Direct Search Methods

- Introduced by Hooke and Jeeves (1961)
- Methods that do not require derivatives of the functions in their search strategy
- Convergence? continuous and differentiable functions
- Two prominent methods
  - Nelder-Mead Simplex Method
  - Hooke-Jeeves Method

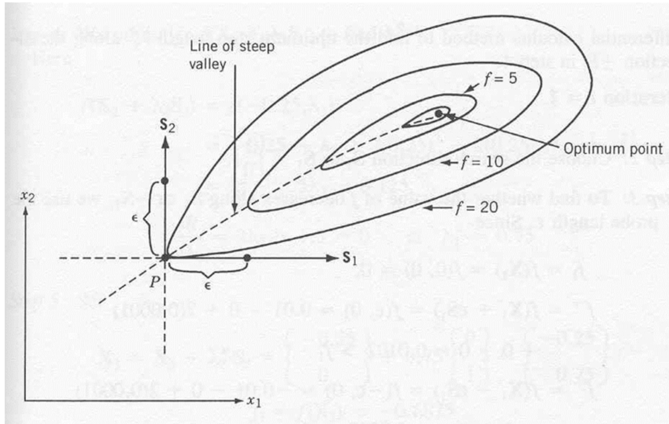
Univariate Search

- Minimize the function with respect to one variable at a time, keeping all other variables fixed
  - perform along the coordinate directions (fixed)

$$\underset{\alpha}{\text{minimize}}\ f\left(x_i^{(k)}+\alpha\right)\rightarrow \text{1-D search}$$

$$+\alpha\,?\quad -\alpha\,?\ \text{reject}$$

$$x_i^{(k+1)}=x_i^{(k)}+\alpha_i,\ i=1,\dots,n$$



- Cyclic Coordinate Search
  - Change only one variable at a time
  - Produce a sequence of improved approximations
- Simple, but not converge rapidly, not even converge!
  - Steep valley

Hooke–Jeeves Method

- Pattern search methods
- Search direction is not always fixed
- Exploratory Search
  - Univariate search is performed with a fixed step size in each coordinate direction
  - Successful? the new design point is called the base point
  - Search fails? the step sizes are reduced by a factor
- Pattern Search

Hooke and Jeeves Pattern Search Method

exploration at point B

$$\boldsymbol{x} \pm s\boldsymbol{e}_i \rightarrow \begin{cases} \text{successful : pattern direction } BP = \boldsymbol{x}_P - \boldsymbol{x}_B \\ \text{not successful : } s = rs\ (r < 1) \end{cases}$$

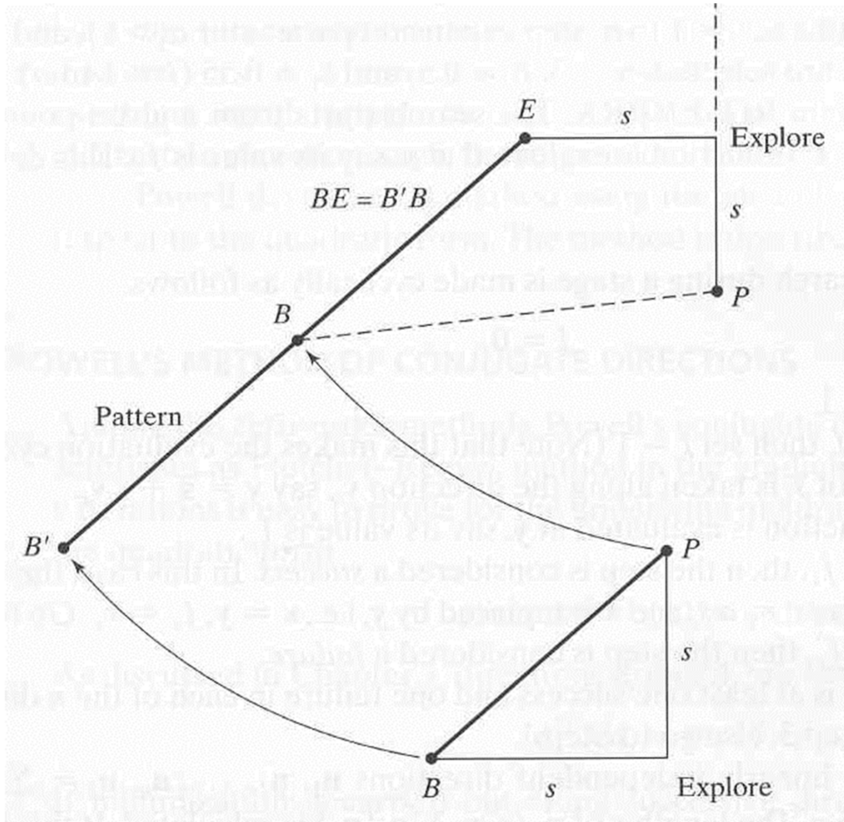
$s = 0.05 \sim 1$: step size $\rightarrow$ line search

