The Korean Communications
in Statistics Vol. 6, No. 1, 1999
pp. 53-67

Improved Two Points Algorithm For D-optimal Design

Yunkee Ahn[1], Man-Jong Lee[2]

Abstract

To improve the slow convergence property of the steepest ascent type algorithm for continuous D-optimal design problems, we develop a new algorithm.

We apply the nonlinear system of equations as the necessary condition of optimality and develop the two-point algorithm that solves the problem of clustering. Because of the nature of the steepest coordinate ascent algorithm, avoiding the problem of clustering itself helps the improvement of convergence speed.

The numerical examples show the performances of the new method is better than those of various steepest ascent algorithms.

1. 서 론

D-최적실험계획법은 무한 측정치를 가진 선형회귀모형의 한 형태로서 이에 관한 알고리듬은 여러 학자들을 통하여 발전되어 왔다. 이에 관한 논문들 중에서 윈(Wynn, 1970)은 유한개의 관찰치를 사용한 D-최적실험계획법을 고려하였고, 훼도로프(Fedorov, 1972)는 1점 디자인을 사용한 연속적인 경우의 최대경사법(steepest ascent or descent method)에 대해서 다루었으며, 애트우드(Atwood, 1973)는 훼도로프의 방법을 발전시켰다.

훼도로프에 의한 최대경사법은 목적함수의 경사도(gradient)를 이용하여 최대경사의 경로를 따라 축차적으로 다음 단계로 가는 점들을 찾는 방법을 통하여 최적해를 찾아가는 방법이다. 이 방법은 조건을 충족하기 쉽고 틀이 갖추어져 있기 때문에 하나의 반복연산단계에서 다음 단계로 발전하는 것이 단순하고 쉽게 이루어지는 것이 특징이라고 할 수 있다. 그러나 이 알고리듬의 약점은 수렴속도가 상당히 느리다는 점이다. 이렇게 수렴속도가 느린 것은 알고리듬의 특성 때문일 것으로 생각된다. 그리고 이 방법은 1차 정보를 이용하기 때문에 해에 접근하는 속도가 느릴 뿐만 아니라, 집락화 문제(clustering problem)에 부딪히게 된다.

애트우드는 훼도로프 알고리듬에서 점 디자인의 선택을 달리함으로써 해에 대한 더 빠른 접근을 시도하였으나, 집락화 문제에서 벗어날 수 없었으며, 이 문제에 대하여 에달과 코타넥(Edahl & Kortanek, 1980)은 2점 디자인을 이용한 알고리듬으로 수렴속도의 향상과 더불어 집락화 문제 해결을 위한 단초를 제공하였다. 그러나 에달과 코타넥의 알고리듬에서도 여전히 집락화 문제는 존재하며 또한 정확한 해를 구하기는 어렵다.

이와 같은 두 가지 문제를 해결하기 위한 생각으로 우선 훼도로프의 최대경사법 알고리듬에 대해 애트우드가 수정한 알고리듬을 기준으로 하여, 키휘와 월포위츠(Kiefer-Wolfowitz, 1960)의 등가

1) (120-749) 서울 서대문구 신촌동 연세대학교 응용통계학과 교수.
2) (130-650) 서울특별시 동대문구 청량사서함 250 한국국방연구원 전력발전연구부 연구위원

법칙을 통하여 최적해를 위한 비선형 연립방정식을 도출하고, 이러한 연립방정식을 풀어서 좀더 정확한 해에 접근할 수 있는 방법을 고려해 보았다. 그러나 이 결과로 조금 더 정확한 해에 접근할 수는 있으나, 집락화 현상으로 인한 연립방정식의 변수증가와 비정칙 가능성을 초래하여 이 경우에도 심각한 장애요인으로 제기되었다. 이를 해결하기 위하여 본 논문에서는 집락식별방법(cluster identification method)을 제시해 보고, 이 방법을 에달과 코타넥의 2점 알고리듬에 적용하여 2점 알고리듬을 개선하였다.

이와 같이 본 논문에서는 연속적인 D-최적실험계획법 분야에서 에달과 코타넥이 연구한 2점 알고리듬을 개선할 수 있는 새로운 알고리듬을 제시하였다. 즉, 2점 알고리듬에 비선형 연립방정식의 풀이단계를 추가하고, 집락식별방법을 적용하여 알고리듬을 개선하였다. 이 결과로 기존의 알고리듬에 비하여 수렴속도를 향상시키고, 또한 많은 디자인 점들을 이용하지 않고서도 더욱 정확한 해를 얻을 수 있는가를 예제를 통하여 비교·분석하였다.

2. D-최적실험계획법 문제의 정형화

Ω 를 주어진 공간에서 공집합이 아닌 콤팩트(compact)한 부분집합이라 하자. 그리고 $f(\cdot)=[f_1(\cdot), f_2(\cdot), \cdots, f_m(\cdot)]^T$ 가 Ω 에서 R^m로 가는 연속적 벡타함수이며 $g(\cdot)$는 Ω 에서 R 로 가는 함수일 때 아래 식(2.1)과 같이 표시하기로 하자.

$$g(x) = \sum_{r=1}^{m} c_r f_r(x) + \psi(x), \qquad (2.1)$$

여기에서 c_r 은 상수이고, 주어진 임의의 값 x 에서 $\psi(x)$ 는 다음과 같은 특성을 가진 확률변수이다.

$$E(\psi(x)) = 0, \quad E(\psi(x)^2) = 1/\omega(x).$$

위에서 E 는 통계적 기대값을 의미하고, $\omega(\cdot)$는 Ω 에서 R 로 가는 연속적인 함수로서, Ω 에 속하는 모든 x 에 대하여 $\omega(x)$는 $0<\omega(x)<\infty$ 범위에 있다고 가정한다. Ω 에 속하는 임의의 점에서 $g(\cdot)$의 확률적 독립 측정치들(stochastically independent measurements)을 관찰함으로써 미지의 상수벡타인 $c = [c_1, c_2, \cdots, c_m]^T$ 를 추정하는 것이 통계학의 기본 문제이다. 최적실험계획법은 여기에서 c 에 대한 최적의 추정치를 얻을 수 있는 x 와 x 에서의 관찰치 수를 구하는 것이다.

