비미분가능 최적화문제의 효율적 수치해에 대한 연구
-A study on the effective numercial method for nondifferentiable optimization problem-

金 埈 弘*
Kim, Jun Hong

Abstract

This study presents a method of realizing the theoretical results of Demyanov in practice on a computer in order to produce a kind of constructive evidence for his theory and a practical method of getting numerical results for quasi-differentiable optimization problems which may arise in industry and science.
A practical result for a restricted nondifferentiable optimization problem is experimented with a simle example.

1. 서론

x-y평면에서 관측된 자료의 점집합 $(x_0,y_0),(x_1,y_1),\cdots,(x_N,y_N)$이 주어졌다 하고, 그 자료에 대한 회귀직선을 구하는 문제를 생각한다. 그 회귀직선 $y=mx+b$에 대해, i번째 자료에 대한 오차 e_i는 $|mx_i+b-y_i|$로 정의된다. 최상의 회귀직선은 경사 m과 절편 b가 모든 m과 b에대해 $(\sum_{i=0}^{N} e_i^2)^{\frac{1}{2}}$를 최소화하는 것이 일반적이다. 그러나, 총오차 $\sum_{i=0}^{N} e_i$를 최소화하는 것을 생각해 볼 수 있다. 전자는 순수측정오차에 해당하는 데이터의 편차를 무시한 경우이고, 후자는 그 점에 대한 영향을 감안한 경우가 된다. 또 다른 예로서, 각 로트가 m_i개의 제품으로 되어있는 n개의 로트를 한대의 크레인이 일정지역에 분배하는 공장을 생각하자. 각 로트 i는 지역 x_i로 운반되는데, 크레인은 제품을 x-y평면상으로만 수송할 수 있으며, 그것은 한번에 한 로트를 분배할 수 있다 하자. 분배거리를 최소화하는 문제는 다음과 같이 제시될 수 있다. Min f: $\mathbb{R}^2 \to \mathbb{R}$, 여기서, $f(x) = \sum_{i=1}^{n} m_i|x_i-x|$, $m_i \in \mathbb{N}$, $x,x_i \in \mathbb{R}^2$. 상기 두 함수는 미분가능하지 않고, 이 최적화문제를 비비문가능 최적화문제라 한다. 일상에서 자주 나타나는 모델이다. 이 문제들은 근사해법으로 그 최적해를 구하지만, exact solution과의 차가 크다. V.F.Demyanov[4][5]는 quasi-differentiable함수의 이론을 사용하여 그러한 유형에 속하는 문제들에 대한 최적해를 구하기 위한 수리적 기초를 제시하였고[5][6][7], 특히, Clarke's theory은 이차도함수에 대한 관점으로 미비분가능함수의 개념으로 확장성을 제시해 주고 있다[2]. quasi-differential이론은 유한개수의 함수에 대해 합, 곱, 몫, 최대치 또는 최소치 등과 같은 연산에 대해 중요한 공식들을 제시하는 반면, 다른 연산들과의 일반적인 포괄성을 보증하여 주고 있다. 그러한 연산공식들은 최적화문제에서

* 수원대학교 산업공학과

steepest descent방향 계산방법에 대해 여러가지 정리들을 제시하고 있고, Pallaschke and Urbanski 등[8][9]은 그에 대한 응용분야로의 적용을 위해 괄목할만한 연구결과를 보여주고 있다. quasi-differential최적화문제의 어려움은 그 앨고리즘적인 계산방법에 있다고 할 수 있다[2][4]. 이 연구에서는 제약이 있는 비미문가능최적화문제에 대한 개선된 앨고리즘과 그 앨고리즘에 대한 수치해석에 따른 효율적인 계산방법을 제시하고, 제약이 있는 비미분최적화문제에 대한 프로그램을 통해 그 적용성을 살펴본다.

2. Quasi-differential함수의 성질

미분가능함수에 대한 계산은 quasi-differentiable calculus에 포함되므로, 이 연구에 필요한 quasi-differentiable calculus를 이용하여 이들 함수의 기본적 정의와 정리에 대해 간단히 언급하겠다. quasi-differentiable functions의 미분은 [5]에서 잘 설명되고 있다.

정의2.1 집합 $S \in R^n$에서 정의된 함수 f가 다음조건을 만족한다면, 주어진 점 $x_0 \in S$에서 quasi-differentiable이라 한다: x_0에서 임의 방향 $g \in R^n$으로 directionally differentiable이고, 방향도함수가 $\frac{\partial}{\partial g} f(x_0) = \lim_{\alpha \to +0} \frac{1}{\alpha} f(x_0 + \alpha g) - f(x_0)$

$$= \max_{w \in \bar{\partial} f(x_0)} <v,g> + \min_{v \in \underline{\partial} f(x_0)} <w,g> \quad \forall g \in R^n, \|g\| = 1$$

으로 주어진 두 개의 convex, compacts set $\underline{\partial}f(x_0), \bar{\partial}f(x_0) \in R^n$이 존재한다면, 집합 $Df(x_0) = [\underline{\partial}f(x_0), \bar{\partial}f(x_0)]$는 점 x_0에서 f의 quasi-differential이라 하고, 집합 $\underline{\partial}f(x_0), \bar{\partial}f(x_0)$를 각각 sub-differential, super-differential이라 한다.

