

전역 최적화 기법 소개 : 결정론적 및 확률론적 방법들

최수형

전북대학교 화학공학부

1. 서론

들을 소개하고자 한다.

최적화는 시스템공학에서 자주 등장하는 문제이며 흔히 다음과 같은 수학적 계획(mathematical programming) 문제로 표현된다.

$$\begin{aligned} \min f(\mathbf{x}) \\ \text{subject to} \end{aligned}$$

$$\begin{aligned} g(\mathbf{x}) \leq 0 \\ h(\mathbf{x}) = 0 \end{aligned}$$

여기서 $\mathbf{x} \in R^n$, $f: R^n \rightarrow R$, $g: R^n \rightarrow R^l$, $h: R^n \rightarrow R^m$, 그리고 $n > m$ 이다. 만약 목적함수(objective function)와 가능 영역(feasible region)이 볼록(convex)하다면, 예를 들어 $f(\mathbf{x})$ 와 $g(\mathbf{x})$ 가 아래로 볼록하고 $h(\mathbf{x})$ 가 선형이라면, 이는 볼록 문제(convex problem)이며 오직 하나의 지역 최소점(local minimum)을 가진다. 그러나 많은 경우, 예를 들어 $h(\mathbf{x})$ 가 비선형이라면, 여러 개의 지역 최소점을 가질 수 있는 비 볼록 문제(nonconvex problem)가 된다. 이때 진정한 최소점을 찾는 것, 즉 전역 최적화(global optimization)가 요구된다.

전역 최적화 기법은 크게 결정론적 접근법(deterministic approach)과 확률론적 접근법(stochastic approach)으로 분류할 수 있다. 결정론적 접근법에 기반을 둔 알고리즘들은 cutting plane [1], generalized Benders decomposition [2, 3, 4], branch and bound [5, 6, 7], interval analysis [8, 9, 10] 등이 있다. 이들은 소위 finite-convergence를 보장한다. 즉 유한회용오차가 주어지면 유한회수의 반복계산을 통해 전역 최적점에 수렴한다는 것이다. 따라서 언젠가는 답이 나오며 얻어진 답의 전역 최적성(global optimality)이 보장된다. 그러나 계산량이 문제 크기에 따라 급격히 증가하므로 대개 작은 문제에만 적용 가능하다. 확률론적 접근법에 기반을 둔 대표적 알고리즘으로는 simulated annealing [11]과 genetic algorithm [12]이 있다. 이들은 계산을 무한히 반복하면 전역 최적점을 찾을 확률이 1에 무한히 접근한다. 즉 전역 최적성은 보장되지 않는다. 그 대신 비교적 큰 문제에도 적용 가능하다. 본 글에서는 전역 최적화에 대한 기초이론을 설명하고 몇 가지 결정론적 및 확률론적 방법

2. 방법론

전역 최적화는 다음 세 가지 일 중 하나를 수행할 수 있으면 이루어진다.

2.1. Find a tight convex hull

목적함수가 선형인 비선형 문제가 있다고 가정하자. 사실 어떠한 문제도 이런 형태로 재구성할 수 있는데 원래 목적함수를 새로운 치환변수로 교체하고 정의식을 제약조건으로 추가하면 된다. 이제 그림 1에 나타낸 것처럼 원래 문제의 가능영역을 볼록하면서 꼭 맞게 둘러싸는 겹침, 즉 최소 볼록 집합을 형성하는 제약조건들을 갖는 새로운 문제를 생각하자. 이 문제는 볼록하므로 현재 널리 사용되고 있는 지역 최적화 기술로 쉽게 풀 수 있으며 그 답은 원래 문제의 전역해와 같다. 이것이 대부분의 결정론적 알고리즘들의 기본 아이디어이다.

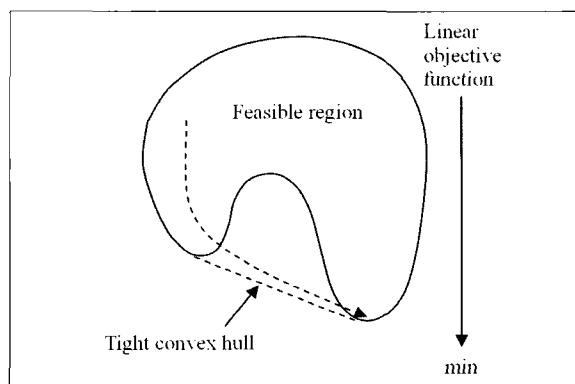


그림 1. Convex hull strategy.

2.2. Find a feasible point

만약 가능영역이 존재하면 그 중 한 점을 찾아내는 방법이 있다고 가정하자. 어떤 feasible point가 주어지면 generalized reduced gradient method를 사용하는 지역 최적화 기법으로 지역 최소점을 찾을 수 있다. 그 다음에는 목적함수가 현재의 지역 최소값보다 작아야 한다는 새로운 제약조건을 추가하고 새로운 feasible point를 찾는다. 이 과정을 feasible point가 존재하지 않을 때까지

반복하면 전역 최소점을 얻는다. 그럼 2에 이 전략을 개략적으로 표현하였다. 이를 수행하기 위해서는 feasible point를 찾기 위한 전역 알고리즘과 local minimum을 찾기 위한 지역 알고리즘이 필요하다. 이 전략을 사용하는 확률론적 방법이 뒤에 소개된다.