2차 적률에 의해 표준화시키는 조건인 $E[(\psi(x)(\omega(x))^{1/2})^2]=1$ 을 만족하는 식 (2.1)을 다음과 같은 식으로 표현할 수 있다.

$$g(x)(\omega(x))^{1/2} = \sum_{r=1}^{m} c_r f_r(x)(\omega(x))^{1/2} + \psi(x)(\omega(x))^{1/2}$$

위의 형태는 가중회귀형태이므로 문제를 간단히 하기 위하여 $\omega(x)=1$이라고 가정할 수 있다. 주어진 자료인 $g(\cdot)$의 독립 측정치들은 Ω 에 속하는 점들인 $x_1, x_2, \cdots, x_n$ 에서 생성되는데, 각각의 점 x_i 에서 구해진 독립 측정치의 수는 각각 n_i 라 하자. 여기서 g_{ij} 를 점 x_i 에서의 j번째

측정치라 할 때, 다음과 같이 변수들을 정의해 보자.

$$N = \sum_{i=1}^{n} n_i , \qquad M = \sum_{i=1}^{n} (n_i/N) f(x_i) f(x_i)^T ,$$

$$b = \sum_{i=1}^{n} n_i f(x_i) \overline{g_i} , \qquad \overline{g_i} = (1/n_i)(\sum_{j=1}^{n_i} g_{ij}) .$$

M 이 정칙행렬(nonsingular matrix)이면, 가우스-마코프 정리(Gauss-Markov theorem)에 의해서 c 의 최량선형불편추정량(best linear unbiased estimator)인 $\hat{c} = (1/N)M^{-1}b$ 가 되고, $\hat{c}$ 의 분산공분산 행렬은 $(1/N)M^{-1}$ 가 된다. 측정치들의 총 숫자인 N 에 대해서 $\hat{c}$ 의 분산공분산 행렬이 작게 될 수 있도록 x_i 와 x_i 에서 측정치의 수 n_i 를 나타내는 디자인 ξ 를 구하는 것이 최적실험계획법이다. D-최적실험계획의 기준은 M^{-1}의 행렬식(det[M^{-1}])을 최소화하는 디자인 ξ 를 찾는 것이다.

이산적 D-최적실험계획의 기준을 만족시키는 최적 디자인을 구하는 방법은 다음과 같이 정형화할 수 있다.

$\{x_1, \cdots, x_n\}$를 디자인 점이라 하고, $\{p_1, \cdots, p_n\}$를 각각의 디자인 점에서의 가중치라 하면, 디자인 ξ 는 다음과 같이 표현할 수 있다.

$$\xi = \{((x_1, p_1), (x_2, p_2), \cdots, (x_n, p_n)) \mid \sum_i p_i = 1,$$

그리고 $i = 1, \cdots, n$ 에 대해서 가중치 $p_i \geq 0$ 이다.$\}$

이산적 D-최적실험계획의 기준은 Ω 에 속하는 $\{x_1, \cdots, x_n\}$과 $\{p_1, \cdots, p_n\}$에 대해서 $det(M^{-1})$를 최소화하는 디자인 ξ 를 찾는 것이다.

최적의 디자인을 구하기 위한 조건들은 다음과 같다.

$$M = \sum_{i=1}^{n} p_i f(x_i) f(x_i)^T : \text{정칙행렬(non-singular matrix)}$$

$$i = 1, \cdots, n \text{ 일 때 } p_i N : \text{정수}$$

위의 조건들 중에서 M 이 정칙행렬이라는 조건은 M 의 역행렬이 존재하기 위해서 필요한 조건이며, $p_i N$ 이 정수라는 조건은 실생활에서 실험계획을 하는데 있어서 디자인 점 x_i 에서의 실험의 수 n_i 를 의미하므로 매우 중요한 조건이라 할 수 있다. 그러나 이러한 조건하에서는 수리적인 접근이 매우 복잡한 정수계획법(integer programming)을 사용할 수 있으나, 반복연산방법을 사용한 수치해석법을 활용하여 근사적으로 해결할 수도 있다.

위에 제시된 문제는 이산적인 방법의 D-최적실험계획 문제로도 해결할 수 있지만, 만일 N 이 크고 고정되어 있다면, 이산적 문제를 연속적인 D-최적실험계획 문제로 전환한 후, 쉐도로프(1972), 애트우드(1973) 등의 학자들이 제시한 최대경사법을 활용한 근사방법(approximation method)을 사용하여 더욱 간편하게 최적의 디자인을 구할 수 있다.

따라서 위와 같이 가정을 하면 연속적인 D-최적실험계획법에 대한 문제를 다음과 같이 정립할 수 있다.

연속적인 D-최적실험계획법에 대한 문제는 $det(M^{-1})$를 최소화하는 디자인 ξ 를 찾는 것이다. 여기서 Ω 에서 x 가 취하는 값이 연속적이기 때문에 디자인 ξ 는 Ω 에서 가중함수 형태이며, M 은 다음과 같이 표시한다.

$$M = \left\{ M(\xi) \in R^{m \times m} \mid M(\xi) = \int_{\Omega} f(x) f(x)^T d\xi(x) \right\}$$

$$\text{단, } \int_{\Omega} d\xi(x) = 1 , \quad \xi(x) \geq 0 , \quad x \in \Omega$$

그리고 앞에서와 같이 M 은 정칙행렬이어야 문제를 풀 수 있다. 그러나 현실적으로 연속적인 디자인의 형태는 존재하기 어려우며, 수학적 접근이 용이한 현실적인 문제를 다루기 위해서는 다음과 같은 조건이 필요하게 된다. 특히 이산형 문제와 다른 점은 $p_i N$ 이 정수라는 조건이 필요하지 않다는 것이다.

(i) Ω 는 콤팩트하고 볼록(convex)하다.

(ii) $f : \Omega \rightarrow R^m$ 이고, 미분이 가능하다.