$P \rightarrow B,\ B \rightarrow B'$

extension step : $\boldsymbol{x}_E = \boldsymbol{x}_B + \alpha(\boldsymbol{x}_B - \boldsymbol{x}'_B)$

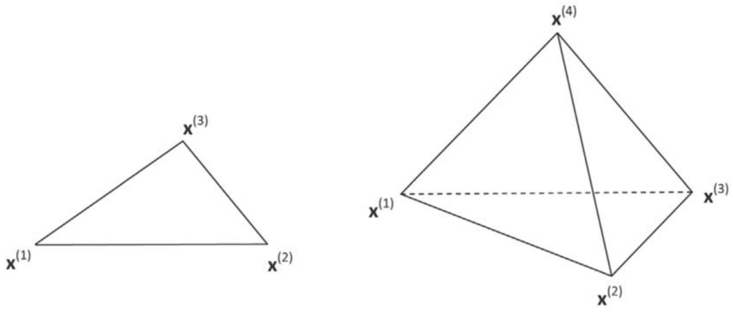
$\alpha = 0.5 \sim 2 \rightarrow$ line search

exploration at point E



Nelder–Mead Simplex Method

- Does not use gradients of the cost function
- Idea of a *simplex*
  - Geometric figure formed by a set of (n+1) points in the n-dimensional space
  - When the points are equidistant, the simplex is said to be *regular*
- Nelder–Mead method (Nelder and ead, 1965)
  - Compute cost function value at the (n+1) vertices of the simplex
  - Move this simplex toward the minimum point
  - reflection, expansion, contraction, and shrinkage



(a) (b)

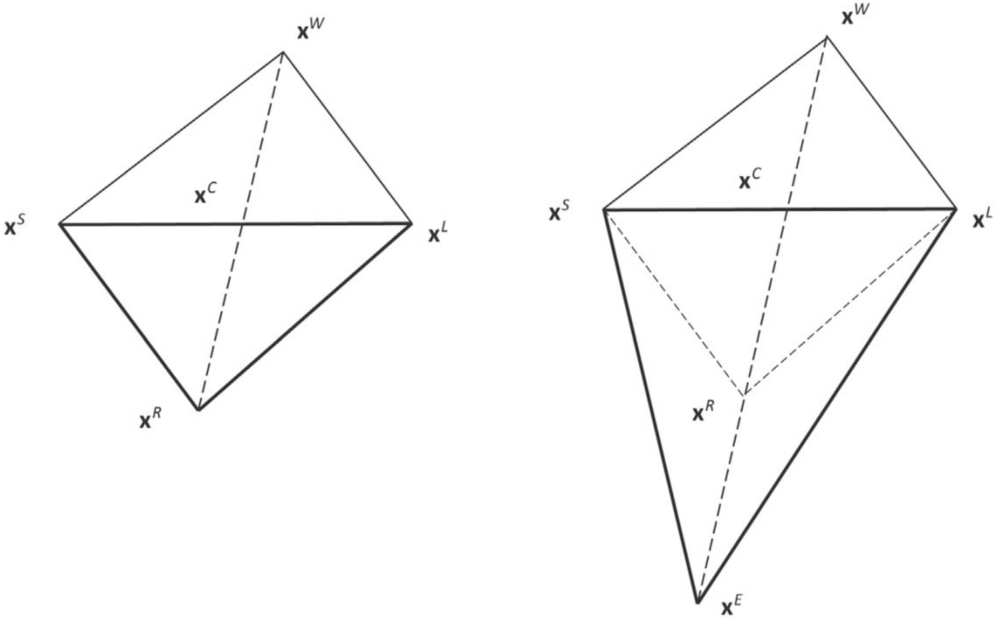
Operations on the Simplex (1)