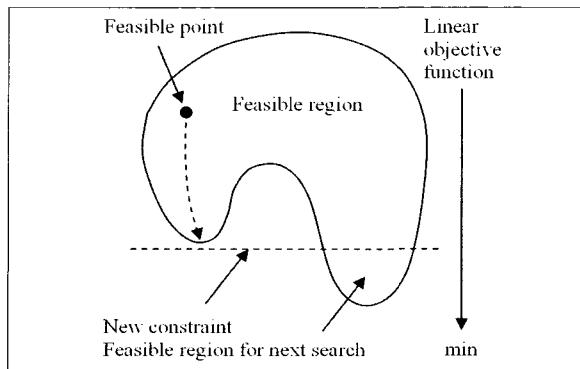


그림 2. Feasible point strategy [13].

2.3. Find all Kuhn-Tucker points

만약 연립 비선형 대수방정식의 모든 실근을 찾을 수 있는 방법이 있다면 이를 전역 최적화에 적용할 수 있다. 모든 Kuhn-Tucker 점들을 구한 뒤 이들의 목적함수 값을 비교하여 전역 최적점을 찾으면 된다. 이 접근법은 본 글의 범위에서 제외키로 한다.

3. 결정론적 접근법

결정론적 접근법에 의한 전역 최적화는 1980년대 이후 활발히 연구되어 왔으며 참고문헌 [10]에 약간의 역사가 요약되어 있다. 그러나 nonconvex nonlinear program의 전역 최적화는 일반적으로 계산시간이 문제크기의 지수함수로 증가하는 NP-hard problem 중에서도 가장 까다로운 것에 속한다. 따라서 현재 알려진 결정론적 알고리즘들은 대개 작은 문제들에만 적용 가능하다. 이는 컴퓨터의 속도와 용량이 증가한다고 해결되는 문제가 결코 아니다. 이에 대한 자세한 내용은 참고문헌 [13]에 설명되어 있다.

3.1. Outer Approximation

임의의 가능영역에 대한 convex hull은 선형 부등식 제약 조건들의 집합으로 구축할 수 있다. 예를 들어 Horst와 Tuy[1]는 그림3에 표시된 것과 같이 꼭지점 \mathbf{x}^* 를 갖는 다면체 뿔 (polyhedral cone) K 와 역 볼록 제약조건(reverse convex constraint) $\mathbf{x} \in \text{int } G$ 로 형성되는 영역 $K \setminus G$ 에 대한 concavity cut을 정의하였다. 이 선형 부등식 제약조건에 의한 것은 가능영역은 건드리지 않으면서 infeasible point \mathbf{x}^* 주위의 일부 영역을 제거한다.

유한하고 닫힌 볼록 영역 내에서 정의된 어떠한 함수도 두 개의 아래로 볼록한 함수들의 차이(difference of two convex (d.c.) functions)로 표시할 수 있다는 점에 주목하자. 이는 주어진 함수에 충분히 아래로 볼록한 함수를 더하고 빼면 된다. 주어진 볼록 영역 내에서 모든 함수들이 d.c.인 문제(d.c. problem)에는 위에 언급한 것과 같은 원리의 cutting plane method가 적용 가능하다[1].

이 방법의 큰 단점은 꼭 맞는 convex hull에 수렴하면서 제약조건의 개수가 계속 증가한다는 것이다. 따라서 유용하지 않은 제약조건들을 수시로 퇴출시키는 전략이 요구된다[1]. 또한 이 방법은 low rank nonconvex problem, 즉 문제를 nonconvex하게 만드는 변수들의 개수가 적은 문제에만 적합하다[14].

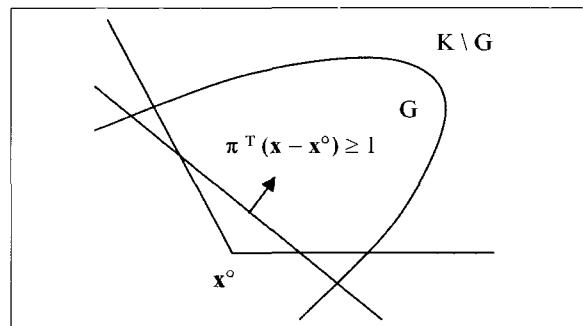


그림 3. (G, K)-cut [1].

3.2. Generalized Benders Decomposition

다음 형태의 문제를 생각해보자.

$$\begin{aligned} & \min_{\mathbf{x}, \mathbf{y}} f(\mathbf{x}, \mathbf{y}) \\ & \text{subject to} \\ & \quad g(\mathbf{x}, \mathbf{y}) \leq 0 \end{aligned}$$

여기서 f 와 g 는 \mathbf{x} 에 대해 아래로 볼록하다. 즉 \mathbf{y} 가 상수이면 convex problem이 된다. 이 문제는 다음과 같이 두 문제로 분해할 수 있다.

Primal:

$$\begin{aligned} & \min_{\mathbf{x}} f(\mathbf{x}, \mathbf{y}) \\ & \text{subject to} \\ & \quad g(\mathbf{x}, \mathbf{y}) \leq 0 \end{aligned}$$

여기서 \mathbf{y} 는 주어진 점에서 고정된다. 이 문제는 볼록하므로 쉽게 풀 수 있다. 단 원래 문제에 비해 가능영역이 좁아졌으므로 이 문제의 해는 전역 최소점에 대한 upper bound이다.