(iii) 다음과 같은 디자인이 존재한다.

$$\xi = \{ ((x_1, p_1), (x_2, p_2), \cdots, (x_n, p_n)) \mid \sum_i p_i = 1, \ n \geq m,$$

$$\text{그리고 } i = 1, \cdots, n \text{ 에 대해서 } p_i > 0 \text{ 이다. } \},$$

$$M(\xi) = \sum_{i=1}^{n} p_i f(x_i) f(x_i)^T \text{ 는 정칙행렬이다.}$$

3. 기존 알고리듬

3.1 1점 알고리듬

훼도로프(1972)는 연속적 D-최적실험계획법에서 최적해를 찾는 방법으로 최대경사법을 사용한 알고리듬을 제안하였다. 훼도로프의 알고리듬은 일종의 반복법을 수행하는 방법으로, 최초의 디자인을 $\xi^{(0)}$로 시작하여 현재의 디자인이 $\xi^{(k)}$라 할 때, 매 반복단계에서 x 에서의 가중치가 1인 하나의 점 디자인(point design) ξ_x ($=\{(x, 1)\}$)를 선정하고, 두 디자인의 최적 조합을 찾은 후, k 단계에서의 새로운 디자인 $\xi^{(k+1)}$를 설정함으로써 최적의 디자인을 찾아가는 과정이라고 할 수 있다.

여기서 $\xi^{(0)}$를 실행 가능한 초기 디자인이라 하고, ξ_x 는 x 에서의 가중치가 1인 점 디자인이라고 하자. 그리고 $\xi^{(1)}(a)$를 $\xi^{(0)}$와 점 디자인 ξ_x 내의 가중치들에 대한 선형결합으로 이루어진 디자인이라고 정의하자. 즉, 개념적으로 아래와 같이 표시할 수 있다.

$$\xi^{(1)}(a) = (1-a)\xi^{(0)} + a\xi_x$$

예를 들어서, $\xi^{(0)}$는 디자인 점의 수가 2개인 디자인이고 ξ_x 를 다음과 같다고 하자.

$$\xi^{(0)} = \{ (x_1, p_1), (x_2, p_2) \mid p_1 + p_2 = 1,\ p_1 > 0,\ p_2 > 0 \}$$
$$\xi_x = \{ (x, 1) \mid x \in \Omega \}$$

여기서 x 가 x_1 혹은 x_2와 같은 디자인 점이라면, 디자인 $\xi^{(1)}(a)$는 디자인 점의 수가 초기 디자인과 같이 2개이지만 같은 디자인 점에서의 가중치는 서로 더해져서 새로운 디자인으로 될 것이다. 그렇지 않고 x 가 x_1 혹은 x_2와 다른 점이라면 디자인 $\xi^{(1)}(a)$는 디자인 점의 수가 하나 늘어나서 3개로 되는 디자인이 된다.

이와 같이 훼도로프 알고리듬은 새로 선정한 점 디자인이 기존의 디자인에서 존재하는 점과 같거나, 계산된 가중치의 값이 0이 되지 않는 한 디자인 점의 수는 계속 늘어나게 된다.

디자인 $\xi^{(1)}(a)$에서 행렬 $M(\xi^{(1)}(a))$의 행렬식은 다음 식 (3.1)로 나타낼 수 있다.

$$det[M(\xi^{(1)}(a))] = (1-a)^m \left\{ 1 + \frac{a}{1-a} d(x, \xi^{(0)}) \right\} det[M(\xi^{(0)})] \qquad (3.1)$$
$$\text{단, } d(x, \xi^{(0)}) = f(x)^T M(\xi^{(0)})^{-1} f(x)$$

애트우드(1973)는 이러한 훼도로프 알고리듬의 일부분을 수정하여 최적해로의 수렴속도를 향상시켰다. 즉, 애트우드는 알고리듬이 기존의 훼도로프 알고리듬에서 점 디자인 ξ_x 의 선택을 다르게 하였다.

$0 < a < 1$ 일 때, 식 (3.1)은 $d(x, \xi^{(0)})$의 증가함수이다. 훼도로프는 x^*가 $d(x, \xi^{(0)})$를 최대로 하는 점이 될 때, 점 디자인 ξ_{x^*}를 선정하도록 주장하였다.

훼도로프 알고리듬에 대한 애트우드의 수정은 단순히 $a < 0$ 일 때 식 (3.1)이 $d(x, \xi^{(0)})$의 감소함수라는 점에 착안하여 $\Omega_{\xi^{(0)}}$ 가 $\xi^{(0)}$의 지주(support)일 때, $\min_{y \in \Omega_{\xi^{(0)}}} d(y, \xi^{(0)})$ 의 해인 y^*를 점 디자인을 위한 후보로 고려한 것이다.

즉, x^*와 y^*를 대입하여 둘 중에 더 큰 비율을 갖는 점 디자인을 선정하는 방법을 제시하였다. 위를 방법을 훼도로프-애트우드 알고리듬(F-A 알고리듬)이라 한다.

3.2 2점 알고리듬

다음의 제시된 예제에서 볼 수 있듯이, F-A 알고리듬을 적용하였을 경우 많은 반복연산단계가 소요되었다. 그것은 같은 집락안에 있는 디자인 점에서 다른 점으로 가중치를 이동시키는 과정이 여러 단계를 거쳐서 매우 느리게 수행되기 때문이다. 만일 이렇게 느린 가중치 이동과정을 알고리듬 내에서 감지하여 한번의 단계에서 집락내에 있는 모든 가중치를 한꺼번에 한 점으로 이동시켜 줄 수 있다면 수렴속도를 향상시킬 수 있을 것이라는 가정 하에 에달과 코타넥(1980)은 다음과 같이 2점 디자인을 사용한 알고리듬을 제시하였다.

에달과 코타넥의 2점 알고리듬의 기본 구상은 F-A 알고리듬에서 새로운 디자인 점을 1점씩 선정하는 단계를 수정하여 아래와 같이 2점의 새로운 디자인 점을 선정함으로써 수렴속도 향상을 시도하였다. 2점 디자인을 사용한 알고리듬을 아래의 E-K 알고리듬을 제시하였다.

