Integer Programming (IP)

$$\left.\begin{array}{l}\text{integer linear programming (ILP) problem}\\ \text{Linear problems with discrete variables}\end{array}\right\} \rightarrow 0-1 \text{ programming problem}$$

Nonlinear discrete problems $\rightarrow$ sequential linearization procedures

$$\begin{cases} \underset{\mathbf{x}}{\text{Minimize}}\ f = \mathbf{c}^T\mathbf{x} \\ \text{subject to} \begin{cases} \mathbf{Ax} \le \mathbf{b} \\ x_i \ge 0 \text{ integer}; \quad i = 1, \ldots, n_d \\ x_{iL} \le x_i \le x_{iU}; \quad i = (n_d + 1), \ldots, n \end{cases} \end{cases}$$

$$x_i = \sum_{j=1}^{q_i} z_{ij} d_{ij} \quad \text{where } z_{ij} = 0 \text{ or } 1 \text{ for all } i \text{ and } j, \ \sum_j z_{ij} = 1;\ i = 1, \ldots, n_d$$

$$f = \mathbf{c}^T\mathbf{x} \rightarrow f = \sum_{i=1}^{n_d} c_i \left[\sum_{j=1}^{q_i} z_{ij} d_{ij} \right] + \sum_{k=n_d+1}^{n} c_k x_k$$

$$\mathbf{Ax} \le \mathbf{b} \rightarrow \sum_{j=1}^{n_d} a_{ij} \left[\sum_{m=1}^{q_i} z_{jm} d_{jm} \right] + \sum_{k=n_d+1}^{n} a_{ik} x_k \le b_i$$

Integer Linear Programming (ILP)

- Linear discrete programming problem

$$\begin{aligned} &\text{minimize} \;\; f(\boldsymbol{x}) = \boldsymbol{c}^T \boldsymbol{x} \\ &\text{subject to} \;\; \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ &\qquad x_i \in X_i = \{d_{i1}, \ldots, d_{il}\} \quad i \in I_d \end{aligned}$$

Cross-sectional areas of truss
Ply thickness of laminated composite plate

  – Integer linear programming
    - Zero/one (binary) ILP $\rightarrow$ Enumeration tree

$$x_i = d_{i1}x_{i1} + \cdots + d_{il}x_{il}, \quad \text{where } x_{i1} + \cdots + x_{il} = 1$$

- Mixed Integer Linear Programming (MILP)
    - $\rightarrow$ Branch and bound

$$\begin{aligned} &\text{minimize} \;\; f(\boldsymbol{x}) = \boldsymbol{c}_1^T \boldsymbol{x} + \boldsymbol{c}_2^T \boldsymbol{y} \\ &\text{subject to} \;\; \boldsymbol{A}_1\boldsymbol{x} + \boldsymbol{A}_2\boldsymbol{y} = \boldsymbol{b} \\ &\qquad x_i \ge 0, \text{integer} \\ &\qquad y_i \ge 0 \end{aligned}$$

Sequential Linear Discrete Programming (1)

$$\underbrace{\begin{aligned}
&\textit{Minimize} \quad f(\boldsymbol{x}) \\
&\text{subject to} \quad g_j(\boldsymbol{x}) \le 0, \quad j = 1, \ldots, m \\
&\qquad h_k(\boldsymbol{x}) = 0, \quad k = 1, \ldots, p \\
&\qquad x_i \in \{d_{i1}, \ldots, d_{iq}\}, \quad i = 1, \ldots, n_0 \\
&\qquad x_i^{(l)} \le x_i \le x_i^{(u)}, \quad i = n_0 + 1, \ldots, n
\end{aligned}}$$

