

## 구조최적설계를 위한 2차계획문제의 효율적인 해법

### An Efficient Solution Algorithm of Quadratic Programming Problems for the Structural Optimization

서경민\* · 류연선\*\*

Seo, Kyung Min · Ryu, Yeon Sun

#### Abstract

Quadratic programming problems(QP) have been widely used as a direction-finding subproblem in the engineering and structural design optimization. To develop an efficient solution algorithm for the QP subproblems, theoretical aspects and numerical behavior of mathematical programming methods that can be used as QP solver are studied and compared. For the solution of both primal and dual QP, Simplex, gradient projection(GRP), and augmented Lagrange multiplier algorithms are investigated and coded.

From the numerical study, it is found that the primal GRP algorithm with potential constraint strategy and the dual Simplex algorithm are more attractive and effective than the others. They have theoretical robustness as well. Moreover, primal GRP algorithm is preferable in case the number of constraints is larger than the number of design variables.

Favorable features of GRP and Simplex algorithm are merged into a combined algorithm, which is useful in the structural design optimization.

#### 요 지

공학 및 구조최적설계에서 광범위하게 이용되고 있는 2차계획문제(QP)의 효율적인 해법을 개발하기 위하여, 먼저 QP의 해법으로 사용가능한 수학적 최적화기법들의 이론적 및 수치적 특성을 비교연구 하였다. 원래 QP 및 쌍대 QP에 대하여 이론적 강건성이 확인된 심플렉스, 경사투영(GRP) 그리고 증대 라그랑지승수 알고리즘의 컴퓨터 프로그램을 작성하고 수치적 수행성이 검토되었다.

연구결과, 잠재제약조건방책을 이용하는 원래 QP의 GRP 알고리즘과 쌍대 QP의 심플렉스 알고리즘이 다른 QP해법에 비하여 효율적이면서 강력한 방법임을 알 수 있었고, 제약함수의 수가 설계변수의 수보다 많을 때는 원래 QP의 GRP 알고리즘이 더욱 효율적이었다.

또 GRP 알고리즘과 심플렉스 알고리즘의 장점을 선별적으로 이용할 수 있는 조합 알고리즘이 제안되었다.

\* 정회원 · 부산수산대학교 해양공학과, 박사과정

\*\* 정회원 · 부산수산대학교 부교수, 해양공학과

# 1. 서론

비선형 계획문제(nonlinear programming problem ; NLP)로 정식화되는 공학적 최적설계문제에 대한 수치적 최적화에 관련하여 현재까지 많은 기법들이 개발되었다. 그 가운데 2차계획문제(quadratic programming problem ; QP)를 풀어서 탐색방향을 결정하는 축차 2차계획법(sequential quadratic programming ; SQP)이 다른 최적화기법에 비하여 효율적임은 널리 알려져 있다.<sup>(1-9)</sup> SQP는 탐색방향 벡터에 관한 QP부분제(QP subproblem)의 구성, QP의 해법, 그리고 탐색방향에 따른 이동거리를 결정하는 3단계로 구성된다. NLP로부터 QP부분제의 구성방법에 관한 연구는 곧 새로운 최적화기법의 개발을 의미하며 이동거리의 결정은 1차원적인 문제로서 그에 관한 많은 연구가 있었다.<sup>(10)</sup>

그러나 QP의 정밀한 해로부터 탐색방향을 결정함으로써 수렴율을 증가시키는 효율적인 QP의 해법에 관한 연구는 주로 특정한 방법에 국한된 실정이므로,<sup>(11)</sup> 각 기법의 계산과정의 비교와 그 이론적 장단점 및 수치적 수행성에 관한 체계적인 비교연구는 대단히 중요한 과제라 하겠다. 따라서 본 연구에서는 QP만을 위한 해법은 물론 일반적인 NLP의 최적화기법들중 QP에 적합한 방법을 도출하고 이들의 효율성을 비교검토하며, 그들의 장점을 취합하고 또 가능한 새로운 기법을 도입함으로써 정확하고도 효율적이며 범용의 경제적인 QP의 해법을 개발하고자 한다.