$$f_1 \le f_2 \le \cdots \le f_n \le f_{n+1} \rightarrow \mathbf{x}^C = \frac{1}{n}\sum_{k=1}^{n}\mathbf{x}^{(k)}$$: centroid of n best points

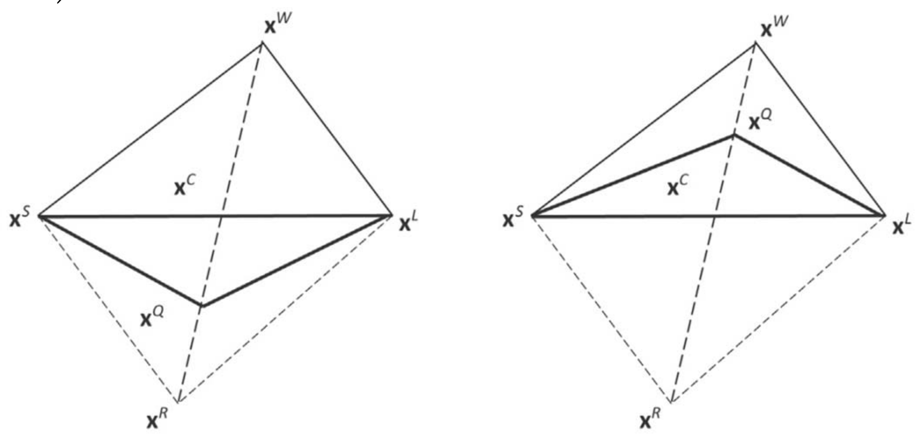
reflection: $\mathbf{x}^R = \mathbf{x}^W + (1+\alpha_R)(\mathbf{x}^C - \mathbf{x}^W) = (1+\alpha_R)\mathbf{x}^C - \alpha_R\mathbf{x}^W,\ 0 < \alpha_R \le 1\,(\alpha_R = 1)$

extension: $\mathbf{x}^E = (1+\alpha_E)\mathbf{x}^C - \alpha_E\mathbf{x}^W,\ \alpha_E \ge 1\,(\alpha_E = 3)$

contraction: $\mathbf{x}^Q = \begin{cases} \mathbf{x}^R + \alpha_Q(\mathbf{x}^R - \mathbf{x}^C) = (1+\alpha_Q)\mathbf{x}^R - \alpha_Q\mathbf{x}^C \\ \mathbf{x}^C + \alpha_Q(\mathbf{x}^C - \mathbf{x}^W) = (1+\alpha_Q)\mathbf{x}^C - \alpha_Q\mathbf{x}^W \end{cases},\ -1 < \alpha_Q < 0\,(\alpha_Q = -0.5)$



(a) (b)

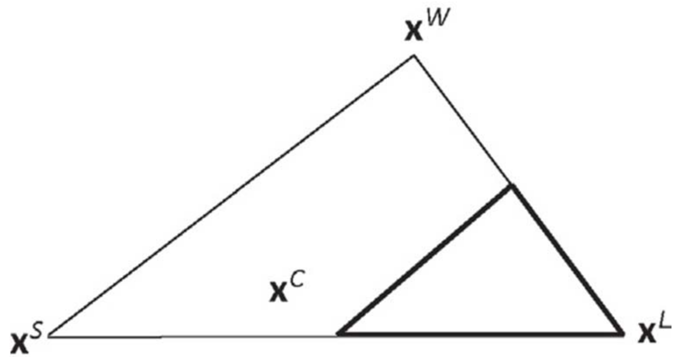


(a) (b)