Master :

$$\min_{\mathbf{y}} y_0$$

subject to

$$\begin{aligned} L^*(\mathbf{y}; \mathbf{u}) &= \min_{\mathbf{x}} [f(\mathbf{x}, \mathbf{y}) + \mathbf{u}^T \mathbf{g}(\mathbf{x}, \mathbf{y})] \leq y_0 \\ &\text{for all } \mathbf{u} \geq \mathbf{0} \end{aligned}$$

$$\begin{aligned} L^*(\mathbf{y}; \mathbf{v}) &= \min_{\mathbf{x}} \mathbf{v}^T \mathbf{g}(\mathbf{x}, \mathbf{y}) \leq 0 \\ &\text{for all } \mathbf{v} \in \{\mathbf{v} \mid \mathbf{v} \geq \mathbf{0}, \sum_i v_i = 1\} \end{aligned}$$

여기서 \mathbf{u} 및 \mathbf{v} 는 실제로는 부분집합만 사용하게 되므로 이 문제에 대한 답은 전역 최소점에 대한 lower bound이다.

이 방법은 \mathbf{x} 와 \mathbf{y} 가 분리 가능할 때, 즉 모든 함수가 \mathbf{x} 의 함수 더하기 \mathbf{y} 의 함수 형태일 때 유용하다. 왜냐하면 primal에서 얻어진 해 \mathbf{x} 와 Lagrange multiplier \mathbf{u} 를 master 첫 번째 제약조건 집합에 그대로 적용할 수 있기 때문이다. 즉 master 내에서의 \mathbf{x} 에 대한 최소화가 불필요해진다.

만약 주어진 \mathbf{y} 에서 primal이 infeasible하면 다음과 같은 infeasibility minimization problem을 풀 수 있다.

$$\min_{\mathbf{x}} \alpha$$

subject to

$$\mathbf{g}(\mathbf{x}, \mathbf{y}) \leq \alpha$$

여기서 $\mathbf{1} = [1 \cdots 1]^T$ 이다. 만약 \mathbf{x} 와 \mathbf{y} 가 분리 가능하다면 이 문제에서 얻어진 해 \mathbf{x} 및 Lagrange multiplier \mathbf{v} 는 master의 두 번째 제약조건 집합을 구축하는데 사용될 수 있다. 즉 여기서도 \mathbf{x} 에 대한 최소화는 불필요하다.

이 알고리즘은 primal과 master를 upper bound와 lower bound가 서로 수렴할 때까지 반복해서 푼다. 이 방법 또한 outer approximation과 같이 제약조건 퇴출 전략이 필요하며 low rank nonconvex problem에만 적합하다.

3.3. Branch and Bound

이것은 다양한 문제의 전역 최적화에 가장 널리 사용되는 기법이다. 다음 형태의 문제를 생각해보자.

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

subject to

$$\begin{aligned} \mathbf{g}(\mathbf{x}) &\leq \mathbf{0} \\ \mathbf{A} \mathbf{x} &= \mathbf{c} \end{aligned}$$

여기서 \mathbf{A} 및 \mathbf{c} 는 각각 상수 행렬 및 벡터이다.

이 알고리즘은 초기 상자 $\mathbf{a} \leq \mathbf{x} < \mathbf{b}$ 에 대하여 위 문제를 다음과 같이 convex problem으로 이완시키는 것으로

로부터 시작한다. 먼저 목적함수 $f(\mathbf{x})$ 를 그림 4에서 점선으로 표시한 것과 같은 convex envelope로 교체한다. 가능한 영역에 대한 convex hull은 모든 nonconvex function $g_i(\mathbf{x})$ 를 그들의 convex envelope로 교체함으로써 얻을 수 있다. 이렇게 얻어진 문제는 convex하므로 지역 최적화 알고리즘으로 최소점을 구할 수 있다. 단 이 문제는 이완된 문제이므로 얻어진 최소점은 원래 문제의 전역 최소점에 대한 lower bound가 된다.

다음 단계는 이 상자를 쪼개는 것이다(branching). 각 subproblem에 대한 해는 해당 영역에 대한 lower bound이다. 이를 중 가장 낮은 값이 원래 문제의 전역 최소점에 대한 lower bound이다. 만약 어떤 해가 원래 제약조건들 까지 만족한다면 그 점에서의 원래 목적함수 값은 upper bound가 된다. 이를 중 가장 낮은 upper bound가 전역 최소점 후보로 저장된다. 한편 모든 subproblem은 infeasible하거나 해가 upper bound보다 높을 경우 버려진다 (bounding). 이 알고리즘은 lower bound가 upper bound에 수렴하면 끝난다.

이 알고리즘의 효율은 주로 convex envelope들이 원래 함수들에 얼마나 가까운가에 달려있다. 이들은 원래 함수보다 항상 작거나 같은 값을 주므로 underestimator라고도 불리며 가장 흔히 사용되는 것들은 다음과 같다.