$\xi^{(k)}$를 현재의 디자인이라 하고, $\xi_{x_1^{(k)}}$과 $\xi_{x_2^{(k)}}$를 두 개의 서로 다른 점 디자인이라 하자. 그리고 $\xi^{(k')}(a_1^{(k)})$와 $\xi^{(k+1)}(a_1^{(k)}, a_2^{(k)})$를 다음과 같이 설명한다.

$$\xi^{(k')}(a_1^{(k)}) = (1-a_1^{(k)})\xi^{(k)} + a_1^{(k)}\xi_{x_1^{(k)}} \tag{3.2}$$

$$\begin{aligned}\xi^{(k+1)}(a_1^{(k)}, a_2^{(k)}) &= (1-a_2^{(k)})\xi^{(k')}(a_1^{(k)}) + a_2^{(k)}\xi_{x_2^{(k)}} \\ &= (1-a_1^{(k)})(1-a_2^{(k)})\xi^{(k)} + a_1^{(k)}(1-a_2^{(k)})\xi_{x_1^{(k)}} + a_2^{(k)}\xi_{x_2^{(k)}}.\end{aligned} \tag{3.3}$$

2점 디자인을 사용한 알고리듬을 D-최적실험계획에 적용하기 위해서는 식 (3.2)와 식 (3.3)을 이용하여, 1점 디자인 형식인 식 (3.1)과 같이 다음과 같은 2변수 최적화 문제를 만족하는 $a_1^{(k)}$와 $a_2^{(k)}$를 구하여야 한다.

$$\min \ det[M(\xi^{(k+1)}(a_1^{(k)}, a_2^{(k)}))^{-1}]$$

에달과 코타넥이 제시한 알고리듬을 E-K 알고리듬이라 하면 다음과 같이 표현할 수 있다.

<u>**<E-K 알고리듬>**</u>

(단계 0) $k=0$이라 하고, 최초의 디자인 $\xi^{(0)}$을 선정한다.

(단계 1) $x_1^{(k)}$과 $x_2^{(k)}$를 다음과 같이 선정한다.

$$x_1^{(k)} = \arg\min_{x\in\Omega_{\xi^{(k)}}}$$

$$d(x, \xi^{(k)})\, x_2^{(k)} = \arg\max_{x\in\Omega} \ d(x, \xi^{(k)}) \ .$$

$x_1^{(k)}$ 과 $x_2^{(k)}$를 사용하고, 앞의 식을 만족하는 $a_1^{(k)}$ 과 $a_2^{(k)}$를 결정한다.

(단계 2) 다음과 같은 새로운 디자인 $\xi^{(k+1)}$ 를 계산한다.

$$\xi^{(k+1)} = (1-a_1^{(k)})(1-a_2^{(k)})\xi^{(k)} + a_1^{(k)}(1-a_2^{(k)})\xi_{x_1^{(k)}} + a_2^{(k)}\xi_{x_2^{(k)}} \ .$$

$k=k+1$ 로 하여 **(단계 1)**로 돌아간다.

위와 같이 두 점을 사용하게 되는 경우, 이 두 점을 x_1과 x_2라 하고, x_1의 가중치가 x_2로 이동할 수 있다고 하자. 이때 가중치는 전반적인 디자인을 경유하여 x_1에서 x_2로 가는 것이 아니라 바로 x_1에서 x_2로 이동할 수 있게 된다.

이것은 1점 접근방법에서 몇 차례의 반복연산과정을 거듭했던 일을 한번에 수행할 수도 있게 됨을 의미한다. 또한 x_1과 x_2가 같은 집락에 있다면, 2점 알고리듬에서 한번의 반복법을 통하여 그 집락으로부터 x_1을 제거할 수 있는 장점도 가지고 있다. 만일 x_2가 새로운 가중치 점이고 x_2 가까이에 x_1이 있다면(즉, 같은 집락안에 있다면), 새로운 점 x_2는 과거의 점 x_1을 흡수할 것이기

때문에 그 집락의 크기는 같은 크기로 남아 있을 것이다.

이러한 접근방법은 가중치가 존재하는 2개의 점들 (x_1, x_2) 를 함께 적용할 수 있게 해 주며, 좌표경사법 형태의 알고리듬으로 하나의 좌표가 아닌 2개의 좌표가 동시에 고려될 수 있다는 것이 획기적인 발전을 보인 점이라 할 수 있다. 또한 가중치가 한 점에서 다른 점으로 더욱 빨리 이동할 수 있기 때문에 수렴속도에서 향상을 보여주었다.

4. 개선된 알고리듬

4.1 최적해를 위한 방정식 도출

키휘와 월포위츠의 등가법칙으로부터, 비선형 연립방정식에 기초를 두고 있는, 최적화를 위해 필요한 조건들을 도출할 수 있다. 키휘와 월포위츠의 등가법칙은 다음과 같다.

<키휘와 월포위츠의 등가법칙>

$d(x, \xi) = f(x)^T M(\xi)^{-1} f(x)$ 일 때, 다음의 주장들은 동등하다.

(i) 디자인 ξ^*는 $det\,[M(\xi)]$를 최대화한다. (즉, $det\,[M(\xi)^{-1}]$를 최소화한다);

(ii) 디자인 ξ^*는 $\max_x d(x, \xi)$ 를 최소화한다.;

(iii) $\max_x d(x, \xi) = m$

또한 키휘와 월포위츠의 등가법칙으로부터 최적 디자인 ξ^*의 점들에서, $d(x, \xi^*)$는 그 최대치인 m 이 되는 것을 알 수 있다.

이와 같은 특성들에 의하여 다음과 같은 필요충분조건들이 만족되면 디자인 ξ^*가 D-최적 디자인임을 알 수 있다.

$$f(x)^T M(\xi^*)^{-1} f(x) \le m \qquad \text{for } all \;\; x \in \Omega \tag{4.1}$$

$$f(x_i)^T M(\xi^*)^{-1} f(x_i) = m \qquad \text{for } i = 1, 2, \cdots, n \tag{4.2}$$

식 (4.1)과 식 (4.2)를 사용하여 D-최적실험계획을 만족시킬 수 있는 필요조건인 비선형 연립방정식 체계를 얻을 수 있다.

$$f(x_i)^T \Big(\sum_{k=1}^{n} p_k f(x_k) f(x_k)^T \Big)^{-1} f(x_i) = m \qquad \text{for } i = 1, \cdots, n \tag{4.3}$$

$$f(x_i)^T \Big(\sum_{k=1}^{n} p_k f(x_k) f(x_k)^T \Big)^{-1} f'(x_i) = 0 \qquad \text{for } i = 1, \cdots, n. \tag{4.4}$$

그리고 이러한 연립방정식의 해가 존재하기 위해서는 $f(x)$가 미분이 가능한 함수이어야 한다. 그런데 훼도로프의 최대경사법의 기본 원리가 목적함수의 경사도를 이용하여 최대경사의 경로를 따라 축차적으로 다음 단계로 가는 점들을 찾는 방법이므로, 위의 비선형 연립방정식 체계를 알고리

듬에 추가하여 수정하는 아래와 같은 알고리듬의 해가 존재하기 위해서는 $f(x)$가 2차 미분이 가능한 함수이어야 한다. 따라서 앞에서의 조건들 중 두 번째 항목을 다음과 같이 수정해야 한다.