Operations on the Simplex (2)

$$\text{shrinking: } \mathbf{x}^{(j)} = \mathbf{x}^{L} + \delta\left(\mathbf{x}^{(j)} - \mathbf{x}^{L}\right),\ 0 < \delta < 1(\delta = 0.5),\ j = 2,\ldots(n+1)$$

$$\text{initial simplex: } \mathbf{x}^{(j)} = \underbrace{\mathbf{x}^{(1)}}_{\text{seed point}} + \delta_j \mathbf{e}^{(j)},\ j = 2,\ldots(n+1)$$



- Termination Criteria
  - Domain convergence test
  - Function value convergence test
  - Limit on number of iterations
- MATLAB
  - fminsearch

Solution of Constrained Problem Using Unconstrained Optimization Methods

- Transformation Method
- constrained problem $\rightarrow$ sequence of appropriately formed unconstrained problems
  - Robust for large problems
  - Few algorithmic parameters to tune
- SUMT (Fiacco and McCormick, 1968)
  - Sequential Unconstrained Minimization Techniques
  - Construct a transformation function (cost + constraint)

$$\Phi(\mathbf{x},\mathbf{r}) = f(\mathbf{x}) + P\left(h(\mathbf{x}), g(\mathbf{x}), \mathbf{r}\right): \text{as } r \rightarrow \infty,\ \mathbf{x}(\mathbf{r}) \rightarrow \mathbf{x}^*$$

  - Penalty Function Method: infeasible points
  - Barrier Function Method: feasible points
  - Augmented Lagrangian (Multiplier) Methods

SUMT: Penalty Function Method

- Quadratic loss function

$$P\left(h(\mathbf{x}), g(\mathbf{x}), \mathbf{r}\right) = r\left\{\sum_{i=1}^{p}\left[h_i(\mathbf{x})\right]^2 + \sum_{i=1}^{m}\left[g_i^+(\mathbf{x})\right]^2\right\} \text{ where } g_i^+(\mathbf{x}) = \max\left(0, g_i(\mathbf{x})\right)$$

- Exterior penalty method
  - Applicable to generally constrained problems with equality and inequality constraints
  - Starting point can be arbitrary
  - Iterates through the infeasible region where the cost and/or constraint functions may be undefined
  - If the iterative process terminates prematurely, the final point may not be feasible and hence not usable

SUMT: Barrier Function Method

$$\begin{cases} \text{Inverse barrier function: } P\left(g\left(\mathbf{x}\right),\mathbf{r}\right) = \dfrac{1}{r}\displaystyle\sum_{i=1}^{p}\dfrac{-1}{g_i\left(\mathbf{x}\right)} \\ \text{Log barrier function: } P\left(g\left(\mathbf{x}\right),\mathbf{r}\right) = \dfrac{1}{r}\displaystyle\sum_{i=1}^{p}\log\left(-g_i\left(\mathbf{x}\right)\right) \end{cases}$$

