

論文94-31B-9-4

가변의 조정변수를 갖는 복합된 1-2 단계 최적화 알고리즘

(A Unified Phase I - Phase II Semi-Infinite Constrained Optimization Algorithm with a Varying Steering Parameter)

梁玄錫*

(Hyun Suk Yang)

要約

복합된 1-2단계 최적화 알고리즘에 조정변수를 첨가하면 효율성이 증대됨은 이미 잘 알려져 있다. 본 논문에서는 현재 벡터점이 모든 제약조건을 만족하는 지역에서 멀리 있으면 조정변수를 크게하여 제약조건을 우선적으로 만족하게 하고 만약 그 지역에 매우 접근하면 조정변수의 값을 줄여 목적함수의 증감도 함께 고려함이 효율적이라는 사실에 입각하여 새로운 알고리즘을 제시한다. 또한 벡터의 진행방향에 따른 목적함수의 증감을 고려하는 조정변수의 선택법을 제시한다. 비록 가변 조정변수를 사용하더라도 이론적인 접근속도는 변하지 않으나 예제를 통한 시물레이션에 의하면 본논문에서 제시한 새로운 알고리즘이 기존의 것보다 효율적임을 보였다.

Abstract

It is known that a unified phase I-phase II semi-infinite optimization algorithm with a steering parameter performs better than the original unified phase I-phase II algorithm. In this paper, the effect of the steering parameter is analyzed and a new algorithm is presented based on the facts that when the point x is far away from the feasible region, reaching to the feasible region is more important than minimizing the cost function and that when the point x is near the region, it is more efficient to try to reach the feasible region and to minimize the cost function concurrently. It is also important to consider the relationship between the feasible direction and the gradient of the cost function. Even though changing the steering parameter does not change the rate of convergence of the algorithm, it is shown from examples that given new algorithm is more efficient than the previous ones.

1. 서론

*正會員, 弘益大學校 電子工學科

(Dept. of Elec. Eng., Hongik Univ.)

* 본 논문은 93년도 한국과학재단 연구비 지원에 의한 결과임

接受日字 : 1994年 2月 1日

시스템의 제어 입력을 구성함에 있어서 주어진 시스템이나 주변기기의 입력, 출력 또는 state의 값이 한정된 범위내에 있기를 요구하거나 이들의 변화율 등이 어떤 값이상을 넘지 않기를 요구하는 경우가 많

다. 이러한 제어 입력은 최적제어 이론을 적용하여 원하는 제어입력을 구한다. 즉, 제어입력은 다음과 같이 최소화 (또는 최대화)하고자 하는 목적함수, $f^0(x)$ 와 입력, 출력 또는 state 값들의 범위 등으로 주어지는 제약함수, $\psi^j(x)$, 로 구성된 최적제어 문제를 해결하여 얻어진다.^{1,5}

$$\min_{x \in R^n} \{ f^0(x) \mid \psi^j(x) \leq 0, j=1, \dots, m \} \quad (1)$$

(1)에서 제약조건이 $\psi^j(x) = \max_{y \in Y_j} \phi^j(x, y) \leq 0, Y_j \in R^n$, 의 형태이면 제약함수 $\psi^j(\cdot)$ 개수는 m 개로 한정되어 있으나 제약함수는 모든 $x \in R^n$ 에서 무한개 함수의 최대값으로 이루어져 있어 이를 Semi-infinite 최적화 문제라 한다.

컴퓨터 공학의 발달로 단위시간당 계산량이 증가하여 (1)의 최적해를 짧은 시간내에 구하는 것이 가능하게 되어 제약조건이 있는 많은 공학문제에서 이방법이 이용되고 있다.^{3,4,6,8} (1)의 최적해를 구함에 있어 그 난이도는 주어진 제약함수의 수, 함수들의 미분 가능성 등에 따라 결정된다.

(1)의 최적화 문제는 모든 함수가 연속미분가능하다고 가정하면 난이도에 있어 가장 기초적인 형태가 된다. 1960년대에서 1970년대 초반에 걸쳐 (1)의 최적해를 효율적으로 구하기 위하여 많은 알고리즘이 개발되었으며^{4,6,9,10} 1970년대 말부터 보다 복잡한 형태의 최적화 문제에 대한 알고리즘도 개발되었다.^{16,17,19} 일반적으로 모든 제약조건을 만족하는 벡터 x 의 집합을 feasible region이라 하는데 이 지역안에 존재하는 벡터를 구하는 것을 1단계라 하고, 여기서 얻은 벡터를 초기치로 이 지역안에서 최적해를 구하는 것을 2단계라 한다. 초기에 제외된 (1)의 해결 방법으로는 feasible direction 이란 방법이 있는데 이의 1단계 과정에서는 (1)의 문제를 변형시켜 feasible region에 속하는 벡터를 임의로 선택하여 2단계 방법을 적용할 수 있게한 방법으로 이를 자세히 살펴보면 다음과 같다. 우선 확장공간 R^{n+1} 안에서 벡터 $z = (x, x_{n+1})$, $x \in R^n$, $x_{n+1} \in R$ 를 정의한다. 그 다음 확장공간에서의 최적화 문제를 다음과 같이 정의한다.

$$\min_{z \in R^{n+1}} \{ x_{n+1} \mid f_j(x) - x_{n+1} \leq 0, j=1, \dots, m \} \quad (2)$$

이문제를 해결하는데 있어 초기벡터 z_0 는 임의의 $x_0 \in R^n$ 에 대하여 $x_{n+1,0} > \max_j f^j(x_0)$ 인 $x_{n+1,0}$ 을 선택하여 $z_0 = (x_0, x_{n+1,0})$ 로 정의한다. 그러면 z_0 에서 (2)의 모든 제약조건이 만족되므로 1단계 과정은 필요치 않게 되어 2단계 과정만을 적용하면 된다. 이러한 과

정을 통하여 얻어진 (x^*, x_{n+1}^*) 이 (2)의 최적해이면 $x_{n+1}^* = \min_x \max_j f^j(x) = \max_j f^j(x^*)$ 을 만족한다.

즉 (1)의 feasible region이 공집합이 아니면 (즉 $\max_j \psi^j(x) \leq 0$ 을 만족하는 벡터 $x \in R^n$ 이 존재하면) $x_{n+1}^* \leq 0$ 이므로 벡터 x^* 는 (1)에서 주어진 모든 제약조건을 만족하게 되어 이점을 2단계 알고리즘의 초기 벡터로 사용할수 있다. 이방법의 단점은 x^* 점이 (1)의 최적해에서 매우 멀리 떨어져 있어 비효율적일수 있다는 것이다. 이러한 단점을 보완한것이 복합된 1-2 단계 알고리즘이고¹⁸ 이 알고리즘에 고정된 조정변수를 첨가하여 보다 효율화 시킨것이 조정변수를 갖는 복합된 1-2 단계 알고리즘이다.¹⁸

본 논문은 2장에서 초기의 복합된 1-2 단계 최적화 알고리즘과 조정변수가 첨가된 알고리즘에 대한 이론들을 참고문헌[5]와 [8]에 근거하여 간략히 서술하였으며 3장에서 조정변수의 값이 변할때 알고리즘에 미치는 영향을 다각도에서 분석하여 이에 따른 새로운 알고리즘을 제시한다. 4장에서는 여러 예제를 통하여 새로운 알고리즘을 테스트하였고 5장에서 결론을 맺었다.