## 2. QP의 정의

### 2.1 유한차원 최적화의 개요

공학설계에서 설계변수가 유한차원벡터일 때 일반적인 최적화문제는 NLP로 정식화된다.

$$\text{fine } \bar{x} = (\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)^T \quad (1a)$$

$$\text{min. } \bar{f}(\bar{x}) = \bar{f}(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n) \quad (1b)$$

$$\text{s.t. } \bar{g}_j(\bar{x}) = \bar{g}_j(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n) \leq 0; \quad j=1, m \quad (1c)$$

$$\bar{h}_j(\bar{x}) = \bar{h}_j(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n) = 0; \quad j=1, l \quad (1d)$$

여기서  $\bar{x}$ 는 n-차원 설계변수벡터,  $\bar{f}$ 는 설계최적화

과정에서 최소화되어야 할 목적함수,  $\bar{g}$  및  $\bar{h}$ 는 설계요구조건에 따라 주어지는 부등호 및 등호제약함수, 그리고 m 및 l은 각각 부등호 및 등호제약조건의 갯수이다.

식 (1)로 표현되는 최적화문제의 해를 구하는 기법은 크게 직접법(direct method)과 간접법(indirect method)으로 대별된다. 일반적인 최적화문제에 적용가능한 직접법은 문제를 취급하는 방법에 따라 변환법(transformation method)과 원시법(primal method)으로 다시 구분된다. 그중 원시법은 최적화문제의 목적함수 및 제약함수를 근사화한 부분제를 풀어서 탐색방향을 결정하고 그것에 따른 이동거리를 구함으로써 최적해를 구하는 방법이다. 원시법은 변환법에 비하여 높은 수렴율 및 계산상의 이점이 있기 때문에 근래의 수치적 최적화기법에 관한 연구는 주로 원시법을 대상으로 이루어지고 있으며 부분제를 구성하는 방법에 따라 다양한 기법들이 개발되어 왔다. 원시법에서 탐색방향을 결정하기 위하여 반드시 풀어야 할 부분제는 대부분 선형계획문제(linear programming problem ; LP) 또는 QP로 나타내어지며, 근래에는 주로 QP를 사용하는 추세이다.<sup>(7,8)</sup>

### 2.2 원래 QP의 유도

식 (1)로 주어진 NLP를 현재의 설계변수벡터  $\bar{x}^q$ 에서 새로운 설계  $\bar{x}^{q+1}$ 로 진행할 탐색방향벡터를  $x = \bar{x}^{q+1} - \bar{x}^q$ 라 두고  $\bar{x}^q$ 에 관해서 목적함수  $\bar{f}$ 와 제약함수  $\bar{g}$  및  $\bar{h}$ 를 선형 테일러 급수전개하면 n-차원의 탐색방향벡터  $x$ 를 설계변수로 하는 LP로 표현된다. 이 LP의 해는 무계(unbound)일 수 있으므로 이를 방지하기 위하여 설계변수  $x$ 에 제약을 가한다.<sup>(12)</sup> 이 제약조건을 취급하는 방법 중의 하나가 이동거리에 관한 2차제약조건(quadratic step size constraint)이다.

LP에 비선형의 2차제약조건을 부가할 경우 다시 NLP로 되며 이동거리의 상한선택이 어렵기 때문에, 이동거리에 대한 2차제약조건을 목적함수에 포함시키면 쿤-터커조건(Kuhn-Tucker condition)의 동치변형에 의하여 결국 선형제약조건만을 갖는 QP부분제가 구성되며 이를 원래 QP(primal QP)라고 한다.<sup>(11)</sup>