$$\Downarrow$$

$$\begin{aligned}
&\textit{Minimize} \quad f(\boldsymbol{x}) \approx f(\boldsymbol{x}^0) + \nabla f(\boldsymbol{x}^0)\delta\mathbf{x} \\
&\text{subject to} \quad g_j(\boldsymbol{x}) \approx g(\boldsymbol{x}^0) + \nabla g(\boldsymbol{x}^0)\delta\mathbf{x} \le 0, \quad j = 1, \ldots, m \\
&\qquad h_k(\boldsymbol{x}) \approx h(\boldsymbol{x}^0) + \nabla h(\boldsymbol{x}^0)\delta\mathbf{x} = 0, \quad k = 1, \ldots, p \\
&\qquad x_i^0 + \delta x_i \in \{d_{i1}, \ldots, d_{iq}\}, \quad i = 1, \ldots, n_0 \\
&\qquad x_i^{(l)} \le x_i^0 + \delta x_i \le x_i^{(u)}, \quad i = n_0 + 1, \ldots, n \\
&\qquad \delta\mathbf{x} = \boldsymbol{x} - \boldsymbol{x}^0
\end{aligned}$$

Sequential Linear Discrete Programming (2)

$$x_i = \sum_{j=1}^{q} y_{ij} d_{ij}, \quad \left(\sum_{j=1}^{q} y_{ij} = 1,\ y_{ij} = 0 \text{ or } 1 \right), \quad i = 1, \ldots, n_0$$

$$\textit{Minimize} \quad f(\boldsymbol{x}) \approx f(\boldsymbol{x}^0) + \sum_{i=1}^{n_0} \frac{\partial f}{\partial x_i} \left(\sum_{j=1}^{q} y_{ij} d_{ij} - x_i^{\,0} \right) + \sum_{i=n_0+1}^{n} \frac{\partial f}{\partial x_i} \left(x_i - x_i^{\,0} \right)$$

$$\text{subject to} \quad g_j(\boldsymbol{x}) \approx g(\boldsymbol{x}^0) + \sum_{i=1}^{n_0} \frac{\partial g}{\partial x_i} \left(\sum_{j=1}^{q} y_{ij} d_{ij} - x_i^{\,0} \right) + \sum_{i=n_0+1}^{n} \frac{\partial g}{\partial x_i} \left(x_i - x_i^{\,0} \right) \le 0, \quad j = 1, \ldots, m$$

$$h_k(\boldsymbol{x}) \approx h(\boldsymbol{x}^0) + \sum_{i=1}^{n_0} \frac{\partial h}{\partial x_i} \left(\sum_{j=1}^{q} y_{ij} d_{ij} - x_i^{\,0} \right) + \sum_{i=n_0+1}^{n} \frac{\partial h}{\partial x_i} \left(x_i - x_i^{\,0} \right) = 0, \quad k = 1, \ldots, p$$

$$\sum_{j=1}^{q} y_{ij} = 1, \quad i = 1, \ldots, n_0$$

$$y_{ij} = 0 \text{ or } 1, \quad i = 1, \ldots, n_0,\ j = 1, \ldots, q$$

$$x_i^{(l)} \le x_i^{\,0} + \delta x_i \le x_i^{(u)}, \quad i = n_0 + 1, \ldots, n$$

$$\Rightarrow \text{mixed-integer LP problem} \quad \left(\text{unknowns}: y_{ij}, x_i \right)$$

Example

$$\mathit{Minimize} \quad f\left(x_1, x_2\right) = 2x_1^{\ 2} + 3x_2^{\ 2}$$

$$\text{subject to} \quad g\left(x_1, x_2\right) = \frac{1}{x_1} + \frac{1}{x_2} - 4 \le 0$$

$$x_1 \in \{0.3, 0.7, 0.8, 1.2, 1.5, 1.8\}$$

$$x_2 \in \{0.4, 0.8, 1.1, 1.4, 1.6\}$$

$$x^{(0)} = \begin{bmatrix} 1.2 \\ 1.1 \end{bmatrix}$$

Simulated Annealing (SA)