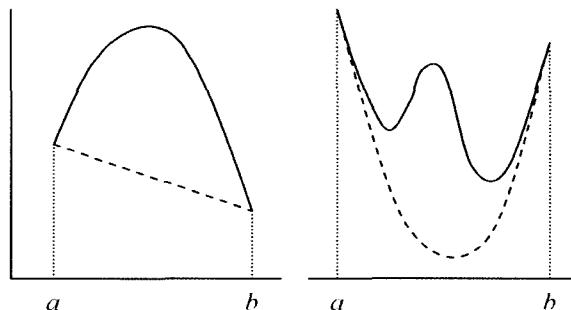


그림 4. Convex envelopes.

3.3.1 Linear underestimators

분리 가능한 함수 $\phi(\mathbf{x}) = \sum_j \phi_j(x_j)$ 내에 reverse convex term이 있다면 이 항에 대해서는 linear underestimator를 사용할 수 있다. 예를 들어, 만약 $\phi_j(x_j) = -x_j^2$, $a_j \leq x_j \leq b_j$ 이면 꼭 맞는 convex envelope는 다음과 같은 일차함수이다.

$$\psi_j(x_j) = -(a_j + b_j)x_j + a_j b_j$$

이 접근법은 분리 가능한 문제에만 적용 가능하다는 점에 주목하자. 그러나 어떠한 문제도 분리 가능한 문제로 변환 할 수 있다. 왜냐하면 분리가 안 되는 곱 $x_1 x_2$ 는 분리 가능한 함수 $w_1^2 - w_2^2$ 로 대체할 수 있기 때문이다. 여기서 w_1 및

w_2 는 다음과 같은 선형 등식 제약조건으로 정의된다.

$$w_1 = (x_1 + x_2) / 2$$

$$w_2 = (x_1 - x_2) / 2$$

3.3.2 Quadratic underestimators

일반 함수에 대해서는 quadratic underestimator를 사용할 수 있다[7]. 어떤 nonconvex function $\phi(\mathbf{x})$, $\mathbf{a} \leq \mathbf{x} \leq \mathbf{b}$ 가 있을 때 convex envelope는 다음과 같이 정의할 수 있다.

$$\begin{aligned}\psi(\mathbf{x}) &= \phi(\mathbf{x}) + \sum_i \alpha_j(x_j - a_j)(x_j - b_j) \\ \alpha_j &\geq \max\{0, -0.5 \min \lambda_k[\mathbf{H}(\phi(\mathbf{x}))]\}\end{aligned}$$

여기서 λ_k ($k = 1, \dots, n$)는 Hessian matrix \mathbf{H} 의 eigenvalue들이다. 이 접근법의 문제는 \mathbf{H} 가 상수가 아닌 한 꼭 맞는 α_j 의 결정은 또다시 nonconvex optimization problem이라는 점이다.

3.4. Interval Analysis

구간해석으로도 convex envelope와 hull을 만들 수 있는데 다음과 같은 구간산수(interval arithmetic)에 기초를 둔다.

$$[a, b] + [c, d] = [a + c, b + d]$$

$$[a, b] - [c, d] = [a - d, b - c]$$

$$[a, b] \times [c, d] = [\min(ac, ad, bc, bd), \max(ac, ad, bc, bd)]$$

$$\begin{aligned}[a, b] \div [c, d] &= [\min(a/c, a/d, b/c, b/d), \\ &\max(a/c, a/d, b/c, b/d)] \text{ if } 0 \in [c, d]\end{aligned}$$

앞서 설명한 linear 및 quadratic underestimator의 경우와는 달리 이제는 branch and bound 알고리즘을 문제 (P)에 직접 적용할 수 있다. 변수 \mathbf{x} 에 대한 구간상자 X 로부터 시작하자. 임의의 $X^k \subset X$ 에 대해 만약 $\text{lb } g_i(X^k) > 0$ or $\text{lb } h_i(X^k) > 0$ or $\text{ub } h_i(X^k) < 0$ 이면 상자 X^k 는 infeasible하다. 또한 $\text{lb } f(X^k) > f(\mathbf{x}^\circ)$ 일 때도 상자 X^k 를 버리는데, 여기서 \mathbf{x}° 는 알려진 feasible point이다. 이 구간 상자들은 반복적으로 branch and bound 과정을 거치며, 전역 lower bound 근처에서 feasible point가 발견되면 끝난다.

구간해석에 기초한 convex envelope 및 hull은 각각 상수 및 구간일 뿐이므로 매우 느슨하다. 따라서 구간해석을 이용하는 branch and bound 알고리즘은 매우 많은 subproblem을 생성하게 된다. 그러나 각각의 subproblem은 매우 효율적으로 풀 수 있는데, 단지 간단한 구간산수만 필요하기 때문이다.

3.5. Handling Equalities

많은 결정론적 알고리즘들이 특정한 형태의 문제들에만 적용 가능하다. 예를 들면 앞서 소개한 generalized Benders decomposition은 부등식 제약조건만 허용하며 underestimator branch and bound는 부등식 및 선형 등식 제약조건만 허용한다. 선형 등식 제약조건은 nonconvexity를 초래하지 않으므로 대개는 이들을 허용하도록 알고리즘을 수정할 수 있다. 그러나 만약 비선형 등식제약조건이 존재한다면 interval branch and bound를 사용하지 않는 한 문제를 변형해야 한다.