집합 A가 convex 및 compact set이라면, $[\underline{\partial}f(x_0)+A, \bar{\partial}f(x_0)-A]$는 x_0에서 함수 f에 대해 역시 quasi-differential이므로, quasi-differential은 유일한 값을 갖지 않는다. 벡터 v_0-w_0는 일반적인 급하강방향과 유사하다. 그러나, 연속인 미분가능함수에 비해, 유사미분가능함수의 급하강방향은 마찬가지로 유일한 값을 갖지 않는다.

Lemma 2.2 $f: R^n \to R$ 는 점 $x_0 \in S$ 에서 quasi-differentiable함수라 하자. 그러면, x_0에서 함수 f의 급하강방향 g^*은 다음과 같다:

$$g^*(x_0) = -\frac{v_0+w_0}{\| v_0+w_0 \|}, \quad \text{where} \quad \| v_0+w_0 \| = \max_{w \in \bar{\partial} f(x_0)} \{ \min_{v \in \underline{\partial} f(x_0)} \| v+w \| \}$$

<u>증명</u>: [5]참조 □

Lemma 2.3 함수 f는 $x_0 \in S \subseteq R^n$ 에서 quasi-differentiable이라 하자.

$$-\bar{\partial}f(x_0) \subset \underline{\partial}f(x_0)$$

은 x_0가 f의 local minimum이 되기위한 필요조건이 된다.

상기조건을 만족하는 점 $x_0 \in S$를 inf-stationary라 한다.

uniformly quasi-differentiable function f에 대한 strictly local minimum의 명제를 위한 충분조건은 다음과 같다:

Lemma 2.4 함수 f는 $x_0 \in S \subseteq \mathbb{R}^n$ 에서 uniformly quasi-differentiable이라 하자.

$$-\overline{\partial}f(x_0) \subset \mathrm{int}(\underline{\partial}f(x_0))$$

은 x_0가 f의 local minimum이 되기 위한 충분조건이다.

이들 가설과 정리에 따라, 미분가능, convex 또는 maximum 함수에 의해 해결되지 않는 최적화문제에 대한 방법을 다음장에 제시한다.

3. Quasi-differentiable function의 최적화를 위한 앨고리즘적 접근

3.1 두 convex set간의 최소길이의 결정

다음과 같은 제약이 있는 최적화문제를 생각하자.

Min f(x),

s.t. $h_i(x) \leq 0$ for i=1,···,m.

여기서, 목적함수 f, 제약식 $h_i(x)$는 $\mathbb{R}^n$에서 정의된 실수함수.

수리계획에의한 미분가능함수 최적화기법으로 가장 일반적인 것은 gradient법이라 할 수 있다. gradient법은 가능해영역내의 수열 $(x_k) \in S$이 구하고자 하는 최적해로 수렴해가는 비증가하는 값 $f(x_k)_{k \in N}$을 갖는다. 보통, 변곡점 x_0는 S에서 선택되고, k번째 반복단계의 점 x_k는 $x_k = x_{k-1} + \alpha_k \cdot g_{k-1}$, $0 < \alpha_k \in \mathbb{R}$, 에 의해 계산된다. 벡터 $g_k \in \mathbb{R}^n$는 x_k의 작은 neighbourhood내에서 이 목적함수치가 감소하는 방향이다. 이것에 대해 두개의 부문제, f, h_i가 smooth인 경우, 방향 g_k와 stepsize α_k를 구하는데는 어려움이 없고, 이 방면에 대해 상당히 효율적인 앨고리즘들이 제시되고 있다[1]. 우리는 여기서 그 가능해집합 $S \subset \mathbb{R}^n$나 목적함수가 연속이고 미분이 가능하지않은 함수인 경우를 생각한다. 이경우 함수 f와 $h_i(x)$는 quasi-differentiable로 restricted quasi-differentiable최적화문제라 한다. 제약이 있는 quasi-differentiable최적화문제는 그 가능해집합 $S=\{x \in \mathbb{R}^n \mid h_i(x) \leq 0,\ i=1,\cdots,m\}$을 $S=\{x \in \mathbb{R}^n \mid \max\{h_i(x)\} \leq 0,\ i=1,\cdots,m\}$로 다시 고쳐 쓸 수 있어 생각보다 덜 복잡하다. 제약이 있는 quasi-differentiable최적화문제는 다음과 같이 재공식화될 수 있다:

Min f(x), $x \in S \subset \mathbb{R}^n$

여기서, $S=\{x \in \mathbb{R}^n \mid \max\{h_i(x)\} \leq 0,\ i=1,\cdots,m\}$.

