25	-
x9	0	-1	0	-1	0	**1**	0	-1	1	0	0	0	0	0	**0**
x10	0	1	0	0	0	0	0	0	0	1	0	0	0	40	-
x3	0	0	1	0	0	0	0	0	0	0	1	0	0	25	-
x12	0	0	0	1	0	0	0	0	0	0	0	1	0	50	-
x13	0	0	0	0	0	1	0	0	0	0	0	0	1	50	50
Obj.	-5	2	0	4	0	**-6**	0	12	0	0	8	0	0	f+800	-

4

	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13	bi	bi/ai
x7	**1**	1	0	0	0	0	1	0	0	0	-1	0	0	25	**25**
x5	0	1	0	1	1	0	0	1	0	0	-1	0	0	25	-
x6	0	-1	0	-1	0	1	0	-1	1	0	0	0	0	0	-
x10	0	1	0	0	0	0	0	0	0	1	0	0	0	40	-
x3	0	0	1	0	0	0	0	0	0	0	1	0	0	25	-
x12	0	0	0	1	0	0	0	0	0	0	0	1	0	50	-
x13	0	1	0	1	0	0	0	1	-1	0	0	0	1	50	-
Obj.	**-5**	-4	0	-2	0	0	0	6	6	0	8	0	0	800	-

3.4 선형 계획 문제의 예 1
- 풀이 과정(7)

5

	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13	bi	bi/ai
x1	1	1	0	0	0	0	1	0	0	0	-1	0	0	25	
x5	0	1	0	**1**	1	0	0	1	0	0	-1	0	0	25	**25**
x6	0	-1	0	-1	0	1	0	-1	1	0	0	0	0	0	
x10	0	1	0	0	0	0	0	0	0	1	0	0	0	40	
x3	0	0	1	0	0	0	0	0	0	0	1	0	0	25	
x12	0	0	0	1	0	0	0	0	0	0	0	1	0	50	50
x13	0	1	0	1	0	0	0	1	-1	0	0	0	1	50	50
Obj.	0	1	0	**-2**	0	0	5	6	6	0	3	0	0	f+925	

선택된 열의 계수가 음수(-1)여서 선택 안됨

6

	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13	bi	bi/ai
x1	1	1	0	0	0	0	1	0	0	0	-1	0	0	25	
x4	0	1	0	1	1	0	0	1	0	0	-1	0	0	25	
x6	0	0	0	0	1	1	0	0	1	0	-1	0	0	25	
x10	0	1	0	0	0	0	0	0	0	1	0	0	0	40	
x3	0	0	1	0	0	0	0	0	0	0	1	0	0	25	
x12	0	-1	0	0	-1	0	0	-1	0	0	1	1	0	25	
x13	0	0	0	0	-1	0	0	0	-1	0	1	0	1	25	
Obj.	0	3	0	0	2	0	5	8	6	0	1	0	0	f+975	

목적 함수의 모든 계수가 0 **이상이므로**
현재의 해가 최적해임($x_2=x_5=0, x_1=x_3=x_4=x_6=25, f=-975$)

따라서 화물 1, 3, 4, 6**을** 25,000ton**씩 적재하였을 때 최대의 운임** 975,000$**를 얻을 수 있음**

3.4 선형 계획 문제의 예 2

- 다음과 같이 등호 제약 조건만으로 이루어진 부정 선형 연립 방정식의 해를 구하는 문제(선형 계획 문제)를 Simplex 방법을 이용하여 푸시오.

$$
\begin{aligned}
&2x_1 + y - z - \zeta_1 = 3 \\
&2x_2 + y - z - \zeta_2 = 3 \\
&x_1 + x_2 = 2 \\
&where, x_1, x_2 y, z, \zeta_1, \zeta_2 \ge 0
\end{aligned}
$$

최적해: $x_1 = x_2 = 1, y = 1, z = 0, \zeta_1 = \zeta_2 = 0$

3.4 선형 계획 문제의 예 2 - 풀이 과정(1)

1. 주어진 문제는 등호 제약 조건만으로 이루어진 부정 선형 연립 방정식의 해를 구하는 문제(선형 계획 문제)임
2. 부정 선형 연립 방정식의 해를 구하기 위하여 Simplex 방법에서 인위 변수 및 인위 목적 함수를 도입하여 초기 기저 가능해를 구하는 방법임

$$\mathbf{B}_{(3\times6)}\mathbf{X}_{(6\times1)} + \underbrace{\mathbf{Y}_{(3\times1)}}_{\text{인위 변수}} = \mathbf{D}_{(3\times1)}$$

3. 인위 목적 함수는 다음과 같이 정의함

$$w = \sum_{i=1}^{3} Y_i = \sum_{i=1}^{3} D_i - \sum_{j=1}^{6}\sum_{i=1}^{3} B_{ij}X_j = w_0 + \sum_{j=1}^{6} C_j X_j$$

여기서, $C_j = -\sum_{i=1}^{3} B_{ij}$: 행렬 B의 j번째 열의 요소를 모두 더해 부호를 바꾼 것(상대 비용 계수)

$$w_0 = \sum_{i=1}^{3} D_i = 3+3+2 = 8$$

: 인위 목적 함수의 초기값으로 행렬 D의 모든 요소를 더한 것

3.4 선형 계획 문제의 예 2
- 풀이 과정(2)

$$2x_1 + y - z - \zeta_1 = 3$$
$$2x_2 + y - z - \zeta_2 = 3$$
$$x_1 + x_2 = 2$$
$$where, x_1, x_2 y, z, \zeta_1, \zeta_2 \geq 0$$

$$\mathbf{B}_{(3\times6)}\mathbf{X}_{(6\times1)} + \underbrace{\mathbf{Y}_{(3\times1)}}_{\text{인위 변수}} = \mathbf{D}_{(3\times1)}$$

$$\begin{bmatrix} 2 & 0 & 1 & -1 & -1 & 0 \\ 0 & 2 & 1 & -1 & 0 & -1 \\ 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1(=X_1) \\ x_2(=X_2) \\ y(=X_3) \\ z(=X_4) \\ \zeta_1(=X_5) \\ \zeta_2(=X_6) \end{bmatrix} + \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \end{bmatrix} = \begin{bmatrix} 3 \\ 3 \\ 2 \end{bmatrix}$$

1

	X1	X2	X3	X4	X5	X6	Y1	Y2	Y3	bi	bi/ai
Y1	**2**	0	1	-1	-1	0	1	0	0	3	**3/2**
Y2	0	2	1	-1	0	-1	0	1	0	3	-
Y3	1	1	0	0	0	0	0	0	1	2	2
A. Obj.	**-3**	-3	-2	2	1	1	0	0	0	w-8	-

인위 목적 함수 식 **각 열의 요소를 모두 더해 부호를 바꾼 것(예**, 1**열**: -(2+0+1)=-3)

3.4 선형 계획 문제의 예 2
- 풀이 과정(3)

2

	X1	X2	X3	X4	X5	X6	Y1	Y2	Y3	bi	bi/ai
X1	1	0	1/2	-1/2	-1/2	0	1/2	0	0	3/2	-
Y2	0	2	1	-1	0	-1	0	1	0	3	3/2
Y3	0	**1**	-1/2	1/2	1/2	0	-1/2	0	1	1/2	**1/2**
A. Obj.	0	**-3**	-1/2	1/2	-1/2	1	3/2	0	0	w-7/2	-

3

	X1	X2	X3	X4	X5	X6	Y1	Y2	Y3	bi	bi/ai
X1	1	0	1/2	-1/2	-1/2	0	1/2	0	0	3/2	3
Y2	0	0	**2**	-2	-1	-1	1	1	-2	2	**1**
X2	0	1	-1/2	1/2	1/2	0	-1/2	0	1	1/2	-
A. Obj.	0	0	**-2**	2	1	1	0	0	3	w-2	-

4

	X1	X2	X3	X4	X5	X6	Y1	Y2	Y3	bi	bi/ai
X1	1	0	0	0	-1/4	1/4	1/4	-1/4	1/2	1	
X3	0	0	1	-1	-1/2	-1/2	1/2	1/2	-1	1	
X2	0	1	0	0	1/4	-1/4	-1/4	1/4	1/2	1	-
A. Obj.	0	0	0	0	0	0	1	1	1	**w-0**	-

인위 목적 함수가 0이므로 초기 기저 가능해가 구해졌음

$$\mathbf{X}^{T}{}_{(1\times5)} = [x_1 \quad x_2 \quad y \quad z \quad \zeta_1 \quad \zeta_2]$$

이로부터 $X_1=1, X_2=1, X_3=1, X_4=X_5=X_6=0$

따라서 주어진 문제의 초기 기저 가능해 중의 하나는 $x_1 = x_2 = 1, v = y - z = 1, \zeta_1 = \zeta_2 = 0$ **임**

선형 계획 문제의 해결을 위한 Simplex Class의 구현 예

```
class Simplex
{
public:
	Simplex();
	virtual ~Simplex();

	int m_nRowNo;                 // Simplex Table의 행의 수(= 제약 조건식의 수)
	int m_nColumnNo;              // Simplex Table의 열의 수(= 원래의 설계 변수의 수 + 완화 변수의 수 + 잉여 변수의 수
	+ 인위 변수의 수)
	int m_nType;                  // Phase I, II 여부

	int* m_pBDVId;                // 기저 변수들의 ID
	double** m_pA;                // 제약 조건식에 해당하는 계수들의 집합
	double* m_pB;                 // 제약 조건식의 값들의 집합
	double* m_pC;                 // 목적 함수식에 해당하는 계수들의 집합
	double* m_pW;                 // 인위 목적 함수식의 설계 변수에 대한 계수들의 집합
	double m_fObj;                // 목적 함수값
	double m_fAObj;               // 인위 목적 함수값

	void InitializeSimplexTable();
	int FindPivotColumn();        // Pivot Column을 결정하는 함수
	int FindPivotRow();           // Pivot Row를 결정하는 함수
	int Pivot(int colNo, int rowNo); // 주어진 Pivot Column과 Pivot Row로 Simplex Table의 Pivoting을 1번 수행하는 함수
	int CheckEndCondition();      // Simplex 방법의 종료 조건을 판단(목적 함수식 또는 인위 목적 함수식에 해당하는
	계수들이 모두 음이 아닌지 확인)하는 함수
	int Solve();                  // Simplex 방법을 실행하는 함수
};
```

Ch4. 쿤-터커(Kuhn-Tucker) 정리를 이용한 최적성 조건

4.1 최적성 조건을 이용한 최적해의 도출

4.2 라그란지 승수(Lagrange multiplier)를 도입한 최적화 기법

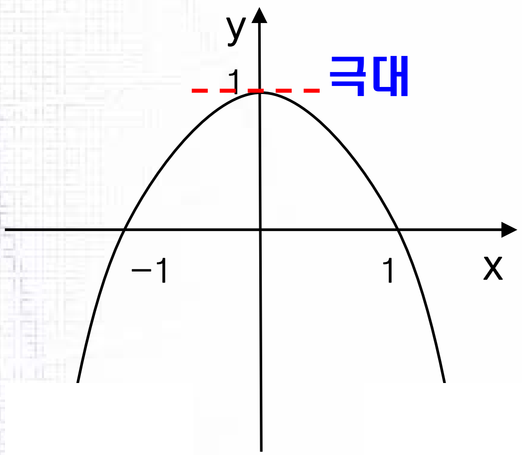
Advanced
Ship
Design
Automation
Laboratory

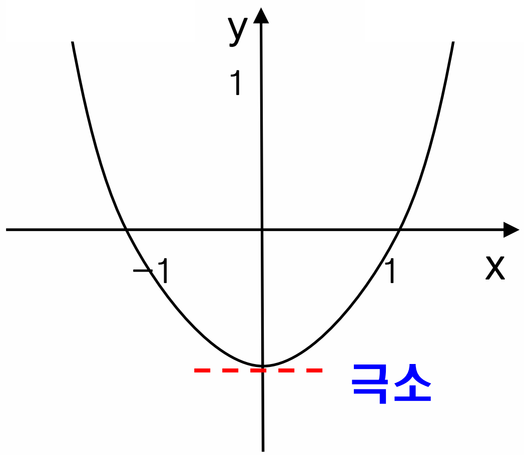
4.1 최적성 조건을 이용한 최적해의 도출

Advanced
Ship
Design
Automation
Laboratory

1변수 함수의 최적성 조건
- 함수의 극대, 극소(고교 과정 Review)

- 고등학교 수학의 정석(수학 II) Review
  - 수학의 정석 "6. 극대, 극소와 미분"(p. 104)





1) 극대값 : $x = x^*$에서 연속 함수 $f(x)$가 증가에서 감소 상태로 변함
2) 극소값 : $x = x^*$에서 연속 함수 $f(x)$가 감소에서 증가 상태로 변함

$$f'(x^*) = 0$$

($x = x^*$에서 극대 또는 극소값을 가질 필요 조건)

1변수 함수의 최적성 조건 - 1계 필요 조건(1)

▪ 변수가 하나일 때, 극값을 가질 필요 조건 : $f'(x^*) = 0$

pf) 주어진 점 x^*에서 $f(x)$의 테일러 급수는 다음과 같다.

$$\boxed{f(x) = f(x^*) + \frac{df(x^*)}{dx}(x - x^*) + \frac{1}{2}\frac{d^2 f(x^*)}{dx^2}(x - x^*)^2 + R}$$

나머지항(Remainder)
: x가 x^*에 충분히 가까우면 그 값이 매우 작음

$x - x^* = d$ 라고 놓으면

$$f(x) = f(x^*) + f'(x^*)d + \frac{1}{2}f''(x^*)d^2 + R$$

함수 값의 변화량 $f(x) - f(x^*) = \Delta f(x)$

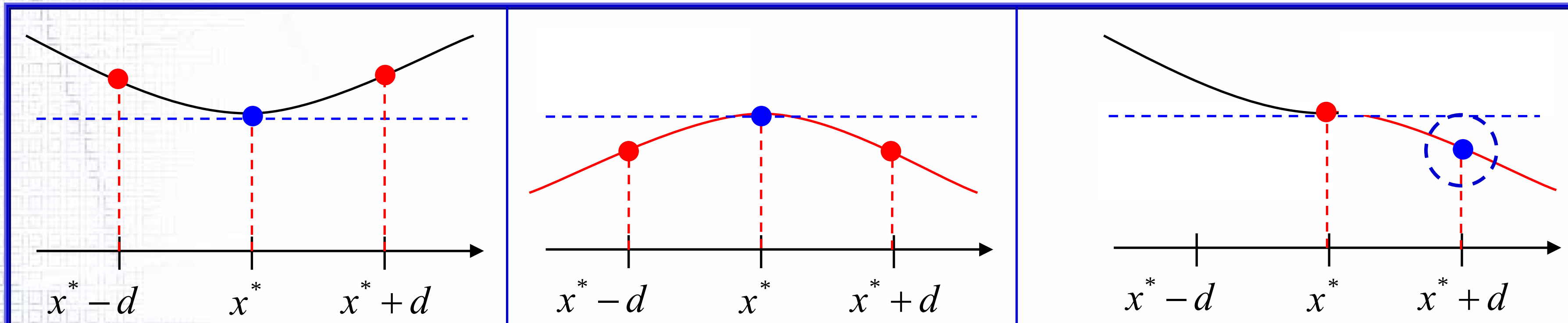
$$\Delta f(x) = f'(x^*)d + \frac{1}{2}f''(x^*)d^2 + R$$

1변수 함수의 최적성 조건 - 1계 필요 조건(2)

$$\Delta f(x) = f(x) - f(x^*) = f'(x^*)d + \frac{1}{2}f''(x^*)d^2 + R$$ ➡

의미 : x^*에서의 함수 값이 $x^* \pm d$에서의 함수 값보다 작으면 x^*는 국부적 후보 최소점이 될 가능성이 있음

$x = x^*$ 에서 국부적 후보 최소점이기 위해서는 $\Delta f \geq 0$



$f(x^*)$ 에서 최소값

$f(x^*)$ 에서 최대값

$f(x^*)$ 보다 함수 값이 작은 점(x^*+d)이 존재하므로 x^*는 최소점이 아님

따라서, $d(= x - x^*)$ 의 부호에 상관없이 $\Delta f \geq 0$ 를 만족하려면 $f'(x^*) = 0$ 이어야 함

이와 유사한 개념으로 $x = x^*$ 에서 국부적 후보 최대점이 되기 위해서는 $\Delta f \leq 0$ 이어야 하며, $d(= x - x^*)$의 부호에 상관없이 $\Delta f \leq 0$를 만족하려면 $f'(x^*) = 0$ 이어야 함

cf) $f'(x^*) = 0$ 를 만족하는 점 : 국부적으로 최소, 최대, 또는 변곡점

총칭하여 상점(Stationary point)이라고 함

1변수 함수의 최적성 조건
- 충분 조건과 2계 필요 조건

$$\Delta f(x) = f(x) - f(x^*) = f'(x^*)d + \frac{1}{2}f''(x^*)d^2 + R$$

$x = x^*$ 에서 국부적 후보 최소점이기 위해서는 $\Delta f \geq 0$

따라서, $d(= x - x^*)$ 의 부호에 상관없이 $\Delta f \geq 0$ 를 만족하려면 $f'(x^*) = 0$ 이어야 함

- 상점($f'(x^*) = 0$를 만족하는 점) 중 어느 점이 최소점인지 결정하는 방법

상점이므로, $f'(x^*) = 0$. 따라서, $\Delta f(x) = \frac{1}{2}f''(x^*)d^2 + R$

2차항이 다른 모든 고차항에 비해 지배적인 항이므로,

$d \neq 0$에 대하여 다음 조건을 만족하면 항상 $\Delta f \geq 0$

$f''(x^*) > 0$ (충분 조건(Sufficient condition))

cf) $f''(x^*) = 0$ 인 경우 국부적으로 최소점인지 알 수 없다.

∵ 1계, 2계 미분이 모두 0이므로, 국부적으로 최소이기 위해서는
3계 미분계수는 0이 되어야 하고, 4계 미분계수가 양수(>0)여야 한다.
즉, R의 부호에 따라 달라지게 된다.

- 2계 필요 조건 :

국부적으로 최소값을 가지면, $f''(x^*) \geq 0$

[※ 단, 역은 성립하지 않는다.]

1변수 함수의 최적성 조건(요약)

- 충분 조건, 필요 조건, 필요 충분 조건

  (고등학교 수학의 정석 '명제와 조건' 참고)

1) A는 B의 충분 조건이다.

   "A이면 B이다." ($A \Rightarrow B$)

2) C는 D의 필요 조건이다.

   "D이면 C이다." ($C \Leftarrow D$)

3) E는 F의 필요 충분 조건이다.

   "E이면 F이고, F이면 E이다." ($E \Leftrightarrow F$)

- 1계 필요 조건(1st Order necessary condition)

  x^*가 국부적 후보 최소점이면, $f'(x^*) = 0$

cf) $f'(x^*) = 0$ 이면, x^*가 상점(극소, 극대, 변곡점)이다.

- 충분 조건(Sufficient condition)

  상점($f'(x^*) = 0$)일 경우

  $f''(x^*) > 0$ 이면 x^*가 국부적 최소점이다.

- 2계 필요 조건(2nd Order necessary condition)

  x^*가 국부적 최소점이면, $f''(x^*) \geq 0$

2변수 함수의 테일러 전개(Taylor Series Expansion)(1)

- 2변수 함수 $f(x_1, x_2)$에 대한 점 (x_1^*, x_2^*)에서의 테일러 전개식

$$\boxed{\begin{aligned} f(x_1,x_2) &= f(x_1^*,x_2^*) + \frac{\partial f}{\partial x_1}(x_1 - x_1^*) + \frac{\partial f}{\partial x_2}(x_2 - x_2^*) \\ &+ \frac{1}{2}\left(\frac{\partial^2 f}{\partial x_1^2}(x_1 - x_1^*)^2 + 2\frac{\partial^2 f}{\partial x_1 \partial x_2}(x_1 - x_1^*)(x_2 - x_2^*) + \frac{\partial^2 f}{\partial x_2^2}(x_2 - x_2^*)^2 \right) + R \end{aligned}}$$

⬇ 이를 다시 표현하면,

$$\boxed{f(\mathbf{x}) = f(\mathbf{x}^*) + \nabla f(\mathbf{x}^*)^T(\mathbf{x} - \mathbf{x}^*) + \frac{1}{2}\left(\mathbf{x} - \mathbf{x}^*\right)^T \mathbf{H}(\mathbf{x}^*)\left(\mathbf{x} - \mathbf{x}^*\right) + R}$$

$$\left(\mathbf{x} = (x_1, x_2)^T \ , \ \mathbf{x}^* = (x_1^*, x_2^*)^T \ , \ \mathbf{H} \in M_{2\times 2} \right)$$

2x2 Matrix의 원소

$$\nabla f(\mathbf{x}^*)^T(\mathbf{x} - \mathbf{x}^*) = \begin{bmatrix} \dfrac{\partial f}{\partial x_1} \\ \dfrac{\partial f}{\partial x_2} \end{bmatrix}^T \begin{bmatrix} x_1 - x_1^* \\ x_2 - x_2^* \end{bmatrix}$$

$$\frac{1}{2}\left(\mathbf{x} - \mathbf{x}^*\right)^T \mathbf{H}(\mathbf{x}^*)\left(\mathbf{x} - \mathbf{x}^*\right) = \frac{1}{2}\left[\frac{\partial^2 f}{\partial x_1^2}(x_1 - x_1^*) + \frac{\partial^2 f}{\partial x_2 \partial x_1}(x_2 - x_2^*) \quad \frac{\partial^2 f}{\partial x_1 \partial x_2}(x_1 - x_1^*) + \frac{\partial^2 f}{\partial x_2^2}(x_2 - x_2^*) \right] \begin{bmatrix} x_1 - x_1^* \\ x_2 - x_2^* \end{bmatrix}$$

$$= \frac{1}{2}\begin{bmatrix} x_1 - x_1^* & x_2 - x_2^* \end{bmatrix} \begin{bmatrix} \dfrac{\partial^2 f}{\partial x_1^2} & \dfrac{\partial^2 f}{\partial x_1 \partial x_2} \\ \dfrac{\partial^2 f}{\partial x_2 \partial x_1} & \dfrac{\partial^2 f}{\partial x_2^2} \end{bmatrix} \begin{bmatrix} x_1 - x_1^* \\ x_2 - x_2^* \end{bmatrix}$$

2변수 함수의 테일러 전개(2)

- 2변수 테일러 전개식의 행렬 표기법

$$f(\mathbf{x}) = f(\mathbf{x}^*) + \nabla f(\mathbf{x}^*)^T (\mathbf{x}-\mathbf{x}^*) + \frac{1}{2}\left(\mathbf{x}-\mathbf{x}^*\right)^T \mathbf{H}(\mathbf{x}^*)\left(\mathbf{x}-\mathbf{x}^*\right) + R$$

2x2 Matrix의 원소

$$\left(\mathbf{x} = (x_1, x_2)^T ,\ \mathbf{x}^* = (x_1^*, x_2^*)^T ,\ \boxed{\mathbf{H} \in M_{2\times 2}} \right)$$

- 다변수 함수의 테일러 전개식의 행렬 표기법 : 2변수 전개식과 동일

$\mathbf{x},\ \mathbf{x}^*, \nabla f$: n 차원 Vector

$\mathbf{H} \in M_{n\times n}$

- $\mathbf{x} - \mathbf{x}^* = \mathbf{d}$ 로 정의 하면, 다음과 같이 표현 가능

$$f(\mathbf{x}^* + \mathbf{d}) = f(\mathbf{x}^*) + \boxed{\nabla f(\mathbf{x}^*)^T} \mathbf{d} + \boxed{\frac{1}{2}\mathbf{d}^T \mathbf{H}(\mathbf{x}^*)\mathbf{d}} + R$$

$$\boxed{\nabla f(\mathbf{x}^*)^T = 0,\ \frac{1}{2}\mathbf{d}^T \mathbf{H}(\mathbf{x}^*)\mathbf{d} > 0} \Rightarrow \mathbf{x} = \mathbf{x}^*$$에서 최소이기 위한 충분 조건

[참고] 헷세 행렬(Hessian Matrix)

- **헷세 행렬**(Hessian Matrix) : Gradient Vector**를 한번 더 미분하면**(**각각**, x_i **에 대하여 미분**), **함수** $f(x)$ **에 대한** 2**계 편도함수의 행렬을 얻게 되는데**, **이를 헷세 행렬이라고 정의함**. **즉**, Gradient Vector**의 각 성분을** $x_1, x_2, \cdots, x_n$ **으로 미분하면 다음을 얻는다**.

$$\frac{\partial^2 f}{\partial \mathbf{x} \partial \mathbf{x}} = \begin{bmatrix} \dfrac{\partial^2 f}{\partial x_1^2} & \dfrac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \dfrac{\partial^2 f}{\partial x_1 \partial x_n} \\ \dfrac{\partial^2 f}{\partial x_2 \partial x_1} & \dfrac{\partial^2 f}{\partial x_2^2} & \cdots & \dfrac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \dfrac{\partial^2 f}{\partial x_n \partial x_1} & \dfrac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \dfrac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$$

$$\mathbf{x} = \begin{pmatrix} x_1 & x_2 & \cdots & x_n \end{pmatrix}^T$$

: n**개의 변수를 가지는** column Vector(**열벡터**)

- **헷세 행렬은 보통** H **또는** $\nabla^2 f$ **로 표시**

$$\mathbf{H} = \left[\frac{\partial^2 f}{\partial x_i \partial x_j} \right] \qquad (i = 1, 2, \cdots, n \; ; \; j = 1, 2, \cdots, n)$$

- **헷세 행렬의 대칭성**

$$\frac{\partial f}{\partial x_i \partial x_j} = \frac{\partial f}{\partial x_j \partial x_i}$$

따라서, **헷세 행렬은 항상 대칭 행렬**

2차 형식(Quadratic Form)

- 2차 형식 : 한 변수의 제곱항과 서로 다른 두 변수의 곱항(다변수 곱항)의 합으로 이루어진 함수

ex) $F(x_1, x_2, x_3) = \frac{1}{2}\left(2x_1^2 + 2x_1x_2 + 4x_1x_3 - 6x_2^2 - 4x_2x_3 + 5x_3^2\right)$

2차 형식은 다음과 같이 행렬로 표현이 가능하다.

A : 대칭 행렬
(Symmetric Matrix)

$$F(x_1, x_2, x_3) = \frac{1}{2}\begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}\begin{bmatrix} 2 & 1 & 2 \\ 1 & -6 & -2 \\ 2 & -2 & 5 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \frac{1}{2}\mathbf{x}^T\boxed{\mathbf{A}}\mathbf{x} \Leftrightarrow \frac{1}{2}\mathbf{d}^T\mathbf{H}\mathbf{d}$$

- 대칭행렬 A의 구성 방법 (A의 (i,j) 성분을 a_{ij}로 나타냄)

1) 대각 성분은 제곱항의 계수 $a_{ii} = \left(x_i^2 \text{의 계수}\right)$

2) 대각 성분 이외의 성분은 다변수 곱항 계수의 ½ $a_{ij} = \left(x_i x_j \text{의 계수}\right) \times \frac{1}{2}$

2차 형식의 형태

- 2차 형식의 정의

1) 양정 형식(Positive Definite)

: $\mathbf{x} \neq 0$ 인 모든 x에 대하여 $\mathbf{x}^T\mathbf{A}\mathbf{x} > 0$

2) 양반정 형식(Positive Semidefinite)

: 모든 x에 대하여 $\mathbf{x}^T\mathbf{A}\mathbf{x} \geq 0$이고

$\mathbf{x}^T\mathbf{A}\mathbf{x} = 0$ 인 $\mathbf{x} \neq 0$ 가 존재

3) 음정 형식(Negative Definite)

: $\mathbf{x} \neq 0$ 인 모든 x에 대하여 $\mathbf{x}^T\mathbf{A}\mathbf{x} < 0$

4) 음반정 형식(Negative Semidefinite)

: 모든 x에 대하여 $\mathbf{x}^T\mathbf{A}\mathbf{x} \leq 0$

5) 부정 형식(Indefinite)

: 어떤 x에 대해서는 양수 또는 음수

- 2차 형식 정의의 이용

① 1변수 함수의 최소 조건

$f'(x^*) = 0$ 인 x^*(상점)에서

$f''(x^*) > 0$ 이면

x^*는 국부적 최소점이다.

② 다변수 함수의 최소 조건

$\nabla f(\mathbf{x}^*)^T = 0 \Rightarrow \nabla f(\mathbf{x}^*) = 0$ 인

$\mathbf{x}^*$(상점)에서 $\mathbf{d}^T\mathbf{H}(\mathbf{x}^*)\mathbf{d} > 0$이면

즉, $\mathbf{d}^T\mathbf{H}(\mathbf{x}^*)\mathbf{d} > 0$ 가 양정 형식이면

$\mathbf{x}^*$는 국부적 최소점이다.

여기서, $\mathbf{H}(\mathbf{x}^*)$가 양정 행렬이면

$\mathbf{d}^T\mathbf{H}(\mathbf{x}^*)\mathbf{d} > 0$

2차 형식 행렬의 형태 결정을 위한 고유치 검사

2차 형식 $F(\mathbf{x}) = \dfrac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x}$ 에 관련된 대칭 $n \times n$ 행렬 $\mathbf{A}$ 의 n개의 고유치를 $\lambda_i, i = 1,\ldots,n$ 이라고 가정

1) $F(\mathbf{x})$ 가 양정(Positive Definite)이기 위한 필요 충분 조건: 모든 고유치가 양수

$$\lambda_i > 0, i = 1,\ldots,n$$

2) $F(\mathbf{x})$ 가 양반정(Positive Semidefinite)이기 위한 필요 충분 조건: 모든 고유치가 음수가 아님

$$\lambda_i \geq 0, i = 1,\ldots,n$$

3) $F(\mathbf{x})$ 가 음정(Negative Definite)이기 위한 필요 충분 조건: 모든 고유치가 음수

$$\lambda_i < 0, i = 1,\ldots,n$$

4) $F(\mathbf{x})$ 가 음반정(Negative Semidefinite)이기 위한 필요 충분 조건: 모든 고유치가 양수가 아님

$$\lambda_i \leq 0, i = 1,\ldots,n$$

5) 적어도 하나의 λ_i 가 음수이고 또 적어도 하나의 λ_i 가 양수이면 $F(\mathbf{x})$ 는 부정(Indefinite)임

고유치 검사에 의한 2차 형식의 행렬의 형태 결정 예제

고유치 구하는 방법

$$Av = \lambda v \quad \Rightarrow \quad (A - \lambda I)v = 0 \quad \Rightarrow \quad \det(A - \lambda I) = 0$$

다음 행렬의 고유치를 구하여 행렬의 형태를 결정하시오.

$$\mathbf{A} = \begin{bmatrix} 4 & 2 & 2 \\ 2 & 4 & 2 \\ 2 & 2 & 4 \end{bmatrix}$$

$$\det \begin{bmatrix} 4-\lambda & 2 & 2 \\ 2 & 4-\lambda & 2 \\ 2 & 2 & 4-\lambda \end{bmatrix} = (2-\lambda)^2(8-\lambda) = 0$$

$$\therefore \lambda = 2(\text{중근}), 8$$

모든 고유치가 양수이므로 주어진 행렬은 양정(Positive Definite) **행렬임**

다변수 함수의 후보 최적성 조건(요약)

- n개의 변수로 이루어진 다변수 함수 $f(\mathbf{x})$ 에 대한 테일러 전개식

$$f(\mathbf{x}) = f(\mathbf{x}^*) + \nabla f(\mathbf{x}^*)^T \mathbf{d} + \frac{1}{2}\mathbf{d}^T \mathbf{H}(\mathbf{x}^*)\mathbf{d} + R$$

- 함수의 변화량으로 위 식을 다시 쓰면,

$$\Delta f = \nabla f(\mathbf{x}^*)^T \mathbf{d} + \frac{1}{2}\mathbf{d}^T \mathbf{H}(\mathbf{x}^*)\mathbf{d} + R$$

- $\mathbf{x}^*$가 국부적으로 후보 최소점이면, $\Delta f \geq 0$이어야 함

1) x*가 국부적 후보 최소점일 필요 조건 :

➡ $\nabla f(\mathbf{x}^*)^T = 0$ 즉, $\dfrac{\partial f(\mathbf{x}^*)}{\partial x_i} = 0$, $(i = 1,2,\Lambda\ n)$

2) $\nabla f(\mathbf{x}^*)^T = 0 \Rightarrow \nabla f(\mathbf{x}^*) = 0$인 상점에서 x*가 국부적 최소점일 조건 :

➡ $\mathbf{d}^T \mathbf{H}(\mathbf{x}^*)\mathbf{d} > 0$ ➡ 위 조건은 $\mathbf{H}(\mathbf{x}^*)$ 가 양정 행렬일 경우 성립한다.

후보 최적성 필요 조건을 이용한 등호 제약 최적화 문제의 해법

Optimum Problem

Minimize $f(x_1, x_2) = (x_1 - 1.5)^2 + (x_2 - 1.5)^2$

Subject to $h(x_1, x_2) = x_1 + x_2 - 2 = 0$

Solution

x_2**를** x_1**의 양함수**(explicit function)**로 표현하면**,

$$x_2 = \Phi(x_1) = -x_1 + 2$$

$$f(x_1, \Phi(x_1)) = (x_1 - 1.5)^2 + (-x_1 + 2 - 1.5)^2$$

$$\frac{df}{dx_1} = 2(x_1 - 1.5) - 2(-x_1 + 0.5) = 0$$

$$\Rightarrow x_1 = 1$$

$$\Rightarrow x_2 = -x_1 + 2 = 1$$

$$\frac{d^2 f}{dx_1^{\ 2}} = 4 > 0$$

$\therefore (x_1,\ x_2) = (1,\ 1)$: **국부적 최소점**

4.2 라그란지 승수(Lagrange multiplier)를 도입한 최적화 기법

Advanced
Ship
Design
Automation
Laboratory

등호 제약 조건이 있는 문제의 후보 최적성 필요 조건
- Lagrange multiplier의 도입

Minimize $f(x_1, x_2)$, *Subject to* $h(x_1, x_2) = 0$

$h(x_1,\ x_2) = 0$ 으로부터 x_2를 x_1으로 표현할 수 있다. 즉, $f(x_1, x_2) = f(x_1, \phi(x_1))$

1변수 함수의 국부적 후보 최소점을 구하기 위해서는

$df(x_1, x_2)/dx_1 = 0$, 그런데 $df(x_1,\ x_2) = \dfrac{\partial f(x_1,\ x_2)}{\partial x_1}dx_1 + \dfrac{\partial f(x_1,\ x_2)}{\partial x_2}dx_2$이므로 $\dfrac{\partial f(x_1,\ x_2)}{\partial x_1} + \dfrac{\partial f(x_1,\ x_2)}{\partial x_2}\dfrac{dx_2}{dx_1} = 0$

$\mathbf{x}^* = (x_1^*,\ x_2^*)$ 를 국부적 후보 최소점이라 가정하면,

$$\frac{\partial f(x_1^*,\ x_2^*)}{\partial x_1} + \frac{\partial f(x_1^*,\ x_2^*)}{\partial x_2}\frac{d\phi(x_1)}{dx_1} = 0 \quad \cdots \text{ 식 (1)}$$

$x_2 = \phi(x_1)$ 는 양함수 형태이나 일반적으로 이렇게 표현이 안됨

등호 제약 조건으로부터 $\boxed{h(x_1^*,\ x_2^*) = 0}$

$$\Rightarrow \frac{dh(x_1^*, x_2^*)}{dx_1} = \frac{\partial h(x_1^*, x_2^*)}{\partial x_1} + \frac{\partial h(x_1^*, x_2^*)}{\partial x_2}\frac{d\phi(x_1)}{dx_1} = 0$$

$$\therefore \frac{d\phi(x_1)}{dx_1} = -\frac{\partial h(x_1^*,\ x_2^*)/\partial x_1}{\partial h(x_1^*,\ x_2^*)/\partial x_2} \quad \cdots \text{ 식 (2)}$$

식 (2)를 식 (1)에 대입하면

$$\therefore \frac{\partial f(x_1^*,\ x_2^*)}{\partial x_1} - \boxed{\frac{\partial f(x_1^*,\ x_2^*)}{\partial x_2}\frac{\partial h(x_1^*,\ x_2^*)/\partial x_1}{\partial h(x_1^*,\ x_2^*)/\partial x_2}} = 0 \cdots \text{ 식 (3)}$$

만일 $v^* = -\dfrac{\partial f(x_1^*,\ x_2^*)/\partial x_2}{\partial h(x_1^*,\ x_2^*)/\partial x_2}$ … 식 (4) 라고 가정하면

식 (3)은 $\boxed{\dfrac{\partial f(x_1^*,\ x_2^*)}{\partial x_1} + v^*\dfrac{\partial h(x_1^*,\ x_2^*)}{\partial x_1} = 0}$

한편, 식 (4)를 다시 정리하면

$$\boxed{\frac{\partial f(x_1^*,\ x_2^*)}{\partial x_2} + v^*\frac{\partial h(x_1^*,\ x_2^*)}{\partial x_2} = 0}$$

요약하면, $\mathbf{x}^* = (x_1^*,\ x_2^*)$가 국부적 후보 최소점이 되기 위해서는 아래의 3가지 조건을 모두 만족해야 함