가장 단순한 방법은 $\mathbf{h}(\mathbf{x}) = \mathbf{0}$ 을 $\mathbf{h}(\mathbf{x}) \leq \mathbf{0}$ and $\mathbf{h}(\mathbf{x}) \geq \mathbf{0}$ 으로 대체하는 것이다. 또한 $\mathbf{h}(\mathbf{x}) \leq \mathbf{0}$ and $\sum_i h_i(\mathbf{x}) \geq 0$ 도 $\mathbf{h}(\mathbf{x}) = \mathbf{0}$ 과 동치임에 주목하자. 그러므로 m 개의 등식 제약조건은 $2m$ 또는 $m + 1$ 개의 부등식 제약조건으로 변환할 수 있다. 이제 generalized Benders decomposition을 등식 제약 문제에도 적용할 수 있다. 또한 linear underestimator branch and bound도 아무 문제에나 적용 가능하다. 왜냐하면 $\mathbf{h}(\mathbf{x})$ 를 분리 가능한 함수로 변환할 수 있기 때문이다.

등식 제약조건을 d.c. 함수를 사용하여, 즉 $\mathbf{h}(\mathbf{x}) = \mathbf{c}(\mathbf{x}) + \mathbf{r}(\mathbf{x}) = \mathbf{0}$ 으로 나타내자. 여기서 $\mathbf{c}(\mathbf{x})$ 는 아래로 볼록(convex)하고 $\mathbf{r}(\mathbf{x})$ 는 위로 볼록(reverse convex)하다. 이들에 대해 새로운 변수 $\mathbf{u} = \mathbf{c}(\mathbf{x})$ 및 $\mathbf{v} = \mathbf{r}(\mathbf{x})$ 를 정의하자. 그러면 비선형 등식 제약조건 $\mathbf{c}(\mathbf{x}) + \mathbf{r}(\mathbf{x}) = \mathbf{0}$ 은 선형 등식 제약조건 $\mathbf{u} + \mathbf{v} = \mathbf{0}$, 볼록 부등식 제약조건 $\mathbf{c}(\mathbf{x}) - \mathbf{u} \leq \mathbf{0}$ 및 $-\mathbf{r}(\mathbf{x}) + \mathbf{v} \leq \mathbf{0}$, 그리고 역 볼록 부등식 제약조건 $-\mathbf{c}(\mathbf{x}) + \mathbf{u} \leq \mathbf{0}$ 및 $\mathbf{r}(\mathbf{x}) - \mathbf{v} \leq \mathbf{0}$ 으로 변환할 수 있다. 여기서 마지막 $2m$ 개의 역 볼록 부등식 제약조건을 합치면 하나의 역 볼록 부등식 제약조건 $\sum_i [-c_i(\mathbf{x}) + r_i(\mathbf{x}) + u_i - v_i] \leq 0$ 이 된다. 그러므로 m 개의 비선형 등식 제약조건은 $2m$ 개의 추가변수를 사용하여 m 개의 선형 등식 제약조건, $2m$ 개의 볼록 부등식 제약조건, 그리고 하나의 역 볼록 부등식 제약조건으로 변환할 수 있다. 따라서 우리가 필요한 것은, 이론적으로는, 단지 볼록 문제에 하나의 역 볼록 제약조건이 추가된 문제를 풀 수 있는 알고리즘이며, 여기에 설명된 모든 결정론적 알고리즘들이 그 일을 할 수 있다. 단 위 과정에서 $2m$ 개의 역 볼록 부등식 제약조건을 하나로 합치는 것보다는 그대로 사용하는 것이 유리한데, 위로 볼록한 항들을 모두 더하면 원래 함수와 convex envelope 사이의 간격이 훨씬 커지기 때문이다.

4. 확률론적 접근법

확률론적 알고리즘들은 결과의 전역 최적성을 보장할 수 없으며 단지 전역 최적성의 확률을 높일 수 있다. 이는 다음과 같은 수렴성 정리(convergence theorem)[15]로부터 알 수 있다.

For minimization of objective function $f(\mathbf{x})$,

- 1) Let $\mathbf{x}^{t+1} = \mathbf{x}^t + N(0, \sigma)$.
- 2) If $f(\mathbf{x}^{t+1}) < f(\mathbf{x}^t)$, accept \mathbf{x}^{t+1} .
Otherwise, $\mathbf{x}^{t+1} = \mathbf{x}^t$.
- 3) Repeat for next t .

Then, for $\sigma > 0$ and $f^{\min} > -\infty$,

$$\lim_{t \rightarrow \infty} p\{f(\mathbf{x}^t) = f^{\min}\} = 1.$$

이것은 정규분포를 기초로 random search를 하면 결과가 전역 최적점일 확률이 궁극적으로 1에 접근함을 의미한다. 그러나 확률론적 알고리즘이 효율적이려면 가장 좋은 해를 이용하는 것(local search)과 탐색공간을 조사하는 것(global search) 사이에 균형이 필요하다[16]. 위의 알고리즘은 지역 탐색(local search)에 치우쳐있다. 전역탐색(global search)을 강화한 대표적인 두 방법을 다음과 같이 요약하였다.