이것은 많은 제약을 갖는 문제를 하나의 제약을 갖는 문제로 고칠 수 있다는 것을 의미한다. quasi-differentiable함수에 대한 도구를 이용하여, 가능해영역을 convex로 표시할 수 있게 된다.

정리 3.1 함수 f, H, $h_i(x)$, i=1,…,n 는 $\mathbb{R}^n$과 $S=\{x \in \mathbb{R}^n \mid \max\{h_i(x)\} \leq 0,\ i=1,\cdots,m\} = H(x) \leq 0$에서 quasi-differentiable이라 하자. 또, $x_0 \in S$, $H(x_0)=0$라 하자. 함수 $\Psi(x) = \max\{f(x)-f(x_0), H(x)\}$의 quasi-differential을 계산한다. 여기서, $\underline{\partial}\Psi(x_0) = \mathrm{co}\{\underline{\partial}f(x_0) - \overline{\partial}H(x_0), \underline{\partial}H(x_0) - \overline{\partial}f(x_0)\}$, $\overline{\partial}\Psi(x_0) = \overline{\partial}f(x_0) + \overline{\partial}H(x_0)$. 이것은 가능해집합 S에서 f의 최소점이 되기 위한 필요조건은 다음을 만족한다:

$$-\overline{\partial}\Psi(x_0) \subset \underline{\partial}\Psi(x_0)$$

증명:[2]참조. □

만약 이 조건이 만족되지 않는다면, Ψ의 급하강방향을 방향 $\overline{g}$로 취하여 f의 급하강방

향을 얻을 수 있다. 이러한 종류의 문제를 풀기 위해 의사미분가능함수의 이론을 바탕으로, 두가지 관점에서 생각한다. 하나는 주어진 점 $x \in R^n$이 최적인가를 결정하는 기준과 또 하나는 최적에 도달되지 않았다면, 현재의 위치에서 비교하여 감소하는 방향을 선택하는 어떤방법을 취해야 하는 것이다. 이것에 대해 다음 두가지 문제를 풀어야 한다: (1)그러한 방향을 어떻게 얻을 것인가? (2)다른 위치가 최적인가를 검토하기 전 그 방향으로 얼마나 진행할 것인가? 이러한 문제는 전형적인 non-smooth문제가 아니라, 그것은 보통 1-dim. nonsmooth최적화문제가 되므로, 실제 적용에 있어서, classical line search법을 수정해 사용할 수 있다. 그래서, quasi-differentiable이론에서 상기 문제는 constructive search를 적용하여 사용한다. 제약이없는 의사미분가능 최적화문제에 이들 일반화된 gradient를 사용하기위해 급하강방향은 다음과 같은 식을 이용한다:우선 두집합 $\underline{\partial}f(x_0)$, $\overline{\partial}f(x_0)$을 구한 후, 급하강방향

$$g(x_0) = \frac{v_0+w_0}{\| v_0+v_0 \|}, \quad v_0,w_0 \in R^n, \quad v_0+w_0 \neq 0$$

$$\text{여기서,} \quad \| v_0+w_0 \| = \max_{w \in \overline{\partial}f(x_0)} \{ \min_{v \in \underline{\partial}f(x_0)} \| v+w \| \}.$$

을 구한다.

$$\text{이것은} \quad g(x_0) = -\frac{v_0-w_0}{\| v_0-w_0 \|}, \quad v_0,w_0 \in R^n, \quad v_0-w_0 \neq 0.$$

$$\text{여기서,} \quad \| v_0-w_0 \| = \max_{w \in \overline{\partial}f(x_0)} \{ \min_{v \in \underline{\partial}f(x_0)} \| v-w \| \} \qquad (*)$$

으로 다시 쓸 수 있다. 우리의 목표는 제약이 있는 비미분가능문제를 푸는 것이므로 이에 대한 문제를 풀기 위해 다음과 같은 정리를 통해 접근할 수 있다:

정리3.2

방정식(*)를 만족하는 점 $w_0 \in -\overline{\partial}f(x_0)$과 $v_0(w_0) \in \underline{\partial}f(x_0)$ 이 존재한다.

<u>증명</u>:

w_0는 방정식(*)를 만족하는 $-\overline{\partial}f(x_0)$에 속하는 임의 점이라 하자. $\underline{\partial}f(x_0)$로 부터의 길이가 $\| v_0-w_0 \|$을 초과하지 않는 점 x의 집합 $P \subset R^n$을 취한다. 즉,

$P := \{x \in R^n \mid \| x-y \| \leq \| v_0-w_0 \|, \ y \in R^n\}$

$= -\underline{\partial}f(x_0) + \| v_0-w_0 \| \cdot B$, 여기서, B는 n-dim. Euclidean ball이다.