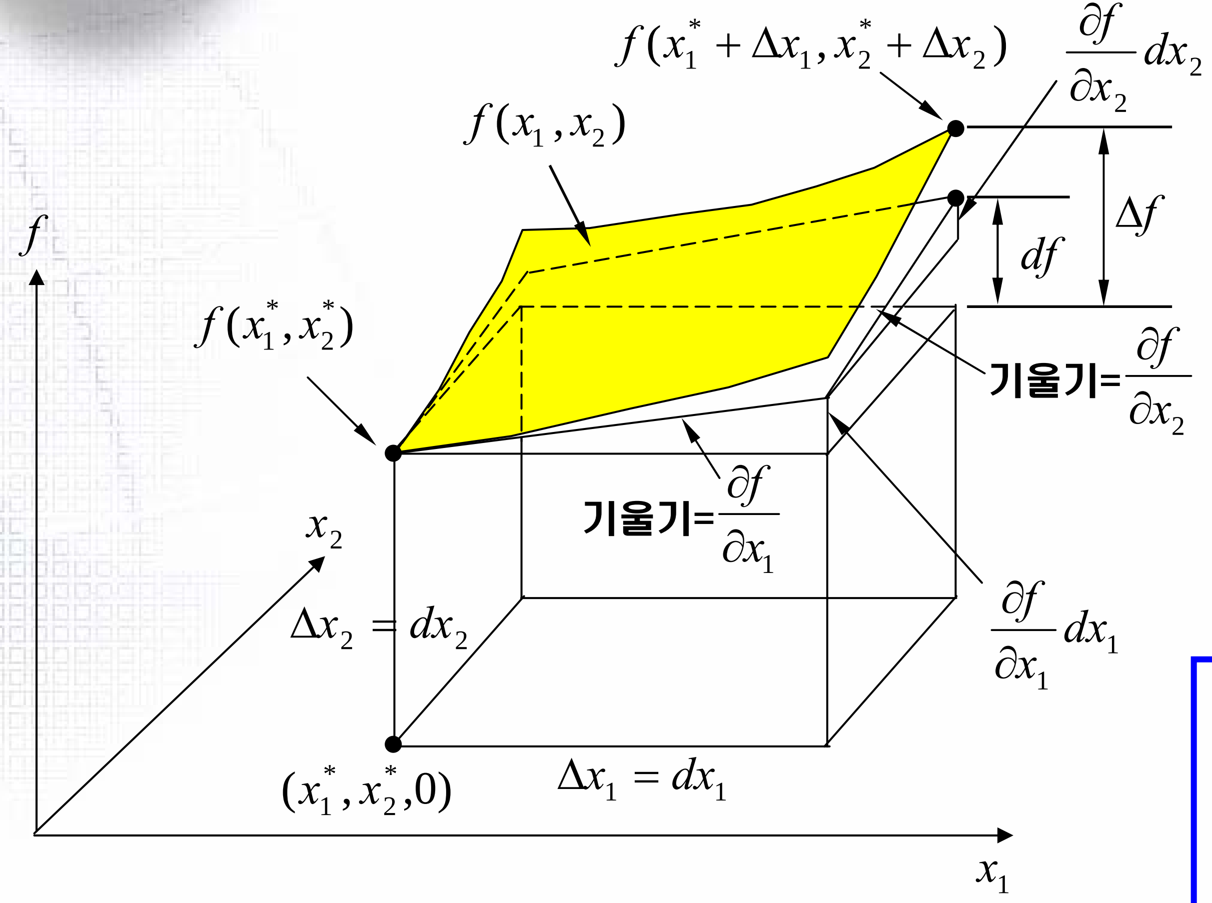
$$\boxed{h(x_1^*,\ x_2^*) = 0}$$

$$\boxed{\frac{\partial f(x_1^*,\ x_2^*)}{\partial x_1} + v^*\frac{\partial h(x_1^*,\ x_2^*)}{\partial x_1} = 0}$$

$$\boxed{\frac{\partial f(x_1^*,\ x_2^*)}{\partial x_2} + v^*\frac{\partial h(x_1^*,\ x_2^*)}{\partial x_2} = 0}$$

위 식에서 v^*를 Lagrange multiplier라고 함

[참고] 2변수 함수의 전미분



주어진 것: $(x_1^*, x_2^*), f(x_1^*, x_2^*)$

실제 구해야 하는 것:

$$f(x_1^* + \Delta x_1, x_2^* + \Delta x_2) = f(x_1^*, x_2^*) + \Delta f$$

근사적으로 구할 수 있는 것:

$$f(x_1^*, x_2^*) + df$$

$\Delta x_1,\ \Delta x_2$**가 아주 작다면**

$\Delta f \cong df$ **라 볼 수 있음**

x_2**방향의 변화량**

$$df = \boxed{\frac{\partial f}{\partial x_1} dx_1} + \boxed{\frac{\partial f}{\partial x_2} dx_2}$$

x_1**방향의 변화량**

등호 제약 조건이 있는 문제의 후보 최적성 필요 조건
- Lagrange 함수의 도입

Minimize $L(x_1,\ x_2,\ v) = f(x_1,\ x_2) + vh(x_1,\ x_2)$

➡ 원래의 목적 함수(f)와 제약 조건(h)을 Lagrange 함수(L)로 표현하여 비제약 함수로 변환함

Necessary Condition		Necessary Condition
$\nabla L(x_1^*, x_2^*, v^*) = \mathbf{0}$ ⬇		
$\dfrac{\partial L(x_1^*, x_2^*, v^*)}{\partial v} = 0$	⇨	$h(x_1^*,\ x_2^*) = 0$
$\dfrac{\partial L(x_1^*, x_2^*, v^*)}{\partial x_1} = 0$	⇨	$\dfrac{\partial f(x_1^*,\ x_2^*)}{\partial x_1} + v^* \dfrac{\partial h(x_1^*,\ x_2^*)}{\partial x_1} = 0$
$\dfrac{\partial L(x_1^*, x_2^*, v^*)}{\partial x_2} = 0$	⇨	$\dfrac{\partial f(x_1^*,\ x_2^*)}{\partial x_2} + v^* \dfrac{\partial h(x_1^*,\ x_2^*)}{\partial x_2} = 0$

$$\left.\begin{aligned}\frac{\partial f(x_1^*,\ x_2^*)}{\partial x_1} + v^* \frac{\partial h(x_1^*,\ x_2^*)}{\partial x_1} &= 0\\ \frac{\partial f(x_1^*,\ x_2^*)}{\partial x_2} + v^* \frac{\partial h(x_1^*,\ x_2^*)}{\partial x_2} &= 0\end{aligned}\right\} \nabla f(\mathbf{x}^*) + v^* \nabla h(\mathbf{x}^*) = 0$$

앞의 방법과는 달리 Lagrange 함수를 도입할 경우
보다 간결하게 후보 최적성 필요 조건을 유도할 수 있음

등호 제약 조건이 있는 최적화 문제에서의 Lagrange 함수의 구성

Original Problem

Minimize $f(\mathbf{x}) = (x_1 - 1.5)^2 + (x_2 - 1.5)^2$

Subject to $h(\mathbf{x}) = x_1 + x_2 - 2 = 0$

Lagrange Function

Minimize $L(\mathbf{x}, \ v) = f(\mathbf{x}) + vh(\mathbf{x})$

$$= (x_1 - 1.5)^2 + (x_2 - 1.5)^2 + v(x_1 + x_2 - 2)$$

Necessary Condition: $\nabla L(\mathbf{x}^*, \ v^*) = 0$

$$\nabla f(\mathbf{x}^*) + v^* \nabla h(\mathbf{x}^*) = 0$$

$$\therefore -\nabla f(\mathbf{x}^*) = v^* \nabla h(\mathbf{x}^*)$$

$$\nabla f(\mathbf{x}) = \begin{bmatrix} 2(x_1 - 1.5) \\ 2(x_2 - 1.5) \end{bmatrix}, \ \nabla h(\mathbf{x}) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

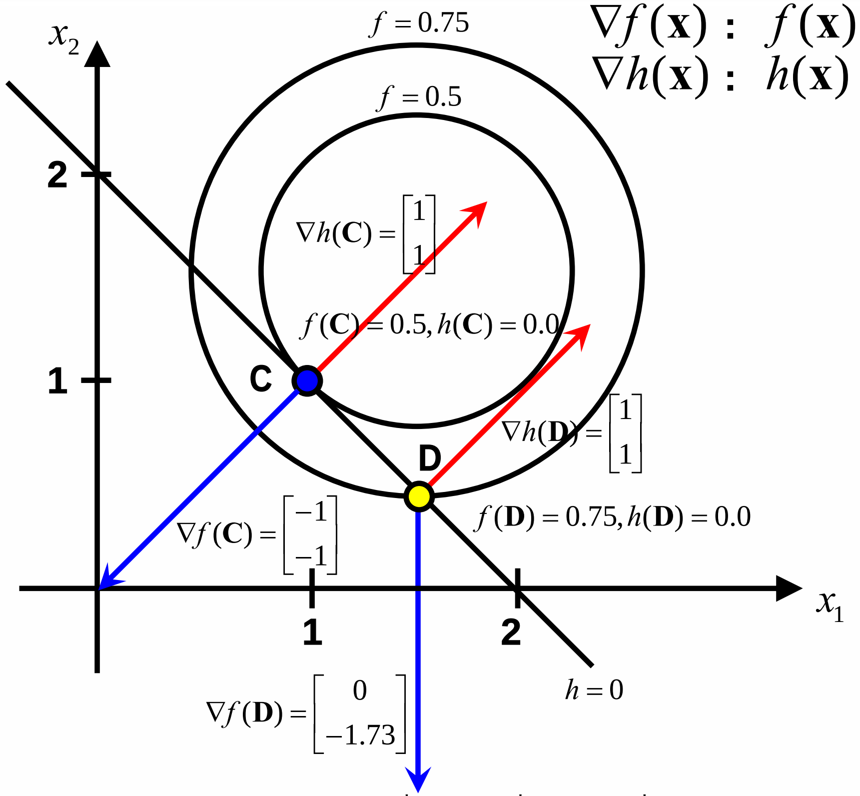
$$-2(x_1^* - 1.5) = v^*, \ -2(x_2^* - 1.5) = v^*$$

$$x_1^* + x_2^* - 2 = 0$$

$\Rightarrow x_1^* = x_2^* = 1, v^* = 1$ (점 C)

$\nabla f(\mathbf{x})$: $f(\mathbf{x})$ 의 증가 방향

$\nabla h(\mathbf{x})$: $h(\mathbf{x})$ 의 증가 방향



후보 최적점 C에서 $-\nabla f(\mathbf{x}^*) = v^* \nabla h(\mathbf{x}^*)$ 의 의미를 살펴 보면,

목적 함수 및 제약 함수의 Gradient 벡터는 동일 작용선 상에 있고, 서로 비례하며 이때 Lagrange multiplier v^*가 비례 상수임

$$\nabla f(\mathbf{C}) = \begin{bmatrix} -1 \\ -1 \end{bmatrix}, \ \nabla h(\mathbf{C}) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \ v^* = 1$$

그러나 점 D에서는 위 식을 만족하지 않으므로 후보 최적점이 아님

Lagrange 함수의 정의
- 등호 제약 조건이 있는 최적화 문제(요약)

- **등호 제약 조건이 있는 최적화 문제**

$$\textit{Minimize} \quad f(\mathbf{x}) = f(x_1, x_2, \ \ldots \ , x_n)$$

$$\textit{Subject to} \quad h_i(\mathbf{x}) = 0, \quad i = 1, \ \ldots \ , p$$

- Lagrange **함수**(L)**의 정의**

$$L(\mathbf{x}, \mathbf{v}) = f(\mathbf{x}) + \sum_{i=1}^{p} v_i h_i(\mathbf{x})$$

$$= f(\mathbf{x}) + \mathbf{v}^T \mathbf{h}(\mathbf{x})$$

v_i = **등호 제약 조건에 대한** Lagrange multiplier**로서 부호에 제한이 없음**

<이유> **원래의 등호 제약 조건("등식")의 양변에 -를 곱해도 주어진 문제의 해는 변하지 않으므로**

후보 최적성 필요 조건을 이용한 부등호 제약 최적화 문제의 해법

Original Problem

Minimize $f(\mathbf{x}) = (x_1 - 1.5)^2 + (x_2 - 1.5)^2$

Subject to $g(\mathbf{x}) = x_1 + x_2 - 2 \le 0$

➡ $g(\mathbf{x}) + s^2 = x_1 + x_2 - 2 + s^2 = 0$

부등호 제약 조건을 등호 제약 조건으로 변환하기 위해 도입한 완화 변수(slack variable)

Lagrange Function

Minimize $L(\mathbf{x},\ u,\ s) = f(\mathbf{x}) + u[g(\mathbf{x}) + s^2]$

$= (x_1 - 1.5)^2 + (x_2 - 1.5)^2$

$+ u(x_1 + x_2 - 2 + s^2)$

Necessary Condition: $\nabla L(\mathbf{x}^*,\ u^*,\ s^*) = 0$

$$\frac{\partial L}{\partial x_1} = 2(x_1 - 1.5) + u = 0,\quad \frac{\partial L}{\partial x_2} = 2(x_2 - 1.5) + u = 0$$

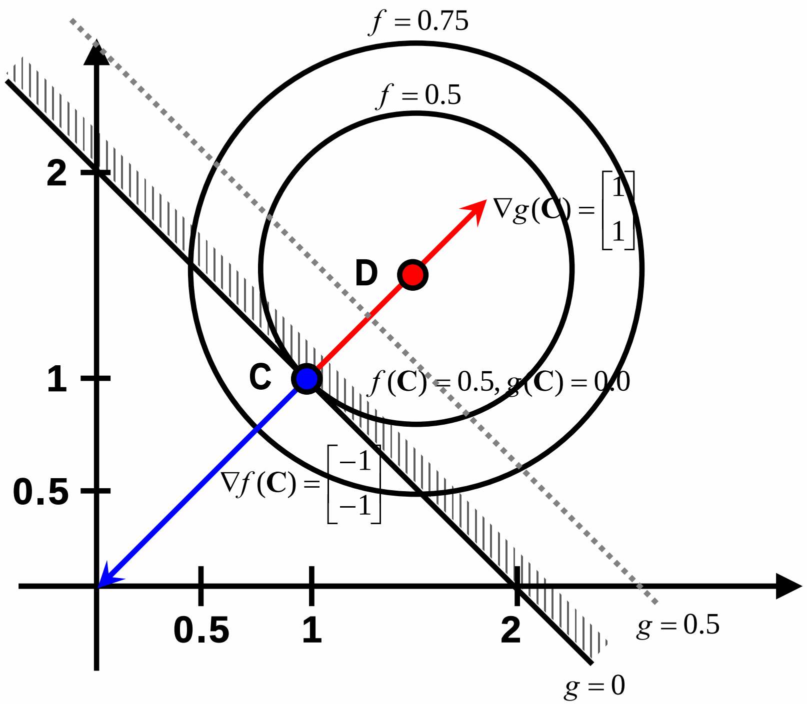
$$\frac{\partial L}{\partial u} = x_1 + x_2 - 2 + s^2 = 0,\quad \frac{\partial L}{\partial s} = 2us = 0 \text{ 단, } u \ge 0$$

(1) s = 0일 때(부등호 제약 조건이 등호 제약 조건으로 변환)

$x_1^* = x_2^* = 1, u^* = 1$ ➡ 후보 최적점(점 C)

(2) u = 0일 때(부등호 제약 조건을 만족함, 즉 부등호 제약 조건이 없는 문제임)

$x_1^* = x_2^* = 1.5, u^* = 0, s^2 = -1$ (점 D: 제약 조건을 위배)



부등호 제약 조건이 있는 문제의 후보 최적성 필요 조건(1)

부등호 제약 조건

$$g_i(\mathbf{x}) \le 0, \quad i = 1, \ldots, m$$

부등호 제약 조건을 등호 제약 조건으로 변경하기 위해서 완화 변수(slack variable) s_i^2 **을 도입하면,**

$$g_i(\mathbf{x}) + s_i^2 = 0, \quad i = 1, \ldots, m$$

부등호 제약 조건이 있는 문제의 Lagrange **함수**

등호 제약 조건이 있는 문제에 대한 Lagrange **함수는 다음과 같음**

$$L(\mathbf{x}, \mathbf{v}) = f(\mathbf{x}) + \sum_{i=1}^{p} v_i h(\mathbf{x}) = f(\mathbf{x}) + \mathbf{v}^T \mathbf{h}(\mathbf{x})$$

v_i : **등호 제약 조건에 대한** Lagrange multiplier**로서 부호에 제한이 없음**

원래의 부등호 제약 조건이 완화 변수의 도입을 통해 등호 제약 조건으로 변경되었기 때문에 위의 Lagrange **함수에 대입하여 정리하면**

$$L(\mathbf{x}, \mathbf{u}, \mathbf{s}) = f(\mathbf{x}) + \sum_{i=1}^{m} u_i (g_i(\mathbf{x}) + s_i^2) = f(\mathbf{x}) + \mathbf{u}^T (\mathbf{g}(\mathbf{x}) + \mathbf{s}^2) \qquad \text{단, } u_i \ge 0$$

u_i : **부등호 제약 조건에 대한** Lagrange multiplier**로서 0보다 크거나 같아야 함**

s_i : **부등호 제약 조건을 등호 제약 조건으로 변환하기 위한 완화 변수**

부등호 제약 조건이 있는 문제의 후보 최적성 필요 조건(2)

부등호 제약 조건이 있는 문제의 Lagrange 함수

$$L(\mathbf{x},\mathbf{u},\mathbf{s}) = f(\mathbf{x}) + \sum_{i=1}^{m} u_i (g_i(\mathbf{x}) + s_i^2) = f(\mathbf{x}) + \mathbf{u}^T (\mathbf{g}(\mathbf{x}) + \mathbf{s}^2)$$

u_i : 부등호 제약 조건에 대한 Lagrange multiplier로서 0보다 크거나 같아야 함

s_i : 부등호 제약 조건을 등호 제약 조건으로 변환하기 위한 완화 변수

부등호 제약 조건이 있는 문제의 후보 최적성 필요 조건

$$\nabla L(\mathbf{x}^*,\mathbf{u}^*,\mathbf{s}^*) = \mathbf{0}$$

⬇

$$\frac{\partial L}{\partial x_j} \equiv \frac{\partial f}{\partial x_j} + \sum_{i=1}^{m} u_i^* \frac{\partial g_i}{\partial x_j} = 0, \qquad j = 1, \ldots, n$$

$$\frac{\partial L}{\partial u_i} \equiv g_i(\mathbf{x}^*) + s_i^{*2} = 0, \qquad i = 1, \ldots, \mathrm{m}$$

$$\frac{\partial L}{\partial s_i} \equiv u_i^* s_i^* = 0, \qquad i = 1, \ldots, m$$

$$u_i^* \geq 0, \qquad i = 1, \ldots, m$$

Lagrange 함수의 정의
- 부등호 제약 조건이 있는 최적화 문제(요약)

- **부등호 제약 조건이 있는 최적화 문제**

$$\text{Minimize} \quad f(\mathbf{x}) = f(x_1, x_2, \ \ldots \ , x_n)$$

$$\text{Subject to} \quad g_i(\mathbf{x}) \le 0, \quad i = 1, \ \ldots \ , m$$

- Lagrange **함수(L)의 정의**

$$L(\mathbf{x}, \mathbf{v}, \mathbf{u}, \mathbf{s}) = f(\mathbf{x}) + \sum_{i=1}^{m} u_i (g_i(\mathbf{x}) + s_i^2)$$

$$= f(\mathbf{x}) + \mathbf{u}^T (\mathbf{g}(\mathbf{x}) + \mathbf{s}^2) \qquad \text{단}, u_i \ge 0$$

u_i : **부등호 제약 조건에 대한** Lagrange multiplier**로서** 0**보다 크거나 같아야 함**

s_i : **부등호 제약 조건을 등호 제약 조건으로 변환하기 위한 완화 변수**

부등호 제약 조건에 대한 Lagrange multiplier의 부호가 양이어야 하는 이유

Original Problem

Minimize $f(\mathbf{x}) = (x_1 - 1.5)^2 + (x_2 - 1.5)^2$

Subject to $g(\mathbf{x}) = x_1 + x_2 - 2 \le 0$

Lagrange Function

Minimize $L(\mathbf{x},\ u,\ s) = f(\mathbf{x}) + u\left[g(\mathbf{x}) + s^2\right]$

$= (x_1 - 1.5)^2 + (x_2 - 1.5)^2$

$+ u(x_1 + x_2 - 2 + s^2)$

Necessary Condition: $\nabla L(\mathbf{x}^*,\ u^*,\ s^*) = 0$

$\nabla f(\mathbf{x}^*) + u^* \nabla g(\mathbf{x}^*) = 0$

$\therefore -\nabla f(\mathbf{x}^*) = u^* \nabla g(\mathbf{x}^*)$

$\nabla f(\mathbf{x}) = \begin{bmatrix} 2(x_1 - 1.5) \\ 2(x_2 - 1.5) \end{bmatrix},\quad \nabla g(\mathbf{x}) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

$-2(x_1^* - 1.5) = u^*,\ -2(x_2^* - 1.5) = u^*$

$x_1^* + x_2^* - 2 + s^2 = 0,\ u^* s^* = 0$

(1) s^* = 0일 때

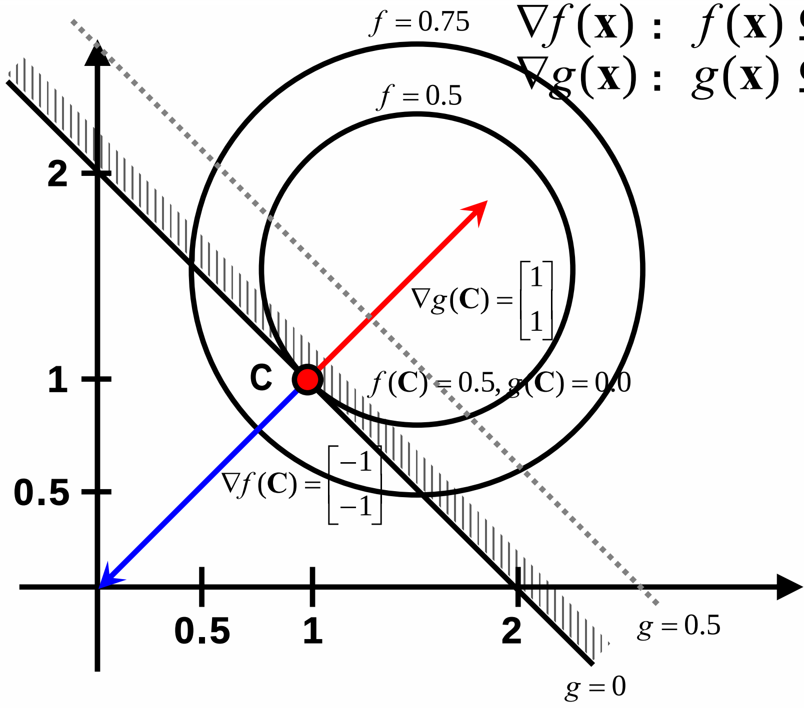
$x_1^* = x_2^* = 1, u^* = 1$ ➡ 국부적 후보 최적점

(2) u^* = 0일 때

$x_1^* = x_2^* = 1.5, u^* = 0, s^2 = -1$ (제약 조건을 위배)

$\nabla f(\mathbf{x})$: $f(\mathbf{x})$ 의 증가 방향

$\nabla g(\mathbf{x})$: $g(\mathbf{x})$ 의 증가 방향



후보 최적점 C에서 $-\nabla f(\mathbf{x}^*) = u^* \nabla g(\mathbf{x}^*)$ 의 의미를 살펴 보면,

1) $u^* > 0$인 경우

목적 함수(f)의 감소 방향 = 제약 조건(g)의 증가 방향

목적 함수 감소 = 제약 조건 증가

2) $u^* < 0$인 경우

목적 함수 감소 = 제약 조건 감소(해 없음)

Lagrange 함수의 정의
- 등호 및 부등호 제약 조건이 있는 최적화 문제(요약)

- 등호 및 부등호 제약 조건이 있는 최적화 문제

$$\begin{aligned} &\textit{Minimize} \quad f(\mathbf{x}) = f(x_1, x_2, \ \dots \ , x_n) \\ &\textit{Subject to} \quad h_i(\mathbf{x}) = 0, \quad i = 1, \ \dots \ , p \\ &\qquad\qquad\quad g_i(\mathbf{x}) \le 0, \quad i = 1, \ \dots \ , m \end{aligned}$$

- Lagrange 함수(L)의 정의

$$\begin{aligned} L(\mathbf{x}, \mathbf{v}, \mathbf{u}, \mathbf{s}) &= f(\mathbf{x}) + \sum_{i=1}^{p} v_i h_i(\mathbf{x}) + \sum_{i=1}^{m} u_i (g_i(\mathbf{x}) + s_i^2) \\ &= f(\mathbf{x}) + \mathbf{v}^T \mathbf{h}(\mathbf{x}) + \mathbf{u}^T (\mathbf{g}(\mathbf{x}) + \mathbf{s}^2) \qquad \text{단}, u_i \ge 0 \end{aligned}$$

v_i : 등호 제약 조건에 대한 Lagrange multiplier로서 부호에 제한이 없음
u_i : 부등호 제약 조건에 대한 Lagrange multiplier로서 0보다 크거나 같아야 함
s_i : 부등호 제약 조건을 등호 제약 조건으로 변환하기 위한 완화 변수

등호 및 부등호 제약 조건이 있는 문제의 후보 최적성 필요 조건 - 쿤-터커(Kuhn-Tucker) 필요 조건

최적화 문제

Minimize $f(\mathbf{x}) = f(x_1, x_2, \Lambda\ , x_n)$

Subject to $h_i(\mathbf{x}) = 0, \quad i = 1,\ldots,p$ 등호 제약 조건

$g_i(\mathbf{x}) = 0, \quad i = 1,\ldots,m$ 부등호 제약 조건

Lagrange 함수의 정의

$$L(\mathbf{x}, \mathbf{v}, \mathbf{u}, \mathbf{s}) = f(\mathbf{x}) + \sum_{i=1}^{p} v_i h_i(\mathbf{x}) + \sum_{i=1}^{m} u_i (g_i(\mathbf{x}) + s_i^2)$$ 일 때

Kuhn-Tucker 필요 조건: $\nabla L(\mathbf{x}, \mathbf{v}, \mathbf{u}, \mathbf{s}) = \mathbf{0}$

$$\frac{\partial L}{\partial x_j} \equiv \frac{\partial f}{\partial x_j} + \sum_{i=1}^{p} v_i^* \frac{\partial h_i}{\partial x_j} + \sum_{i=1}^{m} u_i^* \frac{\partial g_i}{\partial x_j} = 0, \qquad j = 1,\ \ldots\ , n$$

$$\frac{\partial L}{\partial v_i} \equiv h_i(\mathbf{x}^*) = 0, \qquad i = 1,\ \ldots\ , p$$

$$\frac{\partial L}{\partial u_i} \equiv g_i(\mathbf{x}^*) + s_i^{*2} = 0, \qquad i = 1,\ \ldots\ , \mathrm{m}$$

$$\frac{\partial L}{\partial s_i} \equiv u_i^* s_i^* = 0, \qquad i = 1,\ \ldots\ , m$$

$$u_i^* \geq 0, \qquad i = 1,\ \ldots\ , m$$

x*가 국부적 후보 최적해라면 반드시 이 조건을 만족해야 함

즉, Kuhn-Tucker 필요 조건은 x*가 국부적 후보 최적해이기 위한 필요 조건임

따라서 등호 제약 조건 및 부등호 제약 조건을 가진 최적화 문제에 대해, 국부적 후보 최적점을 구하기 위한 조건으로 사용할 수 있음

여기서, 모든 함수 값 및 경사도(Gradient)는 점 $\mathbf{x}^*$에서 계산한다.

제약 최적화 문제
- Kuhn-Tucker 필요 조건을 이용한 국부적 후보 최적해 도출

①

Minimize $f(\mathbf{x}) = x_1^2 + x_2^2 - 3x_1x_2$

$g(\mathbf{x}) = x_1^2 + x_2^2 - 6 \le 0$

②

$L(\mathbf{x}, u, s) = x_1^2 + x_2^2 - 3x_1x_2 + u(x_1^2 + x_2^2 - 6 + s^2)$

③

$$\frac{\partial L}{\partial x_1} = 2x_1 - 3x_2 + 2ux_1 = 0$$

$$\frac{\partial L}{\partial x_2} = 2x_2 - 3x_1 + 2ux_2 = 0$$

$$\frac{\partial L}{\partial u} = x_1^2 + x_2^2 - 6 + s^2 = 0,\ s^2 \ge 0,\ u \ge 0$$

$$\frac{\partial L}{\partial s} = \boxed{2us = 0}$$ ➡ 두가지 경우가 나옴

CASE #1 : $u = 0$ **(제약 조건을 고려하지 않아도 되는 경우)**

$2x_1 - 3x_2 = 0$

$-3x_1 + 2x_2 = 0$ ➡ **점** A: $x_1^* = 0,\ x_2^* = 0,\ f(x_1^*, x_2^*) = 0$

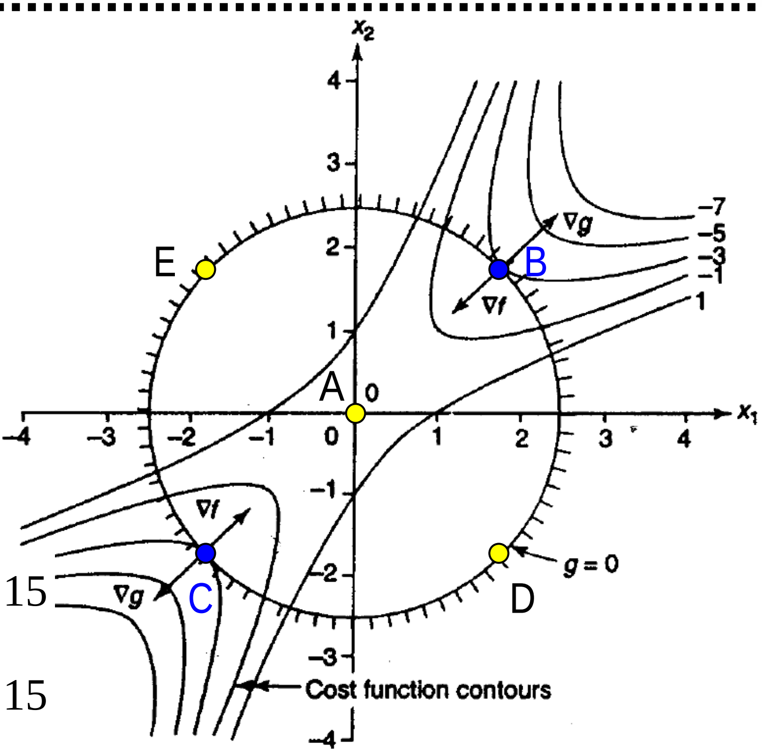
CASE #2 : $s = 0$ **(제약 조건의 경계 상에 해가 있는 경우)**

$x_1 = x_2 = \sqrt{3},\ u = \frac{1}{2}$ ➡ **점** B: $x_1^* = x_2^* = \sqrt{3}, f(x_1^*, x_2^*) = -3$

$x_1 = x_2 = -\sqrt{3},\ u = \frac{1}{2}$ ➡ **점** C: $x_1^* = x_2^* = -\sqrt{3}, f(x_1^*, x_2^*) = -3$

$x_1 = -x_2 = \sqrt{3},\ u = -\frac{5}{2}$ ➡ **점** D: $x_1^* = \sqrt{3}, x_2^* = -\sqrt{3}, f(x_1^*, x_2^*) = 15$

$x_1 = -x_2 = -\sqrt{3},\ u = -\frac{5}{2}$ ➡ **점** E: $x_1^* = -\sqrt{3}, x_2^* = \sqrt{3}, f(x_1^*, x_2^*) = 15$



Kuhn-Tucker 필요 조건을 이용한 프로펠러의 최적 주요 치수 결정 예(1)

Given $P, n, A_E / A_O, V$

Find $J, P_i / D_P$

Maximize $\eta_O = \frac{J}{2\pi} \cdot \frac{K_T}{K_Q}$ → K_T와 K_Q가 모두 J와 P_i/D_p의 함수이므로 목적 함수 역시 J와 P_i/D_p의 함수임

Subject to $\frac{P}{2\pi n} = \rho \cdot n^2 \cdot D_P{}^5 \cdot K_Q$

: 주기관이 전달한 토오크를 프로펠러가 흡수하는 조건

Where, $J = \frac{V(1-w)}{n \cdot D_P}$

$$K_T = f(J, P_i / D_P)$$

$$K_Q = f(J, P_i / D_P)$$

➡ 미지수 2개, 등호 제약 조건 1개인 최적화 문제

P: 프로펠러 전달 마력
n: 프로펠러 회전수
D_P: 프로펠러 직경
P_i: 프로펠러 피치
A_E/A_O: 프로펠러 날개 면적비
V: 선속
η_O: 프로펠러 효율

Kuhn-Tucker 필요 조건을 이용한 프로펠러의 최적 주요 치수 결정 예(2)

$$\frac{P}{2\pi n} = \rho \cdot n^2 \cdot D_P^{\,5} \cdot K_Q \quad \cdots\cdots \quad \text{(a)}$$

: 주기관이 전달한 토오크를 프로펠러가 흡수하는 조건

주기관 전달 마력을 프로펠러가 흡수하는 경우, 그 조건은 식 (a)로부터

$$C = \frac{K_Q}{J^5} = \frac{P \cdot n^2}{2\pi\rho \cdot V_A^{\,5}}$$

$$G(J, P_i / D_P) = K_Q - C \cdot J^5 = 0 \quad \cdots\cdots \quad \text{(b)}$$

프로펠러 단독 효율(η_0)은

$$F(J, P_i / D_P) = \eta_0 = \frac{J}{2\pi} \cdot \frac{K_T}{K_Q} \quad \cdots\cdots \quad \text{(c)}$$

목적 함수 F는 P_i / D_P 와 J의 함수이며 따라서 식 (a)를 만족하면서 단독 효율이 최대가 되는 프로펠러 피치비(P_i / D_P)와 전진비(J)를 구하는 최적화 문제가 됨

Kuhn-Tucker 필요 조건을 이용한 프로펠러의 최적 주요 치수 결정 예(3)

라그란지 승수 λ 를 도입하여 식(b)와 식(c)로부터

$$H(J, P_i/D_P, \lambda) = F(J, P_i/D_P) + \lambda G(J, P_i/D_P) \quad \cdots\cdots \quad \text{(d)}$$

와 같은 함수를 정의하고 이로부터 함수 H가 최대가 되는 미정 계수 P_i/D_P, J 및 λ 를 결정함

$$\frac{\partial H}{\partial J} = \frac{1}{2\pi}\left(\frac{K_T}{K_Q}\right) + \frac{J}{2\pi}\frac{\{(\frac{\partial K_T}{\partial J})\cdot K_Q - (\frac{\partial K_Q}{\partial J})\cdot K_T\}}{K_Q^{\ 2}} + \lambda\{(\frac{\partial K_Q}{\partial J}) - 5\cdot C\cdot J^4\}$$

$$= 0 \quad \cdots\cdots \quad \text{(e)}$$

$$\frac{\partial H}{\partial (P_i/D_P)} = \frac{J}{2\pi}\frac{\{(\frac{\partial K_T}{\partial P_i/D_P})\cdot K_Q - (\frac{\partial K_Q}{\partial P_i/D_P})\cdot K_T\}}{K_Q^{\ 2}} + \lambda\left(\frac{\partial K_Q}{\partial P_i/D_P}\right)$$

$$= 0 \quad \cdots\cdots \quad \text{(f)}$$

$$\frac{\partial H}{\partial \lambda} = K_Q - C\cdot J^5 = 0 \quad \cdots\cdots \quad \text{(g)}$$

Kuhn-Tucker 필요 조건을 이용한 프로펠러의 최적 주요 치수 결정 예(4)

식 (e), (f), (g)에서 λ 를 소거하여 정리하면

$$\left(\frac{\partial K_Q}{\partial (P_i / D_P)}\right)\left\{J \cdot \left(\frac{\partial K_T}{\partial J}\right) - 4K_T\right\}$$
$$+\left(\frac{\partial K_T}{\partial (P_i / D_P)}\right)\left\{5K_Q - J \cdot \left(\frac{\partial K_Q}{\partial J}\right)\right\} = 0 \quad \cdots\cdots \quad \text{(h)}$$

$$K_Q - C \cdot J^5 = 0 \quad \cdots\cdots \quad \text{(i)}$$

식 (h), (i)의 해를 구함으로써 전달 마력을 흡수하는 최대의 효율을 갖는 피치비(P_i / D_P)와 전진비(J)를 구할 수 있다.

$J = \dfrac{V(1-w)}{n \cdot D_P}$ 이므로 J를 구하면 D_p를 구할 수 있고, 또한 P_i/D_p로부터 P_i 역시 구할 수 있다.

즉, 프로펠러 직경(D_P)과 피치(P_i)를 모두 구할 수 있다.