II. 이론적 배경

본 논문에서 고려하는 반무한의 제약조건을 갖는 최적화 문제 (Semi-infinite constrained optimization problem)는 다음과 같다.

$$\min_{x \in R^n} \{ f^0(x) \mid \psi^i(x) \leq 0, i \in M \} \quad (3)$$

여기에서 $f^0(\cdot)$ 은 연속미분가능 함수이고, $M = \{1, \dots, m\}$, $\psi^j(x) = \max_{y_j \in Y_j} \phi^j(x, y_j)$, $j \in M$ 의 형태를 갖고 있으며, $\phi^j(\cdot, \cdot)$ 은 연속미분가능 함수이고 $Y_j = [a_j, b_j] \subset R$ 은 compact 집합으로 가정한다. 한가지 주목할 사항은 비록 $\phi^j(\cdot, y_j)$ 이 모든 $y_j \in Y_j$, $j \in M$ 에 대하여 연속미분가능이지만 $\psi^j(\cdot)$ 는 미분불능이라는 점이다. 이때문에 (3)을 (1)보다 더 복잡한 최적화 문제로 간주한다. 함수 $\psi^0(x) = \max_{y_0 \in Y_0} \phi^0(x, y_0)$ 로 정의하고 모든 $y_0 \in Y_0$ 에 대해서 $\phi^0(x, y_0) = f^0(x)$ 이면 (3)의 $f^0(x)$ 를 $\psi^0(x)$ 로 나타낼수 있다. 변수의 간소화를 위하여 모든 x 에 대하여 제약함수들의 최대값을 $\psi(x) = \max_{j \in M} \psi^j(x)$ 로 정의한다. 최적화 문제 (3)은 m 개의 $\psi^j(x)$ 을 제약조건으로 갖고 있는데 각 함수는 무한개 함수의 최대값인 $\max_{y_j \in Y_j} \phi^j(x, y_j)$ 로 정의되어 이를 반무한의 제약조건이라 부른다. Polak([5])에 의하여 제시된 (3)의 문제를 해결하는 1-2 단계 알고리즘은 다음과 같다.

알고리즘 I:^[5]

변수 : $x_0 \in \mathbb{R}^n, \alpha \in (0, 1), \beta \in (0, 1)$

단계 0: $i = 0$ 로 한다.

단계 1: 최적화 함수 (optimality function)의 값 $\theta_i = \theta(x_i)$ 와 진행방향 (search direction) 벡터 $h_i = h(x_i)$ 를 구한다.

단계 2: 만약 $\psi(x_i) > 0$ 이면

$$\lambda_i = \max \{ \beta^k | k \in N, \psi(x_i) \leq \beta^k \alpha \theta_i \} \quad (4)$$

만약 $\psi(x_i) \leq 0$ 이면

$$\lambda_i = \max \{ \beta^k | k \in N, \psi^0(x_i + \beta^k h_i) - \psi^0(x_i) \leq \beta^k \alpha \theta_i, \psi(x_i + \beta^k h_i) \leq 0 \} \quad (5)$$

단계 3: $x_{i+1} = x_i + \lambda_i h_i$ 로 정의하고 $i + 1$ 을 i 라 한후 단계 1로 간다.

최적화 함수 $\theta(x_i)$ 와 진행방향 $h(x_i)$ 를 구하는 방법은 [5]에서 주어져 있다. 알고리즘 I의 단계2는 만약 x_i 가 feasible region 안에 없으면 식 (4)를 이용한 1단계 과정을, 만약 x_i 가 이 지역 안에 있으면 식 (5)을 이용한 2단계 과정을 실행케한다. 주어진 알고리즘의 단점은 1단계의 과정에서는 (즉 식 (4)에서) 목적함수를 전혀 고려하지 않기 때문에 1단계 과정을 실행할때 목적함수의 값이 증가할수 있다는 점이다. 이러한 단점을 보완하여 (3)의 최적화 문제를 효율적으로 풀기 위하여 어떤 $x_i \in \mathbb{R}^n$ 에 대한 parametrized function을 다음과 같이 정의한다.^[8]

$$F_x(x) = \max \{ \psi^0(x) - \psi^0(x_i) - \gamma \psi_i(x), \psi^i(x) - \psi_i(x), i \in M \} \quad (6)$$

여기에서 $\psi_i(x) = \max \{ 0, \psi(x) \}$ 이고 조정변수 γ 는 영보다 큰 양의 상수이다. 참고문헌 [16]에서 다음과 같은 최적조건 (optimality condition)을 증명하였다.

정리 2.1: x^* 를 (3)의 local minimizer 라고 가정하고 $dF_x(x^*; h)$ 를 x^* 에서 벡터 h 의 방향으로 $F_x(\cdot)$ 의 directional derivative 라고 하면 모든 $h \in \mathbb{R}^n$ 에 대하여 $dF_x(x^*; h) \geq 0$ 이 만족된다. 다시 말하면 $\circ F_x(x^*)$ 이 $F_x(\cdot)$ 의 generalized gradient 일때 $0 \in \circ F_x(x)$ 을 만족한다. 여기에서 $\circ \psi^i(x)$ 가 $\psi^i(x)$ 의 generalized gradient 일때 $\circ F_x(x)$ 는 다음과 같이 정의 한다.

$$\partial F_x(x) = \infty \{ \partial \psi^j(x) | j \in J(x) \} \quad (7)$$

여기에서 $\circ \psi^i(x) = \nabla \psi^i(x)$, "co"는 convex hull을 의미하며

$$J(x) = \begin{cases} I(x), & \psi(x) > 0 \\ I(x) \cup \{0\}, & \psi(x) = 0 \\ \{0\}, & \psi(x) < 0 \end{cases}$$

로써 $I(x) = \{ j \in M | \psi^j(x) = \psi(x) \}$ 이다.