H는 w_0을 포함하는 P의 supporting hyperplane이라 하자. $-\overline{\partial}f(x_0)$에 속하는 임의 한 외부점 w_i는 $H \cap -\overline{\partial}f(x_0) \subseteq H \cap P$ 의 원소임을 보일 수 있다. w_i는 extremal points이므로, 점 $x \in -\overline{\partial}f(x_0)$은 $x = \sum_{i=1}^{k} \lambda_i \cdot w_i$, $\sum_{i=1}^{k} \lambda_i = 1$, $\lambda_i \geq 0$ 로 표현될 수 있다.

$n \in R^n$를 H에 법선이고, $-\overline{\partial}f(x_0)$에 속하는 한 벡터라 하자. 그러면, 임의점 $x \in H$에 대해, $\langle n, w_0-x \rangle = 0$ 가 성립.

$w_i \notin H \cap -\overline{\partial}f(x_0) \Leftrightarrow w_i \in \{x \in R^n \mid \langle n, w_0-x \rangle > 0\}$ 라 가정한다.

그러나, $w_0 \in -\overline{\partial}f(x_0)$이므로, $w_0 = \sum_{i=1}^{k} \lambda_i \cdot w$, $\sum_{i=1}^{k} \lambda_i = 1$, $\lambda_i \geq 0 \Leftrightarrow w_0 - \sum_{i=1}^{k} \lambda_i \cdot w_i = 0$ 이 된다.

그러나, $0 = \langle n,0 \rangle = \langle n, w_0 - \sum_{i=1}^{k} \lambda_i \cdot w_i \rangle = \sum_{i=1}^{k} \lambda_i \cdot \langle N, w_0-w_i \rangle > 0$ 은 모순이다.

따라서, $w_i \in H \cap -\bar{\partial}f(x_0)$. □

집합 $-\bar{\partial}f(x_0)$은 유한개수의 점 $w:=\{w_1,\cdots,w_m\}$으로 주어지므로, $w_i \in co\{ext(-\bar{\partial}f(x_0))\}$이다. 여기서, $ext(-\bar{\partial}f(x_0))$는 $-\bar{\partial}f(x_0)$의 모든 외부의 점집합을 나타낸다. 방정식(*)은 다음과 같이 고쳐 쓸 수 있다.

$$g(x_0) = -\frac{v_0 - w_0}{\| v_0 - w_0 \|}, v_0, w_0 \in \mathbb{R}^n, \ v_0 - w_0 \neq 0,$$

여기서, $\| v_0 - w_0 \| = \max_{w \in W} \{ \min_{v \in \underline{\partial} f(x_0)} \| v - w \| \}$.

정리3.2의 증명을 기초로, 문제는 점 $w \in \mathbb{R}^n$과 다면체 $v \in V:=co\{v_1,\cdots,v_j\}$간의 최소길이를 구하는 것으로 축소된다. 다면체 V는 affine halfspace들의 교집합을 이루는 부등식방정식에 의해서만 아니라, 외부점들이 아닌 유한개의 점들에 의해도 만들어지므로 문제는 좀더 어렵다. 따라서, 함수 h를 최소화하기 위해 다음과 같이 진행하는 앨고리즘을 제시한다:

$h(v)=(\| w-v \|)^2$, $w \in \mathbb{R}^n$, $V \neq 0$, 이고,

v에서 함수 h의 gradient를 $\nabla h'(v)=\left(\frac{dh(v)}{dv_1}, \cdots, \frac{dh(v)}{dv_n} \right)$

라 하자.

<u>(초기화)</u> $k:=0$, $V_0:=\{v_1,\cdots,v_j\}$, $v_k \in V_0$ 는 임의 값.

<u>반복단계1</u>. $\{w\}$와 $\bar{V} \subset v_k$이 초평면 $H_k:=\{x \in \mathbb{R}^n | <\nabla h(v_k), x-x_k>=0 \}$에 의해 결정된 해당하는 affine halfspace의 부분집합이 되는 모든 점들의 집합 $\bar{V}$을 구한다.

즉, $\bar{V} =:\{x \in V_k, x \neq v_k | \pm(<\nabla h(v_k), w-v_k>) = \pm(<\nabla h(v_k), x-v_k>)\}$

$\bar{V}=\phi$ 라면, v_k는 함수 h를 최소화한다.

<u>반복단계2</u>. 함수 $h_k: \bar{V} \to \mathbb{R}$를 최소화한다, 여기서 $v \to \| x-v \|^2$, $x \in H_k$.