Ch5. 제약 비선형 최적화 기법 및 응용 예

5.1 2차 계획법(Quadratic Programming)
5.2 순차적 선형 계획법(Sequential Linear Programming; SLP)
5.3 Penalty Function 방법
5.4 순차적 2차 계획법(Sequential Quadratic Programming; SQP)
5.5 CSD(Constrained Steepest Descent) 방법
5.6 최적화 기법을 이용한 선박의 최적 주요 치수 결정 문제

Advanced
Ship
Design
Automation
Laboratory

5.1 2차 계획법
(Quadratic Programming)

Advanced
Ship
Design
Automation
Laboratory

2차 계획 문제(Quadratic Programming Problem)의 정식화(1)

$$\text{Minimize} \quad f(\mathbf{x}+\Delta\mathbf{x}) \cong f(\mathbf{x}) + \nabla f^T(\mathbf{x})\Delta\mathbf{x} + 0.5\Delta\mathbf{x}^T\mathbf{H}\Delta\mathbf{x}$$

Taylor 급수의 2차항까지 고려한 목적 함수

$$\text{Subject to} \quad h_j(\mathbf{x}+\Delta\mathbf{x}) \cong h_j(\mathbf{x}) + \nabla h_j^T(\mathbf{x})\Delta\mathbf{x} = 0;\ j=1\ to\ p$$

Taylor 급수의 1차항(선형항)만 고려한 등호 제약 조건

$$g_j(\mathbf{x}+\Delta\mathbf{x}) \cong g_j(\mathbf{x}) + \nabla g_j^T(\mathbf{x})\Delta\mathbf{x} \le 0;\ j=1\ to\ m$$

Taylor 급수의 1차항(선형항)만 고려한 부등호 제약 조건

여기서, $\bar{f} = f(\mathbf{x}+\Delta\mathbf{x}) - f(\mathbf{x}),\ e_j = -h_j(\mathbf{x}),\ b_j = -g_j(\mathbf{x}),$

$c_i = \partial f(\mathbf{x})/\partial x_i,\ n_{ij} = \partial h_j(\mathbf{x})/\partial x_i,\ a_{ij} = \partial g_j(\mathbf{x})/\partial x_i,$

$d_i = \Delta x_i$ 라고 가정하면

Matrix form

$$\text{Minimize} \quad \bar{f} = \mathbf{c}^T{}_{(1\times n)}\mathbf{d}_{(n\times 1)} + \frac{1}{2}\mathbf{d}^T{}_{(1\times n)}\mathbf{H}_{(n\times n)}\mathbf{d}_{(n\times 1)}$$

: 2차 형식의 목적 함수

$$\text{Subject to} \quad \mathbf{N}^T{}_{(p\times n)}\mathbf{d}_{(n\times 1)} = \mathbf{e}_{(p\times 1)}$$

: 선형화 된 등호 제약 조건

$$\mathbf{A}^T{}_{(m\times n)}\mathbf{d}_{(n\times 1)} \le \mathbf{b}_{(m\times 1)}$$

: 선형화 된 부등호 제약 조건

2차 계획 문제(Quadratic Programming Problem)의 정식화(2)

$$\text{Minimize} \quad \bar{f} = \mathbf{c}^T{}_{(1\times n)}\mathbf{d}_{(n\times 1)} + \frac{1}{2}\mathbf{d}^T{}_{(1\times n)}\mathbf{H}_{(n\times n)}\mathbf{d}_{(n\times 1)}$$

$$\text{Subject to} \quad \mathbf{N}^T{}_{(p\times n)}\mathbf{d}_{(n\times 1)} = \mathbf{e}_{(p\times 1)}$$

$$\mathbf{A}^T{}_{(m\times n)}\mathbf{d}_{(n\times 1)} \le \mathbf{b}_{(m\times 1)}$$

$$\Big\downarrow \quad \mathbf{H}_{(n\times n)} = \mathbf{I}_{(n\times n)}\text{라고 가정해서 근사화 한다.}$$

$$\text{Minimize} \quad \bar{f} = \mathbf{c}^T{}_{(1\times n)}\mathbf{d}_{(n\times 1)} + \frac{1}{2}\mathbf{d}^T{}_{(1\times n)}\mathbf{I}_{(n\times n)}\mathbf{d}_{(n\times 1)} = \mathbf{c}^T{}_{(1\times n)}\mathbf{d}_{(n\times 1)} + \frac{1}{2}\mathbf{d}^T{}_{(1\times n)}\mathbf{d}_{(n\times 1)}$$

$$\text{Subject to} \quad \mathbf{N}^T{}_{(p\times n)}\mathbf{d}_{(n\times 1)} = \mathbf{e}_{(p\times 1)}$$

$$\mathbf{A}^T{}_{(m\times n)}\mathbf{d}_{(n\times 1)} \le \mathbf{b}_{(m\times 1)}$$

- ➡ $\mathbf{H}_{(n\times n)} = \mathbf{I}_{(n\times n)}$이므로 행렬 $\mathbf{H}_{(n\times n)}$는 양정(Positive Definite) 행렬이며, 모든 제약 조건이 선형임
- ➡ 이 문제는 볼록 계획(Convex Programming) 문제이며, 해가 존재하면 이는 유일한 해 (전역 최적해)임

Simplex 방법을 이용한 2차 계획 문제의 풀이(1)

$$\textit{Minimize} \quad \bar{f} = \mathbf{c}^T{}_{(1\times n)}\mathbf{d}_{(n\times 1)} + \frac{1}{2}\mathbf{d}^T{}_{(1\times n)}\mathbf{H}_{(n\times n)}\mathbf{d}_{(n\times 1)}$$

$$\textit{Subject to} \quad \mathbf{N}^T{}_{(p\times n)}\mathbf{d}_{(n\times 1)} = \mathbf{e}_{(p\times 1)}$$

$$\mathbf{A}^T{}_{(m\times n)}\mathbf{d}_{(n\times 1)} \le \mathbf{b}_{(m\times 1)} \quad \Rightarrow \quad \mathbf{A}^T{}_{(m\times n)}\mathbf{d}_{(n\times 1)} - \mathbf{b}_{(m\times 1)} + \mathbf{s}_{(m\times 1)} = \mathbf{0}$$

완화 변수로 s^2을 사용할 수 있지만
Simplex 방법을 사용하기 위하여
s를 사용하고 추가 제약 조건으로
$s \ge 0$을 도입함

한편, Simplex 방법을 이용할 경우, 설계 변수 d는 음이 아니어야 함
따라서 다음과 같은 추가의 제약 조건이 필요함 $\Rightarrow$ $\mathbf{d}_{(n\times 1)} \ge 0$

Lagrange 함수

$$L = \mathbf{c}^T{}_{(1\times n)}\mathbf{d}_{(n\times 1)} + \frac{1}{2}\mathbf{d}^T{}_{(1\times n)}\mathbf{H}_{(n\times n)}\mathbf{d}_{(n\times 1)}$$
$$+ \mathbf{u}^T{}_{(1\times m)}(\mathbf{A}^T{}_{(m\times n)}\mathbf{d}_{(n\times 1)} + \mathbf{s}_{(m\times 1)} - \mathbf{b}_{(m\times 1)}) - \zeta^T{}_{(1\times n)}\mathbf{d}^{1)}_{(n\times 1)}$$
$$+ \mathbf{v}^T{}_{(1\times p)}(\mathbf{N}^T{}_{(p\times n)}\mathbf{d}_{(n\times 1)} - \mathbf{e}_{(p\times 1)})$$

1): 부등호 제약 조건에 대해 Lagrange 함수를 구성하기 위해 다음과 같이 완화 변수를 도입함

$$-\mathbf{d}_{(n\times 1)} \le \mathbf{0} \Rightarrow -\mathbf{d}_{(n\times 1)} + \boldsymbol{\delta}_{(n\times 1)} = \mathbf{0} \Rightarrow \boldsymbol{\delta}_{(n\times 1)} = \mathbf{d}_{(n\times 1)} \quad \text{단, } \delta_{(n\times 1)} \ge 0$$

따라서 원래 변수의 값을 알면 완화 변수의 값을 바로 알 수 있으므로 완화 변수를 생략해도 됨

따라서 Kuhn-Tucker의 필요 조건 중 $\zeta^T{}_{(1\times n)}\boldsymbol{\delta}_{(n\times 1)} = \mathbf{0}$ 는 다음과 같이 표현할 수 있음

$$\boxed{\zeta^T{}_{(1\times n)}\mathbf{d}_{(n\times 1)} = \mathbf{0}}$$

Simplex 방법을 이용한 2차 계획 문제의 풀이(2)

Lagrange 함수

$$L = \mathbf{c}^T{}_{(1\times n)}\mathbf{d}_{(n\times 1)} + \frac{1}{2}\mathbf{d}^T{}_{(1\times n)}\mathbf{H}_{(n\times n)}\mathbf{d}_{(n\times 1)}$$

$$+\mathbf{u}^T{}_{(1\times m)}(\mathbf{A}^T{}_{(m\times n)}\mathbf{d}_{(n\times 1)} + \mathbf{s}_{(m\times 1)} - \mathbf{b}_{(m\times 1)}) - \zeta^T{}_{(1\times n)}\mathbf{d}_{(n\times 1)}$$

$$+\mathbf{v}^T{}_{(1\times p)}(\mathbf{N}^T{}_{(p\times n)}\mathbf{d}_{(n\times 1)} - \mathbf{e}_{(p\times 1)})$$

Kuhn-Tucker 필요 조건: $\nabla L = \mathbf{0}$

$$\frac{\partial L}{\partial \mathbf{d}_{(n\times 1)}} = \mathbf{c}_{(n\times 1)} + \mathbf{H}_{(n\times n)}\mathbf{d}_{(n\times 1)} + \mathbf{A}_{(n\times m)}\mathbf{u}_{(m\times 1)} - \zeta_{(n\times 1)} + \mathbf{N}_{(n\times p)}\mathbf{v}_{(p\times 1)} = \mathbf{0}$$

$$\frac{\partial L}{\partial \mathbf{u}_{(m\times 1)}} = \mathbf{A}^T{}_{(m\times n)}\mathbf{d}_{(n\times 1)} + \mathbf{s}_{(m\times 1)} - \mathbf{b}_{(m\times 1)} = \mathbf{0}$$

$$\frac{\partial L}{\partial \mathbf{v}_{(p\times 1)}} = \mathbf{N}^T{}_{(p\times n)}\mathbf{d}_{(n\times 1)} - \mathbf{e}_{(p\times 1)} = \mathbf{0}$$

이 식들은 설계 변수 d에 대해 모두 선형이므로, 이 식들로부터 설계 변수 d를 구하는 문제는 등호 제약 조건만으로 이루어진 선형 계획 문제임

$$u_i s_i = 0; i = 1\,to\,m \qquad \zeta_i d_i = 0; i = 1\,to\,n$$

이 식들은 비선형 방정식으로서 해를 구한 후 이를 만족하는지 평가하기 위한 용도로 사용됨

여기서, $u_i, s_i \geq 0;\ i = 1\,to\,m,\ d_i, \zeta_i \geq 0;\ i = 1\,to\,n$

Simplex 방법을 이용한 2차 계획 문제의 풀이(3)

Kuhn-Tucker 필요 조건: $\nabla L = \mathbf{0}$

$$\begin{cases} \dfrac{\partial L}{\partial \mathbf{d}_{(n\times1)}} = \mathbf{c}_{(n\times1)} + \mathbf{H}_{(n\times n)}\mathbf{d}_{(n\times1)} + \mathbf{A}_{(n\times m)}\mathbf{u}_{(m\times1)} - \boldsymbol{\zeta}_{(n\times1)} + \mathbf{N}_{(n\times p)}\mathbf{v}_{(p\times1)} = \mathbf{0} \\ \dfrac{\partial L}{\partial \mathbf{u}_{(m\times1)}} = \mathbf{A}^T{}_{(m\times n)}\mathbf{d}_{(n\times1)} + \mathbf{s}_{(m\times1)} - \mathbf{b}_{(m\times1)} = \mathbf{0} \\ \dfrac{\partial L}{\partial \mathbf{v}_{(p\times1)}} = \mathbf{N}^T{}_{(p\times n)}\mathbf{d}_{(n\times1)} - \mathbf{e}_{(p\times1)} = \mathbf{0} \end{cases}$$

여기서, $u_i, s_i \geq 0;\ i = 1\, to\, m,\ d_i, \zeta_i \geq 0;\ i = 1\, to\, n$

$$\begin{cases} u_i s_i = 0; i = 1\, to\, m \\ \zeta_i d_i = 0; i = 1\, to\, n \end{cases}$$

이 식들은 비선형 방정식으로서 해를 구한 후
이를 만족하는지 평가하기 위한 용도로 사용됨

이 식들은 설계 변수 $\mathbf{d}$에 대해 모두 선형이므로,
이 식들로부터 설계 변수 $\mathbf{d}$를 구하는 문제는 등호 제약 조건만으로 이루어진 선형 계획 문제임

한편, Simplex 방법을 이용할 경우, 설계 변수 d는 음이 아니어야 함
따라서 다음과 같은 추가의 제약 조건이 필요함

$$\mathbf{d}_{(n\times1)} \geq 0$$

만일 설계 변수가 부호에 제한이 없는 경우에는 문제를 풀기 전에 먼저 설계 변수를 아래와 같이 변환해야 함

$$\mathbf{d}_{(n\times1)} = \mathbf{d}_{(n\times1)}{}^{+} - \mathbf{d}_{(n\times1)}{}^{-} \qquad \text{여기서, } \mathbf{d}_{(n\times1)}{}^{+},\ \mathbf{d}_{(n\times1)}{}^{-} \geq 0$$

등호 제약 조건에 대한 Lagrange multiplier $\mathbf{v}_{(p\times1)}$는 부호의 제한이 없으므로 다음과 같이 표현 가능함

$$\mathbf{v}_{(p\times1)} = \mathbf{y}_{(p\times1)} - \mathbf{z}_{(p\times1)}$$

Simplex 방법을 이용한 2차 계획 문제의 풀이(4)

Kuhn-Tucker 필요 조건: $\nabla L = \mathbf{0}$

$$\frac{\partial L}{\partial \mathbf{d}_{(n\times1)}} = \mathbf{c}_{(n\times1)} + \mathbf{H}_{(n\times n)}\mathbf{d}_{(n\times1)} + \mathbf{A}_{(n\times m)}\mathbf{u}_{(m\times1)} - \boldsymbol{\zeta}_{(n\times1)} + \mathbf{N}_{(n\times p)}\mathbf{v}_{(p\times1)} = \mathbf{0}$$

$$\frac{\partial L}{\partial \mathbf{u}_{(m\times1)}} = \mathbf{A}^T{}_{(m\times n)}\mathbf{d}_{(n\times1)} + \mathbf{s}_{(m\times1)} - \mathbf{b}_{(m\times1)} = \mathbf{0}$$

$$\frac{\partial L}{\partial \mathbf{v}_{(p\times1)}} = \mathbf{N}^T{}_{(p\times n)}\mathbf{d}_{(n\times1)} - \mathbf{e}_{(p\times1)} = \mathbf{0}$$

여기서, $u_i, s_i \geq 0;\ i = 1\, to\, m,\ d_i, \zeta_i \geq 0;\ i = 1\, to\, n$

$$u_i s_i = 0; i = 1\, to\, m$$

$$\zeta_i d_i = 0; i = 1\, to\, n$$

Kuhn-Tucker 필요 조건(행렬식 표현)

$$\underbrace{\begin{bmatrix} \mathbf{H}_{(n\times n)} & \mathbf{A}_{(n\times m)} & -\mathbf{I}_{(n\times n)} & \mathbf{0}_{(n\times m)} & \mathbf{N}_{(n\times p)} & -\mathbf{N}_{(n\times p)} \\ \mathbf{A}^T{}_{(m\times n)} & \mathbf{0}_{(m\times m)} & \mathbf{0}_{(m\times n)} & \mathbf{I}_{(m\times m)} & \mathbf{0}_{(m\times p)} & \mathbf{0}_{(m\times p)} \\ \mathbf{N}^T{}_{(p\times n)} & \mathbf{0}_{(p\times m)} & \mathbf{0}_{(p\times n)} & \mathbf{0}_{(p\times m)} & \mathbf{0}_{(p\times p)} & \mathbf{0}_{(p\times p)} \end{bmatrix}}_{= \mathbf{B}_{((n+m+p)\times(2n+2m+2p))}} \underbrace{\begin{bmatrix} \mathbf{d}_{(n\times1)} \\ \mathbf{u}_{(m\times1)} \\ \boldsymbol{\zeta}_{(n\times1)} \\ \mathbf{s}_{(m\times1)} \\ \mathbf{y}_{(p\times1)} \\ \mathbf{z}_{(p\times1)} \end{bmatrix}}_{= \mathbf{X}_{((2n+2m+2p)\times1)}} = \underbrace{\begin{bmatrix} -\mathbf{c}_{(n\times1)} \\ \mathbf{b}_{(m\times1)} \\ \mathbf{e}_{(p\times1)} \end{bmatrix}}_{= \mathbf{D}_{((n+m+p)\times1)}}$$

$$u_i s_i = 0; i = 1\, to\, m$$

$$\zeta_i d_i = 0; i = 1\, to\, n$$

$$\mathbf{B}_{((n+m+p)\times(2n+2m+2p))}\mathbf{X}_{((2n+2m+2p)\times1)} = \mathbf{D}_{((n+m+p)\times1)}$$

Simplex 방법을 이용한 2차 계획 문제의 풀이(5)

Kuhn-Tucker 필요 조건(행렬식 표현)

$$\mathbf{B}_{((n+m+p)\times(2n+2m+2p))}\mathbf{X}_{((2n+2m+2p)\times 1)} = \mathbf{D}_{((n+m+p)\times 1)}$$

Simplex 방법을 이용한 QP 문제의 해법

1. Kuhn-Tucker 필요 조건의 해를 구하는 문제는 등호 제약 조건만으로 이루어진 부정 선형 연립 방정식의 해를 구하는 문제(선형 계획 문제)임
2. 부정 선형 연립 방정식의 해를 구하기 위하여 Simplex 방법에서 인위 변수 및 인위 목적 함수를 도입하여 초기 기저 가능해를 구하는 방법임

$$\mathbf{B}_{((n+m+p)\times(2n+2m+2p))}\mathbf{X}_{((2n+2m+2p)\times 1)} + \mathbf{Y}_{((n+m+p)\times 1)} = \mathbf{D}_{((n+m+p)\times 1)}$$

여기서, D의 어떤 요소가 음이면 해당 식을 -1로 곱하여 양수로 만듦

3. 인위 목적 함수를 다음과 같이 정의함

$$w = \sum_{i=1}^{n+m+p} Y_i = \sum_{i=1}^{n+m+p} D_i - \sum_{j=1}^{2(n+m+p)} \sum_{i=1}^{n+m+p} B_{ij} X_j = w_0 + \sum_{j=1}^{2(n+m+p)} C_j X_j$$

여기서, $C_j = -\sum_{i=1}^{n+m+p} B_{ij}, \; w_0 = \sum_{i=1}^{n+m+p} D_i$ 인위 목적 함수의 초기값

행렬 B의 j번째 열의 요소를 모두 더해 부호를 바꾼 것(상대 비용 계수)

4. Simplex 방법을 이용하여 해를 구하고 다음 식을 만족하는지 확인함

$$u_i s_i = 0;\; i = 1\, to\, m,\;\; \zeta_i d_i = 0;\; i = 1\, to\, n$$

: 이 식은 비선형 방정식으로서 해를 구한 후 이를 만족하는지 평가하기 위한 용도로 사용됨

2차 계획 문제의 풀이 방법

- Kuhn-Tucker 필요 조건을 만족하는 최적해를 직접 구하는 방법
  - $u_i s_i = 0$, $\zeta_i x_i = 0$와 같은 조건을 이용하여 가능한 경우를 파악하고, 각 경우에 대해 해를 구해 최종적으로 Kuhn-Tucker 필요 조건을 만족하는 후보 최적점을 구하는 방법
  - 이 방법을 이용할 경우 사람은 비교적 간단하게 해를 구할 수 있음

- Kuhn-Tucker 필요 조건을 선형 계획 문제로 변환 후 Simplex 방법을 이용하여 최적해를 구하는 방법
  - Kunn-Tucker 필요 조건 중 선형의 식들만을 이용해 선형 계획 문제로 변환한 후 Simplex 방법을 이용해 해를 구하는 방법
  - 해를 구한 후 이들이 Kuhn-Tucker 필요 조건 중 비선형의 식($u_i s_i = 0$, $\zeta_i x_i = 0$)을 만족하는지 평가함
  - 이 방법은 체계적으로 알고리즘화 된 방법이므로 컴퓨터로 구현할 때 유리함

2차 계획 문제 예제
- 부등호 제약 조건이 있는 경우 K-T 필요 조건을 이용한 풀이(1)

Minimize $f(\mathbf{x}) = x_1^2 + x_2^2 - 2x_1 - 2x_2 + 2$

Subject to $g_1(\mathbf{x}) = -2x_1 - x_2 + 4 \le 0$

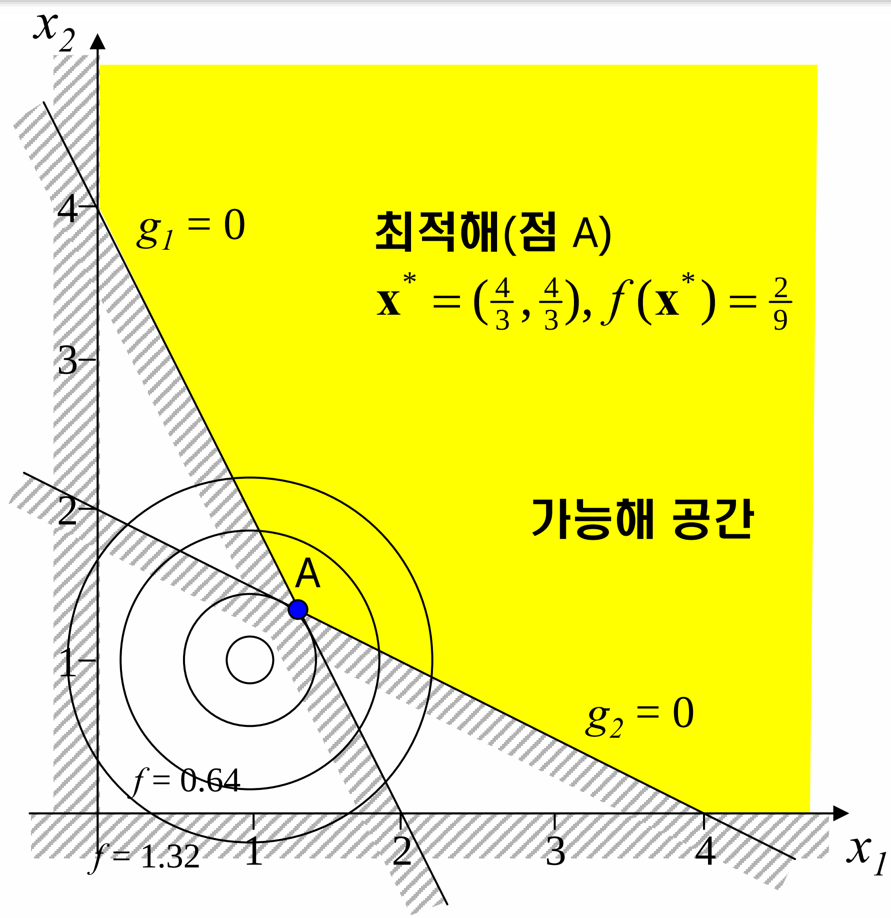
$g_2(\mathbf{x}) = -x_1 - 2x_2 + 4 \le 0$

단, $x_1 \ge 0, x_2 \ge 0$

최적해는 $\mathbf{x}^* = (\frac{4}{3}, \frac{4}{3}), f(\mathbf{x}^*) = \frac{2}{9}$

<u>Lagrange 함수</u>

$$L(\mathbf{x},\mathbf{u},\mathbf{s},\zeta) = x_1^2 + x_2^2 - 2x_1 - 2x_2 + 2$$
$$+ u_1(-2x_1 - x_2 + 4 + s_1^2)$$
$$+ u_2(-x_1 - 2x_2 + 4 + s_2^2)$$
$$- \zeta_1 x_1 - \zeta_2 x_2$$



* Simplex 방법을 이용할 경우, 완화 변수는 제곱 형태를 사용하지 않음

2차 계획 문제 예제
- 부등호 제약 조건이 있는 경우 K-T 필요 조건을 이용한 풀이(2)

Lagrange 함수

$$L(\mathbf{x},\mathbf{u},\mathbf{s},\zeta) = x_1^2 + x_2^2 - 2x_1 - 2x_2 + 2$$
$$+ u_1(-2x_1 - x_2 + 4 + s_1^2)$$
$$+ u_2(-x_1 - 2x_2 + 4 + s_2^2)$$
$$- \zeta_1 x_1 - \zeta_2 x_2$$

⬅

$$f(\mathbf{x}) = x_1^2 + x_2^2 - 2x_1 - 2x_2 + 2$$
$$g_1(\mathbf{x}) = -2x_1 - x_2 + 4 \le 0$$
$$g_2(\mathbf{x}) = -x_1 - 2x_2 + 4 \le 0$$
$$x_1 \ge 0, x_2 \ge 0$$

Kuhn-Tucker 필요 조건: $\nabla L(\mathbf{x},\mathbf{u},\mathbf{s},\zeta) = \mathbf{0}$

$$\frac{\partial L}{\partial x_1} = 2x_1 - 2 - 2u_1 - u_2 - \zeta_1 = 0$$

$$\frac{\partial L}{\partial x_2} = 2x_2 - 2 - u_1 - 2u_2 - \zeta_2 = 0$$

$$\frac{\partial L}{\partial u_1} = -2x_1 - x_2 + 4 + s_1^2 = 0$$

$$\frac{\partial L}{\partial u_2} = -x_1 - 2x_2 + 4 + s_2^2 = 0$$

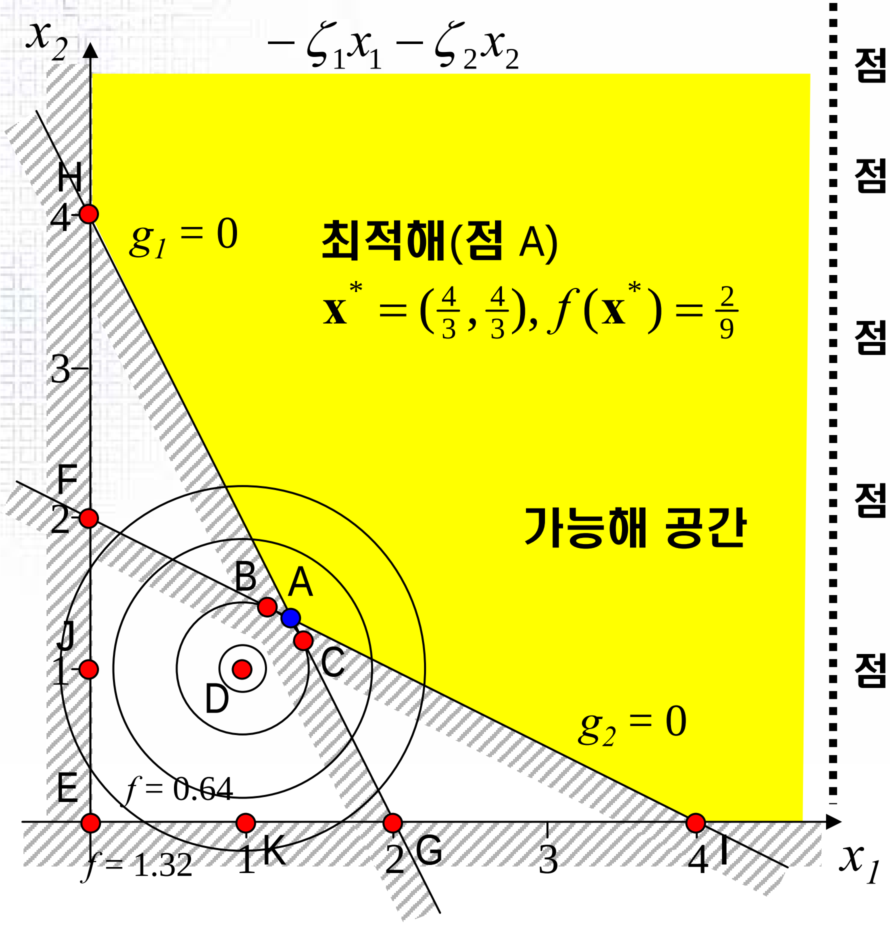
$$\frac{\partial L}{\partial s_1} = 2u_1 s_1 = 0 \qquad \frac{\partial L}{\partial s_2} = 2u_2 s_2 = 0$$

$$\zeta_i x_i = 0,\ i = 1,2 \qquad u_i, \zeta_i \ge 0,\ i = 1,2$$

2차 계획 문제 예제
- 부등호 제약 조건이 있는 경우 K-T 필요 조건을 이용한 풀이(3)

Lagrange 함수

$$L(\mathbf{x},\mathbf{u},\mathbf{s},\zeta) = x_1^2 + x_2^2 - 2x_1 - 2x_2 + 2$$
$$+ u_1(-2x_1 - x_2 + 4 + s_1^2)$$
$$+ u_2(-x_1 - 2x_2 + 4 + s_2^2)$$
$$- \zeta_1 x_1 - \zeta_2 x_2$$



점 A: $s_1=s_2=\zeta_1=\zeta_2=0$일 때(최적해)
$x_1 = x_2 = \frac{4}{3}, u_1 = u_2 = \frac{2}{9}$

점 B: $u_1=s_2=\zeta_1=\zeta_2=0$일 때
$x_1 = \frac{6}{5}, x_2 = \frac{7}{5}, u_2 = \frac{2}{5}, s_1^2 = -\frac{1}{5}$ 음수여서 안됨

점 C: $u_2=s_1=\zeta_1=\zeta_2=0$일 때
$x_1 = \frac{7}{5}, x_2 = \frac{6}{5}, u_1 = \frac{2}{5}, s_2^2 = -\frac{1}{5}$ 음수여서 안됨

점 D: $u_1=u_2=\zeta_1=\zeta_2=0$일 때
$x_1 = x_2 = 1, s_1^2 = s_2^2 = -1$ 음수여서 안됨

점 E: $u_1=u_2=x_1=x_2=0$일 때
$x_1 = x_2 = 0, s_1^2 = s_2^2 = -4,$ 음수여서 안됨
$\zeta_1 = \zeta_2 = -2$

점 E: $u_1=s_2=x_1=x_2=0$일 때
$x_1 = x_2 = 0, s_1^2 = -4,$ 음수여서 안됨
$-x_1 - 2x_2 + 4 + s_2^2 \neq 0$ 제약 조건 위배

점 E: $u_2=s_1=x_1=x_2=0$일 때
$x_1 = x_2 = 0, s_2^2 = -4,$ 음수여서 안됨
$-2x_1 - x_2 + 4 + s_1^2 \neq 0$ 제약 조건 위배

점 E: $s_1=s_2=x_1=x_2=0$일 때
$x_1 = x_2 = 0, -2x_1 - x_2 + 4 + s_1^2 \neq 0,$
$-x_1 - 2x_2 + 4 + s_2^2 \neq 0$ 제약 조건 위배

점 F: $u_1=s_2=\zeta_2=x_1=0$일 때
$x_1 = 0, x_2 = 2, u_2 = 1,$
$s_1^2 = -2, \zeta_1 = -3$ 음수여서 안됨

점 G: $u_2=s_1=\zeta_1=x_2=0$일 때
$x_1 = 2, x_2 = 0, u_1 = 1, s_2^2 = -2,$
$\zeta_2 = -3$ 음수여서 안됨

점 G: $s_1=s_2=\zeta_1=x_2=0$일 때
$x_1 = 2, x_2 = 0,$ 제약 조건 위배
$-x_1 - 2x_2 + 4 + s_2^2 \neq 0$

점 H: $u_2=s_1=\zeta_2=x_1=0$일 때
$x_1 = 0, x_2 = 4, u_1 = 6,$
$s_2^2 = 4, \zeta_1 = -14$ 음수여서 안됨

점 H: $s_1=s_2=\zeta_2=x_1=0$일 때
$x_1 = 0, x_2 = 4,$ 제약 조건 위배
$-x_1 - 2x_2 + 4 + s_2^2 \neq 0$

점 I: $u_1=s_2=\zeta_1=x_2=0$일 때
$x_1 = 4, x_2 = 0, u_2 = 6,$
$s_1^2 = 4, \zeta_2 = -14$ 음수여서 안됨

점 J: $u_1=u_2=\zeta_2=x_1=0$일 때
$x_1 = 0, x_2 = 1, s_1^2 = -3,$
$s_2^2 = -2, \zeta_1 = -2$ 음수여서 안됨

점 K: $u_1=u_2=\zeta_1=x_2=0$일 때
$x_1 = 1, x_2 = 0, s_1^2 = -2,$
$s_2^2 = -3, \zeta_2 = -2$ 음수여서 안됨

2차 계획 문제 예제
- 부등호 제약 조건이 있는 경우 Simplex 방법을 이용한 풀이(1)

Minimize $f(\mathbf{x}) = x_1^2 + x_2^2 - 2x_1 - 2x_2 + 2$

Subject to $g_1(\mathbf{x}) = -2x_1 - x_2 + 4 \le 0$

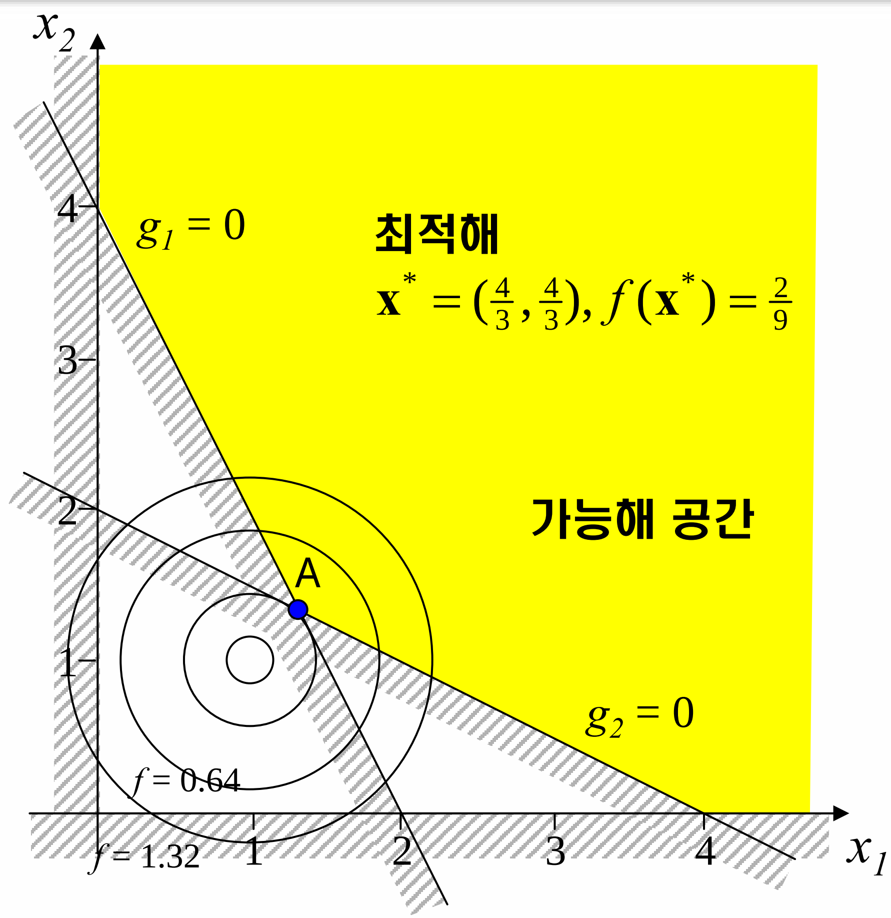
$g_2(\mathbf{x}) = -x_1 - 2x_2 + 4 \le 0$

단, $x_1 \ge 0, x_2 \ge 0$

최적해는 $\mathbf{x}^* = (\frac{4}{3}, \frac{4}{3}), f(\mathbf{x}^*) = \frac{2}{9}$

이를 다시 정리하면

$$f(\mathbf{x}) - 2 = \bar{f}(\mathbf{x}) = x_1^2 + x_2^2 - 2x_1 - 2x_2$$



2차 계획 문제 예제
- 부등호 제약 조건이 있는 경우 Simplex 방법을 이용한 풀이(2)

Minimize $\bar{f}(\mathbf{x}) = x_1^2 + x_2^2 - 2x_1 - 2x_2 = -2x_1 - 2x_2 + x_1^2 + x_2^2$

$$= \begin{bmatrix} -2 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \frac{1}{2} \begin{bmatrix} x_1 & x_1 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Subject to $-2x_1 - x_2 \le -4$

$-x_1 - 2x_2 \le -4$

$x_1 \ge 0, x_2 \ge 0$

$\Big\downarrow$ Matrix 형태로 표현

Minimize $\bar{f} = \mathbf{c}^T{}_{(1\times2)} \mathbf{d}_{(2\times1)} + \frac{1}{2} \mathbf{d}^T{}_{(1\times2)} \mathbf{H}_{(2\times2)} \mathbf{d}_{(2\times1)}$

Subject to $\mathbf{A}^T{}_{(2\times2)} \mathbf{d}_{(2\times1)} \le \mathbf{b}_{(2\times1)}$

$\mathbf{d}_{(2\times1)} \ge 0$

여기서, $\mathbf{d}_{(2\times1)} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \mathbf{c}_{(2\times1)} = \begin{bmatrix} -2 \\ -2 \end{bmatrix}, \mathbf{H}_{(2\times2)} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix},$

$$\mathbf{A}_{(2\times2)} = \begin{bmatrix} -2 & -1 \\ -1 & -2 \end{bmatrix}, \mathbf{b}_{(2\times1)} = \begin{bmatrix} -4 \\ -4 \end{bmatrix}$$

2차 계획 문제 예제
- 부등호 제약 조건이 있는 경우 Simplex 방법을 이용한 풀이(3)

Minimize $\bar{f}(\mathbf{x}) = -2x_1 - 2x_2 + x_1^2 + x_2^2$

Subject to $-2x_1 - x_2 \le -4$

$-x_1 - 2x_2 \le -4$

$x_1 \ge 0, x_2 \ge 0$

<u>Lagrange 함수</u>

$$L(\mathbf{x},\mathbf{u},\mathbf{s},\zeta) = -2x_1 - 2x_2 + x_1^2 + x_2^2$$
$$+u_1(-2x_1 - x_2 + 4 + s_1) + u_2(-x_1 - 2x_2 + 4 + s_2)$$
$$-\zeta_1 x_1 - \zeta_2 x_2$$

단, $u_i, s_i, \zeta_i \ge 0$

$\Big\downarrow$ Matrix 형태로 표현

* Simplex 방법을 이용할 경우,
완화 변수는 제곱 형태를 사용하지 않음

$$L = \begin{bmatrix} -2 & -2 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \frac{1}{2}\begin{bmatrix} x_1 & x_2 \end{bmatrix}\begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$+\begin{bmatrix} u_1 & u_2 \end{bmatrix}\left(\begin{bmatrix} -2 & -1 \\ -1 & -2 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} - \begin{bmatrix} -4 \\ -4 \end{bmatrix}\right) - \begin{bmatrix} \zeta_1 & \zeta_2 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$= \mathbf{c}^T{}_{(1\times2)}\mathbf{d}_{(2\times1)} + \frac{1}{2}\mathbf{d}^T{}_{(1\times2)}\mathbf{H}_{(2\times2)}\mathbf{d}_{(2\times1)} + \mathbf{u}^T{}_{(1\times2)}(\mathbf{A}^T{}_{(2\times2)}\mathbf{d}_{(2\times1)} + \mathbf{s}_{(2\times1)} - \mathbf{b}_{(2\times1)}) - \zeta^T{}_{(1\times2)}\mathbf{d}_{(2\times1)}$$

여기서, $\mathbf{d}_{(2\times1)} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \mathbf{u}_{(2\times1)} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, \mathbf{s}_{(2\times1)} = \begin{bmatrix} s_1 \\ s_2 \end{bmatrix}, \zeta_{(2\times1)} = \begin{bmatrix} \zeta_1 \\ \zeta_2 \end{bmatrix}$ 단, $u_i, s_i, \zeta_i \ge 0$

2차 계획 문제 예제
- 부등호 제약 조건이 있는 경우 Simplex 방법을 이용한 풀이(4)

$$L = \mathbf{c}^T{}_{(1\times2)}\mathbf{d}_{(2\times1)} + \frac{1}{2}\mathbf{d}^T{}_{(1\times2)}\mathbf{H}_{(2\times2)}\mathbf{d}_{(2\times1)} + \mathbf{u}^T{}_{(1\times2)}(\mathbf{A}^T{}_{(2\times2)}\mathbf{d}_{(2\times1)} + \mathbf{s}_{(2\times1)} - \mathbf{b}_{(2\times1)}) - \zeta^T{}_{(1\times2)}\mathbf{d}_{(2\times1)}$$

Kuhn-Tucker 필요 조건: $\nabla L(\mathbf{d},\mathbf{u},\mathbf{s},\zeta) = \mathbf{0}$

$$\frac{\partial L}{\partial \mathbf{d}_{(2\times1)}} = \begin{bmatrix} -2+2x_1-2u_1-u_2-\zeta_1 \\ -2+2x_2-u_1-2u_2-\zeta_2 \end{bmatrix} = \begin{bmatrix} -2 \\ -2 \end{bmatrix} + \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} -2 & -1 \\ -1 & -2 \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} - \begin{bmatrix} \zeta_1 \\ \zeta_2 \end{bmatrix}$$

$$= \mathbf{c}_{(2\times1)} + \mathbf{H}_{(2\times2)}\mathbf{d}_{(2\times1)} + \mathbf{A}_{(2\times2)}\mathbf{u}_{(2\times1)} - \zeta_{(2\times1)} = \mathbf{0}$$

$$\frac{\partial L}{\partial \mathbf{u}_{(2\times1)}} = \begin{bmatrix} -2x_1-x_2+4+s_1 \\ -x_1-2x_2+4+s_2 \end{bmatrix} = \begin{bmatrix} -2 & -1 \\ -1 & -2 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} - \begin{bmatrix} -4 \\ -4 \end{bmatrix}$$

$$= \mathbf{A}^T{}_{(2\times2)}\mathbf{d}_{(2\times1)} + \mathbf{s}_{(2\times1)} - \mathbf{b}_{(2\times1)} = \mathbf{0}$$