(ii) $f: \Omega \rightarrow R^m$ 이고, 연속적인 2차 미분이 가능하다.

4.2 CI 알고리듬

앞에서 x_1과 x_2를 선정하는 방법은 E-K 알고리듬의 (단계 1)에서 두 점을 선정하는 방법 이외의 다른 방법도 고려될 수 있다. 먼저 x_2를 선정하는 방법은, 2점 알고리듬의 수렴을 보장하기 위해서, F-A 알고리듬과 같이 훼도로프 기준에 의하여 $\max_{x\in\Omega} d(x,\xi)$ 를 만족하는 x_2를 선정하는 것이 가장 단순한 방법이다.

다른 점 x_1을 선정하는 방법에 대해서 에달과 코타넥은 두 가지 방법을 제시하였는데, 첫 번째 방법은 앞의 알고리듬에서 소개된 바와 같으며 이 방법을 다시 써보면 아래와 같다.

$$\min_{x\in\Omega_{\xi^{(k)}}} d(x,\xi) \text{ , 단, } \Omega_{\xi^{(k)}} \text{ 는 } \xi \text{ 에 대한 현재의 지주이다.}$$

이것은 x_1과 x_2를 선정하기 위하여 F-A 알고리듬에서 두 번의 반복연산단계를 수행하는 것을 나타낸다. 다시 말해서 하나의 반복연산단계에서 F-A 알고리듬을 두 번 반복하는 것으로 좋은 해가 구해질 수도 있다.

두 번째 방법은 에달과 코타넥의 컴퓨터 사용의 경험을 바탕으로 제시한 방법으로서, 첫 번째 방법에서 선정한 두 점이 같은 집락내에 있을 경우 한 점을 제거하려는 생각에서 제시한 방법이다. 즉, x_1이 집락내의 점일 때 x_2가 같은 집락내의 점이 될 수 있는 기회를 더욱 크게 하여 한 점을 제거하려는 생각에서 출발하였다. 이 두 사람의 경험에 의하면 먼저 한 점을 선정하고 a_1 을 구한 다음, 여기에 1 보다 큰 임의의 상수를 a_1 에 곱하여 a_1 를 수정해주는 방법을 말한다. 이 두 사람은 1보다 큰 임의의 상수를 선택하는 경우 1.5의 수치를 적용하였을 때 좋은 결과를 얻을 수 있었다고 주장하였다. 이것은 전반적인 디자인에서 어떠한 변환을 통하여 가중치를 x_1에서 x_2로 이동하는 방법이라 할 수 있다. 그러나 x_1이 집락내의 점이 아닐 경우에 이 방법은 특별한 효과가 없다고 하였다.

세 번째는 목적함수를 최대로 향상시키기 위하여 ξ 의 현재의 지주인 $\Omega_{\xi^{(k)}}$ 로부터 x_1을 선정하는 방법이다(이 방법은 x_1에 대한 $\Omega_{\xi^{(k)}}$ 의 모든 점들을 하나씩 시험해 봄으로써 수행할 수 있다).

이러한 2점 알고리듬은 가중치 교환을 통하여 수렴속도를 향상시킬 수 있는 장점을 가지고 있다. 그러나 이러한 세 가지 방법들 중 어느 것도 집락의 형성에 대한 예방 및 집락제거에 대한 보장을 하고 있지는 못하다. 만일 2개의 점이 같은 집락에 있다면, 그것들은 거의 같은 $d(x_i,\xi)$ 값을 가질 것이다. 이러한 현상으로 인하여 첫 번째 방법에서는 그러한 두 점을 선택하게 되지는 않을 것이다. 왜냐 하면 이 방법은 모든 점들 중에서 $d(x,\xi)$ 값이 가장 크고, 작은 두 점을 선택하는 것이기 때문이다. 또한 두 번째 방법에서도 F-A 알고리듬의 특성 때문에 같은 집락내에 있는 두 점이

함께 선정되지는 않을 것이다. 마지막으로, 세 번째 방법에서도 같은 집락내에서 새로운 점을 선정하지는 않을 것이다. 왜냐 하면 같은 집락내의 점들은 비슷하게 반응하기 때문에, 다른 집락에 있는 점을 이용하는 것이 더욱 향상된 값을 얻을 수가 있기 때문이다(같은 집락내에서 두 번째 점을 선정할 경우 단지 한 점을 사용했을 때보다 상대적으로 향상되는 부분이 아주 적게 된다).

만일 어떠한 집락도 존재하지 않는다면, 이러한 세 가지 선정방법은 좋은 방법으로 간주될 수 있다. 그러나 집락이 존재하면 수렴속도가 심하게 느려지게 되기 때문에, 집락의 형성을 방해하거나 제거하는 데에 주의를 기울여야 할 것이다. 따라서 어떤 점들이 집락을 형성하고 있는지 판단하여야 한다.

통상적으로 R^n에서 어떤 점들이 집락을 이루고 있는지 공간적으로 식별하는 방법으로 유크리디안 거리를 사용하는 군집분석(classification analysis)을 사용하여 집락식별 문제에 접근할 수 있다. 그러나 이 방법은 최적 디자인이 R^n 에 있는 근접한 두 점을 가지고 있는 경우에는 식별되지 않게 된다. 많은 집락제거 방법들은 이러한 두 점을 합병하여 한 점만 디자인에 포함시킬 수도 있다. 2점 알고리듬은 이러한 잘못된 합병을 피할 수는 있다. 왜냐 하면 그 점들의 가중치들이 발견적 접근방법(heuristic approach)을 적용하지 않고, 최적으로 조정되기 때문이다. 그러나 이러한 방법들은 집락을 제거하기 위한 적극적인 방법이 되지는 못한다.