점 $x \in \mathbb{R}^n$과 초평면 $H=\{y \in \mathbb{R}^n | f(y)=\alpha\}$과의 거리는

$$dist(x,H) = \frac{|f(x)-\alpha|}{\| f \|} \text{ 이므로,}$$

이것은 $\frac{|<\nabla h(v_k),v> - <\nabla h(v_k),v_k>|}{\| \nabla h(v_k) \|}$ 을 최소화하여 계산된다.

$\nabla h(v_k)=0$ 는 $v_k=w$ 가 h의 최소점이라는 것을 의미한다.

따라서, $\nabla h(v_k) \neq 0$ 라 하자. $\| \nabla h(v_k) \| =const.$ $\forall v \in \bar{V}$ 이므로

$$v^* = \arg\min_{v \in \bar{V}} \frac{|<\nabla h(v_k),x-v_k>|}{\| v-v_k \|}$$ 를 계산할 수 있다.

<u>반복단계3</u>. 함수 $t:[0,1] \to \mathbb{R}$를 최소화한다. 여기서, $t \to \| w-t \cdot v_k-(1-t) \cdot v^* \|$ 이다.

해석적방법을 사용하여,

$$t^* = \min \{ 1, \max (0, \frac{< w-v_k, v_k-v^* >}{\| v^*-v_k \|}) \}$$

에 대한 최소치를 얻는다.

반복단계4. $v_{k+1}:=t^* \cdot v_k+(1-t)\cdot v^*$,
$v_{k+1}:=v_o \cup \{v_{k+1}\}$, $k=k+1$,
다음 반복단계를 수행한다.

종료규칙: $w \subseteq V \Leftrightarrow \| v_{k+1}-v_k \| < \varepsilon$ 인 경우 종료한다.

이 앨고리즘으로부터 h의 최소값을 구한다. 그런데, gradient법을 이용한 의사미분계산의 수치적 접근방법을 위해 다음절에서 그 효율적인 방법에 대해 살펴본다.

3.2 급하강 방향의 계산

식(*)의 정의에서 언급한바와 같이, 두집합은 함수의 미분가능 부분을 취하고, quasi-differential함수의 계산은 이들 규칙을 이용하여 더 높은 부분을 만들어 계산할 수 있다.
다음과 같은 간단한 예를생각한다:

$$\max f(x_1, x_2) = \{ x_1 + x_2, \min (x_1, x_2)\}.$$

미분가능부분을 얻기위해 이 식을 더 간단한 부분으로 나눈다. 그리고, 점 $x_0=(0,\cdots,0)$에서 quasi-differential 을 계산한다.
즉, $f(x_1,x_2) = \max (h_1(x_1,x_2),h_2(x_1,x_2))$에서

$$h_1(x_1,x_2) = f_1(x_1,x_2)+f_2(x_1,x_2))$$
$$h_2(x_1,x_2) = \min (f_1(x_1,x_2),f_2(x_1,x_2))$$
$$f_1(x_1,x_2) = x_1$$
$$f_2(x_1,x_2) = x_2.$$

컴퓨터에 이 문제를 시행하기 위해, 임의의 연산을 어떻게 나누는가 하는 기준이 필요하다. 이것은 함수 $h_1,\cdots,h_k$, $f_1,\cdots,f_j$ 를 자동으로 정의하는 방법이 필요하다는 의미이다.

3.3 stepsize 의 계산

의사미분가능함수에서 방향 g_k의 결정에 따라 그 방향으로의 진행길이, 즉 stepsize는 주어진 점 x_k과 주어진 방향 g_k에 의해 결정된다. 이 문제는 보통 line search문제로 진행한다. 이에 대해 좀더 구체적으로 살펴보기 위해, 다음 1-dim. 최적화문제를 생각하자:

$$\text{Min} \quad h_k:U\in\mathbb{R}\rightarrow\mathbb{R} \text{ , 여기서 } t\rightarrow f(x_k+t\cdot g_k).$$

이 문제는 함수 h_k의 최소치로 수렴하는 수열 $(t_i), i\in\mathbb{N}$ 로부터 해를 구할 수 있다.
최적해 t^*는 구하고자 하는 값 α_k이다. 이 부분에서의 진행은 첫째, 앞단계에서 얻은 위치 t_i에 의해 t_{i+1}을 어떻게 계산하는가? 둘째, 계산은 언제 종료되는가? 하는 점을 생각해야 한다. 이러한 점에는 항상 두개의 상호배반적인 문제가 등장한다. 앨고리즘은 최소단계의 수로 최적에 도달하여야 하는 반면, 가능한한 정확한 결과를 얻기 위해 stepsize가 될 수 있는 한 작아야 한다. 이러한 요구에 적합한 방법으로는 미분가능함수에 대해 널리 사용되고 있는 sequential search법이다. 이 방법은 그 도함수를 필요로 하지 않고 함수의 값을 비교하여 interval을 얼마나 줄이는가를 결정한다[1].