알고리즘의 개발을 위하여 $F_x(x)$ 의 선형 일차불록근사치 (linear first order convex approximation)를 다음과 같이 정의한다.

$$F_x(x) = \max \{ \hat{\psi}_i^0(x) - \psi^0(z) - \gamma \psi_i(z), \hat{\psi}_i^i(x) - \psi_i(z), j \in M \} \quad (8)$$

(8)에서 $\hat{\psi}_i^j(x)$, $i = 0, m$,는 z 에서 $\psi^j(x)$ 의 근사치로 다음과 같이 정의한다.

$$\hat{\psi}_i^j(x) = \max_{y \in Y_j} \phi^j(z, y_j) + \langle \nabla_x \phi^j(z, y_j), (x - z) \rangle + (1/2) \|x - z\|^2 \quad (9)$$

이를 이용하여 최적화 함수 $\theta(x)$ 와 진행방향 벡터 $h(x)$ 를 다음과 같이 정의한다.

$$\theta(x) = \min_{h \in \mathbb{R}^n} F_x(x+h), \quad (10)$$

$$h(x) = \arg \min_{h \in \mathbb{R}^n} F_x(x+h)$$

He와 Polak은^[8] (10)을 이용하여 다음과 같은 고정된 조정 변수를 갖는 복합된 1-2 단계 함수를 이용한 알고리즘을 제시하였다.

알고리즘 II:^[8]

변수 : $x_0 \in \mathbb{R}^n, \beta \in (0, 1), \gamma > 0, \alpha \in (0, 1)$.

단계 0: $i = 0$ 로 한다.

단계 1: 최적화 함수 (optimality function)의 값 $\theta_i = \theta(x_i)$ 와 진행방향 (search direction) 벡터 $h_i = h(x_i)$ 를 (10)을 이용하여 구한다.

단계 2: 스텝크기 (stepsize) λ_i 를 다음과 같이 구한다.

$$\lambda_i = \max \{ \beta^k | k \in N, F_x(x_i + \beta^k h_i) \leq \beta^k \alpha \theta_i \} \quad (11)$$

단계 3: $x_{i+1} = x_i + \lambda_i h_i$ 로 정의하고 $i + 1$ 을 i 라 한후 단계 1로 간다.

알고리즘 I과 알고리즘 II의 가장 큰 차이는 알고리즘 I의 제약조건들이 알고리즘 II에서는 모두 목적함수에 포함되어 있어 1단계 과정이 없어졌다는 점이다. 이러한 변환은 알고리즘을 간소화 시켜 프로그램 작성을 용이하게 할뿐만 아니라 많은 경우에 있어 제

산시간의 절감 효과를 보였다. 참고문헌 [8]에서 다음의 결과를 입증하였다.

가정 2.1: 함수 $\phi(\cdot, \cdot)$, $j \in M_0 = \{0\} \cup M$, 이 다음을 만족한다고 가정한다.

(a) 어떤 $0 < c < 1 < C < \infty$ 가 모든 $x, z \in \mathbb{R}^n$, $y_j \in Y_j$, $j \in M_0$ 에 대하여
$$c \|z\|^2 \leq \langle z, (\partial^2 \phi(x, y_j) / \partial x^2) z \rangle \leq C \|z\|^2$$
을 만족한다.

(b) Feasible region 에서 변두리 (boundary)를 뺀 집합 $\{x | \psi(x) < 0\}$ 이 공집합이 아니다. 즉, (3)의 최적해가 존재한다.

가정 2.1의 (a)는 모든 함수 $\phi(\cdot, y_j)$, $y_j \in Y_j$, $j \in M_0 = \{0\} \cup M$, 가 convex임을 가정한다. 그러나 실제 함수가 convex가 아니더라도 local minimizer 근처에서 convex 이면 그 영역 안에서는 아래의 모든 정리를 만족한다. (b)의 가정은 알고리즘 II에서 local minimizer를 찾지 못하고 알고리즘이 끝나는 것을 방지한다. 또한 (a)와 (b)에 의하여 feasible region 밖에 local minimizer 가 있을 가능성을 배제한다.

정리 2.2: 가정 2.1이 만족되고 x^* 가 (3)의 유일한 해라고 가정하자. 그러면 (a) 알고리즘 II에 의하여 얻어질 수 있는 모든 벡터 x_i 의 모임 $\{x_i\}_{i=0}^{\infty}$ 는 x^* 로 접근한다.

(b) $\{x_i\}_{i=0}^{\infty}$ 를 알고리즘 II에 의하여 얻어진 벡터의 모임이라고 가정하자. 그러면 주어진 어떠한 $\epsilon \in (0, 1)$ 에 대하여 $\rho > 0$ 이 존재하여 다음을 만족한다.

(i) 만약 모든 $i \geq 0$ 에 대하여 $\psi(x_i)$ 이고 $x_i \in B^0(x^*, \rho)$ 을 만족한다면 $\psi(x_{i+1}) \leq \delta_1(\epsilon) \psi(x_i)$ 이 성립한다.

(ii) 만약 어떤 $i_0 \geq 0$ 이 존재하여 모든 $1 \geq i_0$ 에 대하여 $\psi(x_i) \leq 0$ 이고 $x_i \in B^0(x^*, \rho)$ 이면 $\psi(x_{i+1}) - \psi(x_i) \leq \delta_2(\epsilon) [\psi(x_i) - \psi(x^*)]$ 을 만족한다.

여기에서 $B^0(x, \rho)$ 는 x 를 원점으로 반지름이 ρ 인 열린볼 (open ball)을 의미한다. 또한

$$\delta_1(\epsilon) = \max(0, 1 - \epsilon \mu^0 \alpha \beta c / C) \in (0, 1),$$

$\delta_2(\epsilon) = 1 - \epsilon \mu^0 \alpha \beta c / C \in (0, 1)$ 으로 정의한다. 최적점 $0 \in \sum_{i=0}^{\infty} \mu^i \partial \psi^i(x^*)$ 을 만족하는데 μ^i 는 이러한 조건을 만족하는 곱수벡터 (multiplier vector)의 첫 번째항 μ^i 중에서 최소치로 정의되는 $(0, 1]$ 사이의 값이다.

(c) $\{x_i\}_{i=0}^{\infty}$ 를 알고리즘 II에 의하여 얻어진 벡터들의 모임이라고 가정하자. 만약 $\gamma > C / (\mu^0 \alpha \beta c)$ 이면 $\rho > 0$ 이 존재하여 모든 $x_i \in B^0(x^*, \rho)$ 에 대하여 $\psi(x_{i+1}) \leq 0$ 을 만족한다.

정리 2.2는 (a)에서 알고리즘 II의 정당성을 입증하였고 (b)는 이의 접근정도(convergence rate)에

관한 결과를 보인 것이다. 또한, (c)는 γ 값이 어느정도로 크면 한정된 계산후에 feasible region 안으로 항상 접근함을 보인 것이다.

III. 조정변수의 영향 및 알고리즘

이 장에서는 알고리즘 II를 좀더 효율적으로 보완하기 위하여 조정변수 γ 의 영향에 대하여 알아보고 이를 근거로 새로운 알고리즘을 제시한다. 우선, 조정변수의 값이 매우 클 경우와 적을 경우의 영향을 다음과 같이 살펴본다.