4.1. Simulated Annealing

주어진 온도 T 에서 평형상태에 있는 원자들의 모임을 생각해보자. 하나의 원자가 이동하면 시스템의 에너지에 변화 ΔE 가 생긴다. 만약 $\Delta E \leq 0$ 이면 이 이동은 발생한다. 만약 $\Delta E > 0$ 이면 이 이동이 발생할 확률은 $\exp(-\Delta E / kT)$ 이다. 여기서 k 는 Boltzmann 상수이다. 이 과정을 다음과 같이 최적화에서 시뮬레이션 할 수 있다.

For minimization of objective function $f(\mathbf{x})$,

- 1) Take \mathbf{x}^{new} randomly.
- 2) If $\Delta f = f(\mathbf{x}^{\text{new}}) - f(\mathbf{x}^{\text{old}}) \leq 0$, accept \mathbf{x}^{new} .
Otherwise,
 - a) Take a random number $w \in [0, 1]$.
 - b) If $w \leq \exp(\Delta f / T)$, then accept \mathbf{x}^{new} .
Otherwise, $\mathbf{x}^{\text{new}} = \mathbf{x}^{\text{old}}$.
- 3) Control T , and repeat.

이 알고리즘은 주로 조합 최적화(combinatorial optimization) 문제에 적용되나 제약조건 없는 함수 최적화에도 적합하다.

4.2. Genetic Algorithm

진화이론 또한 다음과 같이 최적화에 도입될 수 있다.

For optimization of fitness function $f(\mathbf{x})$,

- 1) Select a given size of population $\{\mathbf{x}^i\}$ where \mathbf{x}^i is a chromosome (binary vector).
- 2) At a given crossover probability, crossover \mathbf{x}^p and \mathbf{x}^q to generate $\mathbf{x}^{p'}$ and $\mathbf{x}^{q'}$.
- 3) At a given mutation rate, mutate \mathbf{x}^i .
- 4) Repeat.

이 알고리즘은 가장 좋은 해는 좋은 해들이 상대적으로 높은 비율로 포함된 영역에서 발견될 것이라는 가정에 기초를 두고 있다. 유전 알고리즘은 주어진 문제의 특성을 살린 특별한 자료구조에 맞도록 변형되어 사례별로 널리 적용되고 있다. 변형된 유전 알고리즘은 evolution program이라고도 부른다[17].

원래의 유전 알고리즘은 염색체를 이진수로 나타내므로 조합 최적화에 적합하다. 그러나 이 알고리즘은 제약조건 없는 함수 최적화에도 적용할 수 있다. 이 경우 염색체에 해당하는 것은 공간 내 점들이므로 실수로 표현하는 것이 더 효율적이며 이들에 대해 다음과 같은 연산자들을 사용할 수 있다.

For randomly selected $j \in \{1, \dots, n\}$,

- 1) Simple crossover: $\mathbf{x}^{p'} = [x_1^p, \dots, x_j^p, x_{j+1}^q, \dots, x_n^q]$ and $\mathbf{x}^{q'} = [x_1^q, \dots, x_j^q, x_{j+1}^p, \dots, x_n^p]$
- 2) Arithmetical crossover: $\mathbf{x}^{p'} = w \mathbf{x}^p + (1 - w) \mathbf{x}^q$ and $\mathbf{x}^{q'} = (1 - w) \mathbf{x}^p + w \mathbf{x}^q$
- 3) Uniform mutation: $x_j^i \in [a_j, b_j]$
- 4) Boundary mutation: $x_j^i = a_j$ or b_j
- 5) Non-uniform mutation: Fine tune x_j^i .

4.3. Handling Equalities

확률론적 알고리즘들은 NP-hard problem이라는 고통을 겪지 않아도 되므로 큰 문제에 적합할 것으로 예측된다. 그러나 이들은 random search에 기반을 두고 있고 목적함수 값 계산은 feasible point에서만 의미가 있으므로 제약조건이 없거나 원래부터 부등식 제약조건만 있는 문제에만 적합하다. 따라서 등식 제약조건이 많을 경우 문제 또는 알고리즘을 변형해야 한다.

4.3.1 Penalty function method

제약조건이 있는 최적화 문제는 다음과 같이 제약조건 없는 문제로 변환될 수 있다.

$$\min_x F(\mathbf{x}, r) = f(\mathbf{x}) + 1/(2r) \mathbf{c}(\mathbf{x})^T \mathbf{c}(\mathbf{x})$$

여기서 r 은 penalty parameter (> 0), 그리고 $\mathbf{c}(\mathbf{x})$ 는 모든 active constraint의 합수로 구성된 벡터이다. 만약 r 이 0에 접근하면 \mathbf{x} 는 지역 최소점 \mathbf{x}^* 에 수렴한다. 그러나 Hessian matrix $H(F(\mathbf{x}, r))$ 은 ill-conditioned matrix가 된다. 게다가 등식 제약조건이 너무 많으면 위의 목적함수는 수치적으로도 안정되기 어렵다.