3.4 함수값과 의사미분의 계산을 위한 연산

이 앨고리즘에서 반복단계2의 의사미분에 대한 계산은 실제 함수에 대해 많은 시간을 필

요로 하므로, 계산효율성을 추구할 필요가 있다. 이 방법을 위해 다음과 같은 기법을 이용한다: 함수값을 계산하기 위해, 최적화문제를 설명하는 함수를 recognize, analyse, evaluate해야 한다. 그런데, 문제는 사전에 행렬로 표현될 수 있는 선형함수와는 달리 의사미분가능함수는 일정한 구조를 갖지 않으므로, 프로그램을 수행하기 전에 그 구조를 알 수 없다는 것이다. 따라서, 프로그램을 run하는 동안 함수를 recognize할 필요가 있다. 이것은 Davie and Morrison[3]에 의해 제안된 연산자 해석과 같은 기법을 사용하여 수행할 수 있다. 연산자 해석은 다음과 같이 진행된다

1. Lexical analysis ('Scanning') : logical units인 SCANNER는 변환diagram으로 표시될 수 있는 주어진 syntax에 따라 recognize된다.
2. structural analysis ('Parsing') : PARSER는 language의 구조를 recognize하는 것이다.
3. evaluation : parsing process는 evaluation tree를 제시하고, 내부의 node는 operator에 의해 표시되며, leaf는 operand를 가르킨다. 각 node는 어떤 정보가 들어 있는 배열과, 동시에, 오른쪽 왼쪽으로 계속되는 node들의 address를 제시하는 두 개의 elements들로 표시된다. 이 tree로부터 연산식의 값을 계산하기 위해 cellar기법을 사용한다. 의사미분을 풀기위해 이것을 적용하기 위한 절차는 다음과 같다.

```
Procedure   evaluate(Tree)
begin
  if RIGHT SUBTREE EXISTS, then evaluate (Right Subtree).
  if LEFT  SUBTREE EXISTS, then evaluate (Left Subtree).
     if NODE IS AN OPERAND, then
                         PUT OPERAND INTO THE CELLAR
                         else
                         GET TWO OPERANDS FROM THE CELLAR
                         CARRY OUT THE REQUIRED CALCULATION
                         DELETE THE TWO OPERANDS ON THE TOP.
                         PUT THE RESULT INTO THE CELLAR
     end
end
```

상기 언급한 절차를 의사미분계산에 적용하는데 있어, 이들이 어떻게 진행되는가를 알기 위해 예로서 다음과 같은 의사미분가능함수에 대해 간단히 언급한다.

$f(x_1,x_2) = \max \{ x_1 \cdot x_2 - x_1^2, 2.5 \}$

i번째 단계에서 cellar의 calling sequences와 그 내용은 <표1>과 같다:

의사미분을 계산하기 위해, 함수 f를 parsing하여 differential part로 분할할 필요가 있다. 함수를 계산하는 것과 의사미분을 계산하는 것과의 차이는 의사미분함수인 경우에 실수 뿐만 아니라, 행렬 또는 벡터의 convex hull들이 존재한다는 것이다. 따라서, 한 cellar 대신 두 cellar 가 필요하다, 즉 subdifferential을 위해 하나, superdifferential을 위해 한 cellar가 필요하다. 또, convex hulls의 덧셈, 뺄셈, 스칼라곱을 위해 새로운 절차를 수행해야 한다. quasi-differential함수의 계산절차를 다음 예를 이용해 간단히 살펴본다:

$\max f(x_1,x_2)= \{ x_1+x_2, \min(x_1,x_2) \}$

여기서 $\underline{\partial}_i$, $\overline{\partial}_i$ 는 해당하는 cellar 의 i번째 element이다.

단계	Calling sequence	Cellar
0	evaluate (max) evaluate (2.5)	
1	put(2.5) evaluate (-) evaluate ($\Uparrow$) evaluate (2)	2.5
2	put(2) evaluate (x_1)	2 2.5
3	put(x_1)	x1 2 2.5
4	get (x_1,2) h=$x_1 \Uparrow 2$ delete (x_1,2)	2.5
5	put (h) evaluate (*) evaluate (x_2)	x_1^2 2.5
6	put(x_2) evaluate (x_1)	x_2 x_1^2 2.5
7	put(x_1)	x1 x2 x_1^2 2.5
8	get(x_1, x_2) h=$x_1 \cdot x_2$ delete(x_1, x_2)	x_1^2 2.5
9	put(h)	x1x2 x_1^2 2.5
10	get ($x_1 \cdot x_2$, x_1^2) h=$x_1 \cdot x_2 - x_1^2$ delete(x_1x_2, x_1^2)	2.5
11	put(h)	$x_1x_2-x_1^2$ 2.5
12	get($x_1 \cdot x_2-x_1^2$,2.5) h=max(2.5, $x_1 \cdot x_2-x_1^2$) put(h)	max($x_1x_2-x_1^2$,2.5)