이제 집락을 제거하기 위한 적극적인 방법을 생각해 보자. 우선 첫 번째 점을 선정한 후에는, 그 점을 포함하여 집락을 형성할 수 있는 다른 점들이 있는지 찾아 볼 수 있다. 이렇게 해서 찾은 집락점들 중 하나를 제거함으로써, 가중치를 이동시킬 수 있다. 첫 번째 점을 활용하여 집락점들 중의 하나를 제거할 수 있을지를 알기 위해서는 $\Omega_{\xi^{(k)}}$안에 있는 점들을 각각 하나씩 직접 검사해 보아야 할 것이다. 만일 그러한 점이 존재하지 않는다면, k 번째 디자인에서 결정된 $a_1^{(k)}$에 대한 $d(x,\xi)$ 의 편미분($\partial d(x,\xi)/\partial a_1^{(k)} = d(x,\xi) - d^2(x,x_1^{(k)},\xi)$)을 이용하여 가능한 집락점을 찾아 보아야 한다. 그것은 우리가 현재의 디자인에 있는 모든 점들에 대한 편미분 값들을 검사해서 그 중에 가장 큰 음의 값을 갖는 점을 선정하는 방법을 의미한다. 만일 어떠한 점이 x_1을 포함한 집락에 있다면, x_1에서의 편미분 값은 $m-m^2$ 값과 거의 같은 값이 될 것이기 때문에 그 편미분 값들이 $m-m^2$ 값과 차이가 크면, 현재의 디자인에 집락점이 없다고 판단할 수 있다.

본 논문에서는 이러한 집락식별방법과 에달과 코타넥의 2점 접근방법을 활용하여 새로운 알고리듬을 제시하였다. 이름은 집락을 식별하는 알고리듬이라는 의미로 CI(cluster identification) 알고리듬이라 하였다. CI 알고리듬은 다음과 같다.

<CI 알고리듬>

(단계 0) E-K 알고리듬의 (단계 0)을 적용한다.

(단계 1) $\xi^{(k)}$를 시작점으로 하여, 식 (4.3)과 식(4.4)의 비선형 연립방정식 체계 에 대한 근사해를 구한다.

(단계 2) y_1 과 y_2 를 다음과 같다고 하자.

$$y_1 = \arg\min_{x\in\Omega_{\xi^{(k)}}} d(x,\xi^{(k)}). \qquad y_2 = \arg\max_{x\in\Omega} d(x,\xi^{(k)}).$$

(i) y_1에 의한 a를 계산한다. 만일 y_1이 현재의 지주로부터 제거될 수 있다면 (ii)로 간다.그렇지 않으면 (iii)으로 간다.

(ii) y_3를 다음과 같다고 하자.

$$y_3 = \arg\min_{x\in\Omega_{\xi^{(k)}}} (d(x,\xi^{(k)}) - d^2(y_1, x, \xi^{(k)})).$$

만일 $\min_{x\in\Omega_{\xi^{(k)}}} (d(x,\xi^{(k)}) - d^2(y_1, x, \xi^{(k)})) \leq m(1-m)/2$이면,

$x_1^{(k)} = y_1$ 과 $x_2^{(k)} = y_2$ 로 하여 (vi)으로 간다.

그렇지 않으면 $x_1^{(k)} = y_1$ 그리고 $x_2^{(k)} = y_3$ 로 하여 (vi)으로 간다.

(iii) y_4를 다음과 같다고 하자.

$$y_4 = \arg\min_{x\in\Omega_{\xi^{(k)}}} (d(x,\xi^{(k)}) - d^2(y_1, x, \xi^{(k)})) .$$

만일 y_1 혹은 y_4 가 디자인으로부터 제거될 수 있다면,

$x_1^{(k)} = y_1$ 그리고 $x_2^{(k)} = y_4$ 로 하여 (vi)으로 간다.

(iv) y_5를 다음과 같다고 하자.

$$y_5 = \arg\min_{x\in\Omega_{\xi^{(k)}}} (d(x,\xi^{(k)}) - d^2(y_2, x, \xi^{(k)})) .$$

만일 y_2 혹은 y_5가 디자인으로부터 제거될 수 있다면,

$x_1^{(k)} = y_2$ 그리고 $x_2^{(k)} = y_5$ 로 하여 (vi)으로 간다.

(v) 만일 $\min_{x\in\Omega_{\xi^{(k)}}} (d(x,\xi^{(k)}) - d^2(y_1, x, \xi^{(k)})) \leq m(1-m)/2$ 이면,

$x_1^{(k)} = y_1$ 그리고 $x_2^{(k)} = y_4$ 로 하여 (vi)으로 간다.

그렇지 않고 $\min_{x\in\Omega_{\xi^{(k)}}} (d(x,\xi^{(k)}) - d^2(y_2, x, \xi^{(k)})) \leq m(1-m)/2$이면,

$x_1^{(k)} = y_2$ 그리고 $x_2^{(k)} = y_5$ 로 하여 (vi)으로 간다.

그렇지 않으면 $x_1^{(k)} = y_1$ 그리고 $x_2^{(k)} = y_2$ 로 놓는다.

(vi) $x_1^{(k)}$ 과 $x_2^{(k)}$를 이용하고, E-K 알고리듬에서와 같이 $a_1^{(k)}$ 과 $a_2^{(k)}$를 결정한다.

(단계 3) 다음과 같은 새로운 디자인 $\xi^{(k+1)}$를 계산한다.

$$\xi^{(k+1)} = (1-a_1^{(k)})(1-a_2^{(k)})\xi^{(k)} + a_1^{(k)}(1-a_2^{(k)})\xi_{x_1^{(k)}} + a_2^{(k)}\xi_{x_2^{(k)}}$$

$k = k+1$ 로 하여 **(단계 1)**로 돌아간다.

CI 알고리듬의 (단계 2)에서는 다음과 같이 오목하지 않은 최대화 부 프로그램(nonconcave

maximization subprogram)을 풀어야 한다.

$$y_2 = \arg\max d(x, \xi^{(k)}),\ x \in \Omega .$$

콤팩트한 집합인 Ω 에서 이러한 부 프로그램을 해결하기 위해서는 이산화(discretized) 집합이 되어야 한다. 그러나 만일 우리가 고정된 일양 유한 격자점(fixed uniform finite grid point)인 Ω_D (Ω 의 부분집합)를 사용한다면, 이러한 부 프로그램을 위한 정확한 해를 구할 수 없을 것이다. 반면에 연속적으로 격자점을 순화하는 과정(grid refinement procedure)을 활용한다면, 상대적으로 정확한 해를 얻을 수 있을 것이다. 그러나 이 경우에는 $d(x, \xi^{(k)})$를 계산하기 위하여 많은 시간을 소모해야 할 것이며 이렇게 격자점을 순화하여도 여전히 현재의 지주가 제한을 받는다. 이것이 CI 알고리듬의 (단계 1)에서 비선형 연립방정식의 풀이단계를 사용하는 이유가 된다. 즉, CI 알고리듬의 (단계 2)에서 구해진 유한개의 고정된 격자점을 사용한 근사 디자인(finite fixed grid approximation design)에 대하여 비선형 연립방정식을 적용하는 것은 앞에서 제시한 곤란한 문제를 해결할 수 있을 뿐만 아니라, 많지 않은 점을 사용해서 정확한 해를 구할 수 있기 때문이다.