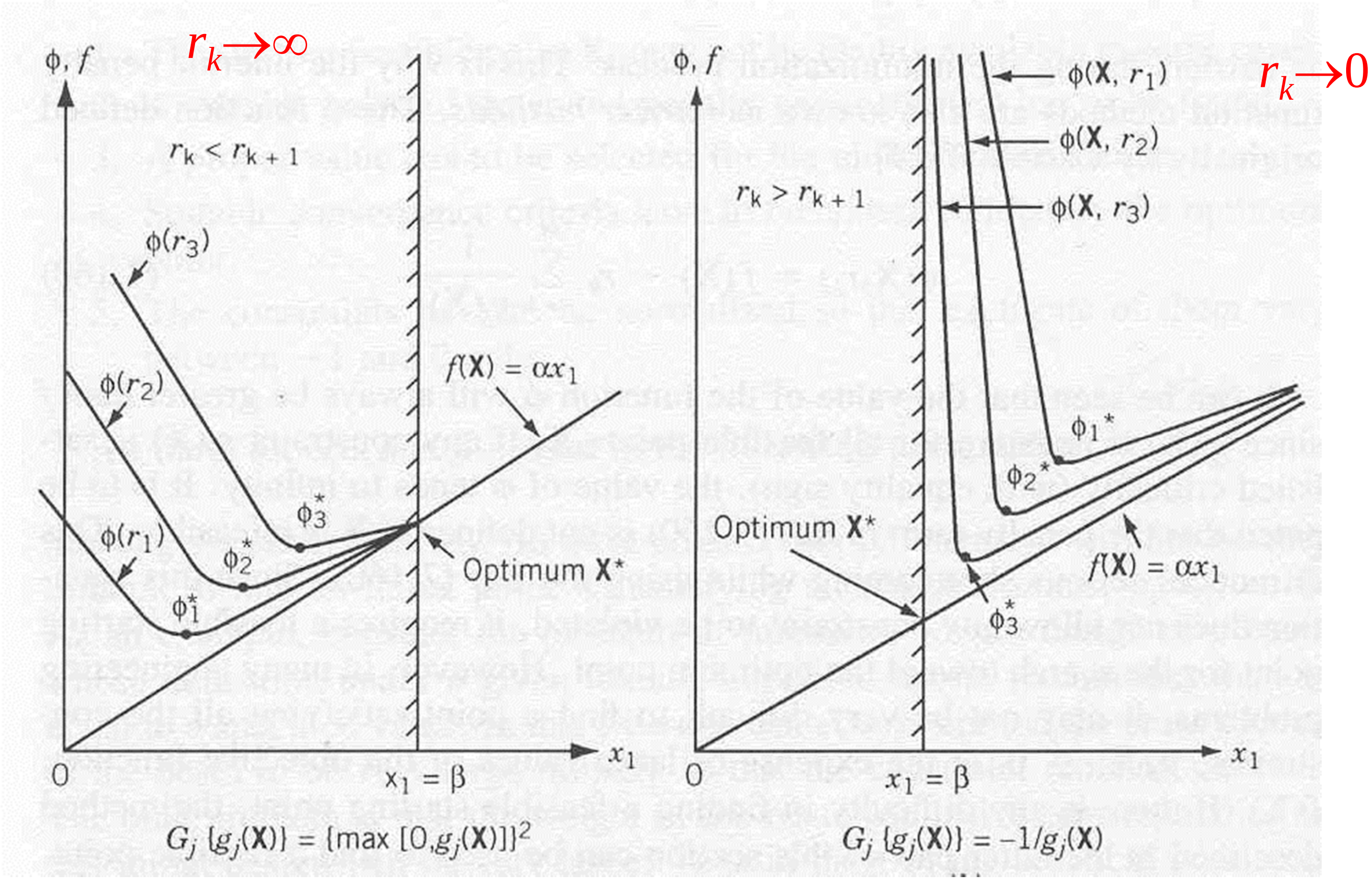
- create a large barrier around the feasible region
- Interior penalty method
  - Applicable to inequality-constrained problems only
  - Starting point must be feasible
  - Always iterates through the feasible region
  - If it terminates prematurely, the final point is feasible and usable

- Weakness of penalty and barrier function method
  - ill-behaved when **r** becomes large, selection of the sequence **r**

Example

Minimize $f(x) = \alpha x_1$

subject to $g(x) = \beta - x_1 \le 0$



Augmented Lagrangian Method (ALM) Multiplier Method

- Fixed parameter SUMT-like penalty method
- No need for the controlling penalty parameter
  - Need not be increased to infinity for convergence
  - transformation function Φ has good conditioning with no singularities
- Arbitrary starting point: no need to be feasible
- Fast rate of convergence than the penalty method
- Possible to achieve active constraints precisely
- Automatic identification of active constraints
  - Nonzero Lagrangian multipliers

$$P\left(h(\mathbf{x}), g(\mathbf{x}), \mathbf{r}, \theta\right) = \frac{1}{2}\sum_{i=1}^{p} r_i'\left(h_i + \theta_i'\right)^2{}_i + \frac{1}{2}\sum_{i=1}^{m} r_i\left[\left(g_i + \theta_i\right)^+\right]^2$$

$$\xrightarrow[r_i = r_i' = r]{\theta_i = \theta' = 0} \text{quadratic loss function}$$