$u_i s_i = 0; i = 1\ to\ 2$

$\zeta_i x_i = 0; i = 1\ to\ 2$

단, $u_i, s_i, \zeta_i \geq 0; i = 1\ to\ 2$

Matrix 형태로 표현

$$\underbrace{\begin{bmatrix} \mathbf{H}_{(2\times2)} & \mathbf{A}_{(2\times2)} & -\mathbf{I}_{(2\times2)} & \mathbf{0}_{(2\times2)} \\ \mathbf{A}^T{}_{(2\times2)} & \mathbf{0}_{(2\times2)} & \mathbf{0}_{(2\times2)} & \mathbf{I}_{(2\times2)} \end{bmatrix}}_{=\mathbf{B}_{(4\times8)}} \underbrace{\begin{bmatrix} \mathbf{d}_{(2\times1)} \\ \mathbf{u}_{(2\times1)} \\ \zeta_{(2\times1)} \\ \mathbf{s}_{(2\times1)} \end{bmatrix}}_{=\mathbf{X}_{(8\times1)}} = \underbrace{\begin{bmatrix} -\mathbf{c}_{(2\times1)} \\ \mathbf{b}_{(2\times1)} \end{bmatrix}}_{=\mathbf{D}_{(4\times1)}}$$

2차 계획 문제 예제
- 부등호 제약 조건이 있는 경우 Simplex 방법을 이용한 풀이(5)

Kuhn-Tucker 필요 조건: $\nabla L(\mathbf{d},\mathbf{u},\mathbf{s},\zeta)=\mathbf{0}$

$$\underbrace{\left[\begin{array}{c:c:c:c} \mathbf{H}_{(2\times2)} & \mathbf{A}_{(2\times2)} & -\mathbf{I}_{(2\times2)} & \mathbf{0}_{(2\times2)} \\ \hdashline \mathbf{A}^T{}_{(2\times2)} & \mathbf{0}_{(2\times2)} & \mathbf{0}_{(2\times2)} & \mathbf{I}_{(2\times2)} \end{array}\right]}_{=\mathbf{B}_{(4\times8)}} \underbrace{\begin{bmatrix} \mathbf{d}_{(2\times1)} \\ \mathbf{u}_{(2\times1)} \\ \zeta_{(2\times1)} \\ \mathbf{s}_{(2\times1)} \end{bmatrix}}_{=\mathbf{X}_{(8\times1)}} = \underbrace{\begin{bmatrix} -\mathbf{c}_{(2\times1)} \\ \mathbf{b}_{(2\times1)} \end{bmatrix}}_{=\mathbf{D}_{(4\times1)}}$$

여기서, $\mathbf{d}_{(2\times1)}=\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \mathbf{c}_{(2\times1)}=\begin{bmatrix} -2 \\ -2 \end{bmatrix}, \mathbf{H}_{(2\times2)}=\begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}, \mathbf{A}_{(2\times2)}=\begin{bmatrix} -2 & -1 \\ -1 & -2 \end{bmatrix}, \mathbf{b}_{(2\times1)}=\begin{bmatrix} -4 \\ -4 \end{bmatrix}$

$$\mathbf{B}_{(4\times8)}=\left[\begin{array}{cc:cc:cc:cc} 2 & 0 & -2 & -1 & -1 & 0 & 0 & 0 \\ 0 & 2 & -1 & -2 & 0 & -1 & 0 & 0 \\ \hdashline -2 & -1 & 0 & 0 & 0 & 0 & 1 & 0 \\ -1 & -2 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right]$$

$$\mathbf{X}^T{}_{(1\times8)}=\begin{bmatrix} x_1 & x_2 & u_1 & u_2 & \zeta_1 & \zeta_2 & s_1 & s_2 \end{bmatrix}, \mathbf{D}^T{}_{(1\times4)}=\begin{bmatrix} 2 & 2 & -4 & -4 \end{bmatrix}$$

2차 계획 문제 예제
- 부등호 제약 조건이 있는 경우 Simplex 방법을 이용한 풀이(6)

Kuhn-Tucker 필요 조건(행렬식 표현)

$$\mathbf{B}_{(4\times 8)}\mathbf{X}_{(8\times 1)} = \underline{\mathbf{D}_{(4\times 1)}}$$

모든 요소가 음이 아니어야 함

$$\begin{bmatrix} 2 & 0 & -2 & -1 & -1 & 0 & 0 & 0 \\ 0 & 2 & -1 & -2 & 0 & -1 & 0 & 0 \\ -2 & -1 & 0 & 0 & 0 & 0 & 1 & 0 \\ -1 & -2 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ u_1 \\ u_2 \\ \zeta_1 \\ \zeta_2 \\ s_1 \\ s_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \\ -4 \\ -4 \end{bmatrix} \Rightarrow \begin{bmatrix} 2 & 0 & -2 & -1 & -1 & 0 & 0 & 0 \\ 0 & 2 & -1 & -2 & 0 & -1 & 0 & 0 \\ 2 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 1 & 2 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ u_1 \\ u_2 \\ \zeta_1 \\ \zeta_2 \\ s_1 \\ s_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \\ 4 \\ 4 \end{bmatrix}$$

양변에 -1을 곱함

구하려는 값

- ➡ **X를 구하는 위 문제는 등호 제약 조건만으로 이루어진 선형 계획 문제임**
- ➡ $u_i s_i = 0, \zeta_i x_i = 0; i = 1\ to\ 2$ **조건(비선형 방정식)은 해를 구한 후 이를 만족하는지 평가하기 위한 용도로 사용됨**

2차 계획 문제 예제
- 부등호 제약 조건이 있는 경우 Simplex 방법을 이용한 풀이(7)

Simplex 방법을 이용한 QP 문제의 해법

1. Kuhn-Tucker 필요 조건의 해를 구하는 문제는 등호 제약 조건만으로 이루어진 부정 선형 연립 방정식의 해를 구하는 문제(선형 계획 문제)임
2. 부정 선형 연립 방정식의 해를 구하기 위하여 Simplex 방법에서 인위 변수 및 인위 목적 함수를 도입하여 초기 기저 가능해를 구하는 방법임

$$\mathbf{B}_{(4\times8)}\mathbf{X}_{(8\times1)} + \underbrace{\mathbf{Y}_{(4\times1)}}_{\text{인위 변수}} = \mathbf{D}_{(4\times1)}$$

3. 인위 목적 함수는 다음과 같이 정의함

$$w = \sum_{i=1}^{4} Y_i = \sum_{i=1}^{4} D_i - \sum_{j=1}^{8}\sum_{i=1}^{4} B_{ij}X_j = w_0 + \sum_{j=1}^{8} C_j X_j$$

여기서, $C_j = -\sum_{i=1}^{4} B_{ij}$: 행렬 B의 j번째 열의 요소를 모두 더해 부호를 바꾼 것(상대 비용 계수)

$$w_0 = \sum_{i=1}^{4} D_i = 2+2+4+4 = 12$$

: 인위 목적 함수의 초기값으로 행렬 D의 모든 요소를 더한 것

4. Simplex 방법을 이용하여 해를 구하고 다음 식을 만족하는지 확인함

$$u_i s_i = 0;\ i = 1\ to\ 2,\ \ \zeta_i x_i = 0;\ i = 1\ to\ 2$$

: 이 식은 비선형 방정식으로서 해를 구한 후 이를 만족하는지 평가하기 위한 용도로 사용됨

2차 계획 문제 예제
- 부등호 제약 조건이 있는 경우 Simplex 방법을 이용한 풀이(8)

$$\mathbf{B}_{(4\times 8)}\mathbf{X}_{(8\times 1)} + \underbrace{\mathbf{Y}_{(4\times 1)}}_{\text{인위 변수}} = \mathbf{D}_{(4\times 1)}$$

$$\begin{bmatrix} 2 & 0 & -2 & -1 & -1 & 0 & 0 & 0 \\ 0 & 2 & -1 & -2 & 0 & -1 & 0 & 0 \\ 2 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 1 & 2 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} x_1(=X_1) \\ x_2(=X_2) \\ u_1(=X_3) \\ u_2(=X_4) \\ \zeta_1(=X_5) \\ \zeta_2(=X_6) \\ s_1(=X_7) \\ s_2(=X_8) \end{bmatrix} + \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \\ 4 \\ 4 \end{bmatrix}$$

1

	X1	X2	X3	X4	X5	X6	X7	X8	Y1	Y2	Y3	Y4	bi	bi/ai
Y1	**2**	0	-2	-1	-1	0	0	0	1	0	0	0	2	**1**
Y2	0	2	-1	-2	0	-1	0	0	0	1	0	0	2	-
Y3	2	1	0	0	0	0	-1	0	0	0	1	0	4	2
Y4	1	2	0	0	0	0	0	-1	0	0	0	1	4	4
A. Obj.	**-5**	-5	3	3	1	1	1	1	0	0	0	0	w-12	-

인위 목적 함수 식 **각 열의 요소를 모두 더해 부호를 바꾼 것(예, 1열**: -(2+0+2+1)=-5)

2차 계획 문제 예제
- 부등호 제약 조건이 있는 경우 Simplex 방법을 이용한 풀이(9)

2

	X1	X2	X3	X4	X5	X6	X7	X8	Y1	Y2	Y3	Y4	bi	bi/ai
X1	1	0	-1	-1/2	-1/2	0	0	0	1/2	0	0	0	1	-
Y2	0	**2**	-1	-2	0	-1	0	0	0	1	0	0	2	**1**
Y3	0	1	2	1	1	0	-1	0	-1	0	1	0	2	2
Y4	0	2	1	1/2	1/2	0	0	-1	-1/2	0	0	1	3	3/2
A. Obj.	0	**-5**	-2	1/2	-3/2	1	1	1	5/2	0	0	0	w-7	-

3

	X1	X2	X3	X4	X5	X6	X7	X8	Y1	Y2	Y3	Y4	bi	bi/ai
X1	1	0	-1	-1/2	-1/2	0	0	0	1/2	0	0	0	1	-
X2	0	1	-1/2	-1	0	-1/2	0	0	0	1/2	0	0	1	-
Y3	0	0	**5/2**	2	1	1/2	-1	0	-1	-1/2	1	0	1	**2/5**
Y4	0	0	2	5/2	1/2	1	0	-1	-1/2	-1	0	1	1	1/2
A. Obj.	0	0	**-9/2**	-9/2	-3/2	-3/2	1	1	5/2	5/2	0	0	w-2	-

4

	X1	X2	X3	X4	X5	X6	X7	X8	Y1	Y2	Y3	Y4	bi	bi/ai
X1	1	0	0	3/10	-1/10	1/5	-2/5	0	1/10	-1/5	2/5	0	7/5	14/3
X2	0	1	0	-3/5	1/5	-2/5	-1/5	0	-1/5	2/5	1/5	0	6/5	-
X3	0	0	1	4/5	2/5	1/5	-2/5	0	-2/5	-1/5	2/5	0	2/5	1/2
Y4	0	0	0	**9/10**	-3/10	3/5	4/5	-1	3/10	-3/5	-4/5	1	1/5	**2/9**
A. Obj.	0	0	0	**-9/10**	3/10	-3/5	-4/5	1	7/10	8/5	9/5	0	w-1/5	-

2차 계획 문제 예제
- 부등호 제약 조건이 있는 경우 Simplex 방법을 이용한 풀이(10)

5

	X1	X2	X3	X4	X5	X6	X7	X8	Y1	Y2	Y3	Y4	bi	bi/ai
X1	1	0	0	0	0	0	-2/3	1/3	0	0	2/3	-1/3	4/3	-
X2	0	1	0	0	0	0	7/15	-2/3	2/5	0	-7/15	2/15	4/3	-
X3	0	0	1	0	2/3	-1/3	-10/9	8/9	-2/3	7/15	10/9	-8/45	2/9	-
X4	0	0	0	1	-1/3	2/3	8/9	-10/9	1/3	-2/3	-8/9	2/9	2/9	-
A. Obj.	0	0	0	0	0	0	0	0	1	1	1	1	**w-0**	-

인위 목적 함수가 0이므로 초기 기저 가능해가 구해졌음

$$\mathbf{X}^T{}_{(1\times 8)} = \begin{bmatrix} x_1 & x_2 & u_1 & u_2 & \zeta_1 & \zeta_2 & s_1 & s_2 \end{bmatrix}$$

이로부터 초기 기저 가능해 중의 하나는 X_1=X_2=4/3, X_3=X_4=2/9, X_5=X_6=X_7=X_8=0

한편, 이들은 제약 조건 $X_i X_{4+i} = 0; i = 1\, to\, 4, X_i \geq 0; i = 1\, to\, 8$ **을 모두 만족한다.**

따라서 주어진 문제의 최적해는 $x_1 = x_2 = \frac{4}{3}, u_1 = u_2 = \frac{2}{9}, \zeta_1 = \zeta_2 = s_1 = s_2 = 0$ **이며,**

이것은 Kuhn-Tucker 필요 조건을 이용, 해를 직접 구한 방법의 결과와 동일함을 알 수 있다.

2차 계획 문제 예제
- 부등호 제약 조건이 있는 경우 Simplex 방법을 이용한 풀이(11)

만약 첫 번째 Table에서 첫 번째 열이 아니라 목적 함수의 계수가 -5으로서 그 값이 동일한 두 번째 열을 Pivot 열로 선택한다면?

1

	X1	X2	X3	X4	X5	X6	X7	X8	Y1	Y2	Y3	Y4	bi	bi/ai
Y1	2	0	-2	-1	-1	0	0	0	1	0	0	0	2	-
Y2	0	2	-1	-2	0	-1	0	0	0	1	0	0	2	1
Y3	2	1	0	0	0	0	-1	0	0	0	1	0	4	4
Y4	1	2	0	0	0	0	0	-1	0	0	0	1	4	2
A. Obj.	-5	-5	3	3	1	1	1	1	0	0	0	0	w-12	-

2

	X1	X2	X3	X4	X5	X6	X7	X8	Y1	Y2	Y3	Y4	bi	bi/ai
Y1	2	0	-2	-1	-1	0	0	0	1	0	0	0	2	1
X2	0	1	-1/2	-1	0	-1/2	0	0	0	1/2	0	0	1	-
Y3	2	0	1/2	1	0	1/2	-1	0	0	-1/2	1	0	3	3/2
Y4	1	0	1	2	0	1	0	-1	0	-1	0	1	2	2
A. Obj.	-5	0	1/2	-2	1	-3/2	1	1	0	5/2	0	0	w-7	-

3

	X1	X2	X3	X4	X5	X6	X7	X8	Y1	Y2	Y3	Y4	bi	bi/ai
X1	1	0	-1	-1/2	-1/2	0	0	0	1/2	0	0	0	1	-
X2	0	1	-1/2	-1	0	-1/2	0	0	0	1/2	0	0	1	-
Y3	0	0	5/2	2	1	1/2	-1	0	-1	-1/2	1	0	1	2/5
Y4	0	0	2	5/2	1/2	1	0	-1	-1/2	-1	0	1	1	1/2
A. Obj.	0	0	-9/2	-9/2	-3/2	-3/2	1	1	5/2	5/2	0	0	w-2	-

2차 계획 문제 예제
- 부등호 제약 조건이 있는 경우 Simplex 방법을 이용한 풀이(12)

4

	X1	X2	X3	X4	X5	X6	X7	X8	Y1	Y2	Y3	Y4	bi	bi/ai
X1	1	0	0	3/10	-1/10	1/5	-2/5	0	1/10	-1/5	2/5	0	7/5	-
X2	0	1	0	-6/10	1/5	-2/5	-1/5	0	-1/5	2/5	1/5	0	6/5	-
X3	0	0	1	-4/5	2/5	1/5	-2/5	0	-2/5	-1/5	2/5	0	2/5	-
Y4	0	0	0	9/10	-3/10	3/5	4/5	-1	3/10	-3/5	-4/5	1	1/5	1/4
A. Obj.	0	0	0	-9/10	3/10	-3/5	-4/5	1	7/10	8/5	9/5	0	w-1/5	-

5

	X1	X2	X3	X4	X5	X6	X7	X8	Y1	Y2	Y3	Y4	bi	bi/ai
X1	1	0	0	3/4	-1/4	1/2	0	-1/2	-1/4	-1/2	0	1/2	3/2	-
X2	0	1	0	-3/8	1/8	-1/4	0	-1/4	-1/8	1/4	0	1/4	5/4	-
X3	0	0	1	5/4	1/4	1/2	0	-1/2	-1/4	-1/2	0	1/2	1/2	-
X7	0	0	0	9/8	-3/8	3/4	1	-5/4	3/8	-3/4	-1	5/4	1/4	-
A. Obj.	0	0	0	0	0	0	0	0	1	1	1	1	**w-0**	-

인위 목적 함수가 0이므로 초기 기저 가능해가 구해졌음

$$\mathbf{X}^T_{(1\times 8)} = [x_1 \quad x_2 \quad u_1 \quad u_2 \quad \zeta_1 \quad \zeta_2 \quad s_1 \quad s_2]$$

이로부터 또 다른 초기 기저 가능해는 X_1=3/2, X_2=5/4, X_3=1/2, X_4=X_5=X_6=0, $X7$=1/4, X_8=0

한편, 이들은 제약 조건 $X_i X_{4+i} = 0; i = 1\, to\, 4, X_i \geq 0; i = 1\, to\, 8$ **을 만족하지 않는다**($X_3 X_7 \neq 0$).

따라서 이들은 주어진 문제의 최적해가 될 수 없다.

➡ Pivot **과정 중 선택 가능한 열 또는 행 또는 "b_i/a_i"의 계수가 중복인 경우, 어떤 것을 선택하느냐에 따라 다른 초기 기저 가능해가 얻어질 수 있음**

➡ **비선형 방정식**(u_i*s_i=0)**을 만족하는 해가 나올 때까지 모든 경우를 확인해 봐야 함**

2차 계획 문제의 해결을 위한 QP Class의 구현 예

```
class QP
{
public:
	QP();
	virtual ~QP();

	Simplex BXYD;              // 선형 방정식 "BX + Y = D"를 해결하기 위한 Simplex
	double** m_pH;             // H를 나타내는 2차원 배열
	double** m_pA;             // A를 나타내는 2차원 배열
	double** m_pN;             // N을 나타내는 2차원 배열
	double* m_pD;              // 선형 방정식을 푼 결과 Search Direction을 저장한 변수
	double* m_pU;              // 선형 방정식을 푼 결과 부등호 제약 조건식에 대한 Lagrange Multipliers를 저장한 변수
	double* m_pXi;             // 선형 방정식을 푼 결과 설계 변수의 양수 조건식에 대한 Lagrange Multipliers를 저장한 변수
	double* m_pS;              // 선형 방정식을 푼 결과 부등호 제약 조건식에 대한 완화 변수를 저장한 변수
	double* m_pY;              // 선형 방정식을 푼 결과 등호 제약 조건식에 대한 Lagrange Multipliers를 저장한 변수
	double* m_pZ;              // 선형 방정식을 푼 결과 등호 제약 조건식에 대한 Lagrange Multipliers를 저장한 변수

	void ConstructSimplexTable();   // 선형 방정식 "BX + Y = D"에 해당하는 Simplex를 구성하는 함수
	int CheckEndCondition();        // QP 방법의 종료 조건을 판단(U*S = 0 & Xi*D = 0)하는 함수
	int Solve();                    // QP를 실행하는 함수
};
```

5.2 순차적 선형 계획법
(Sequential Linear Programming)

Advanced
Ship
Design
Automation
Laboratory

SLP(Sequential Linear Programming)

- 현재의 설계점에서 주어진 목적 함수와 제약 조건을 선형화하여 선형 계획 문제(LP problem)로 만든 후,
- 이를 풀어 설계 변수의 변화 정도를 얻어냄으로써 더 나은 설계점을 구하는 방법

$$\underline{\mathbf{x}^{(k+1)}} = \underline{\mathbf{x}^{(k)}} + \underline{\mathbf{d}^{(k)}}$$

$\mathbf{x}^{(k+1)}$: 개선된 설계점

$\mathbf{x}^{(k)}$: 현재의 설계점

$\mathbf{d}^{(k)}$: LP problem으로부터 구하는 설계 변수의 변화 정도

- 즉, 선형 계획 문제(Linear Programming) 문제를 연속적(Sequential)으로 풀어 최적해를 구하는 방법

제약 최적화 문제의 선형화
(Linear Programming Problem)

$$Minimize \quad f(\mathbf{x}^{(k)}+\Delta\mathbf{x}^{(k)}) \cong f(\mathbf{x}^{(k)})+\nabla f^T(\mathbf{x}^{(k)})\Delta\mathbf{x}^{(k)}$$

Taylor 급수의 1차항(선형항)만 고려한 목적 함수

$$Subject\ to \quad h_j(\mathbf{x}^{(k)}+\Delta\mathbf{x}^{(k)}) \cong h_j(\mathbf{x}^{(k)})+\nabla h_j^T(\mathbf{x}^{(k)})\Delta\mathbf{x}^{(k)} = 0;\ j=1\ to\ p$$

Taylor 급수의 1차항(선형항)만 고려한 등호 제약 조건

$$g_j(\mathbf{x}^{(k)}+\Delta\mathbf{x}^{(k)}) \cong g_j(\mathbf{x}^{(k)})+\nabla g_j^T(\mathbf{x}^{(k)})\Delta\mathbf{x}^{(k)} \le 0;\ j=1\ to\ m$$

Taylor 급수의 1차항(선형항)만 고려한 부등호 제약 조건

여기서, $\bar{f} = f(\mathbf{x}^{(k)}+\Delta\mathbf{x}^{(k)}) - f(\mathbf{x}^{(k)}),\ e_j = -h_j(\mathbf{x}^{(k)}),\ b_j = -g_j(\mathbf{x}^{(k)}),$

$c_i = \partial f(\mathbf{x}^{(k)})/\partial x_i,\ n_{ij} = \partial h_j(\mathbf{x}^{(k)})/\partial x_i,\ a_{ij} = \partial g_j(\mathbf{x}^{(k)})/\partial x_i,$

$d_i = \Delta x_i^{(k)}$ **라고 가정하면**

$$Minimize \quad \bar{f} = \sum_{i=1}^{n} c_i d_i$$

$$Subject\ to \quad \sum_{i=1}^{n} n_{ij} d_i = e_j;\ j=1\ to\ p$$

$$\sum_{i=1}^{n} a_{ij} d_i \le b_j;\ j=1\ to\ m$$

여기서, $d_{il} \le d_i \le d_{iu} (\Delta x_{il}^{(k)} \le \Delta x_i^{(k)} \le \Delta x_{iu}^{(k)})$

<u>Matrix form</u>

$$Minimize \quad \bar{f} = \mathbf{c}^T{}_{(1\times n)}\mathbf{d}_{(n\times 1)}$$: 선형화 된 목적 함수

$$Subject\ to \quad \mathbf{N}^T{}_{(p\times n)}\mathbf{d}_{(n\times 1)} = \mathbf{e}_{(p\times 1)}$$: 선형화 된 등호 제약 조건

$$\mathbf{A}^T{}_{(m\times n)}\mathbf{d}_{(n\times 1)} \le \mathbf{b}_{(m\times 1)}$$: 선형화 된 부등호 제약 조건

➡ **선형 계획 문제**(Linear Problem)

➡ Simplex **방법을 이용해 해결 가능**

순차적 선형 계획법(SLP; Sequential Linear Programming) 예제 - 부등호 제약 조건이 있는 경우 풀이 예(1)

Minimize $f(\mathbf{x}) = x_1^2 + x_2^2 - 3x_1x_2$

Subject to $g_1(\mathbf{x}) = \frac{1}{6}x_1^2 + \frac{1}{6}x_2^2 - 1.0 \le 0$

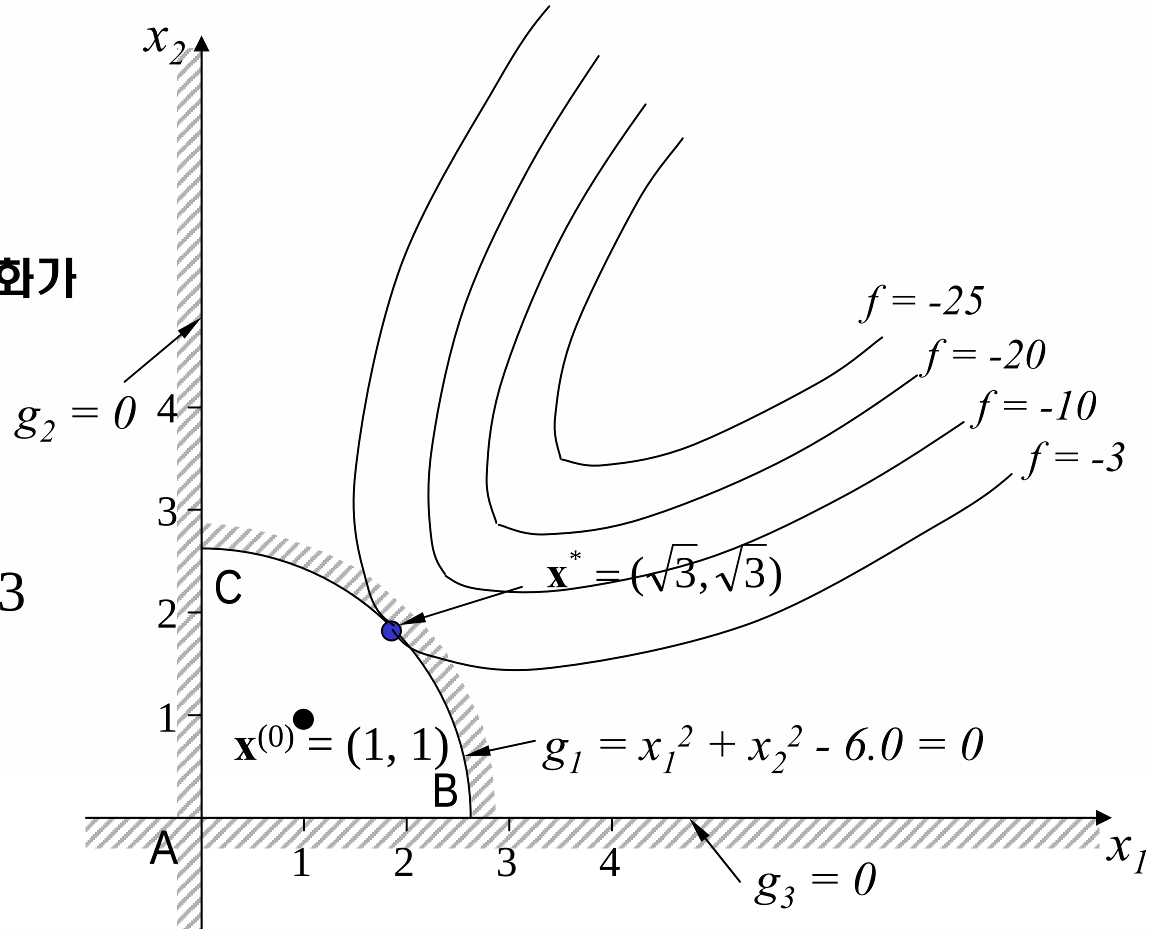
$g_2(\mathbf{x}) = -x_1 \le 0$

$g_3(\mathbf{x}) = -x_1 \le 0$

초기 시작점은 $\mathbf{x}^{(0)} = (1,1),$

$\varepsilon_1 = \varepsilon_2 = 0.001$ **이고**, **15%의 설계 변화가 허용된다고 가정**

최적해는 $\mathbf{x}^* = (\sqrt{3}, \sqrt{3}), f(\mathbf{x}^*) = -3$



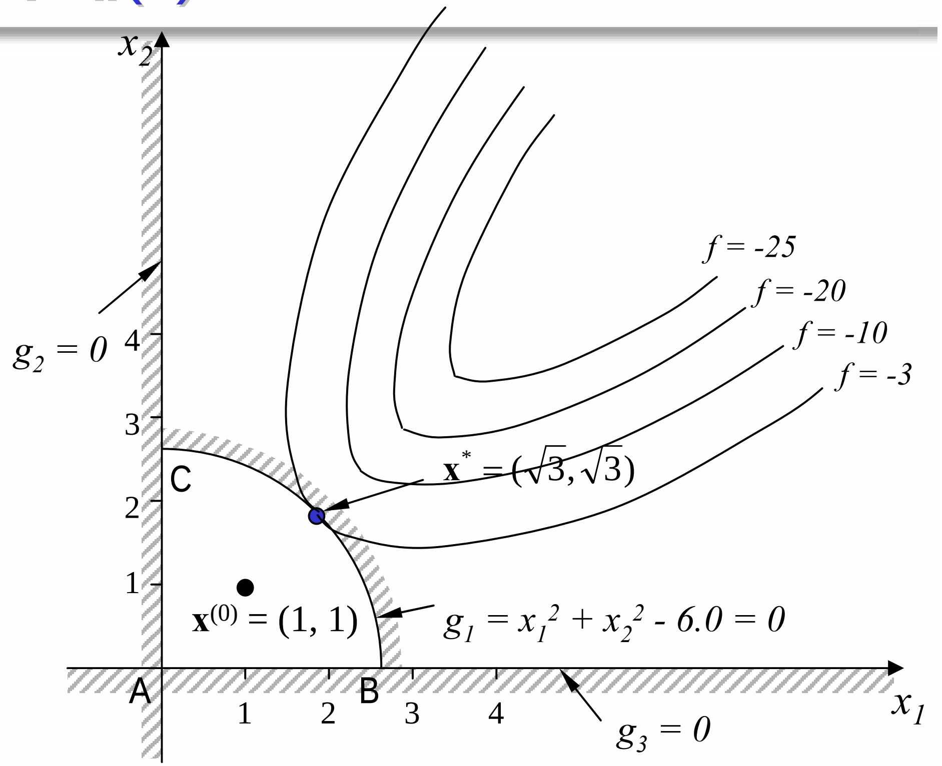
순차적 선형 계획법 예제
- 부등호 제약 조건이 있는 경우 풀이 예(2)

Minimize $f(\mathbf{x}) = x_1^2 + x_2^2 - 3x_1x_2$

Subject to $g_1(\mathbf{x}) = \frac{1}{6}x_1^2 + \frac{1}{6}x_2^2 - 1.0 \le 0$

$g_2(\mathbf{x}) = -x_1 \le 0$

$g_3(\mathbf{x}) = -x_1 \le 0$



(1) **반복 과정** 1(k = 0)

(i) **단계** 1

문제에서 주어진 초기 조건으로부터

$\mathbf{x}^{(0)} = (1,1), \varepsilon_1 = \varepsilon_2 = 0.001$

(ii) **단계** 2: **목적 함수와 제약 조건 함수의 값 및 경사도 계산**

$f(1,1) = -1$

$g_1(1,1) = -\frac{2}{3} < 0$ ➡ **제약 조건 만족**

$g_2(1,1) = -1 < 0$ ➡ **제약 조건 만족**

$g_3(1,1) = -1 < 0$ ➡ **제약 조건 만족**

$\nabla f = (-1,-1), \nabla g_1 = (\frac{1}{3}, \frac{1}{3}), \nabla g_2 = (-1,0), \nabla g_3 = (0,-1)$

순차적 선형 계획법 예제
- 부등호 제약 조건이 있는 경우 풀이 예(3)

(iii) 단계 3: LP 문제의 정의

Minimize $f(\mathbf{x}) = x_1^2 + x_2^2 - 3x_1x_2$

Subject to $g_1(\mathbf{x}) = \frac{1}{6}x_1^2 + \frac{1}{6}x_2^2 - 1.0 \le 0$

$g_2(\mathbf{x}) = -x_1 \le 0$

$g_3(\mathbf{x}) = -x_1 \le 0$

$$\boxed{\begin{aligned} & f(1,1) = -1,\ g_1(1,1) = -\tfrac{2}{3}, \\ & g_2(1,1) = -1,\ g_3(1,1) = -1 \\ & \nabla f = (-1,-1),\ \nabla g_1 = (\tfrac{1}{3},\tfrac{1}{3}), \\ & \nabla g_2 = (-1,0),\ \nabla g_3 = (0,-1) \end{aligned}}$$

Minimize $\bar{f} = -d_1 - d_2$

Subject to $\frac{1}{3}d_1 + \frac{1}{3}d_2 \le \frac{2}{3}$

$-d_1 \le 1$

$-d_2 \le 1$

$$\boxed{\begin{aligned} & -0.15 \le d_1 \le 0.15 \\ & -0.15 \le d_2 \le 0.15 \end{aligned}}$$

문제에서 주어진 설계 변수의 변화 범위(move limit)에 대한 제약 조건

순차적 선형 계획법 예제
- 부등호 제약 조건이 있는 경우 풀이 예(4)

(iv) 단계 4: LP 문제의 풀이를 통한 탐색 방향($\mathbf{d}^{(0)}$)의 결정

Minimize $\bar{f} = -d_1 - d_2$

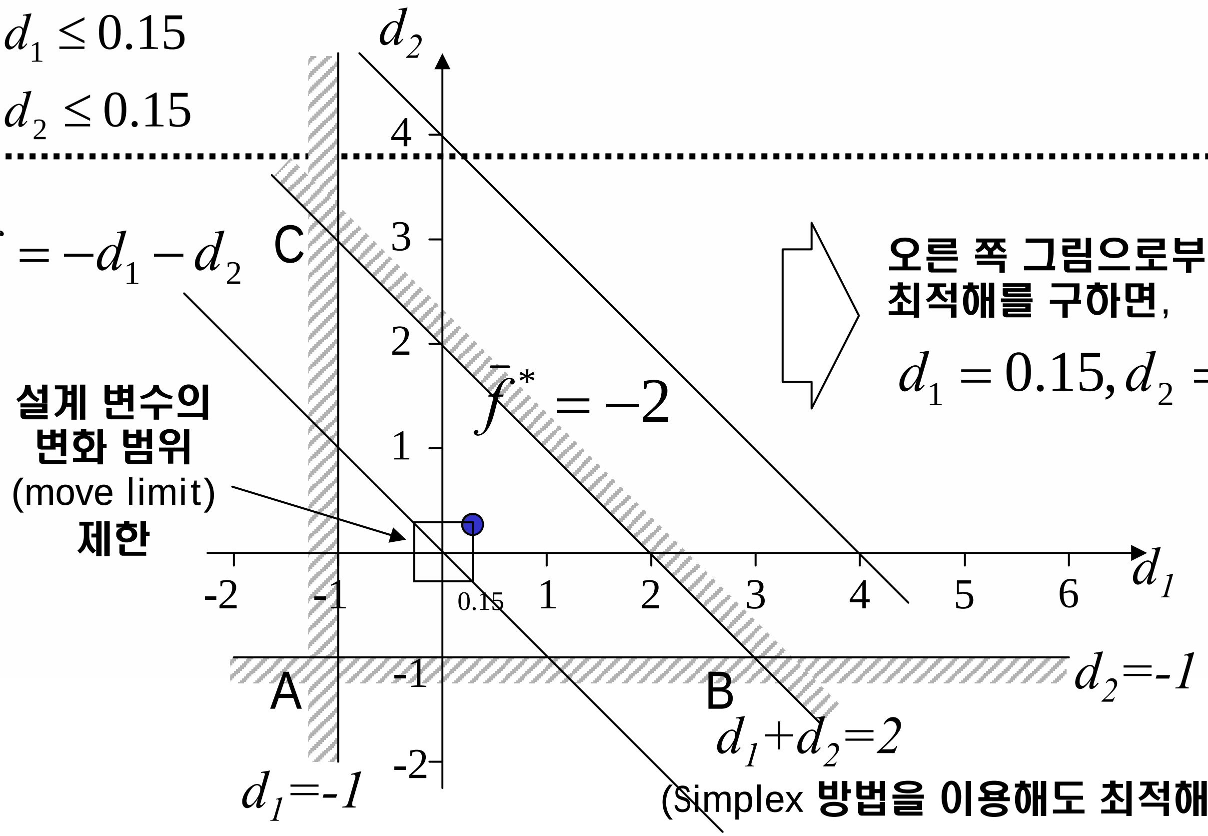
Subject to $\frac{1}{3}d_1 + \frac{1}{3}d_2 \le \frac{2}{3}$

$-d_1 \le 1$

$-d_2 \le 1$

$-0.15 \le d_1 \le 0.15$

$-0.15 \le d_2 \le 0.15$



오른 쪽 그림으로부터
최적해를 구하면,

$d_1 = 0.15, d_2 = 0.15$

(Simplex 방법을 이용해도 최적해를 구할 수 있음)

순차적 선형 계획법 예제
- 부등호 제약 조건이 있는 경우 풀이 예(5)

(v) 단계 5: 수렴 기준의 검토

$$\mathbf{d}^{(0)} = (d_1, d_2) = (0.15, 0.15)$$

$$\left\|\mathbf{d}^{(0)}\right\| = \sqrt{0.15^2 + 0.15^2} = 0.212 > \varepsilon_2 (= 0.001)$$ 이므로 수렴 기준을 만족하지 않음

(vi) 단계 6: 새로운 설계점의 결정 및 반복 횟수의 갱신

$$\mathbf{x}^{(1)} = \mathbf{x}^{(1,1)} = \mathbf{x}^{(0)} + \mathbf{d}^{(0)} = (1,1) + (0.15, 0.15) = (1.15, 1.15)$$

$$k = k + 1 = 1$$

아직 아래와 같은 수렴 기준을 만족하지 않으므로
위와 유사한 과정을 반복적으로 수행한다.

