Direct Search Methods

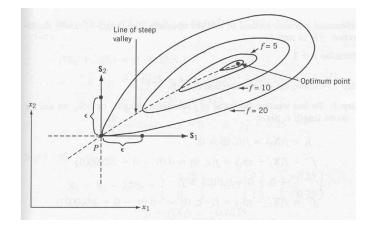
- Introduced by Hooke and Jeeves (1961)
- Methods that do not require derivatives of the functions in their search strategy
- Convergence? continuous and differentiable functions
- Two prominent methods
 - Nelder-Mead Simplex Method
 - Hooke-Jeeves Method

Univariate Search

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

minimize $f(x_i^{(k)} + \alpha) \rightarrow 1$ -D search + α ? $-\alpha$? reject $x_i^{(k+1)} = x_i^{(k)} + \alpha_i, i = 1,...,n$

- 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



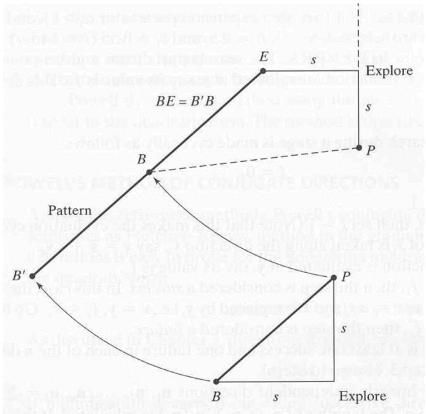
Hooke–Jeeves Method

- 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

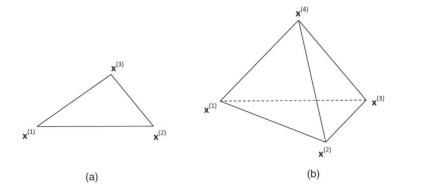
exploration at point B

 $\mathbf{x} \pm s\mathbf{e}_{i} \rightarrow \begin{cases} \text{successful : pattern direction } BP = \mathbf{x}_{P} - \mathbf{x}_{B} \\ \text{not successful : } s = rs(r < 1) \end{cases}$ $s = 0.05 \sim 1 : \text{step size} \rightarrow \text{line search}$ $P \rightarrow B, B \rightarrow B'$ extension step : $\mathbf{x}_{E} = \mathbf{x}_{B} + \alpha(\mathbf{x}_{B} - \mathbf{x}'_{B})$ $\alpha = 0.5 \sim 2 \rightarrow \text{line search}$ 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



Operations on the Simplex (1)

$$f_{1} \leq f_{2} \leq \cdots \leq f_{n} \leq f_{n+1} \rightarrow \mathbf{x}^{C} = \frac{1}{n} \sum_{k=1}^{n} \mathbf{x}^{(k)}: \text{ centroid of } n \text{ best points}$$
reflection: $\mathbf{x}^{R} = \mathbf{x}^{W} + (1 + \alpha_{R}) (\mathbf{x}^{C} - \mathbf{x}^{W}) = (1 + \alpha_{R}) \mathbf{x}^{C} - \alpha_{R} \mathbf{x}^{W}, \quad 0 < \alpha_{R} \leq 1(\alpha_{R} = 1)$
extension: $\mathbf{x}^{E} = (1 + \alpha_{E}) \mathbf{x}^{C} - \alpha_{E} \mathbf{x}^{W}, \quad \alpha_{E} \geq 1(\alpha_{E} = 3)$
contraction: $\mathbf{x}^{Q} = \begin{cases} \mathbf{x}^{R} + \alpha_{Q} (\mathbf{x}^{R} - \mathbf{x}^{C}) = (1 + \alpha_{Q}) \mathbf{x}^{R} - \alpha_{Q} \mathbf{x}^{C} \\ \mathbf{x}^{C} + \alpha_{Q} (\mathbf{x}^{C} - \mathbf{x}^{W}) = (1 + \alpha_{Q}) \mathbf{x}^{C} - \alpha_{Q} \mathbf{x}^{W}, \quad -1 < \alpha_{Q} < 0(\alpha_{Q} = -0.5)$

$$\mathbf{x}^{W} \qquad \mathbf{x}^{W} \qquad \mathbf{x}^{W}$$

Operations on the Simplex (2)

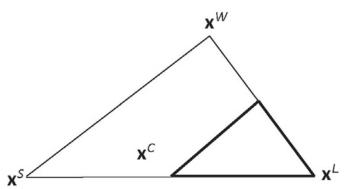
shrinking:
$$\mathbf{x}^{(j)} = \mathbf{x}^L + \delta(\mathbf{x}^{(j)} - \mathbf{x}^L), \ 0 < \delta < 1(\delta = 0.5), \ j = 2, ...(n+1)$$

initial simplex: $\mathbf{x}^{(j)} = \mathbf{x}^{(1)} + \delta_j \mathbf{e}^{(j)}, \ j = 2, ...(n+1)$