Equality Constrained Problem

Minimize $f(\mathbf{x})$

subject to $h_j(\mathbf{x}) = 0 \quad j = 1, \ldots, l$

$$\Phi_E\left(\mathbf{x}, h(\mathbf{x}), r_k\right) = \underbrace{f(\mathbf{x}) + \sum_{j=1}^{p} \lambda_j h_j(\mathbf{x})}_{L(\mathbf{x},\lambda)} + \frac{1}{2} r_k \sum_{j=1}^{p} \left[h_j(\mathbf{x})\right]^2 \Rightarrow \begin{cases} r_k = 0 \rightarrow \text{ Lagrangian} \\ all\ \lambda_j = 0 \rightarrow \text{ penalty function} \end{cases}$$

$$\left.\begin{aligned} \frac{\partial L}{\partial x_i} &= \frac{\partial f}{\partial x_i} + \sum_{j=1}^{p} \lambda_j^* \frac{\partial h_j}{\partial x_i} = 0, \quad i = 1, \ldots, n \\ \frac{\partial T}{\partial x_i} &= \frac{\partial f}{\partial x_i} + \sum_{j=1}^{p} \left(\lambda_j + r_k h_j\right) \frac{\partial h_j}{\partial x_i} = 0, \quad i = 1, \ldots, n \end{aligned}\right\} \rightarrow \lambda_j^* = \lambda_j + r_k h_j$$

$$\text{updating rule} \begin{cases} \lambda_j^{(k+1)} = \lambda_j^{(k)} + r_k h_j\left(\mathbf{x}^{(k)}\right), \quad j = 1, \ldots, l \\ r_{k+1} = c r_k, \quad c \geq 1 \end{cases}$$

Inequality Constrained Problem

$$\text{Minimize} \quad f(\mathbf{x})$$

$$\text{subject to} \quad g_j(\mathbf{x}) \le 0 \qquad j = 1, \dots, m$$

$$\Phi_I(\mathbf{x}, g(\mathbf{x}), r_k) = \underbrace{f(\mathbf{x}) + \sum_{j=1}^{m} \lambda_j \left[g_j(\mathbf{x}) + s_j^{\,2} \right]}_{L(\mathbf{x},\lambda)} + \frac{1}{2} r_k \sum_{j=1}^{m} \left[g_j(\mathbf{x}) + s_j^{\,2} \right]^2$$

$$\rightarrow \begin{cases} \Phi_I(\mathbf{x}, \lambda, r_k) = f(\mathbf{x}) + \sum_{j=1}^{m} \lambda_j \alpha_j + \frac{1}{2} r_k \sum_{j=1}^{m} \alpha_j^{\,2} \\ \alpha_j = \max\left\{ g_j(\mathbf{x}), \quad -\frac{\lambda_j}{r_k} \right\} \end{cases}$$

$$\lambda_j^{(k+1)} = \lambda_j^{(k)} + r_k \alpha_j^{(k)}, \qquad j = 1, \dots, m$$

R.T.Rockafellar, The multiplier method of Hestenes and Powell applied to Convex programming, *Journal of Optimization Theory and Applications*, Vol.12, No.6, pp.555-562, 1973