4.3.2 Feasible point strategy

확률론적 알고리즘에서 등식 제약조건을 다루는 것을 피하기 위해 feasible point 전략을 도입할 수 있다. 여기서 feasible point는 다음과 같은 infeasibility minimization

problem을 풀면 찾을 수 있다.

$$\min \mathbf{c}(\mathbf{x})^T \mathbf{c}(\mathbf{x})$$

여기서 $\mathbf{c}(\mathbf{x})$ 는 위반된 모든 제약조건의 함수들로 구성된 벡터이다. 또 다른 형태의 위반도 최소화 문제는 다음과 같다.

$$\min \max\{\mathbf{g}(\mathbf{x}), \mathbf{h}(\mathbf{x}), -\mathbf{h}(\mathbf{x})\}$$

비 불록 영역의 feasible point를 찾는 것은 NP-complete problem이다[13]. 따라서 NP-hard problem인 비 불록 문제의 전역 최적화보다는 쉽다고 할 수 있다.

위 문제들과 원리는 같으나 앞서 언급한 feasible point 전략을 적용한 예를 들면 다음과 같다[13].

$$\min \alpha \quad (I)$$

subject to

$$\mathbf{g}(\mathbf{x}) \leq \alpha \mathbf{1}$$

$$\mathbf{h}(\mathbf{x}) \leq \alpha \mathbf{1}$$

$$-\mathbf{h}(\mathbf{x}) \leq \alpha \mathbf{1}$$

$$f(\mathbf{x}) \leq f^* - \epsilon + \alpha$$

여기서 $\mathbf{1} = [1 \cdots 1]^T$, f^* 는 현재 알려진 지역 최소점에서의 목적함수 값, 그리고 $\epsilon (> 0)$ 은 최적성 허용오차(optimality tolerance)이다. 만약 $\alpha < \epsilon$ 이라면 이것은 \mathbf{x} 가 이전에 발견한 지역 최소점보다 낮은 feasible point라는 뜻이다. 이로부터 더 낮은 지역 최소점을 구할 수 있으며 같은 과정을 반복하면 전역 최소점을 구할 수 있다. 이 전략의 실행은 그림 5에 개략적으로 표현되어 있다.

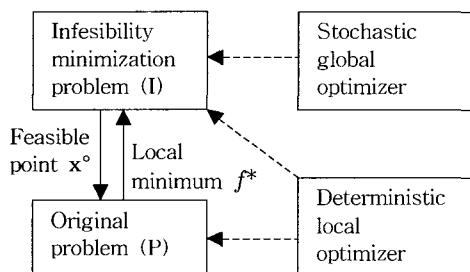


그림 5. A stochastic method based on feasible point strategy.

4.3.3 Decoding strategy

문제 (P)에서 모든 부등식 제약조건 $\mathbf{g}(\mathbf{x}) \leq \mathbf{0}$ 을 등식 제약조건 $\mathbf{g}(\mathbf{x}) + \mathbf{s} = \mathbf{0}$ 으로 변환하자. 여기서 \mathbf{s} 는 음이 아닌 slack variable 벡터이다. 이제 그 문제는 다음 형태로 나타낼 수 있다.

$$\min f(\mathbf{x}, \mathbf{y}) \quad (E)$$

subject to

$$\mathbf{h}(\mathbf{x}, \mathbf{y}) = \mathbf{0}$$

$$\mathbf{a} \leq \mathbf{x} \leq \mathbf{b}$$

$$\mathbf{c} \leq \mathbf{y} \leq \mathbf{d}$$

여기서 \mathbf{x} 는 $n - m$ 개의 설계(독립)변수, \mathbf{y} 는 m 개의 상태(종속)변수, 그리고 $\mathbf{h}: R^n \rightarrow R^m$ ($n > m$)이다. 방정식 풀이에 의해 \mathbf{x} 를 \mathbf{y} 로 해독(decoding)할 수 있다고 가정하면 이 문제는 다음과 같이 볼 수 있다.

$$\min f(\mathbf{x}, \mathbf{y}(\mathbf{x}))$$

subject to

$$\mathbf{a} \leq \mathbf{x} \leq \mathbf{b}$$

이 형태의 문제는 확률론적 알고리즘들에 적합하다. 게다가 원래 문제에 등식 제약조건이 많아 자유도가 작다면 이 문제는 작은 문제가 된다. 그러나 확률론적 알고리즘은 국지적 세부조정에는 비효율적이다. 따라서 결정론적 지역 알고리즘을 함께 사용해야 한다.

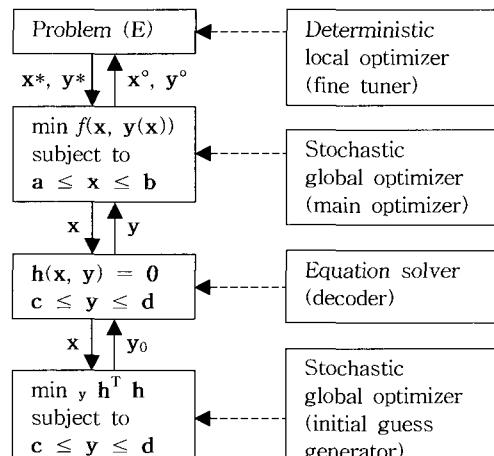


그림 6. A stochastic method based on decoding strategy.