경우 1: 조정변수의 값이 매우 크고 어떤 점 x_i 에서 모든 제약조건이 만족되는 경우, 즉 모든 $i = 1, \dots, m$ 에 대하여 $\psi(x_i) \leq 0$. 이 경우에는 $\psi(x_i) \leq 0$ 이므로 $\psi(x_i) = 0$ 가 된다. 그러면 복합된 1-2 단계 함수 (6)는 $F_{x_i}(x) \{ \psi(x) - \psi(x_i), \psi(x), j=1, \dots, m \}$ 이 되고 알고리즘 II의 (10), (11)에 의하여 얻어진 새로운 점 x_{i+1} 에서는 $\psi(x_i) - \psi(x_i) \leq \lambda_i \alpha \theta_i \leq 0$, $\psi(x_{i+1}) \leq \lambda_i \alpha \theta_i \leq 0$ 를 만족한다. 즉, 이 경우의 새로운 점 x_{i+1} 은 알고리즘 I의 2단계 과정을 이용하여 구한 것과 동일하게 된다.

경우 2: 조정변수의 값이 매우 크고 어떤 점 x_i 에서 제약조건 $\psi(x_i) \leq 0$ 이 만족하지 않을 경우, 즉 어떤 $j \in M$ 에 대해서 $\psi(x_i) \leq 0$. 이 경우에는 $\psi(x_i) = \psi(x_i) \geq \psi(x_i) > 0$ 이 된다. γ 가 매우 큰 양의 값이므로 $\gamma \psi(x_i)$ 는 큰 값의 양의 수이다. 또한 $F_{x_i}(x_i) = 0$ 이므로 모든 $x \in B^0(x_i, \epsilon)$ 에 대하여 $F_{x_i}(x) = \max_{j \in M} \{ \psi(x) - \psi(x_i) \}$ 인 $\epsilon > 0$ 이 존재한다. 이는 모든 $j \in M$ 에 대해서 $\psi(x)$ 이 x 에 대하여 연속이기 때문이다. 여기에서 ϵ 값은 매우 큰 값이 될 수 있으므로 (10)에 의하여 얻을 진행방향 $h(x_i)$ 의 방향은 오직 제약조건을 만족시키기 위한 방향이 된다. 다시 말하면 (10), (11)에 의하여 얻어지는 새로운 점은 $\psi(x_i)$ 가 영보다 어느정도 크다면 목적함수 $\psi(x_i)$ 와는 무관하게 되어 알고리즘 I의 1단계 과정에 해당한다.

경우 3: 조정변수의 값이 영에 가깝고 어떤점 x_i 에서 모든 제약조건이 만족되는 경우, 이 경우에는 $\psi(x_i)$ 이므로 조정변수 γ 는 아무런 영향이 없게 되어 위의 경우 1과 동일한 결과를 얻게 된다.

경우 4: 조정변수의 값이 영에 가깝고 어떤점 x_i 에서 제약조건 $\psi(x_i) \leq 0$ 이 만족되지 않을 경우, 이 경우에는 만약 어떤 $j \in M$ 에 대하여 $\psi(x_i) > 0$ 이 매우 큰 양의 수이면 위의 경우 2와 유사한 결과를 얻게 되나 만약 $\psi(x_i)$ 값이 별로 크지 않은 양의 수일 때는 위의 경우 2에서 주어진 ϵ 의 값이 매우 작을 수도 있어 (10)에서 주어진 진행방향 벡터 $h(x_i)$ 를 구할때

제약조건 뿐만 아니라 목적함수 $\psi(x)$ 도 함께 고려하여 계산된다. 즉 알고리즘 I에서는 일단 모든 제약조건을 만족하는 지역으로 점 x_i 를 끌어 들인후 이 지역을 벗어나지 않는 방향으로 목적함수를 감소시키는 반면에 알고리즘 II에서는 (10), (11)에서 얻어지는 점들이 몇몇 제약조건을 만족하지 않으면서 목적함수와 제약조건 모두를 감소시키는 방향으로 진행할 가능성도 있게 된다.

위에서 언급된 경우 1,2에서 보듯이 γ 값이 큰 경우에는 알고리즘 I과 알고리즘 II가 거의 동일한 벡터 $\{x_i\}_{i=0}^{\infty}$ 을 산출한다. 또한 경우 3,4처럼 조정변수의 값이 작고 초기 벡터 x_0 에서 $\psi(x_0) > 0$ 이면 두 알고리즘은 아주 다른 벡터 $\{x_i\}_{i=0}^{\infty}$ 를 산출하게 된다. 이와 같은 상황에 비추어 볼때 주어진 최적화 문제에 대해 두개의 알고리즘 중 어떤것이 더 효율적이라고 말할수는 없지만 알고리즘 II는 조정변수의 선택에 따라 알고리즘 I을 포함하고 있으므로 적절한 조정변수의 선택은 알고리즘 II를 매우 효율적으로 만들 수 있다. 이는 참고문헌[8]에서 주어진 많은 예제에서 입증되었으나 고정된 조정변수를 사용함으로써 그 장점을 최대화 하지 못하였다. 조정변수의 효율성을 증대하기 위하여 x_i 를 구할때 조정변수의 값을 다음과 같은 근거로 변화시켜 본다. 첫째, x_i 가 feasible region에서 멀리 있는 경우에는 조정변수 γ 를 큰 값으로 선택하여 가능한 빠르게 feasible region으로 향하게 함이 바람직하다. 이 경우에 진행방향 $h(x_i)$ 에 따라 목적함수 $f^0(\cdot)$ 의 값이 증가될수 있다는 단점이 있다. 이점에 대해서는 다음 문단에서 보완책을 제시하기로 한다. 둘째, x_i 가 feasible region에 근접하면 조정변수를 작게하여 진행방향 $h(x_i)$ 를 결정할 때 목적함수를 고려하게 함이 효율적이다. 셋째, 일단 feasible region에 들어간 후에는 조정변수의 영향은 고려할 필요가 없다. 왜냐하면 식 (6)에서 $\psi(x_i) = 0$ 이 되어 조정변수 항목이 없어지기 때문이다. 마지막으로 점 x_i 에서 (10), (11)을 이용하여 새로운 점 x_{i+1} 을 얻었을 때 $\psi(x_{i+1})/\psi(x_i)$ 또는 $\psi(x_{i+1})/\psi(x_0)$ 의 값이 작으면 (즉 $\psi(x)$ 값이 많이 감소하여 feasible region에 많은 근접을 보이면) 조정변수 γ 를 줄여 목적함수의 영향을 좀더 고려하게 함이 효율적이다.