만일 새로 구해진 디자인에서 디자인 점의 수가 바로 앞의 반복연산단계에서의 디자인 점의 수보다 많으면, CI 알고리듬의 (단계 1) 과정은 생략된다. 실제로 알고리듬의 수행과정에서 CI 알고리듬의 (단계 1)은 현재의 디자인에 있는 점들이 바로 전 반복연산단계의 디자인에서의 사용된 점의 수보다 같거나 작을 때만 적용된다. 이러한 방법을 통하여 비선형 연립방정식의 해를 구하는 과정에서의 불필요한 노력을 줄일 수 있고, 전산기 사용시간도 줄일 수 있다.

5. 비교 분석

5.1 세이(Tsay, 1975) 예제 비교 · 분석

앞에서 제시한 알고리듬들을 비교해 보기 위하여 VAX 컴퓨터의 6430 기종에서 포트란(Fortran) 프로그램을 사용하였다. 또한 수행결과를 상호 비교해 보기 위하여, 각각의 알고리듬에서 같은 예제함수를 이용하였다.

<예제함수 1>

$$g(x;\theta) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x_+^2 + \theta_4 (x-0.3)_+^2$$

단, $(x-a)_+^j = (x-a)^j$, $x \geq a$ 인 경우

$= 0$, $x < a$ 인 경우

$$\Omega = [a, b] = [-1, 1]$$

다음의 <표 5.1>에서는 위의 예제함수를 이용한 각각의 알고리듬들의 수행결과들을 비교해 보았다. F-A 알고리듬과 그리고 에달과 코타넥의 E-K 알고리듬 등은 모두 401개의 한정된 격자점과 같은 초기치(0, 100, 200, 300, 400)를 사용하여 최적해를 구할 수 있었다.

<표 5.1>에서 볼 수 있듯이 F-A 알고리듬은 모두 반복연산단계가 1,000단계가 넘게 매우 느린 속도로 최적해를 찾아 가는 과정이 진행되었다. 또한 F-A 알고리듬의 7~10번째 반복연산단계에서 목적함수의 값은 1.86651×10^{7}으로 비슷하게 나타났으며 이것은 이 단계에서 디자인 점들의 집락화 현상으로 인하여 목적함수의 값을 거의 향상시키지 못하였던 것으로 판단할 수 있다.

에달과 코타넥의 E-K 알고리듬에서는 수렴속도가 획기적으로 향상되어 11번째 반복연산단계에서 수렴하였다. 또한 사용한 디자인 점의 수를 비교해 볼 때, F-A 알고리듬의 경우는 점점 증가하여 15개 이상의 디자인 점을 사용하였으나 E-K 알고리듬은 5개로 가장 작았다. 본 논문에서 제시한 CI 알고리듬은 E-K 알고리듬의 수렴단계인 11번째 반복연산단계 보다 빠른 두 번째 반복연산단계에서 수렴하였으며, 이 경우는 첫 번째 반복연산단계를 수행한 결과로 구해진 디자인이 비선형 연립방정식의 좋은 초기치로 사용되었기 때문이다. 사용한 디자인 점의 수도 5개로 E-K 알고리듬에서와 같이 가장 작은 수를 사용하였다.

<표 5.1> 세이의 예제에 의한 D-최적실험계획 결과 비교

구 분	F-A 알고리듬		E-K 알고리듬		CI 알고리듬	
iter.	n	D	n	D	n	D
0	5	1.36125	5	1.36125	5	1.36125
1	6	1.48903	5	1.85381	5	1.85381
2	6	1.63629	5	2.09335	5	2.15025
3	7	1.71664	5	2.11965		
4	7	1.80074	5	2.13895		
5	8	1.85221	5	2.14311		
6	8	1.86639	5	2.14653		
7	8	1.86651	5	2.14840		
8	8	1.86651	6	2.14919		
9	8	1.86651	6	2.14929		
10	7	1.86651	5	2.14987		
11	8	1.89716	5	2.15016		
20	12	2.07487				
30	14	2.11105				
40	15	2.11506				
50	16	2.11849				
100	17	2.13032				
500	15	2.14755				
1000	15	2.14843				

iter. : 반복연산단계의 수

n : 사용한 디자인 점들의 수

D : 정보행렬의 행렬식 $|M|\times 10^{7}$

CI 알고리듬에서 첫 번째 반복연산단계에서 선정한 두 개의 디자인 점은 E-K 알고리듬에서 선정한 두 점과 같았으나 집락식별과정에서 선정된 점 디자인으로 기존의 지주에 있는 디자인 점에서의 가중치를 이동시켜서 기존의 디자인 점 하나를 제거시킬 수 있었고, 이렇게 새로 정해진 디자인을 기준으로 비선형 연립방정식을 풀었을 때 바로 최적해에 도달하였다.

그러나 비선형 연립방정식 풀이과정에서 $\Omega = [a, b]$ 의 양쪽 끝점인 a 혹은 b 에서의 미분값이 Ω 의 범위를 벗어날 수도 있으므로 CI 알고리듬에서 시작점을 외부에서 임의로 지정해 준 격자점을 사용하지 않고, 프로그램에서 생성된 내부점을 이용할 경우(이 경우의 초기치는 401/6, 2×401/6, 3×401/6, 4×401/6, 5×401/6 등으로 5개의 내부점을 이용하였음)에도 기존의 다른 알고리듬들에 비해 상대적으로 빠른 8번째 반복연산단계에서 최적해로 도달하였으며, 이 결과로 구해진 최적해는 CI 알고리듬을 이용하여 5개 격자점을 이용하여 구한 최적해와 일치하였다.

5.2 베트시스(Betsis, 1985) 예제 비교·분석

두 번째의 예제로는 베트시스의 예제에서 사용된 삼각함수를 이용한 최적해를 도출해 보았다.