이 전략은 $\mathbf{y}(\mathbf{x})$ 를 쉽게 계산할 수 있을 때, 예를 들면 \mathbf{h} 가 \mathbf{y} 에 대해 선형일 때 효과적이며, 특별한 문제에만 적용되는 알고리즘들에 널리 사용되고 있다. 일반적인 문제를 위해서는 강건한 방정식 풀이방법이 있어야 하는데, 좋은 initial guess를 구할 수만 있다면 Newton 방식의 알고리즘도 사용할 수 있다. 앞서 설명한 infeasibility minimization problem에 대한 근사해가 좋은 initial guess 역할을 할 수 있으며 확률론적 전역 최적화 알고리즘에 큰 최적성 허용오차를 줌으로써 구할 수 있다. 이제 주어진 \mathbf{x} 에 대한 \mathbf{y} 를 구하기 위해 방정식 $\mathbf{h}(\mathbf{x}, \mathbf{y}) = \mathbf{0}$ 을 풀어야 하는데 근이 없거나 여러 개 존재하더라도 문제되지 않는다. 존재하는 해 중 하나만 구해지겠지만 결과가 확률적이기 때문이다. 이 전략의 실행은 그림 6에 개략적으로 표현되어 있다.

5. 결론

결정론적 방법들은 전역 최소점에 대한 lower bound를 증가시키며 feasible point를 찾는다. 이를 중 추천할 만한 것은 branch and bound로서, 특히 linear underestimator를 사용하되 interval analysis를 통해 subproblem 영역을 좁혀주면 효과적이다. 그러나 여전히 작은 문제에만 적용 가능하다. 전역 최적성을 보장 받으려면 너무나도 비싼 대가를 지불해야 하기 때문이다.

확률론적 방법들은 보다 낮은 지역 최소점을 탐색한다. 즉 upper bound를 감소시킨다. 이들은 전역 최적성을 보장하지 않지만 비교적 큰 문제에도 적용 가능하다. 또한 중간 결과가 최소한 지역 최소점이므로 적당한 시간 내에 유용한 결과를 얻어 사용할 수 있다. 시간이 더 허용되면 더 좋은 해를 찾을 가능성을 위하여 계산을 계속할 수 있다.

참고문헌

1. R. Horst and H. Tuy, *Global Optimization: Deterministic Approaches*, 2nd ed., Springer-Verlag, Berlin, Germany, 1993.
2. A. M. Geoffrion, "Generalized Benders Decomposition," *J. Opt. Theory Applic.*, vol. 10, pp. 237-260, 1972.
3. C. A. Floudas and V. Visweswaran, "A Global Optimization Algorithm (GOP) for Certain Classes of Nonconvex NLPs - I. Theory," *Computers & Chem. Eng.*, vol. 14, pp. 1397-1417, 1990.
4. M. Bagajewicz and V. Manousiouthakis, "On the Generalized Benders Decomposition," *Computers & Chem. Eng.*, vol. 15, pp. 691-700, 1991.
5. R. M. Soland, "An Algorithm for Separable Nonconvex Programming Problems II: Nonconvex Constraints," *Management Science*, vol. 17, pp. 759-773, 1971.
6. H. S. Ryoo and N. V. Sahinidis, "Global Optimization of Nonconvex NLPs and MINLPs with Applications in Process Design," *Computers & Chem. Eng.*, vol. 19, pp. 551-566, 1995.
7. C. S. Adjiman, I. P. Androulakis, C. D. Maranas, and C. A. Floudas, "A Global Optimization Method, α BB, for Process Design," *Computers & Chem. Eng.*, vol. 20, Suppl., pp. S419-S424, 1996.
8. H. Ratschek and J. Rokne, *New Computer Methods for Global Optimization*, Ellis Horwood, Chichester, England, 1988.
9. R. Vaidyanathan and M. El-Halwagi, "Global Optimization of Nonconvex Nonlinear Programs via Interval Analysis," *Computers & Chem. Eng.*, vol. 18, pp. 889-897, 1994.
10. J. R. Han, V. Manousiouthakis, and S. H. Choi, "Global Optimization of Chemical Processes Using the Interval Analysis," *Korean J. of Chem. Eng.*, vol. 14, pp. 270-276, 1997.
11. S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi, "Optimization by Simulated Annealing," *Science*, vol. 220, pp. 671-680, 1983.
12. D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, MA, 1989.
13. S. H. Choi, J. W. Ko, and V. Manousiouthakis, "A Stochastic Approach to Global Optimization of Chemical Processes," *Computers & Chem. Eng.*, vol. 23, pp. 1351-1356, 1999.
14. H. Konno, P. T. Thach, and H. Tuy, *Optimization on Low Rank Nonconvex Structures*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1997.
15. T. Bäck, F. Hoffmeister, and H.-P. Schwefel, "A Survey of Evolution Strategies," *Proceedings of the Fourth International Conference on Genetic Algorithms*, R. K. Belew and L. B. Booker, eds., pp. 2-9, Morgan Kaufmann, San Mateo, CA, 1991.
16. L. B. Booker, "Improving Search in Genetic Algorithms," *Genetic Algorithms and Simulated Annealing*, L. Davis, ed., pp. 61-73, Pitman, London, 1987.
17. Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, 3rd ed., Springer-Verlag, New York, 1996.

저자약력



《최 수 형》

- 1984년 서울대학교 화학공학과 학사.
- 1986년 서울대학교 화학공학과 석사.
- 1990년 University of Missouri-Rolla 박사.
- 1991년~1993년 UCLA 박사후 연구원.
- 1993년~현재 전북대학교 화학공학부 교수.