지금까지 조정변수의 영향과 이에 따른 선택법에 대해 언급하였다. 그러나 현위치 x_i 에서 진행방향 $h(x_i)$ 로 진행할때 제약함수 $\psi(\cdot)$ 의 값은 항상 감소하나 목적함수 $f^0(\cdot)$ 값은 증가할수도, 감소할수도 있다는 점을 전혀 고려하지 않았다. 알고리즘이 효율적이라면 feasible region으로 접근하게 하는

$h(x_i)$ 의 방향이 $f^0(\cdot)$ 의 값을 증가하게 하는 방향일 때는 조정변수의 선택을 달리 해야 한다. 왜냐하면 feasible region에서 멀리 있을때 이지역으로 빨리 끌어들이기 위하여 큰값의 γ 를 택하여 새로운 점 x_{i+1} 을 $f^0(\cdot)$ 의 영향은 무시하고 구했는데 만약 $f^0(x_{i+1})$ 의 값이 $f^0(x_i)$ 보다 매우 커진다면 이는 효율적이지 않기 때문이다. 즉 벡터 $h(x_i)$ 와 $\nabla f^0(x_i)$ 이 이루는 각이 90° 이상이면 진행방향 $h(x_i)$ 는 $f^0(\cdot)$ 를 증가시키는 방향이므로 γ 를 너무 크지 않게 하여 $f^0(\cdot)$ 의 증가를 제한시키고 그각이 90° 이하이면 $h(x_i)$ 방향이 $f^0(\cdot)$ 을 감소시키는 방향이므로 γ 를 좀더 크게 하여 제약함수를 감소시키는 것을 우선으로 하게 함이 조정변수를 사용하는 목적에 부합된다. 이상과 같은 사실에 근거하여 다음과 같은 방법을 제시한다. 우선 두벡터 $h(x_i)$ 와 $\nabla f^0(x_i)$ 이 이루는 각도 θ 의 $\cos \theta$ 를 $\langle h(x_i), \nabla f^0(x_i) \rangle / \|h(x_i)\| \|\nabla f^0(x_i)\|$ 으로 구한다. 이를 이용하여 조정변수의 값을 $\gamma_i = \Gamma_i e^{\cos \theta}$ 로 정의한다. 즉 Γ_i 값에서 $\cos \theta < 0$ (이루는 각이 90° 이상) 이면 조정변수의 값을 줄여 목적함수의 값이 너무 커지지 않게 하고 만약 $\cos \theta > 0$ (이루는 각이 90° 이하) 이면 조정변수의 값을 크게하여 목적함수의 값도 감소시키면서 feasible region으로 더욱 빠르게 접근하도록 한다. 여기에서 Γ_i 는 이전의 문단에서 언급한 네가지 사실에 근거하여 $\psi(x_i)$ 의 값에 따라 선택한 조정변수의 값이고 $c > 0$ 는 사용자가 미리 설정한 값으로 γ_i 의 값이 $\Gamma_i e^c \leq \gamma_i \leq \Gamma_i e^{-c}$ 구간 내에 있도록 한다. 가변의 조정변수를 포함하는 (6) 형태의 함수를 다음과 같이 정의한다.

$$F_{\alpha, \gamma, i}(x) = \max\{\psi^0(x) - \psi^0(x_i) - \gamma_i \psi_+(x_i), \psi^i(x) - \psi_+(x^i), i \in M\} \quad (12)$$

(12)를 이용하여 보완된 알고리즘을 다음과 같이 정의한다.

알고리즘 III :

변수 : $x_0 \in \mathbb{R}^n, 0 < \Gamma_{\min} \leq \Gamma_0 \leq \Gamma_{\max} < \infty, \alpha, \beta \in (0, 1), \rho, \delta \in (0, 0.5), c > 0.$

단계 0: $i = 0$ 로 한다.

단계 1: 최적화 함수 (optimality function) $\theta_i = \psi(x_i)$ 와 진행방향 (search direction) 벡터 $h_i = h(x_i)$ 를 (10)을 이용하여 구한다.

단계 2: 벡터 h_i 와 $\nabla f^0(x_i)$ 이 이루는 각의 $\cos \theta$ 를 구하고 $\gamma_i = \Gamma_i e^{\cos \theta}$ 로 정의한다.

단계 3: 스텝크기 s_i 를 다음과 같이 구한다.

$$\lambda_i = \max\{\beta^k | F_{x_i, \gamma_i}(x_i + \beta^k h_i) \leq \beta^k \alpha \theta_i\} \quad (13)$$

단계 4: $x_{i+1} = x_i + \lambda_i h_i$ 로 정의한다.

단계 5: 만약 $\psi(x_{i+1})=0$ 또는 $\psi(x_0) \neq 0$, $\psi(x_{i+1}) / \psi(x_0) < \delta$ 이면, $\psi(x_0) \Gamma_{i+1} = \Gamma_i$, 만약 $\psi(x_{i+1}) / \psi(x_0) < \rho$ 이면, $\Gamma_{i+1} = \max\{\Gamma_{\min}, \Gamma_i - \min\{\Gamma_0, \Gamma_i^* 0.1\}\}$. 이외의 경우에는, $\Gamma_{i+1} = \min\{\Gamma_{\max}, \Gamma_i + * 0.1\}$

단계 6: $i + 1$ 을 i 라 하고 단계 1로 간다.

단계5에서 사용되는 δ 는 초기위치에서 feasible region 으로 얼마나 접근했는가를 나타내는 기준으로 이지역에 매우 접근하면 $\psi(x_i)$ 의 값이 영에 가까와 상대적으로 조정변수의 영향이 적어 이를 변화시키지 않으며 (예를들어 초기위치에서 feasible region 까지 거리가 99% 이상 줄었을때를 매우 근접했다고 할 경우 0.01을 택한다) ρ 는 각 단계에서 제약함수의 값이 어느 정도 줄면 목적함수를 벡터의 진행방향 설정시 고려하게 하는 변수이다. 이들 변수의 값은 경험이나 함수들의 특성을 고려하여 설정된다. 단계 5에 있는 Γ_i 의 증감은 임의로 설정할수 있다. 본문에서는 감소할때는 초기값과 현재값 중 작은것의 10%, 증가할때는 초기값의 10%씩 가감하고 Γ_i 가 Γ_{\min} 와 Γ_{\max} 의 구간 안에 있도록 조정하였다. 알고리즘 III의 타당성은 다음의 정리들에 의하여 입증된다.