- Termination Criteria
 - Domain convergence test
 - Function value convergence test

seed point

- Limit on number of iterations
- MATLAB
 - fminsearch



Solution of Constrained Problem Using Unconstrained Optimization Methods

- Transformation Method
- constrained problem → 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)

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

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

SUMT: Penalty Function Method

• Quadratic loss function

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

- 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

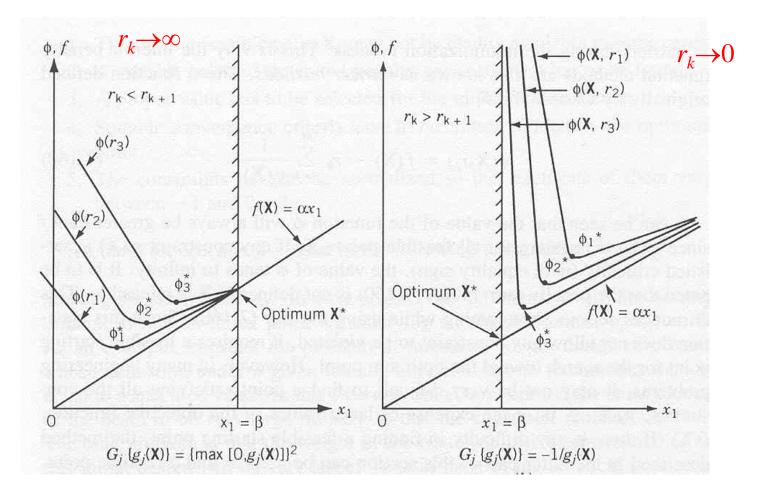
Inverse barrier function:
$$P(g(\mathbf{x}), \mathbf{r}) = \frac{1}{r} \sum_{i=1}^{p} \frac{-1}{g_i(\mathbf{x})}$$

Log barrier function: $P(g(\mathbf{x}), \mathbf{r}) = \frac{1}{r} \sum_{i=1}^{p} \log(-g_i(\mathbf{x}))$

- 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

$$P(h(\mathbf{x}), g(\mathbf{x}), \mathbf{r}, \theta) = \frac{1}{2} \sum_{i=1}^{p} r_{i}' (h_{i} + \theta_{i}')^{2}_{i} + \frac{1}{2} \sum_{i=1}^{m} r_{i} \left[(g_{i} + \theta_{i})^{+} \right]^{2}$$

Equality Constrained Problem

$$\begin{array}{l} \text{Minimize } f\left(\mathbf{x}\right) \\ \text{subject to } h_{j}\left(\mathbf{x}\right) = 0 \quad j = 1, \dots, l \\ \Phi_{E}\left(\mathbf{x}, h\left(\mathbf{x}\right), r_{k}\right) = \underbrace{f\left(\mathbf{x}\right) + \sum_{j=1}^{p} \lambda_{j} h_{j}\left(\mathbf{x}\right) + \frac{1}{2} r_{k} \sum_{j=1}^{p} \left[h_{j}\left(\mathbf{x}\right)\right]^{2} \Rightarrow \begin{cases} r_{k} = 0 \rightarrow \text{Lagrangian} \\ all \ \lambda_{j} = 0 \rightarrow \text{ penalty function} \end{cases} \\ \frac{\partial L}{\partial x_{i}} = \frac{\partial f}{\partial x_{i}} + \sum_{j=1}^{p} \lambda_{j}^{*} \frac{\partial h_{j}}{\partial x_{i}} = 0, \quad i = 1, \dots, n \\ \frac{\partial T}{\partial x_{i}} = \frac{\partial f}{\partial x_{i}} + \sum_{j=1}^{p} \left(\lambda_{j} + r_{k} h_{j}\right) \frac{\partial h_{j}}{\partial x_{i}} = 0, \quad i = 1, \dots, n \end{cases} \\ \text{updating rule} \begin{cases} \lambda_{j}^{(k+1)} = \lambda_{j}^{(k)} + r_{k} h_{j}\left(\mathbf{x}^{(k)}\right), \quad j = 1, \dots, l \\ r_{k+1} = cr_{k}, \quad c \geq 1 \end{cases}$$

Inequality Constrained Problem

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

subject to $g_j(\mathbf{x}) \le 0$ $j = 1,...,m$
 $\Phi_I(\mathbf{x}, g(\mathbf{x}), r_k) = \underbrace{f(\mathbf{x}) + \sum_{j=1}^m \lambda_j \left[g_j(\mathbf{x}) + s_j^2\right] + \frac{1}{2} r_k \sum_{j=1}^m \left[g_j(\mathbf{x}) + s_j^2\right]^2}_{L(\mathbf{x},\lambda)}$
 $\rightarrow \begin{cases} \Phi_I(\mathbf{x}, \lambda, r_k) = f(\mathbf{x}) + \sum_{j=1}^m \lambda_j \alpha_j + \frac{1}{2} r_k \sum_{j=1}^m \alpha_j^2 \\ \alpha_j = \max\left\{g_j(\mathbf{x}), -\frac{\lambda_j}{r_k}\right\} \\ \lambda_j^{(k+1)} = \lambda_j^{(k)} + r_k \alpha_j^{(k)}, \quad j = 1,...,m \end{cases}$