탐색 방향의 크기와 관련된 수렴 기준

$$\left\|\mathbf{d}^{(1)}\right\| = 0.212 > \varepsilon_2 (= 0.001)$$

SLP(Sequential Linear Programming) 알고리즘의 요약

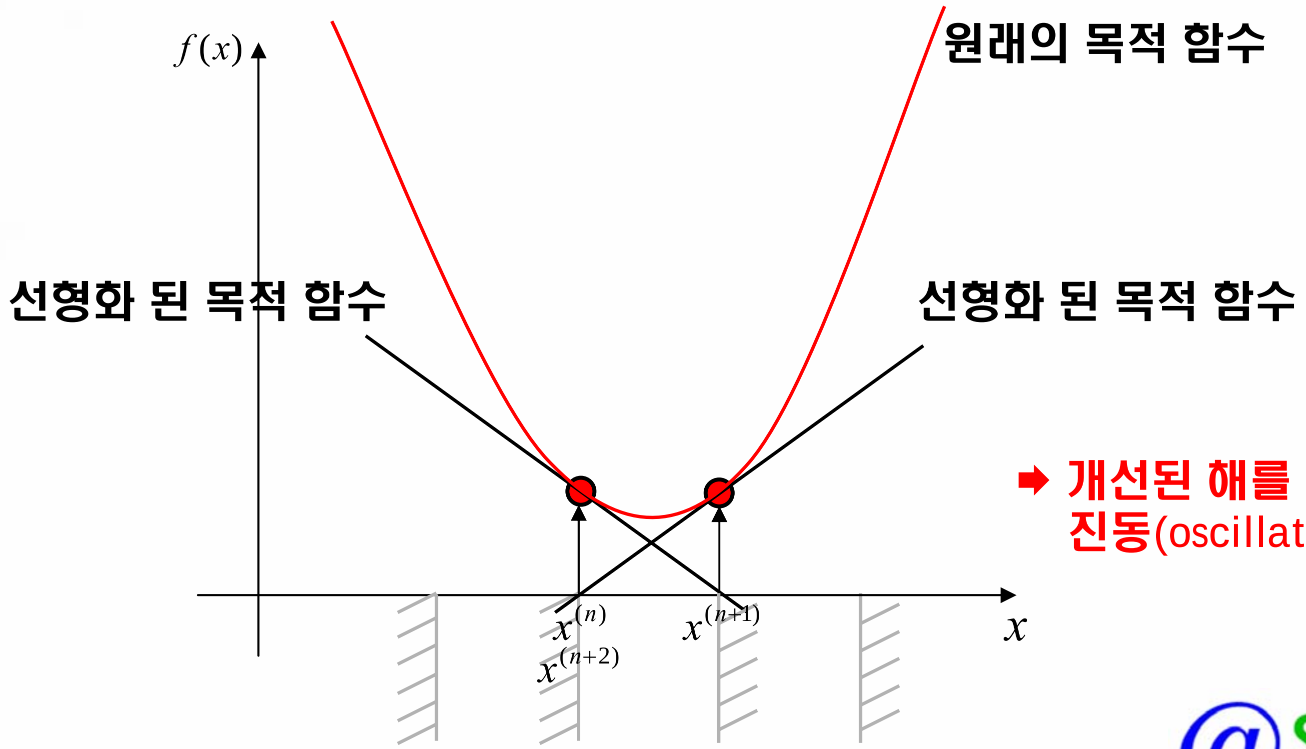
- 단계 1: k=0으로 둔다. $\mathbf{x}^{(0)}$으로 설계 변수의 초기값을 추정한다. 또한 제약 조건의 위배 정도와 수렴 기준으로 작은 수 ε_1, ε_2의 적절한 초기값을 선정한다.
- 단계 2: $\mathbf{x}^{(k)}$에서 목적 함수, 제약 조건과 이들의 경사도(gradient)를 계산한다.
- 단계 3: $\mathbf{x}^{(k)}$의 변화 범위(move limit) $\Delta x_{il}^{(k)}$, $\Delta x_{iu}^{(k)}$를 적절히 선정하고, 선형 계획 문제를 수학적으로 정의한다. 즉,

$$\Delta x_{il}^{(k)} \le \Delta x_{i}^{(k)} \le \Delta x_{iu}^{(k)}$$

- 단계 4: 앞서 정의된 선형 계획 문제를 Simplex 방법으로 풀어 $\mathbf{d}^{(k)}$를 구한다.
- 단계 5: 수렴 여부를 확인한다. 즉, $g_i \le \varepsilon_1 (i = 1 \text{ to } m)$, $|h_i| \le \varepsilon_1 (i = 1 \text{ to } p)$, 그리고 $\|\mathbf{d}^{(k)}\| \le \varepsilon_2$인지 확인하여 그렇다면 현재의 $\mathbf{x}^{(k)}$가 최적해라고 가정하고 종료한다. 그렇지 않다면 다음 단계로 간다.
- 단계 6: 새로운 설계 변수 $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \Delta\mathbf{x}^{(k)}$로, 반복 횟수 $k = k+1$로 수정하고 단계 2로 간다.

SLP 방법의 한계점

- 설계 변수의 변화 범위(move limit)을 사용자가 주어야 함
- 설계 변수의 변화 범위가 작을 경우 최적해를 찾는 데에 많은 시간이 소요됨
- 반면, 설계 변수의 변화 범위가 클 경우 최적해를 못 찾을 수도 있음



➡ 개선된 해를 찾지 못하고 진동(oscillation)함

5.3 Penalty Function 방법

Advanced
Ship
Design
Automation
Laboratory

제약 최적화 문제를 비제약 최적화 문제로 변환하는 방법(1)

제약 최적화 문제

Minimize $f(\mathbf{x})$

Subject to $\mathbf{h}(\mathbf{x}) = \mathbf{0}$ 등호 제약 조건

$\mathbf{g}(\mathbf{x}) \le \mathbf{0}$ 부등호 제약 조건

Lagrange 함수를 이용한 비제약 최적화 문제로의 변환

$$L(\mathbf{x}, \mathbf{v}, \mathbf{u}, \mathbf{s}) = f(\mathbf{x}) + \mathbf{v}^T \mathbf{h}(\mathbf{x}) + \mathbf{u}^T (\mathbf{g}(\mathbf{x}) + \mathbf{s}^2)$$

국부적 후보 최적성 조건인 ∇L=0으로부터 $\mathbf{u}$, $\mathbf{v}$를 계산해야 함

1) 현재의 설계점에서 제약 조건을 만족하는 경우

등호 제약 조건의 경우: $\mathbf{h}(\mathbf{x}) = \mathbf{0}$

부등호 제약 조건의 경우: $\mathbf{u} = \mathbf{0}$ (설계점이 제약 조건의 경계에 있지 않을 때)

$\mathbf{s} = \mathbf{0} \Rightarrow \mathbf{g}(\mathbf{x}) = \mathbf{0}$ (설계점이 제약 조건의 경계 상에 있을 때)

따라서 $L(\mathbf{x}, \mathbf{v}, \mathbf{u}, \mathbf{s}) = f(\mathbf{x}) + \mathbf{v}^T \mathbf{h}(\mathbf{x}) + \mathbf{u}^T (\mathbf{g}(\mathbf{x}) + \mathbf{s}^2) \Rightarrow f(\mathbf{x})$ ➡ 제약 조건을 만족할 때 Lagrange 함수가 원래의 목적 함수와 동일함

2) 현재의 설계점에서 제약 조건을 위배하는 경우

등호 제약 조건의 경우: $\mathbf{v}^T \mathbf{h}(\mathbf{x}) \neq \mathbf{0}$

부등호 제약 조건의 경우: $\mathbf{u}^T (\mathbf{g}(\mathbf{x}) + \mathbf{s}^2) > \mathbf{0}$

따라서 $L(\mathbf{x}, \mathbf{v}, \mathbf{u}, \mathbf{s}) = f(\mathbf{x}) + \underline{\mathbf{v}^T \mathbf{h}(\mathbf{x}) + \mathbf{u}^T (\mathbf{g}(\mathbf{x}) + \mathbf{s}^2)} > f(\mathbf{x})$

➡ 제약 조건을 위배할 때 원래의 목적 함수에 양의 벌칙 값을 더한 것

제약 최적화 문제를 비제약 최적화 문제로 변환하는 방법(2)

제약 최적화 문제

Minimize $f(\mathbf{x})$

Subject to $\mathbf{h}(\mathbf{x}) = \mathbf{0}$ 등호 제약 조건

$\mathbf{g}(\mathbf{x}) \le \mathbf{0}$ 부등호 제약 조건

Penalty 함수를 이용한 비제약 최적화 문제로의 변환

1968년 Fiacco와 McCormick에 의해 제안됨

$$\Phi(\mathbf{x}, \mathbf{r}_k) = f(\mathbf{x}) + \mathbf{r}_k^T \max\{\mathbf{g}(\mathbf{x}), \mathbf{0}\}$$

여기서, $\mathbf{r}_k$는 문제에서 주어지는 양의 상수로서
Iteration이 진행될수록 그 값이 커짐

1) 현재의 설계점에서 제약 조건을 만족하는 경우

$$\mathbf{g}(\mathbf{x}) \le \mathbf{0} \Rightarrow \max\{\mathbf{g}(\mathbf{x}), \mathbf{0}\} = \mathbf{0}$$

$$\Rightarrow \Phi(\mathbf{x}, \mathbf{r}_k) = f(\mathbf{x}) + \mathbf{r}_k^T \max\{\mathbf{g}(\mathbf{x}), \mathbf{0}\}$$

$$\Rightarrow f(\mathbf{x})$$

➡ 제약 조건을 만족할 때 Penalty 함수가 원래의 목적 함수와 동일함

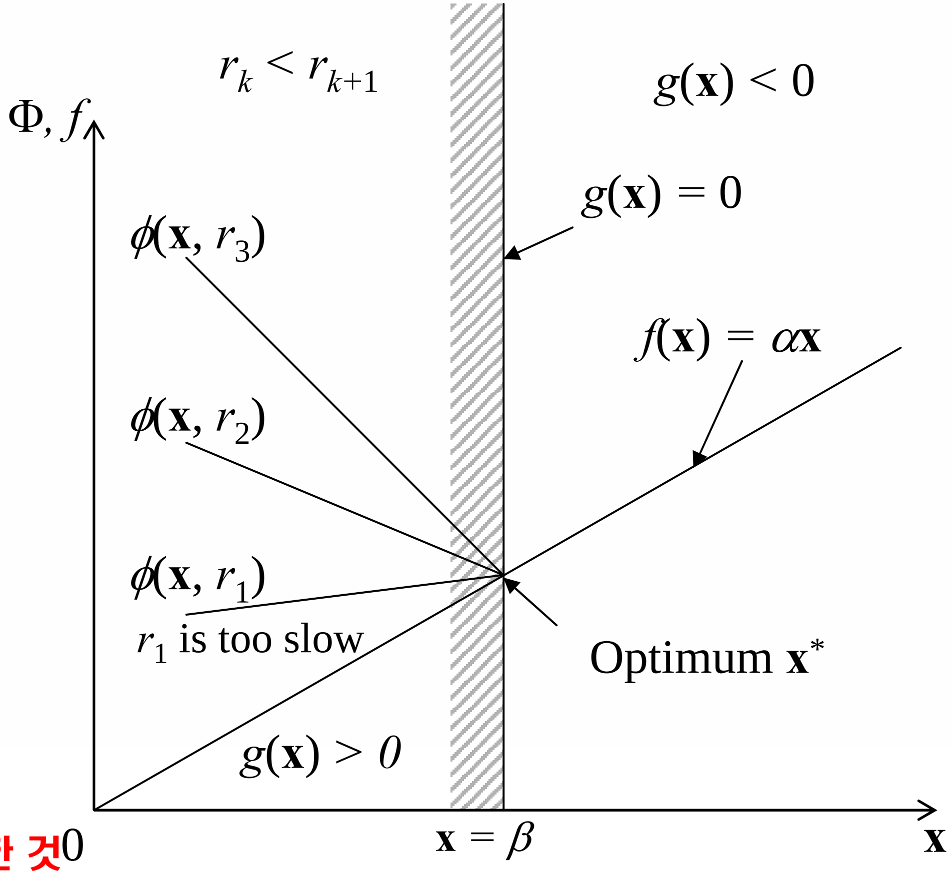
2) 현재의 설계점에서 제약 조건을 위배하는 경우

$$\mathbf{g}(\mathbf{x}) > \mathbf{0} \Rightarrow \max\{\mathbf{g}(\mathbf{x}), \mathbf{0}\} = \mathbf{g}(\mathbf{x})$$

$$\Rightarrow \Phi(\mathbf{x}, \mathbf{r}_k) = f(\mathbf{x}) + \mathbf{r}_k^T \max\{\mathbf{g}(\mathbf{x}), \mathbf{0}\}$$

$$= f(\mathbf{x}) + \mathbf{r}_k^T \mathbf{g}(\mathbf{x}) > f(\mathbf{x})$$

➡ 제약 조건을 위배할 때 원래의 목적 함수에 양의 벌칙 값을 더한 것



제약 최적화 문제를 비제약 최적화 문제로 변환하는 방법(3)

제약 최적화 문제

Minimize $f(\mathbf{x})$

Subject to $\mathbf{h}(\mathbf{x}) = \mathbf{0}$ 등호 제약 조건

$\mathbf{g}(\mathbf{x}) \le \mathbf{0}$ 부등호 제약 조건

Pshenichny의 강하 함수를 이용한 비제약 최적화 문제로의 변환

1978년 Pshenichny와 Danilin에 의해 제안됨

$\Phi(\mathbf{x}) = f(\mathbf{x}) + R \cdot V(\mathbf{x})$ 여기서, $R = \max\{R_0, |\mathbf{v}| + \mathbf{u}\}$: 벌칙 매개 변수로서 모든 Lagrange multiplier의 합

$V(\mathbf{x}) = \max\{0; |\mathbf{h}|; \mathbf{g}\}$: 현재의 설계점에서의 최대 제약 조건 위배 값

1) 현재의 설계점에서 제약 조건을 만족하는 경우

$V(\mathbf{x}) = 0 \Rightarrow R \cdot V(\mathbf{x}) = 0$

$\Rightarrow \Phi(\mathbf{x}) = f(\mathbf{x}) + R \cdot V(\mathbf{x}) \Rightarrow f(\mathbf{x})$ ➡ 제약 조건을 만족할 때 강하 함수가 원래의 목적 함수와 동일함

2) 현재의 설계점에서 제약 조건을 위배하는 경우

$R \cdot V(\mathbf{x}) > 0$

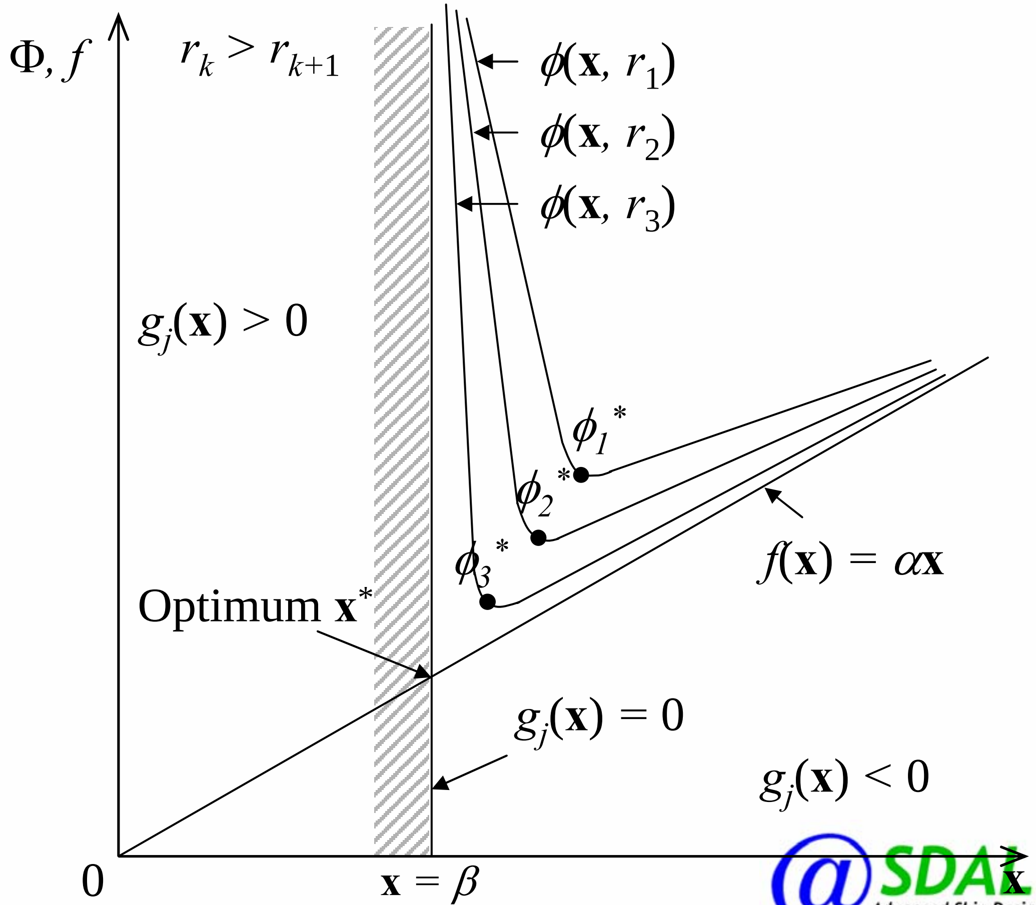
$\Rightarrow \Phi(\mathbf{x}) = f(\mathbf{x}) + R \cdot V(\mathbf{x}) > f(\mathbf{x})$ ➡ 제약 조건을 위배할 때 원래의 목적 함수에 양의 벌칙 값을 더한 것

- 제약 조건에 다가가게 되면 목적 함수에 Penalty 항 부과
- 초기 시작점이 반드시 가능해 영역(feasible region)에 있어야 함

$$\Phi(\mathbf{x}, r_k) = f(\mathbf{x}) - r_k \sum_{j=1}^{m} \frac{1}{g_j(\mathbf{x})}$$

⬇

$$\Phi(\mathbf{x}, r_k) = \alpha\mathbf{x} - r_k \sum_{j=1}^{m} \frac{1}{\beta - \mathbf{x}}$$



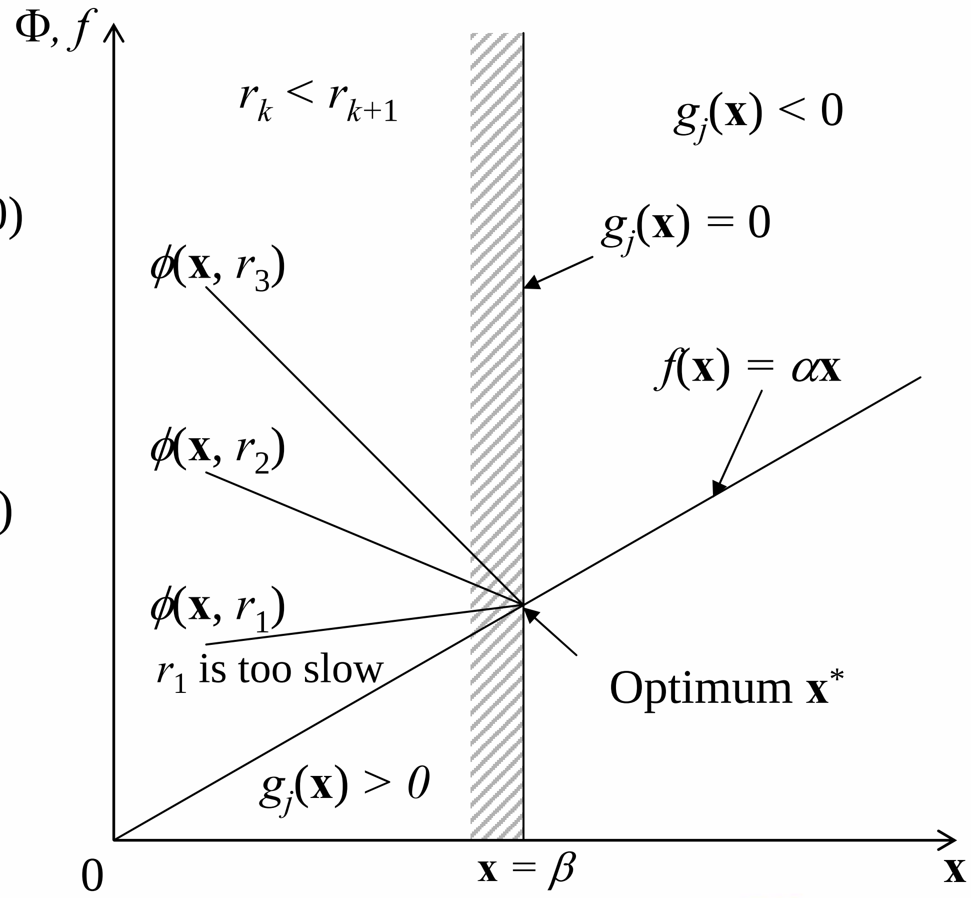
Penalty Function 방법
- External Penalty Function 방법

- 제약 조건이 위배되었을 때만 목적 함수에 Penalty 항 부과

$$\Phi(\mathbf{x},r_k) = f(\mathbf{x}) + r_k \sum_{j=1}^{m} \max(g_j(\mathbf{x}),0)$$

⬇

$$\Phi(\mathbf{x},r_k) = \alpha\mathbf{x} + r_k \max(\beta - \mathbf{x},0)$$



5.4 순차적 2차 계획법 (Sequential Quadratic Programming)

Advanced
Ship
Design
Automation
Laboratory

순차적 2차 계획법(SQP; Sequential Quadratic Programming)

- 현재의 설계점에서 주어진 목적 함수와 제약 조건을 2차 계획 문제(2차 함수 형태의 목적 함수와 선형 함수 형태의 제약 조건을 가진 문제)로 만든 후,
- 이를 해결하여 탐색 방향(search direction)을 구하고,
- 목적 함수를 최소화하는 최적의 이동 거리(step size)를 찾아 더 나은 설계점을 구하는 방법

$$\underbrace{\mathbf{x}^{(k+1)}}_{\text{개선된 설계점}} = \underbrace{\mathbf{x}^{(k)}}_{\text{현재의 설계점}} + \overbrace{\alpha_k}^{\text{탐색 방향으로 목적 함수를 최소화하는 최적의 이동 거리}} \underbrace{\mathbf{d}^{(k)}}_{\text{2차 계획 문제로부터 구한 탐색 방향}}$$

- 즉, 이차 계획 문제(Quadratic Programming) 문제를 연속적(Sequential)으로 풀어 최적해를 구하는 방법

비선형 최적화 문제를 2차 계획 문제(Quadratic Programming)로의 변환

비선형 최적화 문제

$$Minimize \quad f(\mathbf{x})$$

$$Subject\ to \quad h_j(\mathbf{x}) = 0;\ j = 1\ to\ p$$

$$g_j(\mathbf{x}) \le 0;\ j = 1\ to\ m$$

선형 계획(Linear Programming; LP) 문제

$$Minimize \quad f(\mathbf{x}+\Delta\mathbf{x}) \cong f(\mathbf{x}) + \nabla f^T(\mathbf{x})\Delta\mathbf{x}$$

Taylor 급수의 1차항(선형항)만 고려한 목적 함수

$$Subject\ to \quad h_j(\mathbf{x}+\Delta\mathbf{x}) \cong h_j(\mathbf{x}) + \nabla h_j^T(\mathbf{x})\Delta\mathbf{x} = 0;\ j = 1\ to\ p$$

Taylor 급수의 1차항(선형항)만 고려한 등호 제약 조건

$$g_j(\mathbf{x}+\Delta\mathbf{x}) \cong g_j(\mathbf{x}) + \nabla g_j^T(\mathbf{x})\Delta\mathbf{x} \le 0;\ j = 1\ to\ m$$

Taylor 급수의 1차항(선형항)만 고려한 부등호 제약 조건

2차 계획(Quadratic Programming; QP) 문제

$$Minimize \quad f(\mathbf{x}+\Delta\mathbf{x}) \cong f(\mathbf{x}) + \nabla f^T(\mathbf{x})\Delta\mathbf{x} + 0.5\Delta\mathbf{x}^T\mathbf{H}\Delta\mathbf{x}$$

Taylor 급수의 2차항까지 고려한 목적 함수

$$Subject\ to \quad h_j(\mathbf{x}+\Delta\mathbf{x}) \cong h_j(\mathbf{x}) + \nabla h_j^T(\mathbf{x})\Delta\mathbf{x} = 0;\ j = 1\ to\ p$$

Taylor 급수의 1차항(선형항)만 고려한 등호 제약 조건

$$g_j(\mathbf{x}+\Delta\mathbf{x}) \cong g_j(\mathbf{x}) + \nabla g_j^T(\mathbf{x})\Delta\mathbf{x} \le 0;\ j = 1\ to\ m$$

Taylor 급수의 1차항(선형항)만 고려한 부등호 제약 조건

2차 계획 문제(Quadratic Programming Problem)의 정식화

Minimize $f(\mathbf{x}+\Delta\mathbf{x}) \cong f(\mathbf{x}) + \nabla f^T(\mathbf{x})\Delta\mathbf{x} + 0.5\Delta\mathbf{x}^T\mathbf{H}\Delta\mathbf{x}$

Taylor 급수의 2차항까지 고려한 목적 함수

Subject to $h_j(\mathbf{x}+\Delta\mathbf{x}) \cong h_j(\mathbf{x}) + \nabla h_j^T(\mathbf{x})\Delta\mathbf{x} = 0;\ j = 1\ to\ p$

Taylor 급수의 1차항(선형항)만 고려한 등호 제약 조건

$g_j(\mathbf{x}+\Delta\mathbf{x}) \cong g_j(\mathbf{x}) + \nabla g_j^T(\mathbf{x})\Delta\mathbf{x} \le 0;\ j = 1\ to\ m$

Taylor 급수의 1차항(선형항)만 고려한 부등호 제약 조건

여기서, $\bar{f} = f(\mathbf{x}+\Delta\mathbf{x}) - f(\mathbf{x}),\ e_j = -h_j(\mathbf{x}),\ b_j = -g_j(\mathbf{x}),$

$c_i = \partial f(\mathbf{x})/\partial x_i,\ n_{ij} = \partial h_j(\mathbf{x})/\partial x_i,\ a_{ij} = \partial g_j(\mathbf{x})/\partial x_i,$

$d_i = \Delta x_i$ 라고 가정하면

Matrix form

Minimize $\bar{f} = \mathbf{c}^T{}_{(1\times n)}\mathbf{d}_{(n\times 1)} + \frac{1}{2}\mathbf{d}^T{}_{(1\times n)}\mathbf{H}_{(n\times n)}\mathbf{d}_{(n\times 1)}$: 2차 형식의 목적 함수

Subject to $\mathbf{N}^T{}_{(p\times n)}\mathbf{d}_{(n\times 1)} = \mathbf{e}_{(p\times 1)}$: 선형화 된 등호 제약 조건

$\mathbf{A}^T{}_{(m\times n)}\mathbf{d}_{(n\times 1)} \le \mathbf{b}_{(m\times 1)}$: 선형화 된 부등호 제약 조건

2차 계획 문제의 정식화(요약)

$$\text{Minimize} \quad \bar{f} = \sum_{i=1}^{n} c_i d_i + 0.5 \sum_{i=1}^{n} \left(d_i^{\,2} \right)$$

$$\text{Subject to} \quad \sum_{i=1}^{n} n_{ij} d_i = e_j ;\ j = 1 \ to \ p$$

$$\sum_{i=1}^{n} a_{ij} d_i \le b_j ;\ j = 1 \ to \ m$$

<u>Matrix form</u>

$$\text{Minimize} \quad \bar{f} = \mathbf{c}^T{}_{(1\times n)} \mathbf{d}_{(n\times 1)} + 0.5 \mathbf{d}^T{}_{(1\times n)} \mathbf{d}_{(n\times 1)}$$

$$\text{Subject to} \quad \mathbf{N}^T{}_{(p\times n)} \mathbf{d}_{(n\times 1)} = \mathbf{e}_{(p\times 1)}$$

$$\mathbf{A}^T{}_{(m\times n)} \mathbf{d}_{(n\times 1)} \le \mathbf{b}_{(m\times 1)}$$

5.5 CSD(Constrained Steepest Descent) 방법

Advanced
Ship
Design
Automation
Laboratory

CSD(Constrained Steepest Descent) 방법
- Pshenichny의 강하 함수

최적화 문제

$$\textit{Minimize} \quad f(\mathbf{x}) = f(x_1, x_2, \Lambda\ , x_n)$$

$$\textit{Subject to} \quad h_i(\mathbf{x}) = 0, \quad i = 1,\ldots,p \quad \text{등호 제약 조건}$$

$$g_i(\mathbf{x}) = 0, \quad i = 1,\ldots,m \quad \text{부등호 제약 조건}$$

Pshenichny의 강하 함수

$$\Phi(\mathbf{x}^{(k)}) = f(\mathbf{x}^{(k)}) + R \cdot V(\mathbf{x}^{(k)})$$

여기서, R은 양의 상수로 가장 최근의 벌칙 매개 변수(Penalty Parameter, 초기에 사용자에 의해 명시됨)

$V(\mathbf{x})$는 최대 제약 조건 위배 값이며 0보다 크거나 같은 값으로 제약 조건을 모두 만족하는 경우는 0

$$R \ge \underline{r_k} \left(= \sum_{i=1}^{p} \left| v_i^{(k)} \right| + \sum_{i=1}^{m} u_i^{(k)} \right)$$

모든 Lagrange multiplier의 합

$$V(\mathbf{x}^{(k)}) = \max\{0; |h_1|, |h_2|, \Lambda\ , |h_p|; g_1, g_2, \Lambda\ , g_m\}$$ ➡ 모든 제약 조건을 만족하면 이 값은 0

새로운 해의 결정

현재의 설계점보다 강하 함수의 값을 더 감소시키는 새로운 설계점을 다음과 같이 결정

$$\underbrace{\mathbf{x}^{(k+1)}}_{\text{개선된 설계점}} = \underbrace{\mathbf{x}^{(k)}}_{\text{현재의 설계점}} + \alpha_k \mathbf{d}^{(k)}$$

강하 함수를 감소시키는 탐색 방향(2차 계획 문제를 풀어서 구함)

탐색 방향으로 강하 함수를 최대로 감소시킬 수 있는 이동 거리(Step Size)
(1차원 탐색 방법을 이용, 예, 황금 분할 방법)

탐색 방향을 결정하기 위한 2차 계획 문제의 정식화(1)

Minimize $f(\mathbf{x}+\Delta\mathbf{x}) \cong f(\mathbf{x}) + \nabla f^T(\mathbf{x})\Delta\mathbf{x} + 0.5\Delta\mathbf{x}^T\mathbf{H}\Delta\mathbf{x}$

Taylor 급수의 2차항까지 고려한 목적 함수

Subject to $h_j(\mathbf{x}+\Delta\mathbf{x}) \cong h_j(\mathbf{x}) + \nabla h_j^T(\mathbf{x})\Delta\mathbf{x} = 0;\ j = 1\ to\ p$

Taylor 급수의 1차항(선형항)만 고려한 등호 제약 조건

$g_j(\mathbf{x}+\Delta\mathbf{x}) \cong g_j(\mathbf{x}) + \nabla g_j^T(\mathbf{x})\Delta\mathbf{x} \le 0;\ j = 1\ to\ m$

Taylor 급수의 1차항(선형항)만 고려한 부등호 제약 조건

여기서, $\bar{f} = f(\mathbf{x}+\Delta\mathbf{x}) - f(\mathbf{x}),\ e_j = -h_j(\mathbf{x}),\ b_j = -g_j(\mathbf{x}),$

$c_i = \partial f(\mathbf{x})/\partial x_i,\ n_{ij} = \partial h_j(\mathbf{x})/\partial x_i,\ a_{ij} = \partial g_j(\mathbf{x})/\partial x_i,$

$d_i = \Delta x_i$ 라고 가정하면

Matrix form

Minimize $\bar{f} = \mathbf{c}^T{}_{(1\times n)}\mathbf{d}_{(n\times 1)} + \frac{1}{2}\mathbf{d}^T{}_{(1\times n)}\mathbf{H}_{(n\times n)}\mathbf{d}_{(n\times 1)}$: 2차 형식의 목적 함수

Subject to $\mathbf{N}^T{}_{(p\times n)}\mathbf{d}_{(n\times 1)} = \mathbf{e}_{(p\times 1)}$: 선형화 된 등호 제약 조건

$\mathbf{A}^T{}_{(m\times n)}\mathbf{d}_{(n\times 1)} \le \mathbf{b}_{(m\times 1)}$: 선형화 된 부등호 제약 조건

탐색 방향을 결정하기 위한 2차 계획 문제의 정식화(2)

$$\text{Minimize} \quad \bar{f} = \mathbf{c}^T{}_{(1\times n)}\mathbf{d}_{(n\times1)} + \frac{1}{2}\mathbf{d}^T{}_{(1\times n)}\mathbf{H}_{(n\times n)}\mathbf{d}_{(n\times1)}$$

$$\text{Subject to} \quad \mathbf{N}^T{}_{(p\times n)}\mathbf{d}_{(n\times1)} = \mathbf{e}_{(p\times1)}$$

$$\mathbf{A}^T{}_{(m\times n)}\mathbf{d}_{(n\times1)} \le \mathbf{b}_{(m\times1)}$$

$\mathbf{H}_{(n\times n)} = \mathbf{I}_{(n\times n)}$라고 가정해서 근사화 한다.

$$\text{Minimize} \quad \bar{f} = \mathbf{c}^T{}_{(1\times n)}\mathbf{d}_{(n\times1)} + \frac{1}{2}\mathbf{d}^T{}_{(1\times n)}\mathbf{I}_{(n\times1)}\mathbf{d}_{(n\times1)} = \mathbf{c}^T{}_{(1\times n)}\mathbf{d}_{(n\times1)} + \frac{1}{2}\mathbf{d}^T{}_{(1\times n)}\mathbf{d}_{(n\times1)}$$

$$\text{Subject to} \quad \mathbf{N}^T{}_{(p\times n)}\mathbf{d}_{(n\times1)} = \mathbf{e}_{(p\times1)}$$

$$\mathbf{A}^T{}_{(m\times n)}\mathbf{d}_{(n\times1)} \le \mathbf{b}_{(m\times1)}$$

- ➡ $\mathbf{H}_{(n\times n)} = \mathbf{I}_{(n\times n)}$이므로 행렬 $\mathbf{H}_{(n\times n)}$는 양정(Positive Definite) 행렬이며, 모든 제약 조건이 선형임
- ➡ 이 문제는 볼록 계획(Convex Programming) 문제이며, 해가 존재하면 이는 유일한 해 (전역 최적해)임
- ➡ 이 문제를 Simplex 방법을 이용하여 풀 경우 설계 변수 d는 음이 아니어야 함
  (설계 변수가 부호에 제한이 없는 경우 문제를 풀기 전에 먼저 $d = d^+ - d^-$로 변환해야 함)

이동 거리의 결정

강하 조건을 검토하기 위한 시행 설계점

$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{d}^{(k)}$ ➡ 개선된 설계점을 구하기 위해 α_k를 얼마로 해야 하는가?

α_k를 변경시켜 가면서 현재의 설계점보다 강하 함수를 감소시키는 설계점을 찾음
(황금 분할법과 같은 1차원 탐색 방법 이용 가능)

여기서는 다음과 같은 간단한 방법을 이용하여 이동 거리를 구하기로 함

$$\underbrace{\mathbf{x}^{(k+1,j)}}_{\text{시행 설계점}} = \mathbf{x}^{(k)} + A\underbrace{t_j}_{\text{시행 이동 거리}}\mathbf{d}^{(k)},\ t_j = (\tfrac{1}{2})^j;\ j = 0, 1, 2, \ldots,$$

A : 설계점의 최대 변화 정도를 결정하는 양의 상수 (예, 10)

강하 조건

시행 이동 거리(t_j) 중 아래의 조건을 제일 먼저 만족시키는 것을 최종 이동 거리(α_k)라 가정하고, 이때의 시행 설계점을 개선된 설계점이라고 결정함

$$\Phi(\mathbf{x}^{(k+1,j)}) \le \Phi(\mathbf{x}^{(k)}) - t_j \beta_k$$

$$\Rightarrow \underbrace{f(\mathbf{x}^{(k+1,j)}) + R \cdot V(\mathbf{x}^{(k+1,j)})}_{\text{시행 설계점에서의 강하 함수 값}} \le \underbrace{\Phi(\mathbf{x}^{(k)})}_{\text{현재의 설계점에서의 강하 함수 값}} - t_j \beta_k$$

➡ 현재의 설계점에서의 강하 함수 값에서 양수의 값(일종의 마진)을 더 뺀 값보다 강하 함수 값이 더 작은 설계점을 구하기 위한 조건

여기서, $\beta_k = \gamma \left\| \mathbf{d}^{(k)} \right\|^2, 0 < \gamma < 1$

γ는 $0 < \gamma < 1$인 상수

CSD(Constrained Steepest Descent) 방법을 이용한 풀이 예(1)

Minimize $f(\mathbf{x}) = x_1^2 + x_2^2 - 3x_1x_2$

Subject to $g_1(\mathbf{x}) = \frac{1}{6}x_1^2 + \frac{1}{6}x_2^2 - 1.0 \le 0$

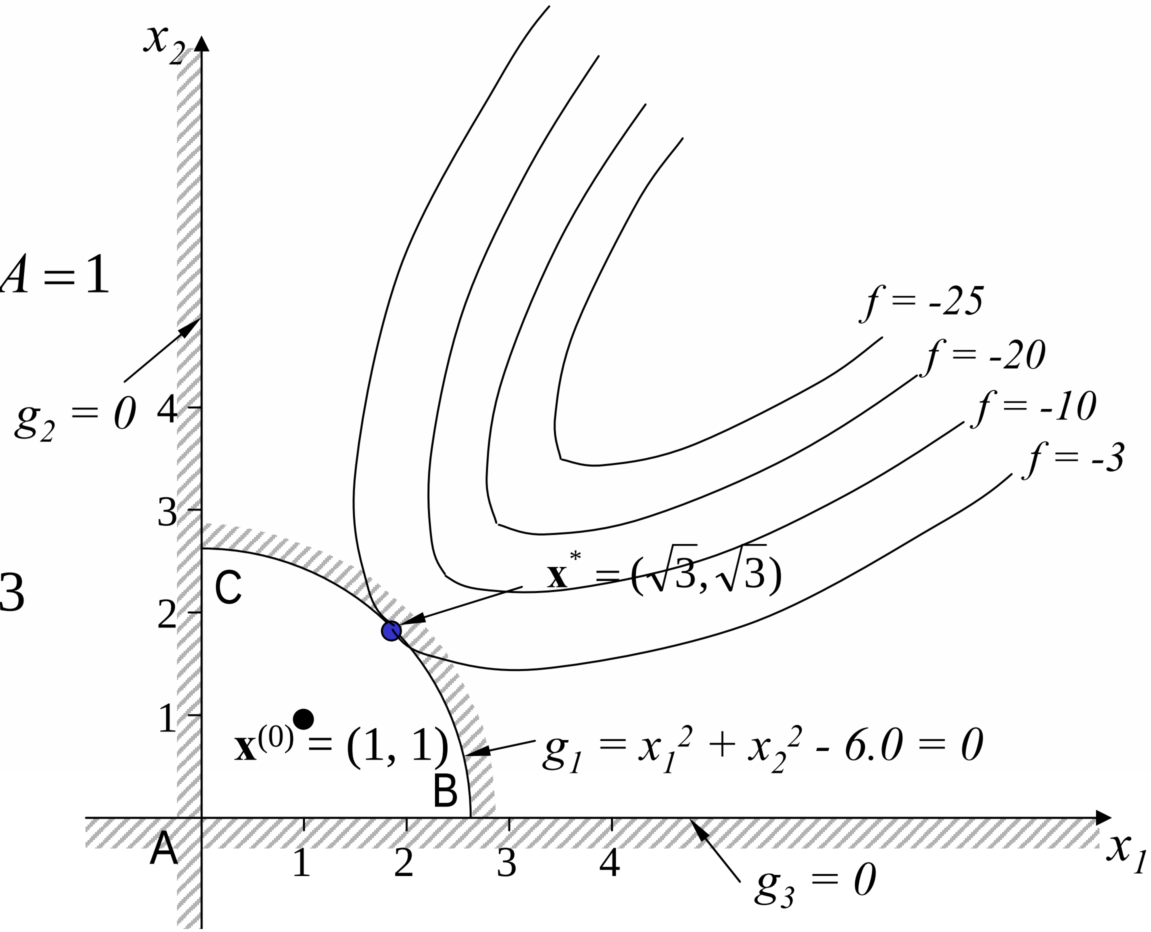
$g_2(\mathbf{x}) = -x_1 \le 0$

$g_3(\mathbf{x}) = -x_1 \le 0$

초기 시작점은 $\mathbf{x}^{(0)} = (1,1)$,

$R_0 = 10, \gamma = 0.5, \varepsilon_1 = \varepsilon_2 = 0.001, A = 1$

이라 가정



최적해는 $\mathbf{x}^* = (\sqrt{3}, \sqrt{3}), f(\mathbf{x}^*) = -3$

이때 Lagrange multiplier**는**

$\mathbf{u}^* = (3,0,0)$

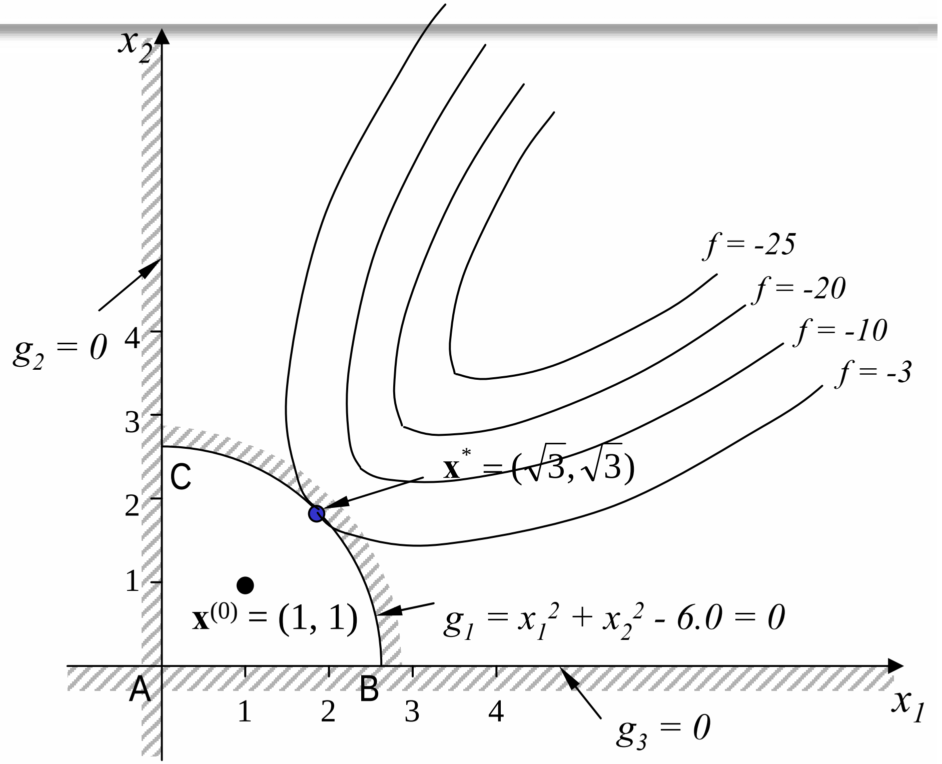
CSD(Constrained Steepest Descent) 방법을 이용한 풀이 예(2)