정리 3.1: (3)의 최적화 문제가 가정 2.1을 만족하며 $\{x_i\}_{i=0}^{\infty}$ 이 알고리즘 III에 의하여 얻어진 점들이라고 가정한다. 그러면 모든 $i \geq 0$ 에 대하여 다음을 만족한다.

$$\begin{aligned} \psi^0(x_{i+1}) &\leq \psi^0(x_i) + \gamma_i \psi_+(x_i), \\ \psi_+(x_{i+1}) &\leq \psi_+(x_i) \end{aligned} \quad (14)$$

(증명) 알고리즘 III의 단계 3.4에 의하여 $F_{x_i, \gamma_i}(x_{i+1}) \leq F_{x_i, \gamma_i}(x_i)$ 을 만족한다. 그러면 (12)의 정의에서 $\psi^0(x_{i+1}) - \psi^0(x_i) - \gamma_i \psi_+(x_i) \leq 0$, $\psi_+(x_{i+1}) - \psi_+(x_i) \leq 0$, $J \in M$. 이 성립된다 (여기에서 (14)의 첫번째 관계는 증명됨). 모든 $J \in M$ 에 대하여 후자가 성립되므로 ψ 의 최대치, 즉 $\psi(x_i) = \max_{J \in M} \psi^0(x_{i+1})$ 에 대하여 $\psi(x_{i+1}) \leq \psi(x_i)$ 가 만족된다. 또한 모든 i 에 대하여 $\psi(x_{i+1}) \geq 0$ 이 성립되므로 $\max\{\psi(x_{i+1}), 0\} \leq \psi(x_i)$ 이 만족되어 (14)가 증명된다.

정리 3.2: $\{x_i\}_{i=0}^{\infty}$ 이 알고리즘 III에 의하여 얻어진 점들이라고 가정하자. x^* 가 이 점들의 accumulation point 일때 $\theta(x^*)=0$ 을 만족한다. 즉 $x_i \rightarrow x^*$ 일때 $\theta(x^*)=0$ 이다.

(증명) 주어진 정리를 두가지 경우로 나누어 증명

한다.

경우 1: 어떤 i_0 에서 $\psi(x_{i_0}) \leq 0$ 이 만족될때. 그러면 $\psi(x_{i_0})=0$ 이므로 정리 3.1의 (14)에 의하여 모든 $i \geq i_0$ 에 대하여 $\psi(x_i)=0$ 이 성립된다. 즉 i_0 번째 이후에는 모든 제약조건이 만족된다. 그러면 알고리즘 III의 단계5에 의하여 모든 $i \geq i_0$ 에 대하여 $\gamma_i = \gamma_{i_0}$ 이 성립된다. 여기에서 조정변수 값은 $\{x_i\}_{i=0}^{\infty}$ 로 한정된 갯수이므로 이들의 최소값과 최대값이 영보다 크고 무한대보다 작은 범위에서 존재하게 된다. 따라서 $\gamma_{\min} = \min_{i=0, \dots, i_0} \gamma_i$, $\gamma_{\max} = \max_{i=0, \dots, i_0} \gamma_i$ 로 정의하면 모든 $i \geq 0$ 에 대해서 $0 < \gamma_{\min} \leq \gamma_i \leq \gamma_{\max} < \infty$ 이 만족된다. 그러면 참고문헌 [14]의 정리3.1에 의하여 $x_i \rightarrow x^*$ 일때 $\theta(x^*)$ 이 증명된다.

경우 2: 모든 $i \geq 0$ 에 대하여 $\psi(x_i) > 0$ 일때. 우선 $x_i \rightarrow x^*$ 일때 $\psi^*(x_i)=0$ 임을 증명하자. 모순을 유도하기 위하여 $x_i \rightarrow x^*$ 일때 어떤 $\epsilon > 0$ 이 있어 $\psi(x_i) - \psi(x^*) > \epsilon$ 을 만족한다고 가정하자. 그러면 모든 $i \geq 0$ 에 대하여 $\gamma_i \geq \Gamma_{\min} e^{-\epsilon}$ 이므로 모든 $x \in R^n$ 에 대하여 다음을 만족한다.

$$\begin{aligned} F_{x_i, \gamma_i}(x) &= \max\{\psi^0(x) - \psi^0(x^*) - \gamma_i \psi_+(x^*), \\ &\psi^j(x) - \psi_+(x^*), j \in M\} \\ &\leq \max\{\psi^0(x) - \psi^0(x^*) - \Gamma_{\min} e^{-\epsilon} \epsilon, \\ &\psi^j(x) - \psi_+(x^*), j \in M\}. \end{aligned}$$

그런데 알고리즘 III에 의하여 얻어진 $\{x_i\}_{i=0}^{\infty}$ 이 $x_i \rightarrow x^*$ 을 만족한다는 것은 $\min_x F_{x_i, \gamma_i}(x) = 0$ 를 의미한다. 왜냐하면 $\min_x F_{x_i, \gamma_i}(x) \leq F_{x_i, \gamma_i}(x) = 0$ 인데 만약 어떤 $\bar{x} \in R^n$ 에 대하여 $\min_x F_{x_i, \gamma_i}(x) = F_{x_i, \gamma_i}(\bar{x}) < 0$ 이면 x^* 에서 알고리즘 III의 단계3에 의하여 새로운 점 \bar{x} 를 산출하므로 $x_i \rightarrow x^*$ 일수 없기 때문이다. 따라서 $\min_x F_{x_i, \gamma_i}(x) = 0$ 인데 이는 x^* 근처의 모든 벡터 x 에서 $\psi(x) - \psi(x^*) \geq 0$ 을 의미한다. 그런데 이는 가정 2.1의 (b)에서 $\{x \in R^n | \psi(x) < 0\}$ 이 공집합이 아니라는 사실에 위배된다. 따라서 $x_i \rightarrow x^*$ 일때 $\psi(x^*)=0$ 을 만족해야 한다.

위의 결과는 어떤 $i_0 \geq 0$ 이 존재하여 모든 $i \geq i_0$ 에 대하여 단계5의 $\psi(x_{i+1}) / \psi(x_0) < \delta$ 을 만족함을 의미한다. 그러면 조정변수 γ_i 는 모든 $i \geq i_0$ 에 대하여 상수인 γ_{i_0} 값을 갖게 된다. 이를 이용하여 경우 2를 다음과 같이 증명한다. 역시 모순을 유도하기 위해서 $x_i \rightarrow x^*$ 일때 $\theta(x^*) \neq 0$ 라고 가정하자. $\theta(x)$ 는 음의 수이므로 ([5]의 Proposition 5.4와 5.5 참조) $\theta(x^*) = -\delta < 0$ 로 나타낼수 있다. 그러면 참고문헌[14]의 Proposition 2.3과 $\theta(\cdot)$ 의 연속성에 의하여 어떤 $\epsilon > 0$, $i_1 \geq i_0$ 가 존재하여 모든 $i \geq i_1$ 에 대하여 $F_{x_i, \gamma_i}(x_{i+1}) < -\epsilon \delta / 2$ 이 만족하게 된다. 다시말하면 모든

$i \geq i_1$ 에 대하여 $\psi(x_{i+1}) - \psi(x_i) = \psi(x_{i+1}) - \psi(x_i) \leq -\epsilon\delta/2$ 이 성립된다. 즉 $\psi(x_i)$ 는 최소한 $\epsilon\delta/2$ 의 값만큼 계속적으로 감소 (monotone decreasing sequence)하게 된다. 그러면 $i \rightarrow \infty$ 일때 $\psi(x_i) \rightarrow -\infty$ 를 의미하는데 이는 $\psi(x_i)$ 가 항상 양의 수임에 모순된다. 따라서 $x_i \rightarrow x^*$ 일때 $\theta(x^*)=0$ 가 되어야한다.