<예제함수 2>

$$g(x;\theta) = \theta_0 + \theta_1 \sin x + \theta_2 \cos x + \theta_3 \sin 2x + \theta_4 \cos 2x + \theta_5 \sin 3x$$

$$\Omega = [a, b] = [0, 2\pi]$$

이 문제는 6개의 디자인 점으로 구성된 이론적인 최적해가 존재하고, 그 값은 다음과 같다.

$$x_i : \quad (2i+1)\pi/6, \quad i = 0, 1, \cdots, 5,$$

$$p_i : \quad 1/6, \quad i = 0, 1, \cdots, 5$$

<표 5.2>에서는 위의 예제함수를 이용한 각각의 알고리듬들의 수행결과들을 비교해 보았다. F-A 알고리듬과 그리고 E-K 알고리듬 등은 모두 61개의 한정된 격자점과 같은 초기치(0, 12, 24, 36, 48, 60)를 사용하여 최적해를 구할 수 있었다.

<표 5.2>에서 각각의 알고리듬 수행결과를 비교해 보면, F-A 알고리듬은 544단계에서 수렴하였다. E-K 알고리듬에서는 수렴속도가 획기적으로 향상되어 35번째 반복연산단계에서 최적해로 수렴하였다. 본 논문에서 제시한 CI 알고리듬은 E-K 알고리듬에 비해 수렴속도에서 큰 증가를 보였으며, 그 결과 두 번째 반복연산단계에서 최적해로 수렴하였다. 이 경우도 앞에서 세이의 예제함수를 적용했을 때와 마찬가지로 첫 번째 반복연산단계를 수행한 결과로 구해진 디자인이 비선형 연립방정식의 좋은 초기치로 사용되었기 때문이다. 사용한 디자인 점의 수는 6개로 F-A 알고리듬 및 E-K 알고리듬의 경우와 같은 수를 사용하였다.

<표 5.2> 베트시스 예제에 의한 D-최적실험계획 결과

구 분	F-A 알고리듬		E-K 알고리듬		CI 알고리듬	
iter.	n	D	n	D	n	D
0	6	0.87213	6	0.87213	6	0.87213
1	7	1.18371	6	1.45828	6	1.45828
2	8	1.72058	7	2.89648	6	6.25000
3	8	2.36708	8	3.91501		
4	9	2.78991	9	4.82548		
5	10	3.36931	10	5.07235		
6	10	3.44474	11	5.27907		
7	11	3.67033	10	5.30732		
8	12	3.93034	11	5.39036		
9	12	4.01902	11	5.51977		
10	12	4.01989	9	5.53353		
17	12	4.57786	12	6.03326		
20	12	4.84500	10	6.11119		
30	14	5.47176	6	6.24999		
35	13	5.47486	6	6.25000		
50	14	5.60952				
100	16	5.81941				
544	6	6.25000				

iter. : 반복연산단계의 수

n : 사용한 디자인 점들의 수

D : 정보행렬의 행렬식 $|M| \times 10^2$

6. 결 론

연속적 D-최적실험계획 문제를 풀기 위한 최대경사법 형태의 알고리듬에서 수렴속도를 향상시키기 위하여, 본 논문에서는 두 가지 개선 방법을 기존방법에 추가한 새로운 방법을 제시하였다.

비선형 연립방정식에서의 변수와 방정식의 수를 감소시키기 위하여, 2점 디자인을 사용한 알고리듬인 E-K 알고리듬에 집락식별방법을 추가함으로써 집락화 문제를 개선하였으며, 이와 같은 비선형 연립방정식과 집락식별방법을 적용하여 CI 알고리듬을 만들어 제시하였다. 이 경우 최대경사법의 속성 때문에, 집락화 문제를 피하는 것이 수렴속도를 향상시키는 데 도움이 되었다.

그리고 집락화 현상이 발생하지 않는 경우에는 비선형 연립방정식의 풀이단계와 집락식별과정으로 인하여 E-K 알고리듬에 비하여 CI 알고리듬의 수렴시간이 더 길어질 수 있으므로, 바로 전의 반복연산단계에 비하여 디자인 점의 수가 늘어나게 되는 반복연산단계에서는 비선형 연립방정식의 풀이단계를 생략하여 수렴속도 향상을 위한 방법으로 프로그램에 적용하였다.

예제를 통하여 기존의 방법들과 새로운 접근방법을 수행한 결과에 대하여 비교·분석을 하였으며, 그 결과로 본 논문에서 제시한 방법이 기존의 방법들에 비하여 적은 디자인 점들을 이용하고도 더욱 빠르게 최적해를 구할 수 있는 방법임을 보였다.

이러한 알고리듬은 응용측면에서 환경오염을 측정하기 위한 장소를 결정하는 문제, 그리고 잠수함의 소나에서 센서의 위치를 결정하는 문제 등 다양한 과학분야에서 활용될 수 있을 것이다.

앞으로 이론적인 측면에서 디자인 점을 벡타로 전개하여 비선형 연립방정식의 필요충분조건을 만족시킬 수 있는 알고리듬을 개발할 수 있다면 D-최적실험계획 분야에서 획기적인 발전을 기대할 수 있을 것이다.

참 고 문 헌

[1] Atwood, C.L. (1973). Sequences Converging to D-Optimal Designs of Experiments, *The Annals of Statistics*, 1(2), 342-352.

[2] Betsis, D. (1985). Studies in D-Optimal Experimental Design, TRITA-NA_8501, Department of Numerical Analysis and Computing Science, *Royal Institute of Technology*, S-100 44 Stockholm, Sweden.

[3] Edahl, R. and Kortanek, K.O. (1980). A Steepest Descent Clustering Algorithm for Determinant-Maximizing Regression Experimental Designs, *NSF Grant ENG-7825488*, Carnegie-Mellon University.

[4] Fedorov, V.V. (1972). *Theory of Optimal Experiments*, Academic Press, New York and London.

[5] Kiefer. J. and Wolfowitz. J. The equivalence of two Extremum problems. *Canadian Journal of Mathematics* 12, 363(1960)

[6] Tsay, J.Y. (1975). *Linear Optimal Experimental Designs*, University of Cincinnati Medical Center.

[7] Wynn, H.P. (1970) The Sequential Generation of D-Optimum Experimental Designs, *Annals of Mathematical Statistics* 41, 1655-1664.