Minimize $f(\mathbf{x}) = x_1^2 + x_2^2 - 3x_1x_2$

Subject to $g_1(\mathbf{x}) = \frac{1}{6}x_1^2 + \frac{1}{6}x_2^2 - 1.0 \le 0$

$g_2(\mathbf{x}) = -x_1 \le 0$

$g_3(\mathbf{x}) = -x_1 \le 0$



(1) **반복 과정** 1(k = 0)

(i) **단계** 1: **강하 함수**(descent function)**의 정의**

$$\Phi_k = f_k + R \cdot V_k$$

문제에서 주어진 초기 조건으로부터

$$\mathbf{x}^{(0)} = (1,1), R_0 = 10, \gamma = 0.5, \varepsilon_1 = \varepsilon_2 = 0.001$$

(ii) **단계** 2: **최대 위배 제약 조건 값의 계산**

$$f(1,1) = -1$$

$g_1(1,1) = -\frac{2}{3} < 0$ ➡ **제약 조건 만족**

$g_2(1,1) = -1 < 0$ ➡ **제약 조건 만족**

$g_3(1,1) = -1 < 0$ ➡ **제약 조건 만족**

$V(\mathbf{x}^k) = \max\{0; |h_1|, |h_2|, \Lambda\ , |h_p|; g_1, g_2, \Lambda\ , g_m\}$**로부터**

$$V_0 = \max\{0; -\tfrac{2}{3}, -1, -1\} = 0$$

$$\nabla f = (-1,-1), \nabla g_1 = (\tfrac{1}{3}, \tfrac{1}{3}), \nabla g_2 = (-1,0), \nabla g_3 = (0,-1)$$

CSD(Constrained Steepest Descent) 방법을 이용한 풀이 예(3)

(iii) 단계 3: QP 문제의 풀이를 통한 탐색 방향($\mathbf{d}^{(0)}$)의 결정

$$\textit{Minimize} \quad f(\mathbf{x}) = x_1^2 + x_2^2 - 3x_1x_2$$

$$\textit{Subject to} \quad g_1(\mathbf{x}) = \frac{1}{6}x_1^2 + \frac{1}{6}x_2^2 - 1.0 \le 0$$

$$g_2(\mathbf{x}) = -x_1 \le 0$$

$$g_3(\mathbf{x}) = -x_2 \le 0$$

①

$$f(1,1) = -1,\ g_1(1,1) = -\tfrac{2}{3},$$
$$g_2(1,1) = -1,\ g_3(1,1) = -1$$
$$\nabla f = (-1,-1), \nabla g_1 = (\tfrac{1}{3},\tfrac{1}{3}),$$
$$\nabla g_2 = (-1,0), \nabla g_3 = (0,-1)$$

$$\textit{Minimize} \quad \bar{f} = (-d_1 - d_2) + 0.5(d_1^2 + d_2^2)$$

$$\textit{Subject to} \quad \tfrac{1}{3}d_1 + \tfrac{1}{3}d_2 \le \tfrac{2}{3}$$

$$-d_1 \le 1$$

$$-d_2 \le 1$$

여기서,
$d_1 = x_1 - 1,\ d_2 = x_2 - 1$

②

Lagrange 함수

$$L = (-d_1 - d_2) + 0.5(d_1^2 + d_2^2) + u_1[\tfrac{1}{3}(d_1 + d_2 - 2) + s_1^2] + u_2(-d_1 - 1 + s_2^2) + u_3(-d_2 - 1 + s_3^2)$$

③

Kuhn-Tucker 필요 조건: $\nabla L(\mathbf{d},\mathbf{u},\mathbf{s}) = \mathbf{0}$

$$\frac{\partial L}{\partial d_1} = -1 + d_1 + \tfrac{1}{3}u_1 - u_2 = 0$$

$$\frac{\partial L}{\partial d_2} = -1 + d_2 + \tfrac{1}{3}u_1 - u_3 = 0$$

$$\tfrac{1}{3}(d_1 + d_2 - 2) + s_1^2 = 0$$

$$(-d_1 - 1) + s_2^2 = 0$$

$$(-d_2 - 1) + s_3^2 = 0$$

$$u_i s_i = 0$$

$$u_i \ge 0,\ i = 1,2,3$$

최적해를 구하면

$$\mathbf{d}^{(0)} = (d_1, d_2) = (1,1),$$

$$\mathbf{u}^{(0)} = (u_1, u_2, u_3) = (0,0,0)$$

* Simplex 방법을 이용해도 최적해를 구할 수 있음 (뒤에서 설명함)

CSD(Constrained Steepest Descent) 방법을 이용한 풀이 예(4)

(iv) 단계 4: 수렴 기준의 검토

$$\mathbf{d}^{(0)} = (d_1, d_2) = (1,1)$$

$\left\|\mathbf{d}^{(0)}\right\| = \sqrt{2} > \varepsilon_2 (= 0.001)$ 이므로 수렴 기준을 만족하지 않음

(v) 단계 5: 벌칙 매개 변수 R의 계산

$\mathbf{u}^{(0)} = (u_1, u_2, u_3) = (0,0,0)$ 이고 $r_k = \sum_{i=1}^{p} \left| v_i^{(k)} \right| + \sum_{i=1}^{m} u_i^{(k)}$ 로부터 $r_0 = \sum_{i=1}^{m} u_i^{(0)} = 0$

따라서 $R = \max\{R_0, r_0\} = \max\{10,0\} = 10$

(vi) 단계 6: 탐색 방향($\mathbf{d}^{(0)}$)으로 강하 함수를 최소화 하는 이동 거리의 계산

이동 거리를 결정하기 위해 황금 분할법을 이용해도 되나 여기서는 간단한 선 탐색법을 이용하기로 함

$\Phi_k = f_k + R \cdot V_k$ 로부터

$$\Phi_0 = f_0 + R \cdot V_0 = -1 + (10) \cdot (0) = -1$$

$$\beta_0 = \gamma \left\|\mathbf{d}^{(0)}\right\|^2 = 0.5 \cdot (\sqrt{2})^2 = 1$$

CSD(Constrained Steepest Descent) 방법을 이용한 풀이 예(5)

(vi) 단계 6: 탐색 방향($\mathbf{d}^{(0)}$)으로 강하 함수를 최소화 하는 이동 거리의 계산

시행 이동 거리 $t_0 = 1$이라 가정하고 아래의 시행 설계점에 대해 강하 함수를 계산하면 다음과 같다.

$$\mathbf{x}^{(1,0)} = \mathbf{x}^{(0)} + A \cdot t_0 \cdot \mathbf{d}^{(0)} = (1,1) + 1 \cdot 1 \cdot (1,1) = (2,2)$$

$$f_{1,0}(\mathbf{x}^{(1,0)}) = f_{1,0}(2,2) = -4$$

$$g_1(\mathbf{x}^{(1,0)}) = g_1(2,2) = \tfrac{1}{6}x_1^2 + \tfrac{1}{6}x_2^2 - 1 = \tfrac{1}{3}$$

$$g_2(\mathbf{x}^{(1,0)}) = g_2(2,2) = -x_1 = -2$$

$$g_3(\mathbf{x}^{(1,0)}) = g_3(2,2) = -x_2 = -2$$

$$V_{1,0} = V(\mathbf{x}^{(1,0)}) = \max\{0; \tfrac{1}{3}, -2, -2\} = \tfrac{1}{3}$$

$$\Phi_{1,0} = f_{1,0} + R \cdot V_{1,0} = -4 + (10) \cdot (\tfrac{1}{3}) = -\tfrac{2}{3}$$

한편, $\Phi_0 - t_0\beta_0 = -1 - (1) \cdot (1) = -2$ 로부터

$\Phi_{1,0}(= -\tfrac{2}{3}) > \Phi_0 - t_0\beta_0(= -2)$이므로 강하 조건은 만족되지 않는다.

따라서 이동 거리를 반으로 줄여서 강하 조건을 다시 검토한다.

CSD(Constrained Steepest Descent) 방법을 이용한 풀이 예(6)

(vi) **단계** 6: **탐색 방향**($\mathbf{d}^{(0)}$)**으로 강하 함수를 최소화 하는 이동 거리의 계산**

시행 이동 거리 $t_1 = 0.5$**라 가정하고 아래의 시행 설계점에 대해 강하 함수를 계산하면 다음과 같다.**

$$\mathbf{x}^{(1,1)} = \mathbf{x}^{(0)} + A \cdot t_1 \cdot \mathbf{d}^{(0)} = (1,1) + 1 \cdot 0.5 \cdot (1,1) = (1.5,1.5)$$

$$f_{1,1}(\mathbf{x}^{(1,1)}) = f_{1,1}(1.5,1.5) = -2.25$$

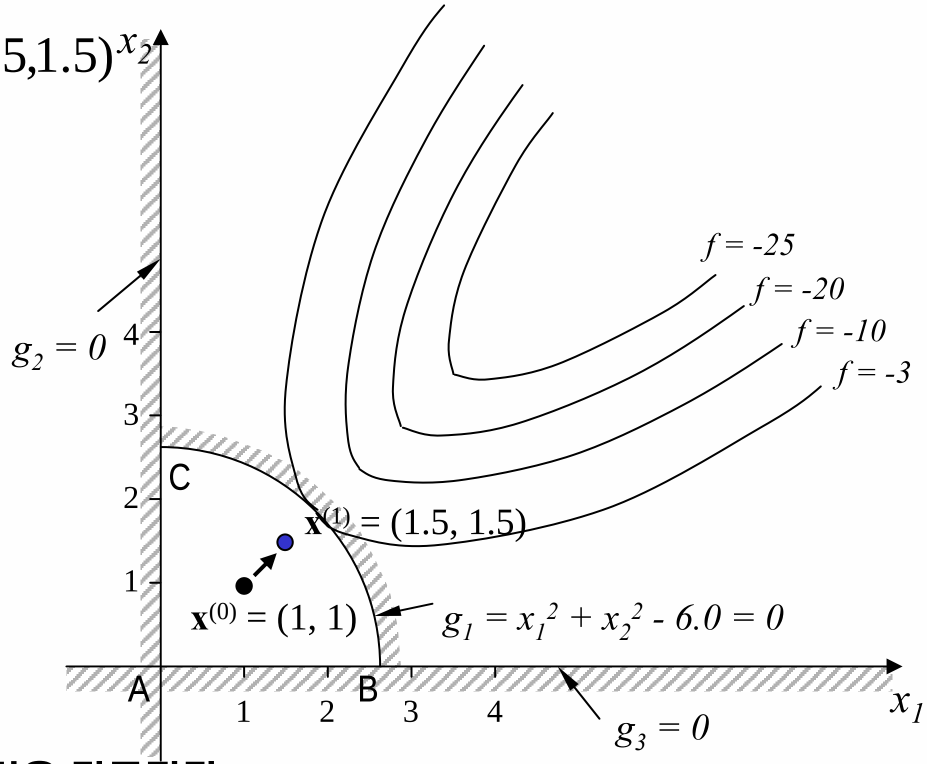
$$g_1(\mathbf{x}^{(1,1)}) = g_1(1.5,1.5) = \tfrac{1}{6}x_1^2 + \tfrac{1}{6}x_2^2 - 1 = -\tfrac{1}{4}$$

$$g_2(\mathbf{x}^{(1,1)}) = g_2(1.5,1.5) = -x_1 = -1.5$$

$$g_3(\mathbf{x}^{(1,1)}) = g_3(1.5,1.5) = -x_2 = -1.5$$

$$V_{1,1} = V(\mathbf{x}^{(1,1)}) = \max\{0; -\tfrac{1}{4}, -1.5, -1.5\} = 0$$

$$\Phi_{1,1} = f_{1,1} + R \cdot V_{1,1} = -2.25 + (10)\cdot(0) = -2.25$$



한편, $\Phi_0 - t_1\beta_0 = -1 - (0.5)\cdot(1) = -1.5$ **로부터**

$\Phi_{1,1}(= -2.25) < \Phi_0 - t_1\beta_0(= -1.5)$ **이므로 강하 조건은 만족된다.**

따라서 $\alpha_0 = t_1 = 0.5$ **이고**

$\mathbf{x}^{(1)} = \mathbf{x}^{(1,1)} = \mathbf{x}^{(0,0)} + \alpha_0\mathbf{d}^{(0)} = \mathbf{x}^{(0,0)} + A \cdot t_1 \cdot \mathbf{d}^{(0)} = (1,1) + 1 \cdot 0.5 \cdot (1,1) = (1.5,1.5)$ **이다.**

(vii) **단계** 7: **벌칙 매개 변수 및 반복 횟수의 갱신** $R_1 = R = 10, k = k + 1 = 1$

CSD(Constrained Steepest Descent) 방법을 이용한 풀이 예(7)

(2) **반복 과정** 2(k = 1)

(ii) **단계** 2: **최대 위배 제약 조건 값의 계산**

이전 단계로부터

$$\mathbf{x}^{(1,1)} = (1.5, 1.5)$$

$$f(\mathbf{x}^{(1,1)}) = f(1.5, 1.5) = -2.25$$

$$g_1(\mathbf{x}^{(1,1)}) = g_1(1.5, 1.5) = \tfrac{1}{6}x_1^2 + \tfrac{1}{6}x_2^2 - 1 = -\tfrac{1}{4}$$ ➡ **제약 조건 만족**

$$g_2(\mathbf{x}^{(1,1)}) = g_2(1.5, 1.5) = -x_1 = -1.5$$ ➡ **제약 조건 만족**

$$g_3(\mathbf{x}^{(1,1)}) = g_3(1.5, 1.5) = -x_2 = -1.5$$ ➡ **제약 조건 만족**

$$V_1 = V(\mathbf{x}^{(1)}) = \max\{0; \tfrac{1}{4}, -1.5, -1.5\} = 0$$

한편,

$$\nabla f(\mathbf{x}^{(1)}) = (2x_1 - 3x_2, 2x_2 - 3x_1) = (-1.5, -1.5)$$

$$\nabla g_1(\mathbf{x}^{(1)}) = (\tfrac{1}{3}x_1, \tfrac{1}{3}x_2) = (0.5, 0.5), \nabla g_2 = (-1, 0), \nabla g_3 = (0, -1)$$

CSD(Constrained Steepest Descent) 방법을 이용한 풀이 예(8)

(iii) 단계 3: QP 문제의 풀이를 통한 탐색 방향($\mathbf{d}^{(1)}$)의 결정

Minimize $f(\mathbf{x}) = x_1^2 + x_2^2 - 3x_1x_2$

Subject to $g_1(\mathbf{x}) = \frac{1}{6}x_1^2 + \frac{1}{6}x_2^2 - 1.0 \le 0$

$g_2(\mathbf{x}) = -x_1 \le 0$

$g_3(\mathbf{x}) = -x_1 \le 0$

①

$f(1.5,1.5) = -2.25, g_1(1.5,1.5) = -\frac{1}{4},$
$g_2(1.5,1.5) = -1.5, g_3(1.5,1.5) = -1.5$
$\nabla f = (-1.5,-1.5), \nabla g_1 = (0.5,0.5),$
$\nabla g_2 = (-1,0), \nabla g_3 = (0,-1)$

Minimize $\bar{f} = (-1.5d_1 - 1.5d_2) + 0.5(d_1^2 + d_2^2)$

Subject to $0.5d_1 + 0.5d_2 \le \frac{1}{4}$

$-d_1 \le 1.5$

$-d_2 \le 1.5$

여기서,
$d_1 = x_1 - 1.5,\ d_2 = x_2 - 1.5$

②

Lagrange 함수

$$L = (-1.5d_1 - 1.5d_2) + 0.5(d_1^2 + d_2^2)$$
$$+ u_1[0.5(d_1 + d_2 - 0.5) + s_1^2]$$
$$+ u_2(-d_1 - 1.5 + s_2^2)$$
$$+ u_3(-d_2 - 1.5 + s_3^2)$$

③

Kuhn-Tucker 필요 조건: $\nabla L(\mathbf{d},\mathbf{u},\mathbf{s}) = \mathbf{0}$

$$\frac{\partial L}{\partial d_1} = -1.5 + d_1 + 0.5u_1 - u_2 = 0$$
$$\frac{\partial L}{\partial d_2} = -1.5 + d_2 + 0.5u_1 - u_3 = 0$$
$$0.5(d_1 + d_2 - 0.5) + s_1^2 = 0$$
$$(-d_1 - 1.5) + s_2^2 = 0$$
$$(-d_2 - 1.5) + s_3^2 = 0$$
$$u_i s_i = 0$$
$$u_i \ge 0,\ i = 1,2,3$$

최적해를 구하면

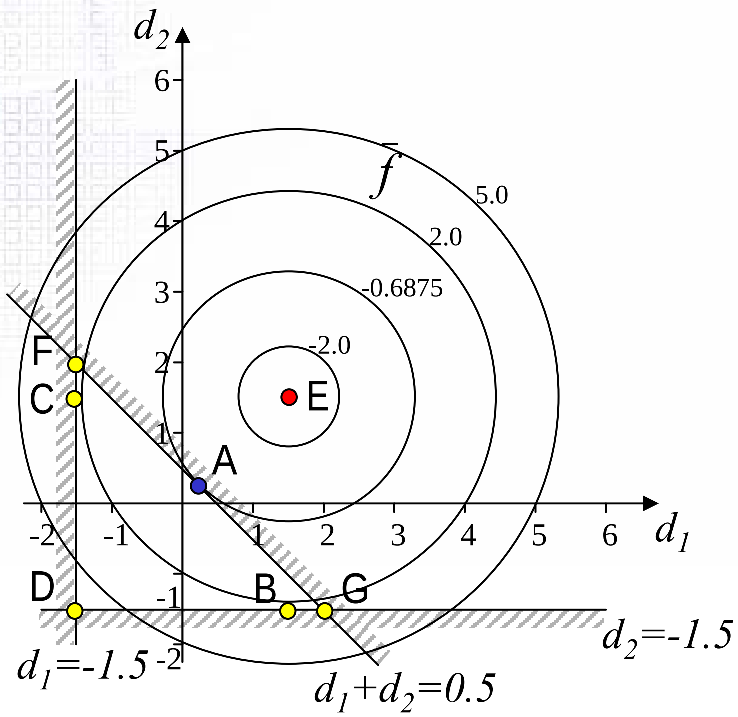
$$\mathbf{d}^{(1)} = (d_1, d_2) = (0.25, 0.25),$$
$$\mathbf{u}^{(1)} = (u_1, u_2, u_3) = (2.5, 0, 0)$$

CSD(Constrained Steepest Descent) 방법을 이용한 풀이 예(9)

(iii) 단계 3: QP 문제의 풀이를 통한 탐색 방향($\mathbf{d}^{(1)}$)의 결정

<u>Lagrange 함수</u>

$$L = (-1.5d_1 - 1.5d_2) + 0.5(d_1^2 + d_2^2)$$
$$+ u_1[0.5(d_1 + d_2 - 0.5) + s_1^2]$$
$$+ u_2(-d_1 - 1.5 + s_2^2)$$
$$+ u_3(-d_2 - 1.5 + s_3^2)$$



점 A: s_1=u_2=u_3=0일 때

$$d_1 = d_2 = 0.25, u_1 = 2.5, s_2 = s_3 = \pm\sqrt{1.75}$$ ➡ **최적해**

점 B: u_1=u_2=s_3=0일 때

$$d_1 = 1.5, d_2 = -1.5, u_3 = -3, s_1 = \pm\sqrt{0.25}, s_2 = \pm\sqrt{3}$$

음수여서 안됨

점 C: u_1=s_2=u_3=0일 때

$$d_1 = -1.5, d_2 = 1.5, u_2 = -3, s_1 = \pm\sqrt{0.25}, s_3 = \pm\sqrt{3}$$

음수여서 안됨

점 D: u_1=s_2=s_3=0일 때

$$d_1 = -1.5, d_2 = -1.5, u_2 = u_3 = -3, s_1 = \pm\sqrt{1.75}$$

음수여서 안됨

점 D: s_1=s_2=s_3=0일 때

$$d_1 = -1.5, d_2 = -1.5, s_1 = \pm\sqrt{1.75}(\neq 0)$$

점 E: u_1=u_2=u_3=0일 때

$$d_1 = d_2 = 1.5, s_1^2 = -1.25, s_2 = s_3 = \pm\sqrt{3}$$

음수여서 안됨

점 F: s_1=s_2=u_3=0일 때

$$d_1 = -1.5, d_2 = 2, u_1 = -1, u_2 = -3.5, s_3 = \pm\sqrt{3.5}$$

음수여서 안됨

점 G: s_1=u_2=s_3=0일 때

음수여서 안됨

$$d_1 = 2, d_2 = -1.5, u_1 = -1, u_3 = -3.5, s_2 = \pm\sqrt{3.5}$$

CSD(Constrained Steepest Descent) 방법을 이용한 풀이 예(10)

(iv) 단계 4: 수렴 기준의 검토

$$\mathbf{d}^{(1)} = (d_1, d_2) = (0.25, 0.25)$$

$\left\|\mathbf{d}^{(1)}\right\| = \sqrt{0.25^2 + 0.25^2} = 0.3535 > \varepsilon_2 (= 0.001)$ 이므로 수렴 기준을 만족하지 않음

(v) 단계 5: 벌칙 매개 변수 R의 계산

$\mathbf{u}^{(1)} = (u_1, u_2, u_3) = (2.5, 0, 0)$ 이고 $r_k = \sum_{i=1}^{p} \left|v_i^{(k)}\right| + \sum_{i=1}^{m} u_i^{(k)}$ 로부터 $r_1 = \sum_{i=1}^{m} u_i^{(1)} = 2.5$

따라서 $R = \max\{R_1, r_1\} = \max\{10, 2.5\} = 10$

(vi) 단계 6: 탐색 방향($\mathbf{d}^{(1)}$)으로 강하 함수를 최소화 하는 이동 거리의 계산

이동 거리를 결정하기 위해 황금 분할법을 이용해도 되나 여기서는 간단한 선 탐색법을 이용하기로 함

$\Phi_k = f_k + RV_k$ 로부터

$$\Phi_1 = f_1 + R \cdot V_1 = -2.25 + (10) \cdot (0) = -2.25$$

$$\beta_1 = \gamma \left\|\mathbf{d}^{(1)}\right\|^2 = 0.5 \cdot (\sqrt{0.25^2 + 0.25^2})^2 = 0.0625$$

CSD(Constrained Steepest Descent) 방법을 이용한 풀이 예(11)

(vi) 단계 6: 탐색 방향($\mathbf{d}^{(1)}$)으로 강하 함수를 최소화 하는 이동 거리의 계산

시행 이동 거리 $t_0 = 1$이라 가정하고 아래의 시행 설계점에 대해 강하 함수를 계산하면 다음과 같다.

$$\mathbf{x}^{(2,0)} = \mathbf{x}^{(1)} + A \cdot t_0 \cdot \mathbf{d}^{(1)} = (1.5,1.5) + 1 \cdot 1 \cdot (0.25,0.25) = (1.75,1.75)$$

$$f_{2,0}(\mathbf{x}^{(2,0)}) = f_{2,0}(1.75,1.75) = -3.0625$$

$$g_1(\mathbf{x}^{(2,0)}) = g_1(1.75,1.75) = \tfrac{1}{6}x_1^2 + \tfrac{1}{6}x_2^2 - 1 = 0.0208$$

$$g_2(\mathbf{x}^{(2,0)}) = g_2(1.75,1.75) = -x_1 = -1.75$$

$$g_3(\mathbf{x}^{(2,0)}) = g_3(1.75,1.75) = -x_2 = -1.75$$

$$V_{2,0} = V(\mathbf{x}^{(2,0)}) = \max\{0; 0.0208, -1.75, -1.75\} = 0.0208$$

$$\Phi_{2,0} = f_{2,0} + R \cdot V_{2,0} = -3.0625 + (10) \cdot (0.0208) = -2.8541$$

한편, $\Phi_1 - t_0\beta_1 = -2.25 - (1) \cdot (0.0625) = -2.3125$ 로부터

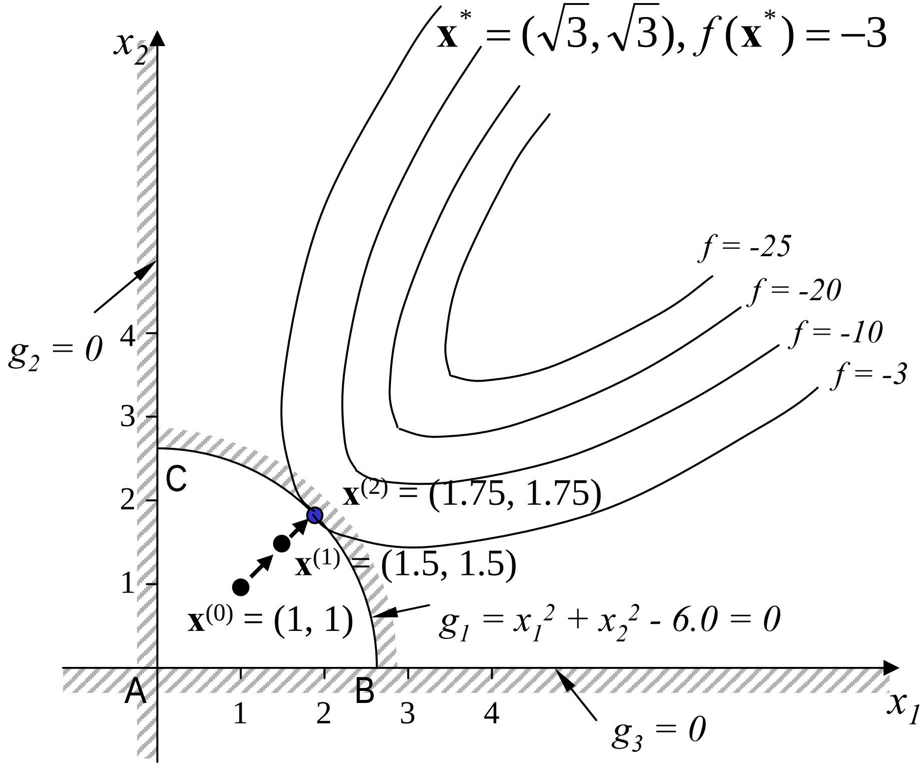
$\Phi_{2,0}(= -2.8541) < \Phi_1 - t_0\beta_1(= -2.3125)$ 이므로 강하 조건은 만족된다.

CSD(Constrained Steepest Descent) 방법을 이용한 풀이 예(12)

(vi) 단계 6: 탐색 방향($\mathbf{d}^{(1)}$)으로 강하 함수를 최소화 하는 이동 거리의 계산

따라서 $\alpha_1 = A \cdot t_0 = 1.0$ 이고

$$\mathbf{x}^{(2)} = \mathbf{x}^{(2,0)} = \mathbf{x}^{(1,1)} + \alpha_1 \mathbf{d}^{(1)}$$
$$= \mathbf{x}^{(1,1)} + A \cdot t_0 \cdot \mathbf{d}^{(1)}$$
$$= (1.5,1.5) + 1.0 \cdot (0.25,0.25)$$
$$= (1.75,1.75)$$ 이다.



(vii) 단계 7: 벌칙 매개 변수 및 반복 횟수의 갱신

$$R_2 = R = 10, k = k + 1 = 2$$

아직 아래와 같은 수렴 기준을 만족하지 않으므로 위와 유사한 과정을 반복적으로 수행한다.

<u>탐색 방향의 크기와 관련된 수렴 기준</u>

$$\left\| \mathbf{d}^{(1)} \right\| = 0.3535 > \varepsilon_2 (= 0.001)$$

<u>최대 제약 조건 위배와 관련된 수렴 기준</u>

$$V_{2,0} = V(\mathbf{x}^{(2,0)}) = 0.0208 \geq \varepsilon_1 (= 0.001)$$

Simplex 방법을 이용, 탐색 방향을 결정하기 위한 2차 계획 문제의 풀이(1)

Minimize $f(\mathbf{x}) = x_1^2 + x_2^2 - 3x_1x_2$

Subject to $g_1(\mathbf{x}) = \frac{1}{6}x_1^2 + \frac{1}{6}x_2^2 - 1.0 \le 0$

$g_2(\mathbf{x}) = -x_1 \le 0$

$g_3(\mathbf{x}) = -x_2 \le 0$

①

$f(1,1) = -1,\ g_1(1,1) = -\frac{2}{3},$
$g_2(1,1) = -1,\ g_3(1,1) = -1$
$\nabla f = (-1,-1), \nabla g_1 = (\frac{1}{3},\frac{1}{3}),$
$\nabla g_2 = (-1,0), \nabla g_3 = (0,-1)$

탐색 방향($\mathbf{d}^{(0)}$)의 결정을 위한 2차 계획(QP) 문제의 정의

Minimize $\bar{f} = (-d_1 - d_2) + 0.5(d_1^2 + d_2^2)$

Subject to $\frac{1}{3}d_1 + \frac{1}{3}d_2 \le \frac{2}{3}$

$-d_1 \le 1$

$-d_2 \le 1$

여기서, $d_1 = x_1 - 1,\ d_2 = x_2 - 1$

⬇ 도식화

②

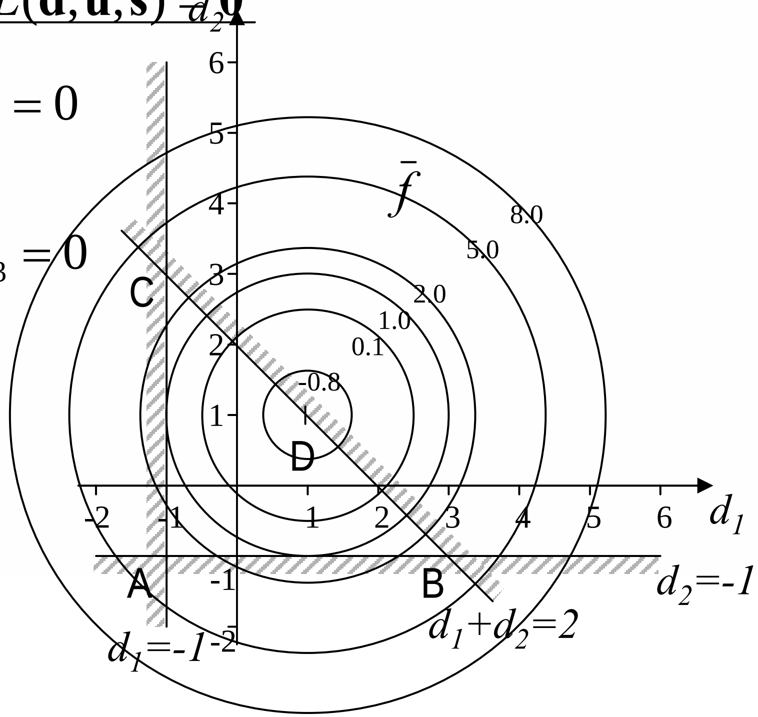
Lagrange 함수

$$L = (-d_1 - d_2) + 0.5(d_1^2 + d_2^2)$$
$$+ u_1[\tfrac{1}{3}(d_1 + d_2 - 2) + s_1]$$
$$+ u_2(-d_1 - 1 + s_2)$$
$$+ u_3(-d_2 - 1 + s_3)$$

③

Kuhn-Tucker 필요 조건: $\nabla L(\mathbf{d},\mathbf{u},\mathbf{s}) = \mathbf{0}$

$$\frac{\partial L}{\partial d_1} = -1 + d_1 + \tfrac{1}{3}u_1 - u_2 = 0$$
$$\frac{\partial L}{\partial d_2} = -1 + d_2 + \tfrac{1}{3}u_1 - u_3 = 0$$
$$\tfrac{1}{3}(d_1 + d_2 - 2) + s_1 = 0$$
$$(-d_1 - 1) + s_2 = 0$$
$$(-d_2 - 1) + s_3 = 0$$
$$u_i s_i = 0$$
$$u_i, s_i \ge 0,\ i = 1,2,3$$



Simplex 방법을 이용, 탐색 방향을 결정하기 위한 2차 계획 문제의 풀이(2)

Minimize $\bar{f} = (-d_1 - d_2) + 0.5(d_1^2 + d_2^2)$

Subject to $\frac{1}{3}d_1 + \frac{1}{3}d_2 \le \frac{2}{3}$

$-d_1 \le 1$

$-d_2 \le 1$

$\Big\downarrow$ Matrix 형태로 표현

Minimize $\bar{f} = \mathbf{c}^T{}_{(1\times2)}\mathbf{d}_{(2\times1)} + \dfrac{1}{2}\mathbf{d}^T{}_{(1\times2)}\mathbf{H}_{(2\times2)}\mathbf{d}_{(2\times1)}$

Subject to $\mathbf{A}^T{}_{(3\times2)}\mathbf{d}_{(2\times1)} \le \mathbf{b}_{(3\times1)}$

$\mathbf{d}_{(2\times1)} \ge 0$

여기서, $\mathbf{d}_{(2\times1)} = \begin{bmatrix} d_1 \\ d_2 \end{bmatrix}, \mathbf{c}_{(2\times1)} = \begin{bmatrix} -1 \\ -1 \end{bmatrix}, \mathbf{H}_{(2\times2)} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$

$$\mathbf{A}_{(2\times3)} = \begin{bmatrix} \frac{1}{3} & -1 & 0 \\ \frac{1}{3} & 0 & -1 \end{bmatrix}, \mathbf{b}_{(3\times1)} = \begin{bmatrix} \frac{2}{3} \\ 1 \\ 1 \end{bmatrix}$$

Simplex 방법을 이용, 탐색 방향을 결정하기 위한 2차 계획 문제의 풀이(3)

$$\bar{f} = (-d_1 - d_2) + 0.5(d_1^2 + d_2^2)$$
$$\tfrac{1}{3}d_1 + \tfrac{1}{3}d_2 \le \tfrac{2}{3}$$
$$-d_1 \le 1$$
$$-d_2 \le 1$$

Lagrange 함수

$$L = (-d_1 - d_2) + 0.5(d_1^2 + d_2^2)$$
$$+ u_1[\tfrac{1}{3}(d_1 + d_2 - 2) + s_1]$$
$$+ u_2(-d_1 - 1 + s_2) + u_3(-d_2 - 1 + s_3)$$

Matrix 형태로 표현

$$L = \begin{bmatrix} -1 & -1 \end{bmatrix}\begin{bmatrix} d_1 \\ d_2 \end{bmatrix} + \frac{1}{2}\begin{bmatrix} d_1 & d_2 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} d_1 \\ d_2 \end{bmatrix}$$
$$+ \begin{bmatrix} u_1 & u_2 & u_3 \end{bmatrix}\left(\begin{bmatrix} \frac{1}{3} & \frac{1}{3} \\ -1 & 0 \\ 0 & -1 \end{bmatrix}\begin{bmatrix} d_1 \\ d_2 \end{bmatrix} + \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix} - \begin{bmatrix} \frac{2}{3} \\ 1 \\ 1 \end{bmatrix}\right)$$
$$= \mathbf{c}^T{}_{(1\times2)}\mathbf{d}_{(2\times1)} + \frac{1}{2}\mathbf{d}^T{}_{(1\times2)}\mathbf{H}_{(2\times2)}\mathbf{d}_{(2\times1)} + \mathbf{u}^T{}_{(1\times3)}(\mathbf{A}^T{}_{(3\times2)}\mathbf{d}_{(2\times1)} + \mathbf{s}_{(3\times1)} - \mathbf{b}_{(3\times1)})$$

여기서, $\mathbf{d}_{(2\times1)} = \begin{bmatrix} d_1 \\ d_2 \end{bmatrix}, \mathbf{u}_{(3\times1)} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}, \mathbf{s}_{(3\times1)} = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix}$

부등호 제약 조건을 등호 제약 조건으로 변환하기 위해
도입된 완화 변수(slack variable)

Simplex 방법을 이용, 탐색 방향을 결정하기 위한 2차 계획 문제의 풀이(4)