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

$$\begin{aligned} \text{Minimize } f(\mathbf{x}) \\ \text{subject to } g_{j}(\mathbf{x}) &\leq 0, \quad j = 1, \dots, m \\ h_{j}(\mathbf{x}) &= 0, \quad j = 1, \dots, p \end{aligned} \\ \Phi_{I}(\mathbf{x}, h(\mathbf{x}), g(\mathbf{x}), r_{k}) &= f(\mathbf{x}) + \sum_{j=1}^{m} \lambda_{j} \alpha_{j} + \sum_{j=1}^{p} \lambda_{m+j} h_{j}(\mathbf{x}) + \frac{1}{2} r_{k} \sum_{j=1}^{m} \alpha_{j}^{2} + \frac{1}{2} r_{k} \sum_{j=1}^{p} h_{j}^{2}(\mathbf{x}) \end{aligned} \\ \text{updating rule} \begin{cases} \lambda_{j}^{(k+1)} &= \lambda_{j}^{(k)} + r_{k} \max\left\{g_{j}(\mathbf{x}^{(k)}), -\frac{\lambda_{j}^{(k)}}{r_{k}}\right\}, \quad j = 1, \dots, m \\ \lambda_{m+j}^{(k+1)} &= \lambda_{m+j}^{(k)} + r_{k} h_{j}(\mathbf{x}^{(k)}), \quad j = 1, \dots, p \end{cases} \end{aligned}$$

Example 10.6: Excel/Solver

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

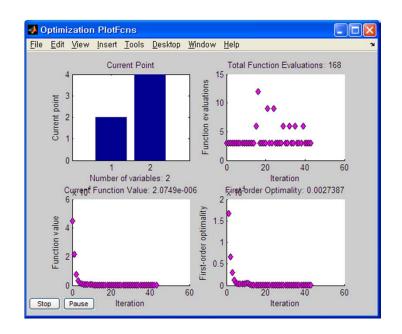
ō	배 찾기 옵션								
	최대 계산 시간(T):	100 초		확인					
	최대 계산 횟수(l):	100		취소					
	정밀도(P):	0,000001		모델 읽기(<u>L</u>)					
	허용 한도(E):	5 %		모델 저장(<u>S</u>)					
	수렴도(V):	0,0001		도움말(<u>H</u>)					
			단위 자동 설정(<u>U</u>) 동간 결과 보기(<u>R</u>))		➡ 시나	리오 저장			
	근사 방법 ○ 1차식(<u>A</u>) ○ 2차식(<u>Q</u>) □분 계수 ○ 전진(E) ○ 중앙(<u>C</u>) 		탐색 방법 ⊙ 뉴튼법(<u>N)</u> ○ 공액 경사법(<u>O</u>)			터>가상분석> 리오 관리자			
	quasi-Newton method		con	conjugate gradient me					
memory	y ma	more		less					
iteratior	tion fewer		more						

Example 11.8: Matlab/fminunc (1)

🕴 Optimization Tool										
Eile Help										
Problem Setup and Results			Options				\longrightarrow			
Solver:	fminunc	- Unconstrained nonlinear minimization			O Specify:		^			
Algorithm:	: Medium scale		Type:		forward difference	s	~			
Problem		Hessian update: BFGS			~					
Objective function: @(x) 50+(x(2)-x(1)^2)^2+(2-x(1))^2		Initial quasi-Newton Hessian: 💿 Scaled identity								
Derivatives: Approximated by solver				O Identity						
Start point: [5 -5]		[5 -5]			O User-supplied:					
Run solver and view results			□ Algorithm settings							
		Subproblem algorithm		olesky factorization						
Start Pause Stop			Pre	conditioned CG						
Current iteration: 43 Clear Results			Pre	conditioner bandwi	dth: 0					
Optimization running, Optimization terminated,		Typical X values:		e default: ones(num						
Objective function value: 2,074903093076608E-6			O Sp							
Local minimum found. Optimization completed because the size of the gradient is less than the default value of the function tolerance. Final point: 1 2 1,999 3,994		Inner iteration stopping criteria								
		Preconditioned conjugate gradient:								
		Maximum iterations: Use default: max(1,floor(numberOfVariables/2)) 								
				O S	pecify:					
		Tolerance:	() Us	e default: 0,1						
			O S	pecify:						
		Plot functions								
		3,994	Current point	🗹 Func	tion count 🛛 💽	Function value				
		Current step	First	order optimality						
		Custom function:								
		Output function								
			Custom function:							
			□ Display to command window							
			Level of display: off							
٤										
			<		7.IIII.		>			

$f(\mathbf{x}) = 50(x_2 - x_1^2)^2 + (2 - x_1)^2$ from $\mathbf{x}_0 = (5, -5)$

>> optimtool



Example 11.8: Matlab/fminunc (2)