- S. Kirkpatrick, C. D. Gelatt and M. P. Vecchi, “Optimization by simulated annealing”, Science, 220, pp.671-680, 1983
- stochastic approach for locating a good approximation to the global minimum of a function
- comes from the *annealing process* in metallurgy
- involves heating and controlled cooling of a material to increase the size of its crystals and reduce their defects
- At high temperatures, the atoms become loose from their initial configuration and move randomly, perhaps through the states of higher internal energy, to reach a configuration having absolute minimum energy
- cooling process needs to be slow, and enough time needs to be spent at each temperature, giving more chance for the atoms to find configurations of lower internal energy

Simulated Annealing (SA)

- 금속의 풀림: 고체를 녹을 때까지 가열하고 난 후 그것을 완전한 격자상태의 결정체가 될 때까지 식히는 물리적 과정. 이런 과정 중에 그 고체의 자유에너지는 최소화
- 개선되지 않는 이웃해로의 이동을 확률적으로 허용
  - 지역최적해에 머무는 것을 방지, 이웃해 탐색방법의 하나
- 물리적 어닐링과 시뮬레이티드 어닐링의 관계

어닐링	시뮬레이티드 어닐링
물질	최적화문제
물리적 상태	가능해
에너지	비용함수
기저상태	최적화
냉각	국부탐색법

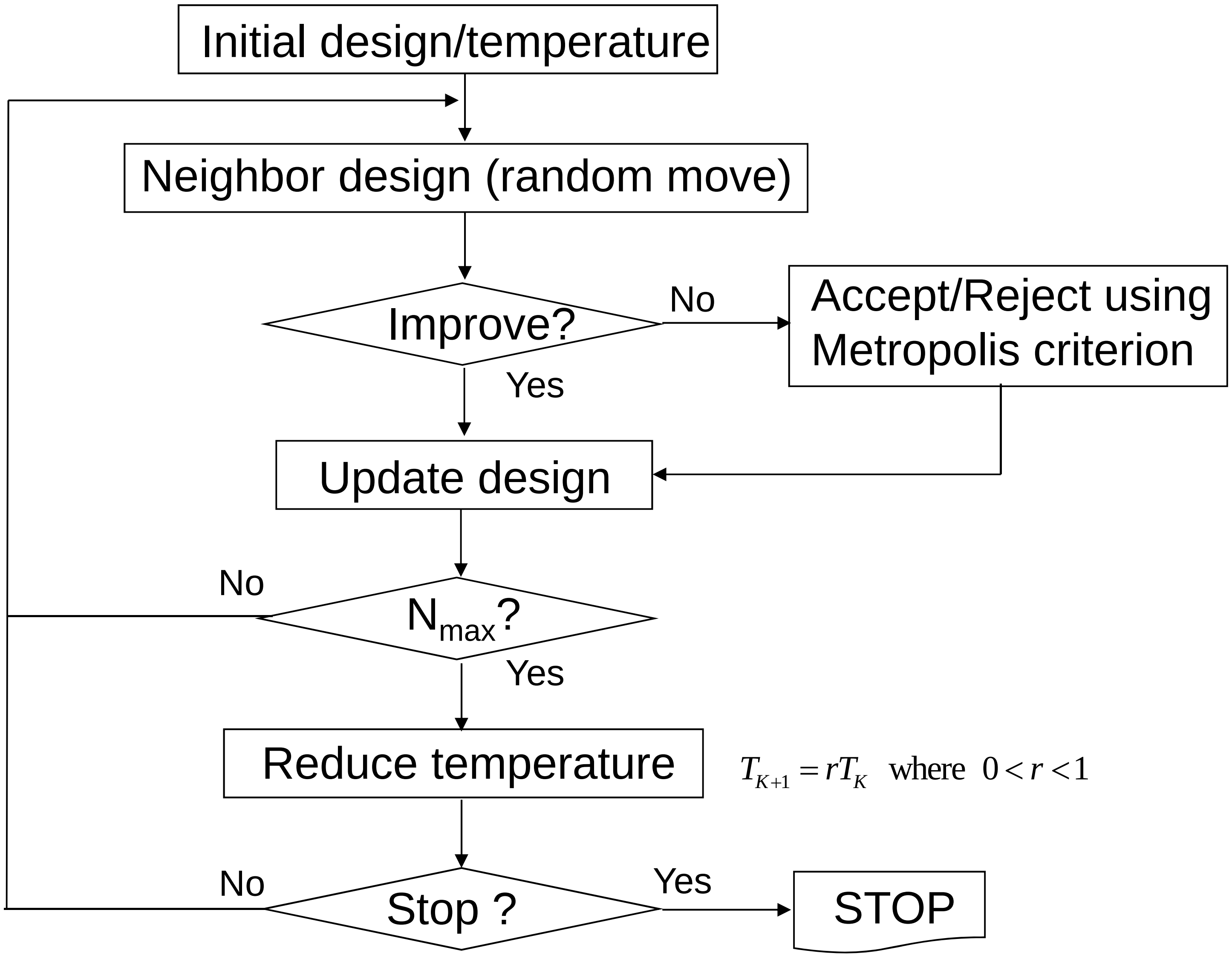
Metropolis Algorithm

- At each iteration, an "atom" is randomly displaced a small amount (random move).
- The energy is calculated for each atom and the difference with the energy at its original location is calculated.

$$\Delta E = f\left(\mathbf{x}^{(k)}\right) - f\left(\mathbf{x}^{(0)}\right)$$

- Boltzmann(-Gibbs) probability factor $p(\Delta E) = e^{-\frac{\Delta E}{kT}}$ is calculated.
  - T: temperature of the body, k: Boltsmann's constant
  - ΔE: energy difference between the two atom states
- If $\Delta E \leq 0$, then the new location is accepted.
- Otherwise, a random number(z) is generated between 0 and 1.
  - If $p(\Delta E) > z$, the higher energy state is accepted.
  - Otherwise, the old atom location is retained and the algorithm generates a new location.

SA: Algorithms



$$T_{K+1} = rT_K \quad \text{where} \quad 0 < r < 1$$

SA: Characteristics

- The quality of the final solution is not affected by the initial guess, except that the computational effort may increase with wrong starting designs.