Kuhn-Tucker 필요 조건: $\nabla L(\mathbf{d},\mathbf{u},\mathbf{s}) = \mathbf{0}$

$$L = \mathbf{c}^T{}_{(1\times2)}\mathbf{d}_{(2\times1)} + \frac{1}{2}\mathbf{d}^T{}_{(1\times2)}\mathbf{H}_{(2\times2)}\mathbf{d}_{(2\times1)} + \mathbf{u}^T{}_{(1\times3)}(\mathbf{A}^T{}_{(3\times2)}\mathbf{d}_{(2\times1)} + \mathbf{s}_{(3\times1)} - \mathbf{b}_{(3\times1)})$$

$$\frac{\partial L}{\partial \mathbf{d}_{(2\times1)}} = \begin{bmatrix} -1+d_1+\frac{1}{3}u_1-u_2 \\ -1+d_2+\frac{1}{3}u_1-u_3 \end{bmatrix}$$

$$= \begin{bmatrix} -1 \\ -1 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} d_1 \\ d_2 \end{bmatrix} + \begin{bmatrix} \frac{1}{3} & -1 & 0 \\ \frac{1}{3} & 0 & -1 \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}$$

$$= \mathbf{c}_{(2\times1)} + \mathbf{H}_{(2\times2)}\mathbf{d}_{(2\times1)} + \mathbf{A}_{(2\times3)}\mathbf{u}_{(3\times1)} = \mathbf{0}$$

$$\frac{\partial L}{\partial \mathbf{u}_{(3\times1)}} = \begin{bmatrix} \frac{1}{3}(d_1+d_2-2)+s_1 \\ -d_1-1+s_2 \\ -d_2-1+s_3 \end{bmatrix} = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} \\ -1 & 0 \\ 0 & -1 \end{bmatrix}\begin{bmatrix} d_1 \\ d_2 \end{bmatrix} + \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix} - \begin{bmatrix} \frac{2}{3} \\ 1 \\ 1 \end{bmatrix}$$

$$= \mathbf{A}^T{}_{(3\times2)}\mathbf{d}_{(2\times1)} + \mathbf{s}_{(3\times1)} - \mathbf{b}_{(3\times1)} = \mathbf{0}$$

$u_i s_i = 0; i = 1\ to\ 3$ 단, $u_i, s_i \geq 0; i = 1\ to\ 3$

한편, 설계 변수 d는 부호의 제한이 없으므로

Simplex 방법을 이용하기 위해 다음과 같이 표현해야 함

$$\mathbf{d}_{(2\times1)} = \mathbf{d}^+_{(2\times1)} - \mathbf{d}^-_{(2\times1)}$$

Matrix 형태로 표현

$$\underbrace{\begin{bmatrix} \mathbf{H}_{(2\times2)} & -\mathbf{H}_{(2\times2)} & \mathbf{A}_{(2\times3)} & \mathbf{0}_{(2\times3)} \\ \mathbf{A}^T{}_{(3\times2)} & -\mathbf{A}^T{}_{(3\times2)} & \mathbf{0}_{(3\times3)} & \mathbf{I}_{(3\times3)} \end{bmatrix}}_{=\mathbf{B}_{(5\times10)}} \underbrace{\begin{bmatrix} \mathbf{d}^+_{(2\times1)} \\ \mathbf{d}^-_{(2\times1)} \\ \mathbf{u}_{(3\times1)} \\ \mathbf{s}_{(3\times1)} \end{bmatrix}}_{=\mathbf{X}_{(10\times1)}} = \underbrace{\begin{bmatrix} -\mathbf{c}_{(2\times1)} \\ \mathbf{b}_{(3\times1)} \end{bmatrix}}_{=\mathbf{D}_{(5\times1)}}$$

Simplex 방법을 이용, 탐색 방향을 결정하기 위한 2차 계획 문제의 풀이(5)

Kuhn-Tucker 필요 조건: $\nabla L(\mathbf{d}^+, \mathbf{d}^-, \mathbf{u}, \mathbf{s}) = \mathbf{0}$

$$\underbrace{\left[\begin{array}{c:c:c:c} \mathbf{H}_{(2\times2)} & -\mathbf{H}_{(2\times2)} & \mathbf{A}_{(2\times3)} & \mathbf{0}_{(2\times3)} \\ \hdashline \mathbf{A}^T{}_{(3\times2)} & -\mathbf{A}^T{}_{(3\times2)} & \mathbf{0}_{(3\times3)} & \mathbf{I}_{(3\times3)} \end{array}\right]}_{=\mathbf{B}_{(5\times10)}} \underbrace{\begin{bmatrix} \mathbf{d}^+_{(2\times1)} \\ \mathbf{d}^-_{(2\times1)} \\ \mathbf{u}_{(3\times1)} \\ \mathbf{s}_{(3\times1)} \end{bmatrix}}_{=\mathbf{X}_{(10\times1)}} = \underbrace{\begin{bmatrix} -\mathbf{c}_{(2\times1)} \\ \mathbf{b}_{(3\times1)} \end{bmatrix}}_{=\mathbf{D}_{(5\times1)}}$$

여기서, $\mathbf{d}^+_{(2\times1)} = \begin{bmatrix} d_1^+ \\ d_2^+ \end{bmatrix}, \mathbf{d}^-_{(2\times1)} = \begin{bmatrix} d_1^- \\ d_2^- \end{bmatrix}, \mathbf{c}_{(2\times1)} = \begin{bmatrix} -1 \\ -1 \end{bmatrix}, \mathbf{H}_{(2\times2)} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \mathbf{A}_{(2\times3)} = \begin{bmatrix} \frac{1}{3} & -1 & 0 \\ \frac{1}{3} & 0 & -1 \end{bmatrix}, \mathbf{b}_{(3\times1)} = \begin{bmatrix} \frac{2}{3} \\ 1 \\ 1 \end{bmatrix}$

$$\mathbf{B}_{(5\times10)} = \left[\begin{array}{cc:cc:ccc:ccc} 1 & 0 & -1 & 0 & \frac{1}{3} & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & \frac{1}{3} & 0 & -1 & 0 & 0 & 0 \\ \hdashline \frac{1}{3} & \frac{1}{3} & -\frac{1}{3} & -\frac{1}{3} & 0 & 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right]$$

$\mathbf{X}^T{}_{(1\times10)} = \begin{bmatrix} d_1^+ & d_2^+ & d_1^- & d_2^- & u_1 & u_2 & u_3 & s_1 & s_2 & s_3 \end{bmatrix}, \mathbf{D}^T{}_{(1\times5)} = \begin{bmatrix} 1 & 1 & \frac{2}{3} & 1 & 1 \end{bmatrix}$

Simplex 방법을 이용, 탐색 방향을 결정하기 위한 2차 계획 문제의 풀이(6)

Kuhn-Tucker 필요 조건(행렬식 표현)

$$\mathbf{B}_{(5\times10)}\mathbf{X}_{(10\times1)} = \mathbf{D}_{(5\times1)}$$

$$\begin{bmatrix} 1 & 0 & -1 & 0 & \frac{1}{3} & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & \frac{1}{3} & 0 & -1 & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{3} & -\frac{1}{3} & -\frac{1}{3} & 0 & 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} d_1^+ \\ d_2^+ \\ d_1^- \\ d_2^- \\ u_1 \\ u_2 \\ u_3 \\ s_1 \\ s_2 \\ s_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ \frac{2}{3} \\ 1 \\ 1 \end{bmatrix}$$

구하려는 값

➡ X를 구하는 위 문제는 등호 제약 조건만으로 이루어진 선형 계획 문제임

➡ $u_i s_i = 0; i = 1\, to\, 3$ 조건(비선형 방정식)은 해를 구한 후 이를 만족하는지 평가하기 위한 용도로 사용됨

Simplex 방법을 이용, 탐색 방향을 결정하기 위한 2차 계획 문제의 풀이(7)

Simplex 방법을 이용한 QP 문제의 해법

1. Kuhn-Tucker 필요 조건의 해를 구하는 문제는 등호 제약 조건만으로 이루어진 부정 선형 연립 방정식의 해를 구하는 문제(선형 계획 문제)임
2. 부정 선형 연립 방정식의 해를 구하기 위하여 Simplex 방법에서 인위 변수 및 인위 목적 함수를 도입하여 초기 기저 가능해를 구하는 방법임

$$\mathbf{B}_{(5\times10)}\mathbf{X}_{(10\times1)} + \underbrace{\mathbf{Y}_{(5\times1)}}_{\text{인위 변수}} = \mathbf{D}_{(5\times1)}$$

3. 인위 목적 함수는 다음과 같이 정의함

$$w = \sum_{i=1}^{5} Y_i = \sum_{i=1}^{5} D_i - \sum_{j=1}^{10}\sum_{i=1}^{5} B_{ij}X_j = w_0 + \sum_{j=1}^{10} C_j X_j$$

여기서, $C_j = -\sum_{i=1}^{5} B_{ij}$: 행렬 B의 j번째 열의 요소를 모두 더해 부호를 바꾼 것(상대 비용 계수)

$$w_0 = \sum_{i=1}^{5} D_i = 1+1+\tfrac{2}{3}+1+1 = \tfrac{14}{3}$$

: 인위 목적 함수의 초기값으로 행렬 D의 모든 요소를 더한 것

4. Simplex 방법을 이용하여 해를 구하고 다음 식을 만족하는지 확인함

$u_i s_i = 0; i = 1\,to\,3$: 이 식은 비선형 방정식으로서 해를 구한 후 이를 만족하는지 평가하기 위한 용도로 사용됨

Simplex 방법을 이용, 탐색 방향을 결정하기 위한 2차 계획 문제의 풀이(8)

$$\mathbf{B}_{(5\times10)}\mathbf{X}_{(10\times1)} + \underbrace{\mathbf{Y}_{(5\times1)}}_{\text{인위 변수}} = \mathbf{D}_{(5\times1)}$$

➡

$$\begin{bmatrix} 1 & 0 & -1 & 0 & \frac{1}{3} & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & \frac{1}{3} & 0 & -1 & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{3} & -\frac{1}{3} & -\frac{1}{3} & 0 & 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} d_1^+(=X_1) \\ d_2^+(=X_2) \\ d_1^-(=X_3) \\ d_2^-(=X_4) \\ u_1(=X_5) \\ u_2(=X_6) \\ u_3(=X_7) \\ s_1(=X_8) \\ s_2(=X_9) \\ s_3(=X_{10}) \end{bmatrix} + \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \\ Y_5 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ \frac{2}{3} \\ 1 \\ 1 \end{bmatrix}$$

1

	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	Y1	Y2	Y3	Y4	Y5	bi	bi/ai
Y1	1	0	-1	0	1/3	-1	0	0	0	0	1	0	0	0	0	1	-
Y2	0	1	0	-1	1/3	0	-1	0	0	0	0	1	0	0	0	1	-
Y3	1/3	1/3	-1/3	-1/3	0	0	0	**1**	0	0	0	0	1	0	0	2/3	**2/3**
Y4	-1	0	1	0	0	0	0	0	1	0	0	0	0	1	0	1	-
Y5	0	-1	0	1	0	0	0	0	0	1	0	0	0	0	1	1	-
A. Obj.	-1/3	-1/3	1/3	1/3	-2/3	1	1	**-1**	-1	-1	0	0	0	0	0	w-14/3	-

인위 목적 함수 식 **각 열의 요소를 모두 더해 부호를 바꾼 것(예, 1열: -(1+0+1/3-1+0)=-1/3)**

Simplex 방법을 이용, 탐색 방향을 결정하기 위한 2차 계획 문제의 풀이(9)

2

	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	Y1	Y2	Y3	Y4	Y5	bi	bi/ai
Y1	1	0	-1	0	1/3	-1	0	0	0	0	1	0	0	0	0	1	-
Y2	0	1	0	-1	1/3	0	-1	0	0	0	0	1	0	0	0	1	-
X8	1/3	1/3	-1/3	-1/3	0	0	0	1	0	0	0	0	1	0	0	2/3	-
Y4	-1	0	1	0	0	0	0	0	**1**	0	0	0	0	1	0	1	**1**
Y5	0	-1	0	1	0	0	0	0	0	1	0	0	0	0	1	1	-
A. Obj.	0	0	0	0	-2/3	1	1	0	**-1**	-1	0	0	1	0	0	w-4	-

3

	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	Y1	Y2	Y3	Y4	Y5	bi	bi/ai
Y1	1	0	-1	0	1/3	-1	0	0	0	0	1	0	0	0	0	1	-
Y2	0	1	0	-1	1/3	0	-1	0	0	0	0	1	0	0	0	1	-
X8	1/3	1/3	-1/3	-1/3	0	0	0	1	0	0	0	0	1	0	0	2/3	-
X9	-1	0	1	0	0	0	0	0	1	0	0	0	0	1	0	1	-
Y5	0	-1	0	1	0	0	0	0	0	**1**	0	0	0	0	1	1	**1**
A. Obj.	-1	0	1	0	-2/3	1	1	0	0	**-1**	0	0	1	1	0	w-3	-

4

	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	Y1	Y2	Y3	Y4	Y5	bi	bi/ai
Y1	**1**	0	-1	0	1/3	-1	0	0	0	0	1	0	0	0	0	1	**1**
Y2	0	1	0	-1	1/3	0	-1	0	0	0	0	1	0	0	0	1	-
X8	1/3	1/3	-1/3	-1/3	0	0	0	1	0	0	0	0	1	0	0	2/3	2
X9	-1	0	1	0	0	0	0	0	1	0	0	0	0	1	0	1	-
X10	0	-1	0	1	0	0	0	0	0	1	0	0	0	0	1	1	-
A. Obj.	**-1**	-1	1	1	-2/3	1	1	0	0	0	0	0	1	1	1	w-2	-

Simplex 방법을 이용, 탐색 방향을 결정하기 위한 2차 계획 문제의 풀이(10)

5

	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	Y1	Y2	Y3	Y4	Y5	bi	bi/ai
X1	1	0	-1	0	1/3	-1	0	0	0	0	1	0	0	0	0	1	-
Y2	0	**1**	0	-1	1/3	0	-1	0	0	0	0	1	0	0	0	1	**1**
X8	0	1/3	0	-1/3	-1/9	1/3	0	1	0	0	-1/3	0	1	0	0	1/3	1
X9	0	0	0	0	1/3	-1	0	0	1	0	1	0	0	1	0	2	-
X10	0	-1	0	1	0	0	0	0	0	1	0	0	0	0	1	1	-
A. Obj.	0	**-1**	0	1	-1/3	0	1	0	0	0	1	0	1	1	1	w-1	-

6

	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	Y1	Y2	Y3	Y4	Y5	bi	bi/ai
X1	1	0	-1	0	1/3	-1	0	0	0	0	1	0	0	0	0	1	-
X2	0	1	0	-1	1/3	0	-1	0	0	0	0	1	0	0	0	1	-
X8	0	0	0	0	-2/9	1/3	1/3	1	0	0	-1/3	-1/3	1	0	0	0	-
X9	0	0	0	0	1/3	-1	0	0	1	0	1	0	0	1	0	2	-
X10	0	0	0	0	1/3	0	-1	0	0	1	0	1	0	0	1	2	-
A. Obj.	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	**w-0**	-

$\mathbf{X}^T_{(1\times10)} = [d_1^+ \quad d_2^+ \quad d_1^- \quad d_2^- \quad u_1 \quad u_2 \quad u_3 \quad s_1 \quad s_2 \quad s_3]$

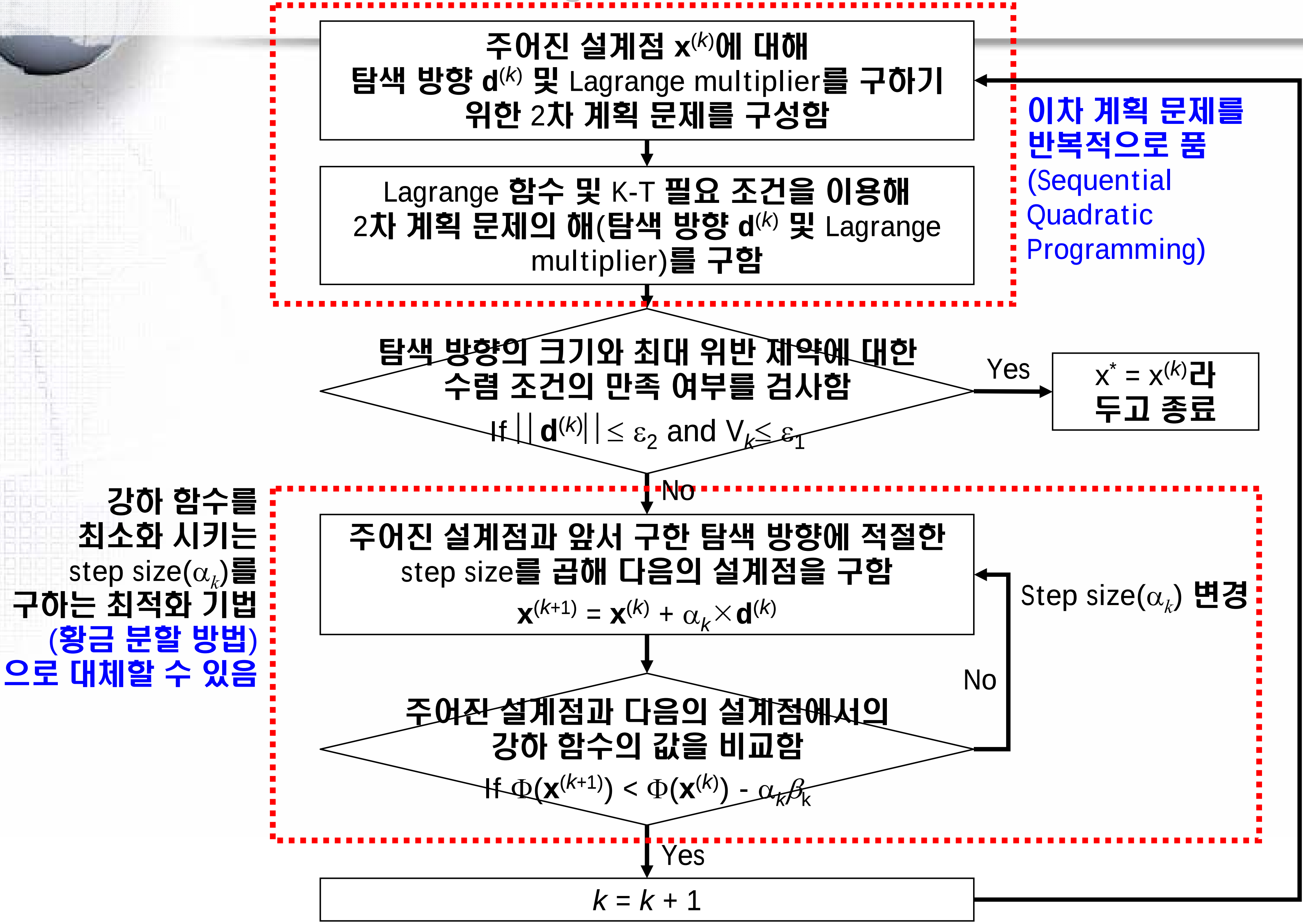
인위 목적 함수가 0이므로
초기 기저 가능해가 구해졌음

이로부터 초기 기저 가능해 중의 하나는 X_1=1, X_2=1, X_3=X_4=X_5=X_6=X_7=X_8=0, X_9=X_{10}=2
한편, 이들은 제약 조건 $X_i X_{3+i} = 0; i = 5\ to\ 7, X_i \ge 0; i = 1\ to\ 10$ 을 모두 만족한다.
따라서 주어진 문제의 최적해는 $d_1 = d_2 = 1, u_1 = u_2 = u_3 = 0, s_1 = 0, s_2 = s_3 = 2$ 이며,
이것은 Kuhn-Tucker 필요 조건을 이용, 해를 직접 구한 방법의 결과와 동일함을 알 수 있다.

➡ Pivot 과정 중 선택 가능한 열 또는 행 또는 "b_i/a_i"의 계수가 중복인 경우,
어떤 것을 선택하느냐에 따라 다른 초기 기저 가능해가 얻어질 수 있음

➡ 비선형 방정식(u_i*s_i=0)을 만족하는 해가 나올 때까지 모든 경우를 확인해 봐야 함

CSD 알고리즘의 Flow Diagram



CSD 알고리즘의 요약

- 단계 1: k=0으로 둔다. $\mathbf{x}^{(0)}$으로 설계 변수의 초기값을 추정한다. 벌칙 매개 변수 R_0, 0과 1 사이의 상수 $\gamma(0 < \gamma < 1)$와 허용되는 제약 조건의 위배 정도와 수렴 기준으로 작은 수 ε_1, ε_2의 적절한 초기값을 선정한다. 일반적으로 $R_0 = 1$과 $\gamma = 0.2$가 적절한 선택이다.

- 단계 2: $\mathbf{x}^{(k)}$에서 목적 함수, 제약 조건과 이들의 경사도(gradient)를 계산한다. 또한 최대 위배 제약 조건 V_k를 계산한다.

- 단계 3: 목적 함수, 제약 조건과 이들이 경사도를 이용하여 2차 계획 문제를 정의하고, 이를 풀어 탐색 방향 $\mathbf{d}^{(k)}(= \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)})$와 Lagrange multiplier $\mathbf{v}^{(k)}$, $\mathbf{u}^{(k)}$를 구한다.

- 단계 4: 수렴 기준 $||\mathbf{d}^{(k)}|| \le \varepsilon_2$을 만족하는지 확인한다. 그리고 최대 위반 제약 조건 $V_k \le \varepsilon_1$을 확인한다. 만일 수렴 기준을 만족하면 현재의 $\mathbf{x}^{(k)}$가 최적해라고 가정하고 종료한다. 그렇지 않다면 다음 단계로 간다.

- 단계 5: Lagrange multiplier의 합 r_k를 계산하여 $R = \max\{R_k, r_k\}$로 둔다.

- 단계 6: 새로운 설계 변수 $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{d}^{(k)}$로 둔다. 여기서 $\alpha = \alpha_k$는 적절한 이동 거리이다. 이동 거리는 탐색 방향 $\mathbf{d}^{(k)}$를 따라 목적 함수(강하 함수, descent function)를 최소화 하여 구한다. 황금 분할 탐색과 같은 방법을 이동 거리를 결정하는 데 이용할 수 있다.

- 단계 7: 현재의 벌칙 매개 변수를 $R_k = R$로 저장한다. 반복 횟수를 $k = k+1$로 수정하고 단계 2로 간다.

제약 최적화 문제의 해결을 위한 SQP-CSD Class의 구현 예

- Part 1: Simplex 방법
  - Phase I/Phase II
- Part 2: QP(Quadratic Programming)
  - 주어진 문제의 목적 함수식을 2차식으로, 제약 조건식들을 1차식으로 근사화하여 탐색 방향을 구함
  - Kuhn-Tucker 필요 조건을 이용하여, 선형 및 비선형 방정식을 구성하여 이를 풀어 탐색 방향을 구함
    - 선형 방정식: Simplex 방법을 이용하여 해를 구함
    - 비선형 방정식: Simplex 방법으로부터 구한 해가 이를 만족하는지 판단하는 용도로 사용함
- Part 3: SQP-CSD 방법
  - SQP(Sequential Quadratic Programming): QP를 연속적(Sequential)으로 풀어 현재의 설계점에서의 탐색 방향을 구함
  - CSD(Constrained Steepest Descent): 원래의 목적 함수식과 제약 조건식들로부터 강하 함수를 구성하고 이를 최소화 하는 이동 거리를 구함

Simplex Class의 구현 예

```
class Simplex
{
public:
	Simplex();
	virtual ~Simplex();

	int m_nRowNo;			// Simplex Table의 행의 수(= 제약 조건식의 수)
	int m_nColumnNo;		// Simplex Table의 열의 수(= 원래의 설계 변수의 수 + 완화 변수의 수 + 잉여 변수의 수
	+ 인위 변수의 수)
	int m_nType;			// Phase I, II 여부

	int* m_pBDVId;			// 기저 변수들의 ID
	double** m_pA;			// 제약 조건식에 해당하는 계수들의 집합
	double* m_pB;			// 제약 조건식의 값들의 집합
	double* m_pC;			// 목적 함수식에 해당하는 계수들의 집합
	double* m_pW;			// 인위 목적 함수식의 설계 변수에 대한 계수들의 집합
	double m_fObj;			// 목적 함수값
	double m_fAObj;			// 인위 목적 함수값

	void InitializeSimplexTable();
	int FindPivotColumn();		// Pivot Column을 결정하는 함수
	int FindPivotRow();		// Pivot Row를 결정하는 함수
	int Pivot(int colNo, int rowNo);	// 주어진 Pivot Column과 Pivot Row로 Simplex Table의 Pivoting을 1번 수행하는 함수
	int CheckEndCondition();	// Simplex 방법의 종료 조건을 판단(목적 함수식 또는 인위 목적 함수식에 해당하는
	계수들이 모두 음이 아닌지 확인)하는 함수
	int Solve();			// Simplex 방법을 실행하는 함수
};
```

QP Class의 구현 예

```
class QP
{
public:
        QP();
        virtual ~QP();

        Simplex BXYD;                    // 선형 방정식 "BX + Y = D"를 해결하기 위한 Simplex
        double** m_pH;                   // H를 나타내는 2차원 배열
        double** m_pA;                   // A를 나타내는 2차원 배열
        double** m_pN;                   // N을 나타내는 2차원 배열
        double* m_pD;                    // 선형 방정식을 푼 결과 Search Direction을 저장한 변수
        double* m_pU;                    // 선형 방정식을 푼 결과 부등호 제약 조건식에 대한 Lagrange Multipliers를 저장한 변수
        double* m_pXi;                   // 선형 방정식을 푼 결과 설계 변수의 양수 조건식에 대한 Lagrange Multipliers를 저장한
        변수
        double* m_pS;                    // 선형 방정식을 푼 결과 부등호 제약 조건식에 대한 완화 변수를 저장한 변수
        double* m_pY;                    // 선형 방정식을 푼 결과 등호 제약 조건식에 대한 Lagrange Multipliers를 저장한 변수
        double* m_pZ;                    // 선형 방정식을 푼 결과 등호 제약 조건식에 대한 Lagrange Multipliers를 저장한 변수

        void ConstructSimplexTable();    // 선형 방정식 "BX + Y = D"에 해당하는 Simplex를 구성하는 함수
        int CheckEndCondition();         // QP 방법의 종료 조건을 판단(U*S = 0 & Xi*D = 0)하는 함수
        int Solve();                     // QP를 실행하는 함수
};
```

SQP-CSD Class의 구현 예

```
class SQPCSD
{
public:
	SQPCSD();
	virtual ~SQPCSD();

	int m_nDVNo;                                    // 설계 변수의 수
	int m_nCFNo;                                    // 제약 조건의 수
	int m_nIterationNo;                             // 반복 회수(k)
	double m_nStepSize;                             // 임시 이동 거리(t)
	double m_nPenaltyParameter;                     // 벌칙 매개 변수(R)
	double* DV;                                     // 설계 변수들을 저장한 배열

	double ObjectiveFunction(int DVNo, double* DV); // 목적 함수식
	void Constraints(int CFNo, double* CF);         // 제약 조건식
	void OFGradients(double* OFG);                  // 목적 함수식에 대한 Gradient Vector
	void CFGradients(double** CFG);                 // 제약 조건식에 대한 Gradient Vector

	int DetermineSearchDirection();                 // SPQ 방법에 따라 탐색 방향을 결정하는 함수
	double CalculatePenaltyParameter();             // 벌칙 매개 변수(R)를 계산하는 함수
	double CalculateMaxConstraintsViolation()       // 제약 조건의 최대 위배 값(V)을 결정하는 함수
	double ConstructDescentFunction();              // 강하 함수를 구성하는 함수
	int DetermineStepSize();                        // CSD 방법에 따라 이동 거리를 결정하는 함수
	int CheckEndCondition();                        // SQP-CSD 방법의 종료 조건을 판단
	int Solve();                                    // SQP-CSD 방법을 실행하는 함수
};
```

5.6 최적화 기법을 이용한 선박의 최적 주요 치수 결정 문제

Advanced
Ship
Design
Automation
Laboratory

상선의 주요 치수 결정을 위한 최적화 문제

- **주요 치수 선정 기준(목적 함수)**
  - 최소 건조비 또는 최소 중량 또는 최소 운송비
- **주어진 값(선주 요구 조건)**
  - 재화 중량(Deadweight; DWT)
  - 화물창 용적(Cargo Capacity; CC_{req})
  - 최대 흘수(T_{max})
  - 선속(V)
- **구하는 값(설계 변수)**
  - 선박의 길이(Length; L)
  - 선박의 폭(Breadth; B)
  - 선박의 깊이(Depth; D)
  - 방형 계수(Block Coefficient; C_B)

- **제약 조건**
  - 부력·중량 평형 조건(선박의 경하 중량 추정 필요)
  - 화물창 용적 요구 조건(화물창 용적 계산 필요)
  - 최소 요구 건현 조건(건현 계산 필요)

상선의 최적 주요 치수 결정 문제

구하는 값(설계 변수) L, B, D, C_B

길이 폭 깊이 방형 계수

주어진 값(선주 요구 조건) $DWT, CC_{req}, T_{\max}(=T), V$

재화 중량 요구 화물창 용적 최대 흘수 선속

부력(buoyancy)-중량(displacement) 평형 조건(등호 제약 조건)

$$L \cdot B \cdot T \cdot C_B \cdot \rho_{sw} \cdot C_\alpha = DWT_{given} + LWT(L, B, D, C_B)$$

$$= DWT_{given} + C_s \cdot L^{1.6}(B+D) + C_o \cdot L \cdot B + C_{ma} \cdot NMCR$$

$$= DWT_{given} + C_s \cdot L^{1.6}(B+D) + C_o \cdot L \cdot B$$

$$+ C_{power} \cdot (L \cdot B \cdot T \cdot C_B)^{2/3} \cdot V^3$$

요구되는 화물창 용적(cargo capacity) 조건(부등호 제약 조건)

$$CC_{req} \le C_{CH} \cdot L \cdot B \cdot D$$

최소 요구 건현 조건(부등호 제약 조건)

$$D \ge T + C_{FB} \cdot D$$

목적 함수(주요 치수 선정 기준)

$$Building\ Cost = C_{PS} \cdot C_s \cdot L^{1.6}(B+D) + C_{PO} \cdot C_o \cdot L \cdot B + C_{PM} \cdot C_{ma} \cdot NMCR$$

➡ **미지수 4개, 등호 제약 조건 1개, 부등호 제약 조건 2개인 최적화 문제**

상선의 최적 주요 치수 결정 문제의 수학적 정식화(1)

Find L, B, D, C_B

Minimize $Building\ Cost = C_{PS} \cdot C_s \cdot L^{1.6}(B+D) + C_{PO} \cdot C_o \cdot L \cdot B + C_{PM} \cdot C_{ma} \cdot NMCR$

Subject to * 부력(buoyancy)-중량(displacement) 평형 조건

$$
\begin{aligned}
L \cdot B \cdot T \cdot C_B \cdot \rho_{sw} \cdot C_\alpha &= DWT_{given} + LWT(L, B, D, C_B) \\
&= DWT_{given} + C_s \cdot L^{1.6}(B+D) + C_o \cdot L \cdot B + C_{ma} \cdot NMCR \\
&= DWT_{given} + C_s \cdot L^{1.6}(B+D) + C_o \cdot L \cdot B \\
&\quad + C_{power} \cdot (L \cdot B \cdot T \cdot C_B)^{2/3} \cdot V^3
\end{aligned}
$$

* 요구되는 화물창 용적 조건

$$CC_{req} \le C_{CH} \cdot L \cdot B \cdot D$$

* 최소 요구 건현 조건

$$D \ge T + C_{FB} \cdot D$$

➡ **미지수 4개, 등호 제약 조건 1개, 부등호 제약 조건 2개인 최적화 문제**

상선의 최적 주요 치수 결정 문제의 수학적 정식화(2) - 목적 함수

- **선박의 건조비(Building Cost) 최소**

  또는
- **요구 운임(RFR; Required Freight Rate, 단위 수송 화물 당 수송 원가) 최소**

  또는
- **연료 소모율(Fuel Consumption) 최소**

- **위 목적 함수는 L, B, D, C_B의 함수(closed form)로 표현 가능**
  - **선박 건조비의 계산 예**

$$Building\ Cost = C_{PS} \cdot C_s \cdot L^{1.6}(B+D) + C_{PO} \cdot C_o \cdot L \cdot B + C_{PM} \cdot C_{ma} \cdot NMCR$$

상선의 최적 주요 치수 결정 문제의 수학적 정식화(3) - 제약 조건

- 부력(Buoyancy)-중량(Displacement) 평형 조건

$$L \cdot B \cdot T \cdot C_B \cdot \rho \cdot (1+\alpha) = DWT_{given} + LWT$$

그런데 경하 중량(LWT; lightweight)은 선각 중량, 의장 중량, 기관부 중량으로 세분화할 수 있음
따라서 위 식은 아래와 같이 표현할 수 있음

$$L \cdot B \cdot T \cdot C_B \cdot \rho \cdot (1+\alpha) = DWT_{given} + LWT$$

$$= C_s \cdot L^{1.6}(B+D) + C_o \cdot L \cdot B + C_{ma} \cdot NMCR + DWT$$

한편, NMCR은 저항 및 마력 추정을 통해 주기관을 선정하여 결정해야 하나
초기 단계에서는 기준선의 admiralty 계수(C_{ad})로부터 개략적으로 추정할 수 있음

$C_{ad} = \dfrac{\Delta^{2/3} \times V^3}{DHP}$ 로부터 $DHP = \dfrac{\Delta^{2/3} \times V^3}{C_{ad}}$ 이며, 따라서

$NMCR = C_1 \cdot DHP = C_1 \cdot \dfrac{\Delta^{2/3} \times V^3}{C_{ad}} = C_2 \cdot \Delta^{2/3} \times V^3$ 이며, 최종적으로

$$L \cdot B \cdot T \cdot C_B \cdot \rho \cdot (1+\alpha) = DWT_{given} + LWT(L, B, D, C_B)$$

$$= DWT_{given} + C_s \cdot L^{1.6}(B+D) + C_o \cdot L \cdot B + C_{ma} \cdot C_2 \cdot \Delta^{2/3} \cdot V^3$$

$$= DWT_{given} + C_s \cdot L^{1.6}(B+D) + C_o \cdot L \cdot B$$

$$+ \underline{C_{power}} \cdot (L \cdot B \cdot T \cdot C_B)^{2/3} \cdot V^3$$

$$C_{power} = C_{ma} \cdot C_2$$

상선의 최적 주요 치수 결정 문제의 수학적 정식화(4) - 제약 조건

- 요구되는 화물창 용적(CC; Cargo Capacity) 조건

$$CC = f(L, B, D, C_B, L_{APT}, L_{ER}, L_{FPT}, DB_{Height})$$

* L_{APT}: 선미창 길이, L_{ER}: 기관부 길이, L_{FPT}: 선수창 길이, DB_{Height}: 이중저 높이

화물창 용적의 간이 추정 예 1

$$CC = C_{CH} \cdot L \cdot B \cdot D$$

* C_{CH}: 화물창 용적 계수

$$C_{CH} = \left(\frac{CC}{L \cdot B \cdot D} \right)_p$$

기준선의 값

상선의 최적 주요 치수 결정 문제의 수학적 정식화(5) - 제약 조건

- **요구되는 화물창 용적**(CC; Cargo Capacity) **조건**

$$CC = f(L, B, D, C_B, L_{APT}, L_{ER}, L_{FPT}, DB_{Height})$$

* L_{APT}: **선미창 길이**, L_{ER}: **기관부 길이**, L_{FPT}: **선수창 길이**, DB_{Height}: **이중저 높이**

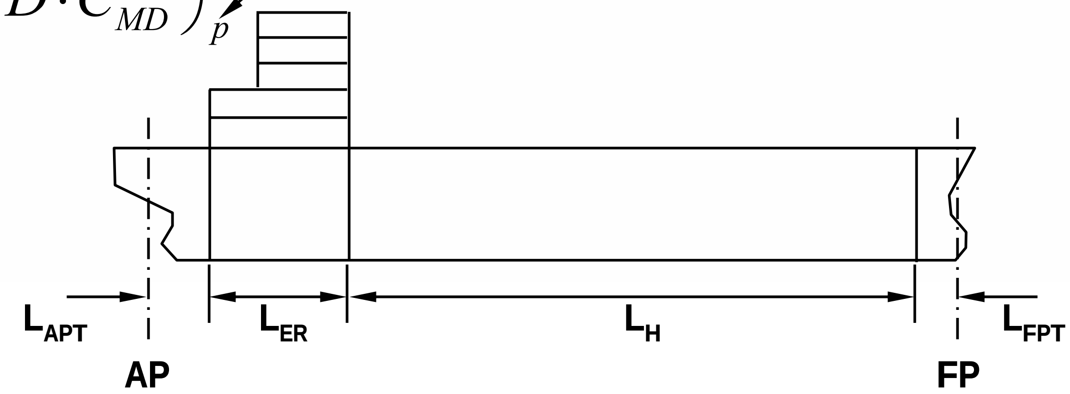
화물창 용적의 간이 추정 예 2

$$CC = C_{CH} \cdot (\underbrace{L_{BP} - L_{APT} - L_{ER} - L_{FPT}}_{\text{화물창 길이 } L_H}) \cdot B \cdot D \cdot C_{MD}$$

* C_{CH}: **화물창 용적 계수**, C_{MD}: **갑판까지의 중앙횡단면 계수**

$$C_{CH} = \left(\frac{CC}{L_H \cdot B \cdot D \cdot C_{MD}} \right)_p$$

기준선의 값



상선의 최적 주요 치수 결정 문제의 수학적 정식화(6) - 제약 조건

- **최소 요구 건현 조건**

$$D \geq T + Freeboard(L, B, D, C_B)$$

최소 요구 건현은 초기 단계에서 아래와 같이 개략적으로 계산할 수 있음

$$D \geq T + C_{FB} \cdot D \qquad C_{FB} = \left(\frac{Freeboard}{D} \right)_p$$

기준선의 값

실제로는 다음과 같은 식으로부터 계산을 해야 함

<u>ICLL 1966 **요구 건현**</u>

$$Freeboard = f(L_f, D, C_B, Superstructure_{Length}, Superstructure_{Height}, Sheer)$$

상선의 최적 주요 치수 결정 문제의 수학적 정식화(7) - 제약 조건

- 초기 복원성 요구 조건

$$0.04B \le GM \le 4\pi^2(0.4B)^2/(gTr^2)$$

* Tr: 횡동요 주기(Roll Period)

$$GM = T(0.9 - 0.3C_M - 0.1C_B) + B\left(\frac{0.08}{\sqrt{C_M}} \cdot \frac{B}{T}\right) - D\left(\frac{1.6}{L^{0.2}}\right)$$

- 조종성 관점에서의 비만 계수(Obesity Coefficient) 요구 조건

$$\underbrace{C_B/(L/B)}_{\text{비만 계수}} \le 0.15$$

- Watson & Gilfillan에 의한 C_B 추천 값

$$C_B \le 0.70 + 0.125\tan^{-1}((23 - 100Fn)/4)$$

재화 중량 150,000톤 살물선(기준선)의 자료를 이용한 재화 중량 160,000톤 살물선의 최적 주요 치수 결정 문제(1)