Mixed Equality-Inequality Constrained Problem

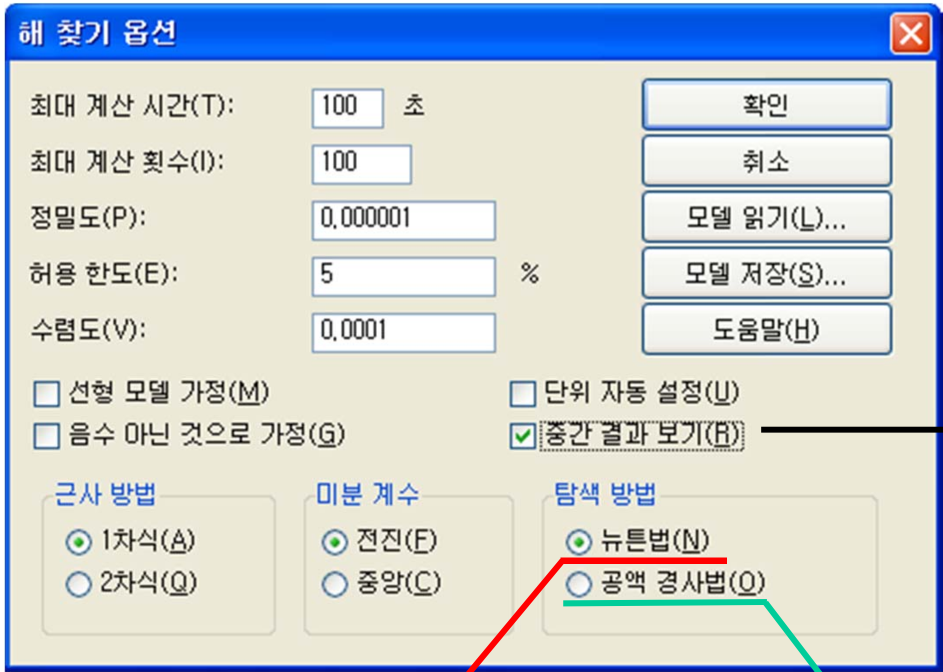
$$\begin{aligned}
\text{Minimize}\quad & f(\mathbf{x}) \\
\text{subject to}\quad & g_j(\mathbf{x}) \le 0, \quad j = 1, \ldots, m \\
& h_j(\mathbf{x}) = 0, \quad j = 1, \ldots, p
\end{aligned}$$

$$\Phi_I\left(\mathbf{x}, h(\mathbf{x}), g(\mathbf{x}), r_k\right) = f(\mathbf{x}) + \sum_{j=1}^{m} \lambda_j \alpha_j + \sum_{j=1}^{p} \lambda_{m+j} h_j(\mathbf{x}) + \frac{1}{2} r_k \sum_{j=1}^{m} \alpha_j^{\,2} + \frac{1}{2} r_k \sum_{j=1}^{p} h_j^{\,2}(\mathbf{x})$$

$$\text{updating rule}\ \begin{cases} \lambda_j^{(k+1)} = \lambda_j^{(k)} + r_k \max\left\{ g_j\left(\mathbf{x}^{(k)}\right),\ -\dfrac{\lambda_j^{(k)}}{r_k} \right\}, & j = 1, \ldots, m \\ \lambda_{m+j}^{(k+1)} = \lambda_{m+j}^{(k)} + r_k h_j\left(\mathbf{x}^{(k)}\right), & j = 1, \ldots, p \end{cases}$$

Example 10.6: Excel/Solver

$$f(x_1, x_2, x_3) = x_1^2 + 2x_2^2 + 2x_3^2 + 2x_1x_2 + 2x_2x_3 \quad \text{from} \quad \mathbf{x}_0 = (2, 4, 10)$$