IV. 시뮬레이션

여기에서는 알고리즘 II와 알고리즘 III를 비교 분석한다. 두알고리즘은 C 언어로 작성되었으며 SUN SPARC II Workstation 에서 수행되었다.

예제 1: (Rozen-Suzuki 문제^[10])

$$\begin{aligned} \min \quad & x_1^2 + x_2^2 + 2x_3^2 + x_4^2 - 5x_1 - 5x_2 - 21x_3 - 7x_4, \\ \text{with} \quad & 2x_1^2 + x_2^2 + x_3^2 + 2x_4 - x_3 - x_4 - 5 \leq 0, \\ & x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_1 - x_2 + x_3 - x_4 - 8 \leq 0, \\ & x_1^2 + 2x_2^2 + x_3^2 + 2x_4^2 - x_1 - x_4 - 10 \leq 0. \end{aligned}$$

이문제에서 모든함수는 convex이고 최소값은 (0.1, 2, -1) 에서 -44이다. Feasible 초기점은 (0,0,0,0), infeasible 초기점은 (2,4,8,1)을 택했다. 알고리즘 III에서 쓰이는 변수는 $\Gamma_0 = 2.0$, $\Gamma_{\min} = 0.3$, $\Gamma_{\max} = 4.0$, $c = 1.0$ 이다. Feasible 초기점에서는 $\psi(x_0) = 0$ 이므로 조정변수의 영향이 없게되어 알고리즘 II와 알고리즘 III는 동일하게 된다. 이경우의 목적함수와 진행횟수의 관계는 그림 1에서 보여진다. 그림 2,3은 infeasible 초기점을 이용한 목적함수, 제약함수와 진행횟수의 관계를 보인 것으로 알고리즘 III가 알고리즘 II보다 훨씬 빨리 최적점에 도달함을 보인다. 계산 시간은 알고리즘 II가 0.16초, 알고리즘 III가 0.11초로 약 40%의 감소가 있었다.

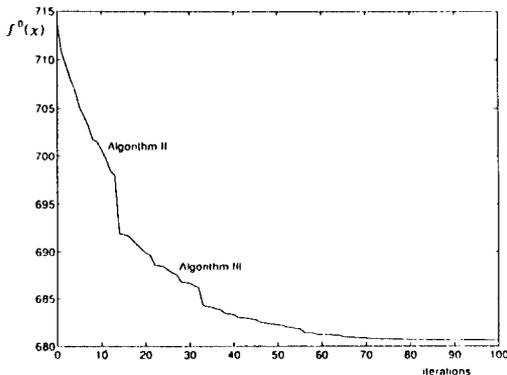


그림 1. 목적함수와 진행횟수의 관계: Feasible 초기점 사용시

Fig. 1. Cost vs. iteration plot with a feasible initial point.

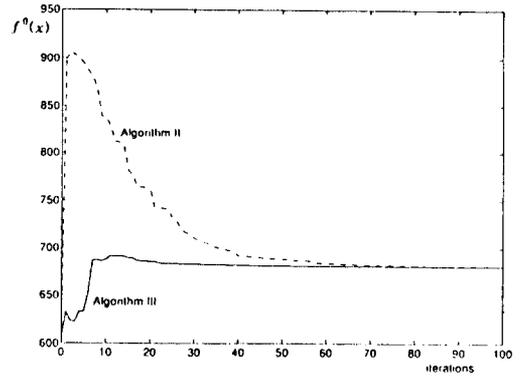


그림 2. 목적함수와 진행횟수의 관계: Infeasible 초기점 사용시

Fig. 2. Cost vs. iteration plot with an infeasible initial point.

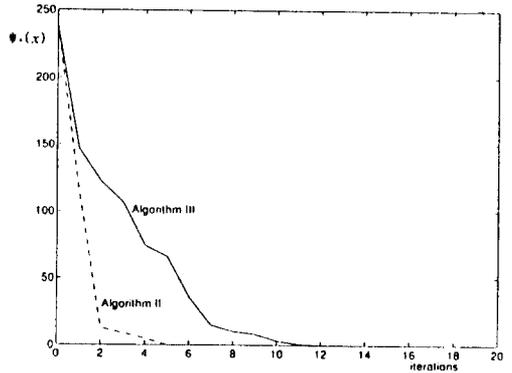


그림 3. 제약함수와 진행횟수의 관계: Infeasible 초기점 사용시

Fig. 3. Constraints vs. iteration plot with an infeasible initial point.

예제 2: (Wong 문제^[9])

$$\begin{aligned} \min \quad & (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 + 3(x_4 - 11)^2 \\ & 10x_5^6 + 7x_6^2 + x_7^4 - 4x_6x_7 - 10x_6 - 8x_7, \\ \text{with} \quad & 2x_1^2 + 3x_2^4 + x_3 + 4x_4^2 + 5x_5 - 127 \leq 0, \\ & 7x_1 + 3x_2 + 10x_3^2 + x_4 - x_5 - 282 \leq 0, \\ & 23x_1 + x_2^2 + 6x_6^2 - 8x_7 - 196 \leq 0, \\ & 4x_1^2 + x_2^2 - 3x_1x_2 + 2x_3^2 + 5x_6 - 11x_7 \leq 0. \end{aligned}$$

이문제에서는 모든 제약함수는 convex이나 목적함수는 nonconvex이다. 최소점은 (2.33, 1.95, -0.48, 4.37, -0.62, 1.04, 1.59)에서 680.63이다. 알고리즘 II와 알고리즘 III의 초기점으로 infeasible 점 (3.3, 0.5, 1.3, 0)을 사용하였다. 알고리즘 III의 변수로는 $\Gamma_0 = 2.0$, $\Gamma_{\min} = 0.3$, $\Gamma_{\max} = 4.0$, $c = 2.0$

을 선택하였다. 그림 4.5는 각각 목적함수, 제약함수와 진행횟수의 관계를 보인 것인데 알고리즘 III이 훨씬 효율적임을 나타낸다. 계산시간은 알고리즘 II가 0.47초, 알고리즘 III이 0.39초로 약 15%의 감소가 있었다.