재화 중량 150,000톤 살물선(기준선)의 주요 요목

항목		실적선	설계선	비고
주요제원	L_{OA}	abt. 274.00 m	max. 284.00 m 이내	
	L_{BP}	264.00 m		
	B_{mld}	45.00 m	45.00 m	
	D_{mld}	23.20 m		
	T_{mld}	16.90 m	17.20 m	
	T_{scant}	16.90 m	17.20 m	
Deadweight		150,960 ton	160,000 ton	at 17.20 m
Speed		13.5 kts	13.5 kts	90 % MCR (with 20 % SM)
M/E	TYPE	B&W 5S70MC		
	NMCR	17,450 HP × 88.0 RPM		Derating Ratio = 0.9
	DMCR	15,450 HP × 77.9 RPM		E.M = 0.9
	NCR	13,910 HP × 75.2 RPM		
FOC	SFOC	126.0 g/HP.H		NCR 기준
	TON/DAY	41.6		
Cruising Range		28,000 N/M	26,000 N/M	
중앙 단면 형상		Single Hull Double Bottom/Hopper /Top Side Wing Tank	Single Hull Double Bottom/Hopper /Top Side Wing Tank	
Capacity	Cargo	abt. 169,380 m^3	abt. 179,000 m^3	Hatch Coaming 포함
	Fuel Oil	abt. 3,960 m^3		Total
	Fuel Oil	abt. 3,850 m^3		Bunker Tank Only
	Ballast	abt. 48,360 m^3		F.P 및 A.P Tank 포함

재화 중량 150,000톤 살물선(기준선)의 자료를 이용한 재화 중량 160,000톤 살물선의 최적 주요 치수 결정 문제(2)

재화 중량 150,000톤 살물선(기준선)의 추가 자료 및 기타 자료

항목	값
경하 중량(LWT)	18,269 ton
선각 중량(Ws)	15,289 ton
의장부 중량(Wo)	1,694 ton
기관부 중량(Wm)	1,281 ton
건현(Fb)	6.996 m
방형 계수(Cb)	0.8214
Admiralty 계수(Cad)	644.4139
선각 강재비 관련 계수(C_{PS})	972.80
의장부 비용 관련 계수(C_{PO})	20,256
기관부 비용 관련 계수(C_{PM})	7,760

기타 사항

1. 추가 제약 조건으로서 “조종성 관점에서의 비만 계수 요구 조건”과 “Watson & Gilfillan에 의한 Cb 추천 값”을 고려할 것

재화 중량 160,000톤 살물선의 최적 주요 치수 결정 문제에 대한 최적화 결과

Minimize ship building cost

		Unit	MFD	MS	GA	HYBRID w/o Refine	HYBRID with Refine
GIVEN	DWT	ton	160,000				
	Cargo Capacity	m^3	179,000				
	T_{max}	m	17.2				
	V	knots	13.5				
L		m	265.54	265.18	264.71	264.01	263.69
B		m	45.00	45.00	45.00	45.00	45.00
D		m	24.39	24.54	24.68	24.71	24.84
C_B		-	0.8476	0.8469	0.8463	0.8427	0.8420
D_P		m	8.3260	8.3928	8.4305	8.4075	8.3999
P_i		m	5.8129	5.8221	5.7448	5.7491	5.7365
A_E/A_O		-	0.3890	0.3724	0.3606	0.3618	0.3690
Building Cost		$	59,889,135	59,888,510	59,863,587	59,837,336	59,831,834
Iteration No		-	10	483	96	63	67
CPU Time		sec	4.39	209.58	198.60	184.08	187.22

* MFD: Method of feasible directions, MS: Multi-start local optimization method, GA: Genetic algorithm, HYBRID: Global-local hybrid optimization method
* **테스트 시스템**: Pentium 3 866MHz, 512MB RAM

Ship Class의 구현 예(1)

```
class Ship
{
public:
	Ship();
	virtual ~Ship();

	// 선주 요구 조건(Given)
	double m_fDWT;                    // 재화 중량
	double m_fCCrequirement;          // 요구 화물창 용적
	double m_fTmax;                   // 흘수
	double m_fVs;                     // 선속 in Knots

	// 선박 주요 치수(Find)
	double m_fL;                      // 수선간 길이(LBP)
	double m_fB;                      // 형 폭(Bmld)
	double m_fD;                      // 형 깊이(Dmld)
	double m_fCb;                     // 방형 계수(Block Coefficient)

	double m_fCC;                     // 화물창 용적
	double m_fFB;                     // 건현
	double m_fDisplacement;           // 배수량
	double m_fLWT;                    // 경하 중량(Light Weight)

	// 기준선 정보로부터 계산되는 변수
	double m_fAppendageFactor;        // Appendage Factor(1 + alpha)
	double m_fCs;                     // 선각 중량 계수
	double m_fCo;                     // 의장부 중량 계수
	double m_fCma;                    // 기관부 중량 계수
	double m_fCch;                    // 화물창 용적 계수
	double m_fCfb;                    // 건현 계수
	double m_fCps;                    // 건조비 추정을 위한 선각 중량 관련 계수
	double m_fCpo;                    // 건조비 추정을 위한 의장부 중량 관련 계수
	double m_fCpm;                    // 건조비 추정을 위한 기관부 중량 관련 계수

	// 계속
};
```

Ship Class의 구현 예(2)

```
class Ship
{
public:
    // 선박 주요 치수에 대한 상하한값
    double m_fLlower, m_fLupper;                 // 수선간 길이에 대한 하/상한값
    double m_fBlower, m_fLupper;                 // 형 폭에 대한 하/상한값
    double m_fDlower, m_fDupper;                 // 형 깊이에 대한 하/상한값
    double m_fCblower, m_fCbupper;               // 방형 계수에 대한 하/상한값

    void CalculateParentShipData();              // 기준선 정보로부터 관련 계수를 계산하는 함수
    double CalculateBuildingCost();              // 건조비를 계산하는 함수
    double CalculateWS();                        // 선각 중량을 계산하는 함수
    double CalculateWO();                        // 의장부 중량을 계산하는 함수
    double CalculateWM();                        // 기관부 중량을 계산하는 함수
    double CalculateCC();                        // 화물창 용적을 계산하는 함수
    double CalculateFB();                        // 건현을 계산하는 함수

    double BuoyancyDisplacementCondition();      // 부력-중량 평형 조건을 계산하는 함수
    double CCRequirementCondition();             // 화물창 요구 조건을 계산하는 함수
    double FBRequirementCondition();             // 건현 요구 조건을 계산하는 함수
    double ObesityCoefficientCBCondition();      // 조종성 관점에서의 비만 계수 요구 조건
    double WGCBCondition();                      // Watson & Gilfillan에 의한 Cb 추천값
    void DVUpperLowerCondition(int DVNo, double* CF);  // 설계 변수의 상하한값에 대한 조건을 계산하는 함수
};
```

Ch6. 제약 비선형 최적화 프로그램 소개

6.1 제약 비선형 최적화 프로그램의 구성

6.2 제약 비선형 최적화 프로그램의 사용 방법

6.3 제약 비선형 최적화 프로그램을 이용한 최적 설계 예

Advanced
Ship
Design
Automation
Laboratory

6.1 제약 비선형 최적화 프로그램의 구성(1)

- 비선형 최적화 라이브러리 'EzOptimizer'
  - 다양한 최적화 알고리즘을 포함
  - C++ 언어로 구현
  - 최적화 모듈을 사용자의 프로그램에 쉽게 이식 가능
  - 최적화 과정을 제어할 수 있는 다양한 옵션 지정 가능
  - 헤더 파일(EzOptimizer.h)과 라이브러리 파일(EzOptimizerLibraryD.lib/EzOptimizerLibraryR.lib)로 구성

- 'EzOptimizer'를 위한 프리-컴파일러 'EzPreCompiler'
  - 'EzOptimizer'의 보다 편리한 사용을 위해 개발
  - 키워드(Keyword) 인식 방식
  - 'EzPreCompile'용 입력 파일을 C++ 소스 파일로 변환
  - 특정 키워드에 의한 다중 최적화 블록도 처리 가능
  - DOS 및 Windows 버전 개발

6.1 제약 비선형 최적화 프로그램의 구성(2)

입력 파일

EzPreCompiler

C++ 소스 파일

EzOptimizer

최적화 결과

Pre-Compiler for EzOptimizer

- 키워드 인식 방식
- DOS 및 Windows 환경 지원

Multi-Objective Hybrid NLP* Library

내장된 최적화 알고리즘

- Method of Feasible Directions
- Sequential Quadratic Programming
- Sequential Linear Programming
- Genetic Algorithm
- Hybrid Optimization
- Multi-Objective Programming
- ...

* NLP: Non-Linear Programming

* 1: 노명일, 협동 최적화 방법에 의한 다분야 최적화 기법에 관한 연구, 서울대학교 조선해양공학과 석사학위논문, 2000.2

* 2: Kyu-Yeul Lee, Seon-Ho Cho, Myung-Il Roh, "An Efficient Global-Local Hybrid Optimization Method Using Design Sensitivity Analysis", International Journal of Vehicle Design(SCIE), Vol. 28, No. 4, pp.300~317, 2002.7

6.2 제약 비선형 최적화 프로그램의 사용 방법

1. 사용자가 작성하는 프로그램의 디렉토리에 "EzOptimizer.h"와 "EzOptimizerD.lib"(또는 "EzOptimizerR.lib")를 복사

2. 사용자가 작성하는 프로그램의 Source Code 상단에 다음과 같은 내용을 추가

```
#include "EzOptimizer.h"
```

3. EzOptimizer를 사용하는 프로그램 작성

4. 사용자가 작성하는 프로그램의 프로젝트 파일에 "EzOptimizerD.lib"(또는 "EzOptimizerR.lib")를 포함

5. 사용자가 작성한 프로그램을 컴파일, 링크 후 실행

6.3 제약 비선형 최적화 프로그램을 이용한 최적 설계 예 - 수학 최적화 문제

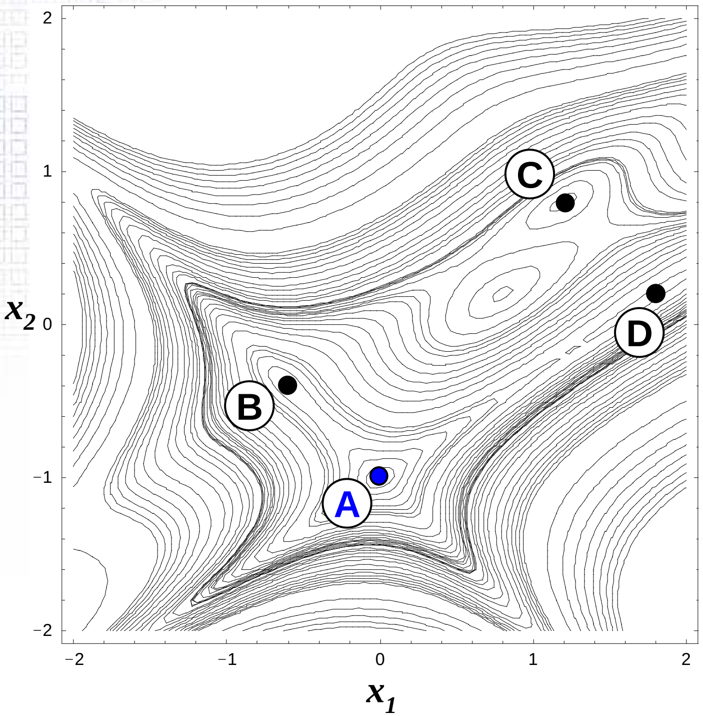
Goldstein-Price Function

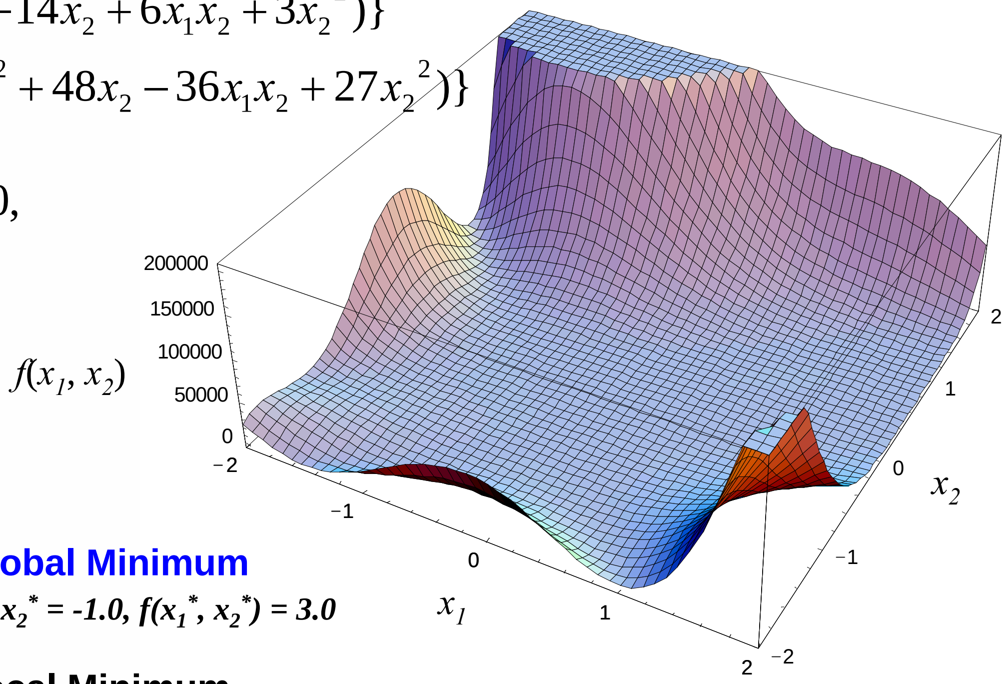
Minimize

$$f(x_1,x_2)=\{1+(x_1+x_2+1)^2\cdot(19-14x_1+3x_1{}^2-14x_2+6x_1x_2+3x_2{}^2)\}$$
$$\cdot\{30+(2x_1-3x_2)^2\cdot(18-32x_1+12x_1{}^2+48x_2-36x_1x_2+27x_2{}^2)\}$$

Subject to

$$g_1(x_1,x_2)=-2-x_1\le 0,\ g_2(x_1,x_2)=-2-x_2\le 0,$$
$$g_3(x_1,x_2)=x_1-2\le 0,\ g_4(x_1,x_2)=x_2-2\le 0$$





A : Global Minimum
$x_1{}^* = 0.0,\ x_2{}^* = -1.0,\ f(x_1{}^*, x_2{}^*) = 3.0$

B : Local Minimum
$x_1{}^* = -0.6,\ x_2{}^* = -0.4,\ f(x_1{}^*, x_2{}^*) = 30.0$

C : Local Minimum
$x_1{}^* = 1.2,\ x_2{}^* = 0.8,\ f(x_1{}^*, x_2{}^*) = 840.0$

D : Local Minimum
$x_1{}^* = 1.8,\ x_2{}^* = 0.2,\ f(x_1{}^*, x_2{}^*) = 84.0$

6.3 제약 비선형 최적화 프로그램을 이용한 최적 설계 예
- EzPreCompiler용 입력 파일 작성

```
$$ EzOptimizer Start
[Optimization Method]
    MFD
[Print Option]
    SMALL
[Design Variables]
    x1, 0.0, -2.0, 2.0        ← 제약 조건을 설계 변수에 대한 상·하한값으로 표현
    x2, 0.0, -2.0, 2.0
[Objective Function]
    MINIMIZE f = (1.0+pow(x1+x2+1.0, 2.0)*(19.0-14.0*x1+3.0*pow(x1, 2.0)-
    14.0*x2+6.0*x1*x2+3.0 *pow(x2, 2.0))) * (30.0+pow(2.0*x1-3.0*x2, 2.0)*
    (18.0-32.0*x1+12.0*pow(x1, 2.0)+48.0*x2-36.0*x1*x2+27.0*pow(x2, 2.0)))
$$ EzOptimizer End
```

6.3 제약 비선형 최적화 프로그램을 이용한 최적 설계 예
- EzPreCompiler에 의해 자동 생성된 C++ 파일

```
EzOptimizerConfiguration MyOptimizerConfiguration00(MFD, SMALL, 2, 0, MINIMIZE);
EzOptimizer MyOptimizer00(MyOptimizerConfiguration00);

int ResultFlag = 0;
double* Design_Value = MyOptimizer00.GetDesignValueAddress();
double* Lower_Value = MyOptimizer00.GetLowValueAddress();
double* Upper_Value = MyOptimizer00.GetUpperValueAddress();
double* Objective_Value = MyOptimizer00.GetObjectiveValueAddress();

Low_Value[ 0] = -2.000000; Design_Value[ 0] = 0.000000; Upper_Value[ 0] = 2.000000;
Low_Value[ 1] = -2.000000; Design_Value[ 1] = 0.000000; Upper_Value[ 1] = 2.000000;
```

➡ 제약 조건을 설계 변수에 대한 상·하한값으로 표현

```
while ((ResultFlag = MyOptimizer00.Optimization()) == 1) {
    *Objective_Value = (1.0+pow(Design_Value[0]+Design_Value[1]+1.0, 2.0)*(19.0-
    14.0*Design_Value[0]+3.0*pow(Design_Value[0], 2.0)-
    14.0*Design_Value[1]+6.0*Design_Value[0]*Design_Value[1] +3.0 *pow(Design_Value[1], 2.0))) *
    (30.0+pow(2.0*Design_Value0]-3.0*Design_Value[1], 2.0)* (18.0-
    32.0*Design_Value[0]+12.0*pow(Design_Value[0], 2.0)+48.0*Design_Value[1]-
    36.0*Design_Value[0]*Design_Value[1]+27.0*pow(Design_Value[1], 2.0)));
}

if (ResultFlag == 0) { x1 = Design_Value[ 0]; x2 = Design_Value[ 1]; f = *Objective_Value; }
else if (ResultFlag == -1) { MyOptimizer00.GetErrorMessage(); }
```

6.3 제약 비선형 최적화 프로그램을 이용한 최적 설계 예
- EzOptimizer를 이용한 최적화 결과

최적화 알고리즘에 따른 결과 비교

	True Solution	MFD	MS	GA	HYBRID w/o Refine	HYBRID with Refine
x1	0.0000	-0.6000	0.0000	0.0081	0.0000	0.0000
x2	-1.0000	-0.4000	-1.0000	-1.0032	-1.0000	-1.0000
f	3.0000	30.0000	3.0000	3.0262	3.0000	3.0000
Iteration No	-	5	154	78	33	36
CPU Time(s)		0.03	0.78	1.23	0.73	0.75

Local Minimum ⬆

⬆
개선된 유전 알고리즘으로부터 얻어진 근사 최적점에 대해
local optimization을 **수행함으로써 정확한 전역 최적해 도출**

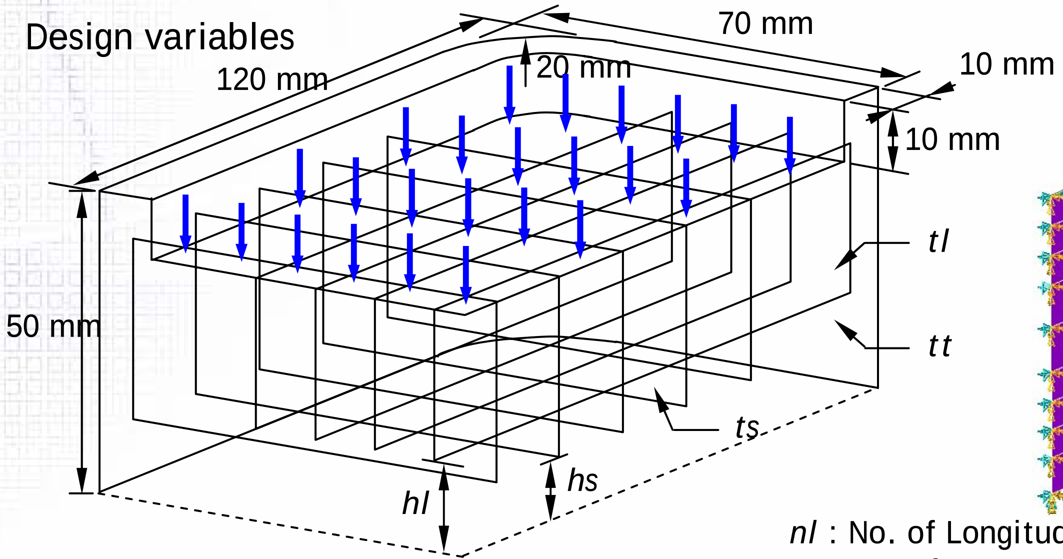
* MFD: Method of feasible directions, MS: Multi-start local optimization method, GA: Genetic algorithm, HYBRID: Global-local hybrid optimization method
* **테스트 시스템**: Pentium 3 866MHz, 512MB RAM

6.3 제약 비선형 최적화 프로그램을 이용한 최적 설계 예
- Ribbed Tray 설계 문제

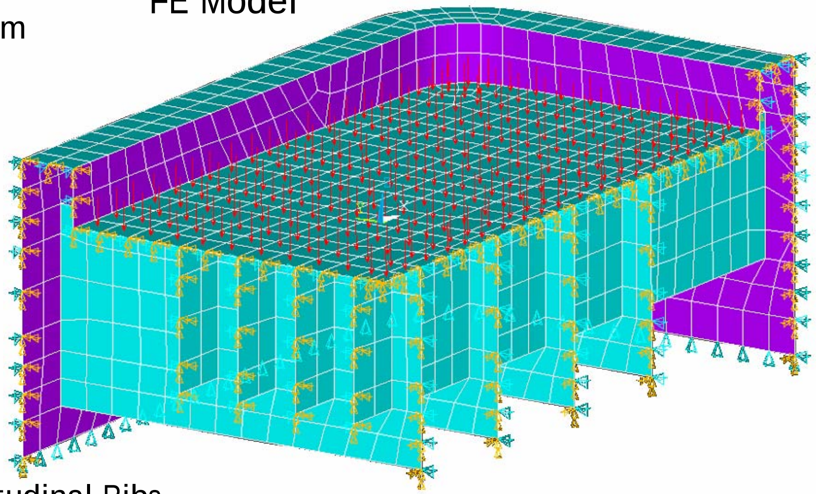
Minimize $\quad Volume = f(nl, ns, tl, ts, tt, hl, hs)$ **Objective Function**

Subject to $\quad \delta_{\max} \leq 1.0(mm) \quad \sigma_{\max} \leq 22.5(MPa)$ **Constraints**

Design variables



FE Model



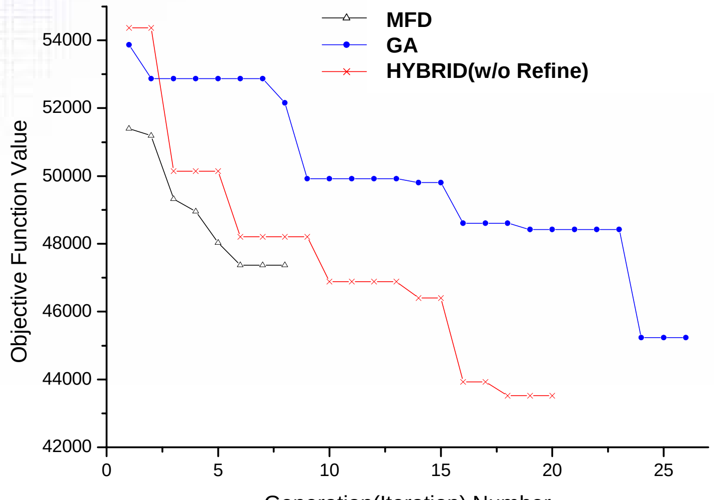
E = 3.1 × 10^3 *MPa*
ν = 0.35
Surface Load on Cavity = 0.15 *MPa*

nl : No. of Longitudinal Ribs
ns : No. of Transversal Ribs
tl : Thickness of Longitudinal Ribs
ts : Thickness of Transversal Ribs
tt : Thickness of Tray
hl : Gap at Bottom of Longitudinal Ribs
hs : Gap at Bottom of Transversal Ribs

* Kyu-Yeul Lee, Seon-Ho Cho, Myung-Il Roh, "An Efficient Global-Local Hybrid Optimization Method Using Design Sensitivity Analysis", International Journal of Vehicle Design(SCIE), Vol. 28, No. 4, pp.300~317, 2002.7

6.3 제약 비선형 최적화 프로그램을 이용한 최적 설계 예
- Ribbed Tray 문제에 대한 최적화 결과(1)

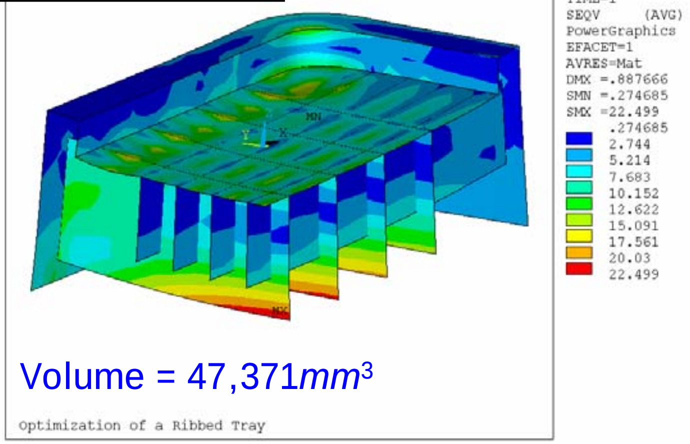
	Unit	MFD	GA	HYBRID	
				w/o Refine	with Refine
Volume	mm^3	47,370.662	45,236.305	43,520.642	42,507.595
nl		4	3	4	4
ns		4	4	4	4
tl	*mm*	1.307021	1.199413	1.007820	0.995354
ts	*mm*	1.313513	1.355816	1.332356	1.327935
tt	*mm*	1.067905	1.156403	1.025415	1.008237
hl	*mm*	16.779161	16.434995	16.998045	18.129324
hs	*mm*	6.073858	5.619746	5.482893	5.389428
Iteration No	-	8	26	20	25
CPU Time	*sec*	2,420.38	13,091.02	19,767.95	20,973.59



* MFD: Method of feasible directions, MS: Multi-start local optimization method, GA: Genetic algorithm, HYBRID: Global-local hybrid optimization method
* **테스트 시스템**: Pentium 3 866MHz, 512MB RAM

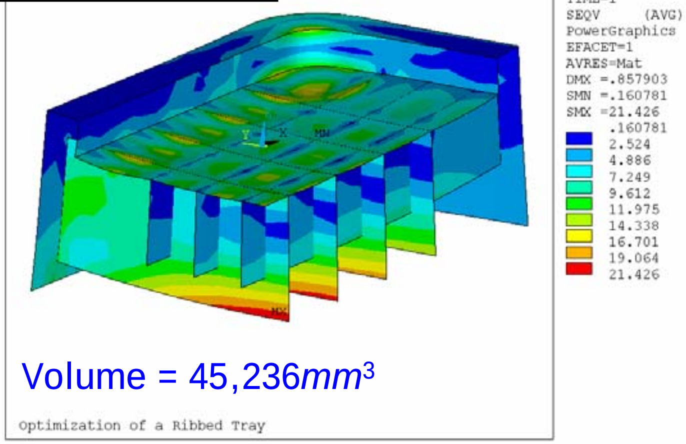
6.3 제약 비선형 최적화 프로그램을 이용한 최적 설계 예
- Ribbed Tray 문제에 대한 최적화 결과(2)

MFD



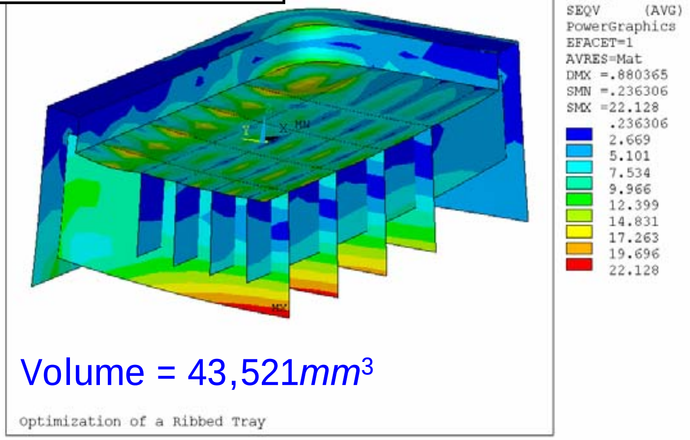
Volume = 47,371mm^3

Conventional GA



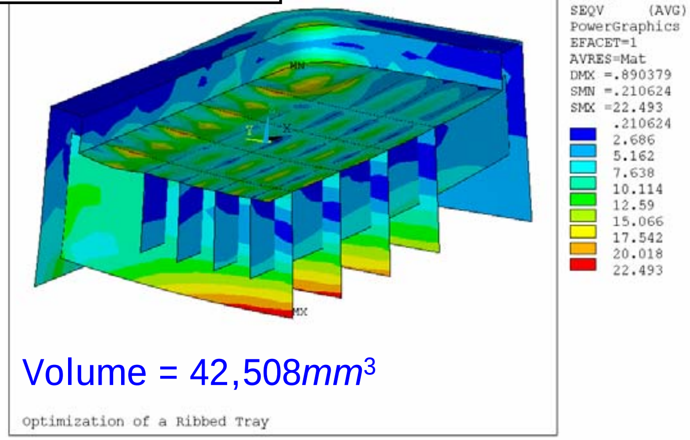
Volume = 45,236mm^3

Proposed Method (w/o Refinement)



Volume = 43,521mm^3

Proposed Method (with Refinement)



Volume = 42,508mm^3

6.3 제약 비선형 최적화 프로그램을 이용한 최적 설계 예
- 건조비를 최소화 하는 중앙 단면 설계

Find $x_i, i = 1, \Lambda, 16$

Minimize *Building Cost*

Subject to

$$t_{i,\min} - x_i \le 0,\ i = 6, \Lambda, 16$$

: minimum plate thickness

$$Z_{\min}^{deck} - Z^{deck} \le 0$$

: minimum section modulus at deck

$$Z_{\min}^{bottom} - Z^{bottom} \le 0$$

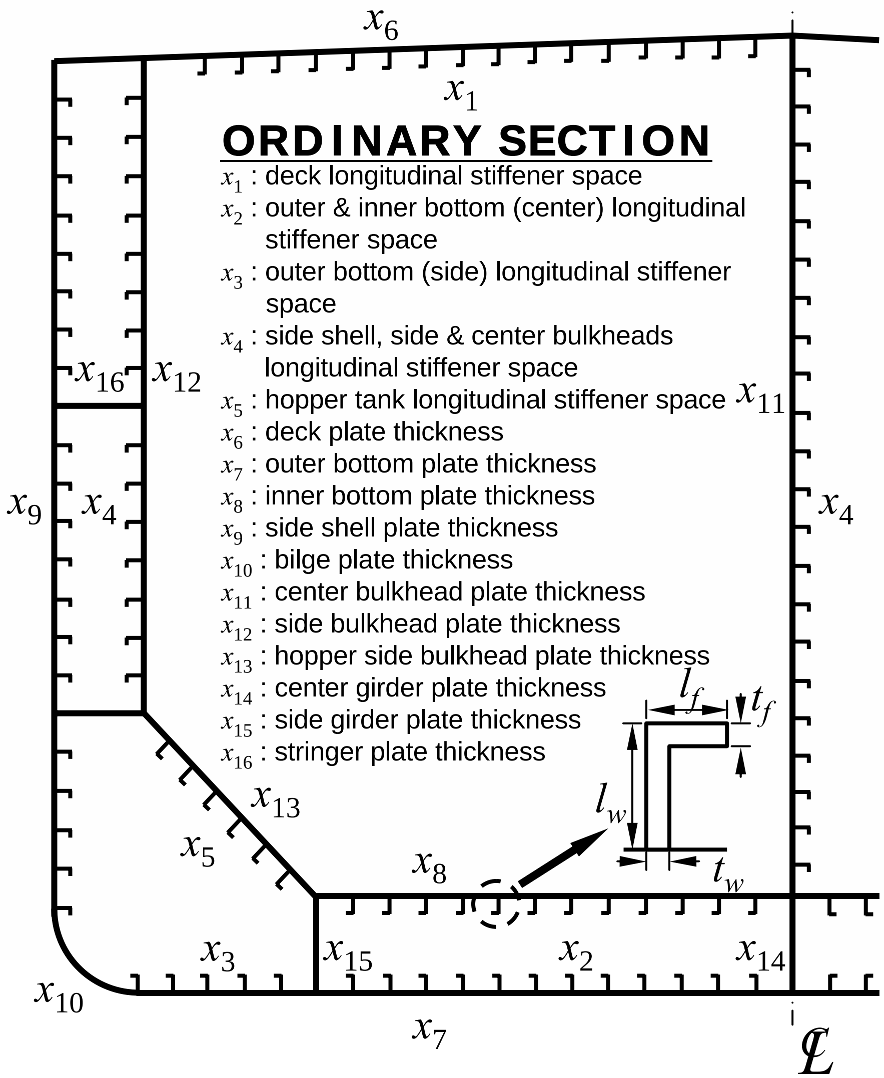
: minimum section modulus at bottom

$$\sigma^{deck} - \eta^{deck} \sigma_c^{deck} \le 0$$

: critical buckling stress at deck

$$\sigma^{bottom} - \eta^{bottom} \sigma_c^{bottom} \le 0$$

: critical buckling stress at bottom



* Kyu-Yeul Lee, Myung-Il Roh, "An Efficient Genetic Algorithm Using Gradient Information for Ship Structural Design Optimization", Journal of Ship Technology Research, Vol. 48, No. 4, pp.161-170, 2001.

건조비를 최소화 하는 중앙 단면 설계 문제의 수학적 정식화(1)

- 목적 함수
  - Midship Section의 단위길이당 건조비

$$Building\ Cost = Material\ Cost + Labour\ Cost\ \ [\$/m]$$

$$Material\ Cost = Weight\,[ton] \times Unit\ Material\ Cost\,[\$/ton]$$

: midship section의 단위 길이당 재료비

$$LabourCost = Welding\ Cost + Painting\ Cost$$

: midship section의 단위 길이당 인건비

$$Weight = \rho \cdot A\ \ [ton]$$

: midship section의 단위 길이당 중량

$$Welding\ Cost = 2 \cdot Fw \cdot N\ \ [\$/m]$$

: midship section의 단위 길이당 용접비

$$Painting\ Cost = Fp \cdot G\ \ [\$/m]$$

: midship section의 단위 길이당 도장비

ρ : steel's mass density, 7.85[ton/m^3]
A : area of the midship section
Fw: welding cost per unit length
N : number of the plates and stiffeners
Fp : painting cost per unit area
G : summation of the girth length of the plates and stiffeners

건조비를 최소화 하는 중앙 단면 설계 문제의 수학적 정식화(2)

- 제약 조건
  - Minimum Plate Thickness에 관한 조건

$$t_{i,\min} - x_i \le 0\,,\;\; i = 6, \Lambda\;, 16$$

$$t_{i,\min} = \max\left(\frac{15.8 \cdot s_i \cdot \sqrt{p_i}}{\sqrt{\sigma_i}} + t_k, c_i + \frac{k_i \cdot L}{\sqrt{f_i}} + t_k \right) [mm]$$

  - Minimum Section Modulus에 관한 조건(Deck, Bottom)

$$Z_{\min}^{deck} - Z^{deck} \le 0 \quad Z_{\min}^{bottom} - Z^{bottom} \le 0$$

$$Z_{\min}^{deck} = \max\left(\frac{C_W \cdot L^2 \cdot B \cdot (C_B + 0.7)}{1.39}, \frac{|M_S + M_W|}{175 \cdot 1.39} \right) [cm^3]$$

$$Z_{\min}^{bottom} = \max\left(\frac{C_W \cdot L^2 \cdot B \cdot (C_B + 0.7)}{1.28}, \frac{|M_S + M_W|}{175 \cdot 1.28} \right) [cm^3]$$

건조비를 최소화 하는 중앙 단면 설계 문제의 수학적 정식화(3)

- Critical Buckling Stress에 관한 조건(Deck, Bottom)

$$\sigma^{deck} - \eta^{deck}\sigma_c^{deck} \le 0 \qquad \sigma^{bottom} - \eta^{bottom}\sigma_c^{bottom} \le 0$$

$$\sigma_c^{deck} = \begin{cases} \sigma_{el}^{deck} & \text{when } \sigma_{el}^{deck} < 177.5 \\ 355 \cdot \left(1 - \dfrac{355 \cdot \sigma_{el}^{deck}}{4}\right) & \text{when } \sigma_{el}^{deck} > 177.5 \end{cases}$$

$$\sigma_c^{bottom} = \begin{cases} \sigma_{el}^{bottom} & \text{when } \sigma_{el}^{bottom} < 117.5 \\ 235 \cdot \left(1 - \dfrac{355 \cdot \sigma_{el}^{bottom}}{4}\right) & \text{when } \sigma_{el}^{bottom} > 117.5 \end{cases}$$

$$\sigma^{deck} = \frac{M_S + M_W}{I_N} \cdot z_n^{deck} \cdot 10^5 \ [N/mm^2]$$

$$\sigma^{bottom} = \frac{M_S + M_W}{I_N} \cdot z_n^{bottom} \cdot 10^5 \ [N/mm^2]$$

@SDAL Advanced Ship Design Automation Lab. http://asdal.snu.ac.kr

중앙 단면 설계 문제에 대한 최적화 결과

	Unit	Actual Ship	MFD	MS	GA	HYBRID	
						w/o Refine	with Refine
Building Cost	*$/m*	-	21,035.254748	20,637.828634	20,597.330090	20,422.478135	20,350.286893
x_1	*mm*	800.0	787.038274	811.324938	780.000000	810.000000	810.3701321
x_2	*mm*	800.0	762.891023	799.038243	750.000000	800.000000	800.1282732
x_3	*mm*	780.0	743.313979	787.034954	770.000000	790.000000	789.0923943
x_4	*mm*	835.0	814.142029	833.909455	820.000000	830.000000	834.838424
x_5	*mm*	770.0	756.434513	772.349435	790.000000	780.000000	780.002092
x_6	*mm*	16.5	16.983723	16.203495	16.000000	16.000000	16.390923
x_7	*mm*	16.0	16.829142	16.043803	16.500000	16.000000	15.989044
x_8	*mm*	15.5	16.020913	15.390394	16.000000	15.500000	15.432091
x_9	*mm*	17.0	17.329843	17.039439	16.500000	16.500000	17.139433
x_{10}	*mm*	14.5	15.001923	14.324335	15.000000	15.000000	14.780908
x_{11}	*mm*	13.5	14.192834	14.240495	14.000000	13.500000	13.550214
x_{12}	*mm*	14.5	15.123051	15.403945	14.500000	14.500000	14.500130
x_{13}	*mm*	17.0	16.902832	16.849387	16.500000	17.000000	17.010902
x_{14}	*mm*	14.0	14.784034	14.739454	15.500000	14.500000	14.309324
x_{15}	*mm*	14.0	15.129430	14.448504	15.500000	14.500000	14.588917
x_{16}	*mm*	14.5	14.824045	14.940584	15.000000	15.000000	14.789992
Iteration No	-	-	8	912	93	64	70
CPU Time	*sec*	-	2.90	293.28	272.91	265.06	267.92

* 최적 설계값을 실제로 사용하기 위해서는 부재의 두께와 간격의 경우 생산성을 고려하여 적절한 조정(예, 반올림 등)이 필요함

참고 문헌

- **이규열**, **노명일**, **차주환**, **전산선박설계**, 3th Ed., 2003.9
- Arora, J. S., Introduction to Optimum Design, 2nd Ed., Elsevier, 2004