- Because of the discrete nature of the function and constraint evaluation, the convergence or transition characteristics are not affected by the continuity or differentiability of the functions.
- The convergence is also not influenced by the convexity status of the feasible space.
- The design variables need not be positive.
- The method can be used to solve mixed-integer, discrete, or continuous problems.
- For problems involving behavior constraints, an equivalent unconstrained function is to be formulated.

Other methods (1)

- Rounding-off procedure
  - first obtain an optimum solution sing a continuous approach
  - Then, using heuristics, the variables are rounded off to their nearest available discrete values to obtain a discrete solution
  - main concern: selection of variables to be increased and the variables to be decreased
- Dynamic rounding-off algorithm
  - round off variables in a sequence rather than all of them at the same time
  - choose a design variable that minimizes the Lagrangian and remove that variable from the design variable set

Other methods (2)

- Neighborhood search method
  - When the number of discrete variables is small and each discrete variable has only a few choices
  - explicitly enumerate all of the possibilities
- Methods for linked discrete variables
  - For example, choice of materials, framed structural members
  - problems with linked variables are discrete and the problem functions are not differentiable with respect to them
  - simulated annealing, genetic algorithms and other nature inspired methods
  - Two or more algorithms may be combined to develop strategies that are more effective than the use of a purely discrete algorithm

Selection of a Method

Method	MV-OPT problem type solved	Can find feasible discrete solution?	Can find global minimum for convex problem?	Need gradients?
BBM	1–5	Yes	Yes	No/Yes
SA	1–5	Yes	Yes	No
Genetic algorithm	1–5	Yes	Yes	No
Sequential linearization	1	Yes	Yes	Yes
Dynamic round-off	1	Yes	No guarantee	Yes
Neighborhood search	1	Yes	Yes	Yes

Classification