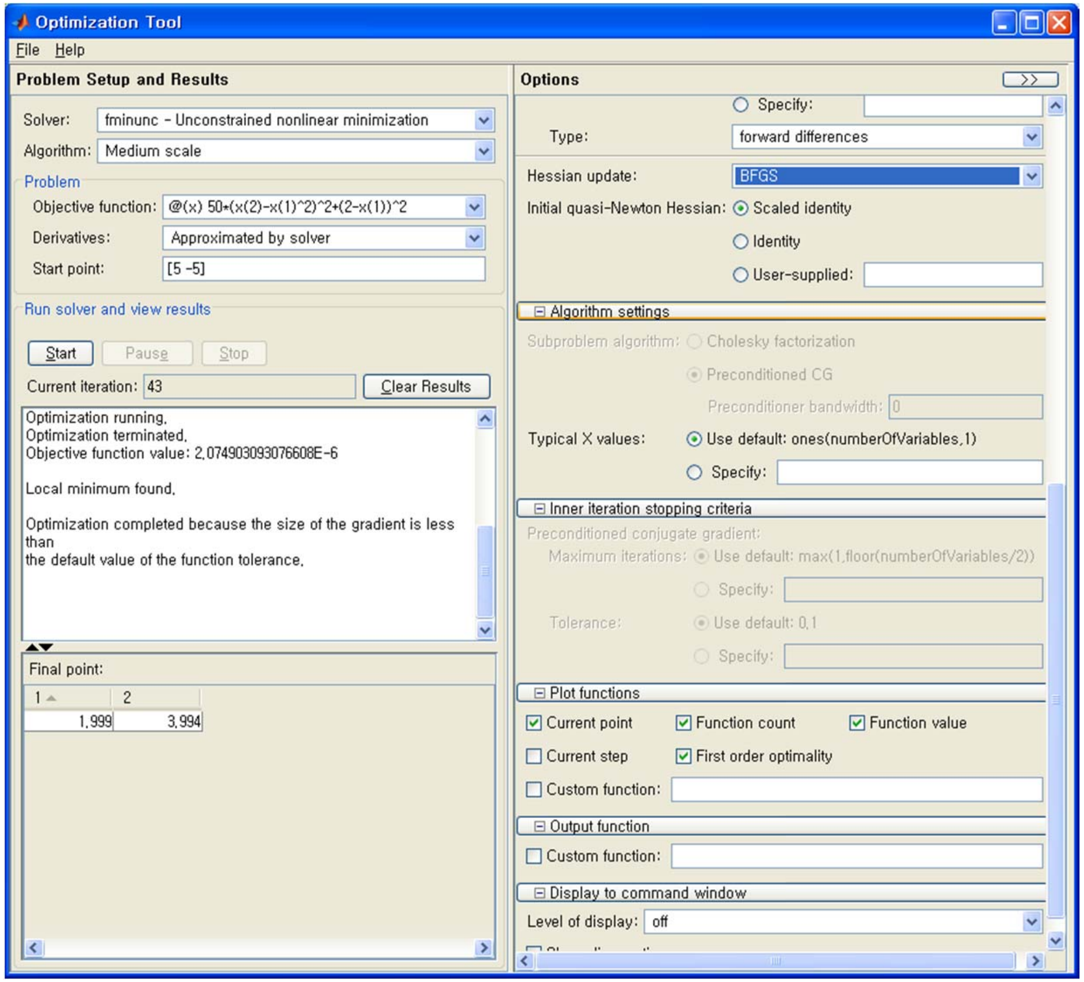
시나리오 저장

데이터>가상분석>
시나리오 관리자

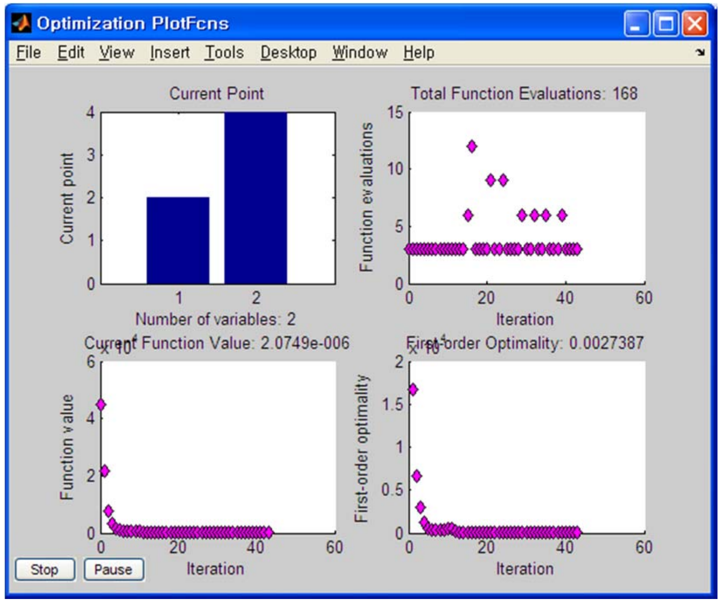
	quasi-Newton method	conjugate gradient method
memory	more	less
iteration	fewer	more

Example 11.8: Matlab/fminunc (1)

$$f(\boldsymbol{x}) = 50\left(x_2 - x_1^2\right)^2 + \left(2 - x_1\right)^2 \quad \text{from } \boldsymbol{x}_0 = (5, -5)$$



>> optimtool



Example 11.8: Matlab/fminunc (2)