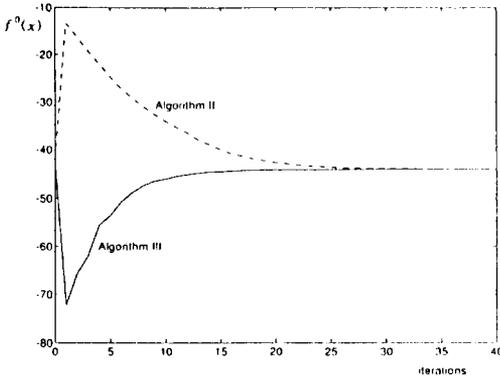


그림 4. 목적함수와 진행횟수의 관계: Infeasible 초기점 사용시

Fig. 4. Cost vs. iteration plot with an infeasible initial point.

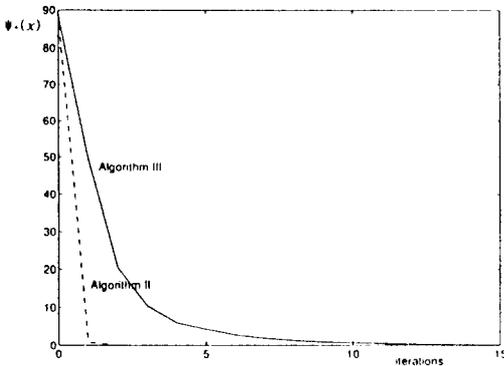


그림 5. 제약함수와 진행횟수의 관계: Infeasible 초기점 사용시

Fig. 5. Constraints vs. iteration plot with an infeasible initial point.

V. 결론

본 논문에서는 고정된 조정변수를 갖는 복잡한 1-2 단계 최적화 알고리즘에서 조정변수의 값에 따른 영향을 고려하였고 이에 근거하여 새로운 알고리즘을 제시하였다. Feasible region 에서 멀리 떨어져 있으면 조정 변수의 값을 크게하여 가능한 빨리 가능지

역으로 접근하게하고 어느 정도 근접하였을 경우에는 조정변수의 값을 감소시켜 feasible region 으로 접근함과 동시에 최적화 하고자 하는 목적함수를 원하는 방향으로 이끄는 것이 효율적이다. 또한 진행방향 $h(x_i)$ 에 따른 목적함수의 증감을 고려하여 조정변수를 선택함이 효율적이다. 이러한 분석에 따라 제시한 알고리즘은 모든 문제에 있어서 기존의 알고리즘 보다 우수하다고 할수는 없지만 (최악의 경우 (worst case)를 고려해야 하기 때문) 여러 예제의 시뮬레이션 결과를 통하여 기존의 알고리즘보다 효율적임을 보였다. 또한 사용자가 주어진 문제에 대해 많은 정보가 있는 경우에 적절한 변수 (ρ, γ, c 등의 변수)를 선택하면 계산 시간을 더욱 줄일수 있다.

參考文獻

- [1] Bhatti, M.A., Polak, E. and Pister, K.S., "Optimization of Control Devices in Base Isolation Systems for Seismic Design", Proc. Internat. IUTAM Symposium on Structural Control, University of Waterloo, Ontario, Canada, 1980.
- [2] He, L., and Polak, E., "Effective Diagonalization Strategy for the Solution of a Class of Optimal Design Problems", IEEE Trans. on Automatic Control, Vol.AC-35, No.3, pp.258-267, 1990.
- [3] Mayne, D.Q., Polak, E., and Sangiovanni-Vincentelli, A., "Computer Aided Design via Optimization: A Review", Automatica, 18 pp.147-154, 1982.
- [4] Mayne, D.Q., Polak, E., and Trahan, R., "An outer approximation algorithms for computeraided design problems", Journal of Applied Math. and Optimization, 28 pp.331-352, 1979.
- [5] Polak, E., "On the Mathematical Foundations of Nondifferentiable Optimization in Engineering Design", SIAM Review, Vol.29, No.1, pp.21-91, 1987.
- [6] Polak, E., "Computational Methods

- in Optimization: A Unified Approach". Academic Press, New York, NY, 1971.
- [7] Polak, E., and He, L., "Rate-Preserving Discretization Strategies for Semi-Infinite Programming and Optimal Control". UCB/ERL Memo M89/112, Electric Research Laboratory, University of California, Berkeley, 1989.
- [8] Polak, E., and He, L., "Unified Phase I-Phase II Method of Feasible Directions for Semi-Infinite Optimization", Vol.69, No.1, pp.83-107, April 1991.
- [9] Assadi, J.A., "Computational Comparison of Some Nonlinear Programs", Mathematical Programming, Vol.4, pp.144-154, 1973.
- [10] Conn, A.R., "Constrained Optimization Using a Nondifferentiable Penalty Function", *SIAM Journal on Numerical Analysis*, Vol.10, No.4, pp.760-784, 1973.
- [11] Klessig, R., and Polak, E., "An Adaptive Precision Gradient Methods for Optimal Control", *SIAM Journal on Control*, Vol.10, pp.80-93, 1973.
- [12] Pironneau, O., and Polak, E., "On the Rate of Convergence of Certain Methods of Centers", *Mathematical Programming*, Vol.2, pp.230-258, 1972.
- [13] Topkis, D.M. and Veinott, A F.Jr., "On the Convergence of Some Feasible Direction Algorithms for Nonlinear Programming", *SIAM Journal on Control*, Vol.5, pp.268-279, 1967.
- [14] Zoutendijk, G., "Methods of Feasible Directions", Elsevier, Amsterdam, Holland, 1960.
- [15] Zuhovitskii, S.I., Polyak, R.A., and Primak, M.E., "An Algorithm for the Solution of Convex Problems", *Doklady Akademii Nauk USSR*, Vol.153, pp.991-1000, 1963.
- [16] Clarke, F.H., "Optimization and Nonsmooth Analysis", Wiley-Interscience, New York, NY, 1983.
- [17] Higgins, J.E., and Polak, E., "Minimizing Pseudo-Convex Functions on Convex Compact Sets", *Journal of Optimization Theory and Applications*, Vol.65, No.1, pp.1-28, 1990.
- [18] Polak, E., Trahan, R., and Mayne, D.Q., "Combined Phase I-Phase II Methods of Feasible Directions", *Mathematical Programming*, Vol.17, pp.61-73, 1979.
- [19] Tanaka, Y., Fukushima, M., and Ibaraki, T.A., "A Comparative Study of Several Semi-Infinite Nonlinear Programming Algorithms", *European Journal of Operational Research*, Vol.36, pp.92-100, 1988.

 著者紹介



梁玄錫(正會員)

1961年 4月 7日生. 1984年 5月 Purdue University 전기전자공학과 졸업(공학석사).
 1991年 12月 University of California, Berkeley 전기전자공학과 졸업(공학박사).
 1992年 3月 ~ 현재 홍익대학교 전자공학과 조교수. 주관심 분야는 최적제어, CAD-Control System, 비선형 제어 이론 및 알고리즘 등임.