< 표 1 >

	$\underline{\partial}$	$\bar{\partial}$
$\bar{\partial}_1 = \{0\}$ $\underline{\partial}_1 = \{\nabla f_1\}$	$\begin{pmatrix}1\\0\end{pmatrix}$	$\begin{pmatrix}0\\0\end{pmatrix}$
$\bar{\partial}_2 = \{0\}$ $\underline{\partial}_2 = \{\nabla f_2\}$	$\begin{pmatrix}1\\0\end{pmatrix}$ $\begin{pmatrix}1\\0\end{pmatrix}$	$\begin{pmatrix}0\\0\end{pmatrix}$ $\begin{pmatrix}0\\0\end{pmatrix}$
$\bar{\partial}_1 = \bar{\partial}_2 + \bar{\partial}_1$ $\underline{\partial}_1 = \underline{\partial}_2 + \underline{\partial}_1$	$\begin{pmatrix}1\\1\end{pmatrix}$	$\begin{pmatrix}0\\0\end{pmatrix}$
$\bar{\partial}_2 = \{0\}$ $\underline{\partial}_2 = \{\nabla f_1\}$	$\begin{pmatrix}1\\0\end{pmatrix}$ $\begin{pmatrix}1\\1\end{pmatrix}$	$\begin{pmatrix}0\\0\end{pmatrix}$ $\begin{pmatrix}0\\0\end{pmatrix}$
$\bar{\partial}_3 = \{0\}$ $\underline{\partial}_3 = \{\nabla f_2\}$	$\begin{pmatrix}0\\1\end{pmatrix}$ $\begin{pmatrix}1\\0\end{pmatrix}$ $\begin{pmatrix}1\\1\end{pmatrix}$	$\begin{pmatrix}0\\0\end{pmatrix}$ $\begin{pmatrix}0\\0\end{pmatrix}$ $\begin{pmatrix}0\\0\end{pmatrix}$
$\bar{\partial}_2 = co\{\bar{\partial}_2 - \underline{\partial}_3, \bar{\partial}_3 - \underline{\partial}_2\}$ $\underline{\partial}_2 = \underline{\partial}_3 + \underline{\partial}_2$	$\begin{pmatrix}1\\1\end{pmatrix}$ $\begin{pmatrix}1\\1\end{pmatrix}$	$\begin{pmatrix}0 & -1\\-1 & 0\end{pmatrix}$ $\begin{pmatrix}0\\0\end{pmatrix}$
$\bar{\partial}_1 = \bar{\partial}_2 \cdot \bar{\partial}_1$ $\underline{\partial}_1 = co\{\underline{\partial}_1 - \bar{\partial}_2 \cdot \underline{\partial}_2 - \bar{\partial}_1\}$	$\begin{bmatrix}1 & 1 & 2\\1 & 2 & 1\end{bmatrix}$	$\begin{bmatrix}0 & -1\\-1 & 0\end{bmatrix}$

점들의 집합에 대한 convex hull을 아주 빨리 결정하는 앨고리즘을 알지 못하므로, convex hull의 내부점들을 제거하지 않는다. 즉, 계산과정에서 얻는 점 모두를 store한다. 급하강방향을 결정하기 위해 convex hull 간의 개선된 길이가 필요하자. 그러나, 이러한 것은 별 어려움 없이 해결될 수 있다.

4. 계산결과 및 결론

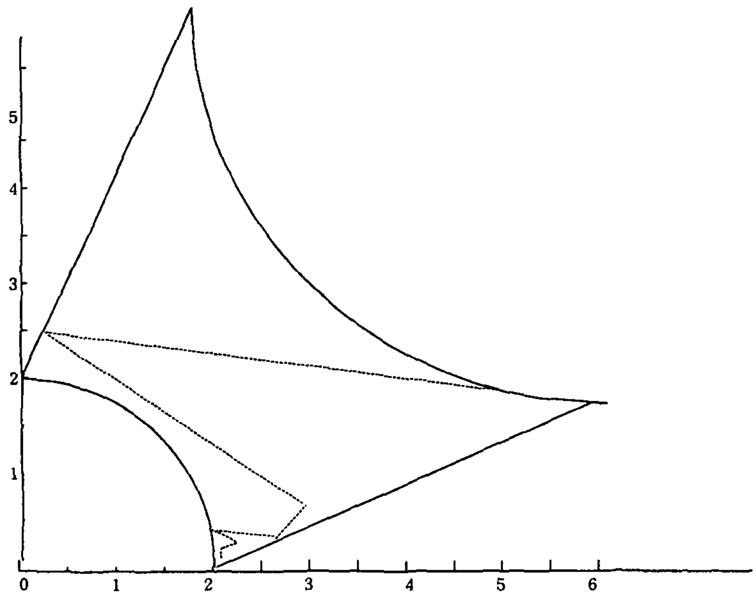
이 연구에서 제시한 앨고리즘을 수행하기 위해 다음과 같은 제약이 있는 최적화문제의 예에 적용하였다:

$$\min f(x) = |x_1| + x_2^2, \; x=(x_1,x_2)$$
$$\text{s.t. } 2x_1 - x_2 \geq -2$$
$$\frac{1}{2}x_1 - x_2 \leq 1$$
$$x_1^2 + x_2^2 \geq 4$$
$$(x_1-7)^2 + (x_2-7)^2 \geq 26$$

다음과 같은 계산단계의 결과를 얻는다:

i	x_i	$f(x_i)$	g_i	α_i
0	(6.00 ,2.00)	10	(-0.99,0.09)	5.74
1	(0.28, 2.57)	6.89	(0.81, -0.58)	3.21
2	(2.88, 0.69)	3.36	(-0.58,-0.81)	0.48
3	(2.60, 0.30)	2.69	(-0.95, 0.29)	0.69
4	(1.93, 0.51)	2.20	(0.38,-0.92)	0.33
5	(2.06, 0.21)	2.10	(0.92, -0.38)	0.07
6	(1.99 , 0.18)	2.02	(0.14, -0.98)	0.11
7	(2.00 ,0.07)	2.01	(-0.99,0.14)	0.01
8	(1.99, 0.07)	2.00	(0.06,-0.99)	0.04
9	(2.00 ,0.02)	2.00	(0.02,-0.99)	0.01
10	(2.00, 0.01)	2.00		

아래 그림은 이 계산단계에 대한 결과를 보여 준다.



의사미분가능함수의 이론을 근거로 제약이있는 비미문가능함수 최소화문제를 풀기위해 위에서 제시한 수치해석적인 앨고리즘을 통해, 그 implementation을 효율적으로 수행하는 체계적인 연산절차를 이용하였다. 해석적으로 어려운 개념에 속하는 Demyanov의 이론[5]은 일차도함수에 대한 앨고리즘에 대해 큰 어려움이 없이 컴퓨터프로그램에 의한 수행가능성을 보여주었다. 의사미분가능함수의 경우에 대한 global optimization과 같은 다른 문제에도 확대될 수 있다. 그러나, 이차이상의 도함수의 형태를 갖는 앨고리즘에 대해, 급하강방향을 계산하기 위한 집합들을 요구하지 않는 다른 방법들을 사용해야 할 것이다. 앨고리즘은 의사미분가능함수의 일반적인 최적화문제에 대해 우선적으로 잘 수행된다고 생각된다. 이 implementation은 programming language FORTRAN 77 수행되었고, 수치결과는 앨고리즘의 실험결과를 위해서만 제시되었다. 보다 상업적인 목적을 위해서는 실행시간과 main memory를 줄이기 위해 좀더 다듬어 져야 한다.

참 고 문 헌

[1]Bazaraa,M.S., Shetty,C.M. "*Nonlinear Programming*", John Wiley and Sons, 1979.

[2]Clarke,F.H., *"Generalized gradients and applications"*, Trans. Amer. Math. Soc., Vol.205, 1975.

[3]Davie,A.J.T., Morrison,R., *"Recursive descent compiling"*, Halsted Press, New York, 1978.

[4]Demyanov,V.F., Rubinov,A.M. *"On quasi-differentiable mappings"*, Optimization, Vol.14, No.1, 1983.

[5]Demyanov,V.F., Rubinov,A.M., *"Quasi-differentiable calculus"*, Optimization Software, Springer-Verlag, Berlin-Heidelberg-New York-Tokyo, 1986.

[6]Demyanov,V.F., Vasilev,L.V.," *Non-differentiable optimization"*, Optimization Software, Springer-Verlag Berlin, 1985.

[7]Demyanov,V.F., L.N.Polyakova, *"Minimization of a quasi-differentiable funtion on a quasi-differential set"*, USSR Computational Mathematics and Mathematical Physics, Vol.20, No.4, P.34-43, 1981.

[8]Pallaschke,D., R.Urbanski, *"Reduction of quasi-differentials and minimal representations"*, Mathematical Programming, Vol.66, 1995, pp.161-180,

[9]Pallaschke,D., R.Urbanski, *"A continuum of minimal pairs of compact convex sets which are not connected by translations"*, Journal of convex analysis, Vol.3, No.1, 1996, pp.83-95

[10]Rockafellar,R.T., *"Convex analysis"*, Princeton Unversity Press, Princeton, 1979.