$$\min. f(x) = 0.5x^T A x + a^T x \quad (2a)$$

$$\text{s.t. } g(x) \equiv B^T x + b \leq 0 \quad (2b)$$

$$h(x) \equiv C^T x + c = 0 \quad (2c)$$

여기서 A는 양정성 대칭가중행렬(positive definite symmetric weighting matrix)로서 식 (1)의 NLP에 대한 라그랑지함수(Lagrange function)의 헤세행렬(Hessian matrix)을 근사화한 것으로 생각할 수 있다. 또  $x \in R^n$ ,  $a \in R^n$ 은 목적함수의 경사도벡터( $\nabla f^T$ ), B 및 C는 부등호 및 등호제약함수의 경사도벡터( $\nabla g_i^T$  및  $\nabla h_i^T$ )를 열(column)로 하는  $n \times m$  및  $n \times l$ 행렬, 그리고 b 및 c는 부등호 및 등호제약함수의 값을 성분으로 하는 벡터( $\bar{g}$  및  $\bar{h}$ )로서 현재의 설계  $x^0$ 에서 계산된 값들이다. 또 식 (2a)의 행렬 A가 양정행렬이므로 QP부문제는 볼록계획문제(convex programming problem)가 되며 국부적 최적해(local optimum)를 구하면 그것이 바로 전역적 최적해(global optimum)이다.

### 2.3 쌍대 QP의 유도

일반적으로 설계변수에 대한 목적함수의 최소화 문제를 제약조건에 대응하는 라그랑지승수에 대한 최대화문제로 변환하는 것이 가능하며 이러한 성질을 쌍대성(duality)이라고 한다. 식 (2)의 최적설계 문제 및 설계변수 x를 각각 원래문제(primal problem) 및 원래변수(primal variable)라 하고 라그랑지승수에 관한 최대화문제 및 그때의 설계변수인 라그랑지승수를 각각 쌍대문제(dual problem) 및 쌍대변수(dual variable)라고 한다. 쌍대성을 식 (2)의 원래 QP에 적용하여 쌍대문제를 유도하려면 원래 QP에 대한 최적해의 필요조건을 이용한다. 즉, 식 (2)의 라그랑지함수 L의 원래변수 x에 관한 경사도 벡터가 0이 되어야 한다는 것이다.

$$\nabla L = Ax + a + Bu + Cv = 0 \quad (3)$$

여기서 u 및 v는 각각 식 (2b)의 부등호 및 식 (2c)의 등호제약조건에 대응하는 라그랑지승수이다. 식 (3)를 x에 관하여 풀어서 식 (2)의 원래문제에 대입하고 라그랑지승수 u와 v에 관해 정리하면 라그랑지승수를 설계변수(쌍대변수)로 하는 쌍대 QP(dual QP)로 변환된다.

$$\min. f^d(w) = 0.5w^T D w + d^T w \quad (4a)$$

$$\text{s.t. } g^d(w) \equiv -w_i \leq 0, i = 1, m \quad (4b)$$

여기서,  $w = [u \ v]^T$

$$D = \begin{bmatrix} B^T A^{-1} B & B^T A^{-1} C \\ C^T A^{-1} B & C^T A^{-1} C \end{bmatrix}$$

$$d = \begin{bmatrix} B^T A^{-1} a - b \\ C^T A^{-1} a - c \end{bmatrix}$$

원래문제를 쌍대문제로 변환할 경우, 원래문제의 부등호 및 등호제약조건인 갯수(m+l)가 쌍대변수의 갯수가 되며 원래문제의 부등호제약조건에 대응하는 라그랑지승수에 대한 조건인 식 (4b)가 쌍대문제의 제약조건으로 된다. 일반적으로 구조최적설계를 위한 원시법에서 잠재제약조건방책이 이용될 경우, 설계의 개선이 진행됨에 따라 활동제약조건(active constraint)의 갯수는 전체 제약조건인 갯수보다 매우 적어진다. 따라서 원래변수의 갯수는 설계 전과정을 통하여 변환이 없으나 쌍대변수와 쌍대제약함수의 갯수가 점차적으로 줄어들며 제약함수의 형태도 식 (4b)와 같이 간단해진다.

### 3. QP의 해법

일반적인 최적화 알고리즘 중에서 LP의 해법으로 개발된 심플렉스(Simplex) 알고리즘과 NLP의 직접법 중 변환법에 속하는 ALM-BFGS(augmented Lagrange multiplier and Broydon-Fletcher-Goldfarb-Shanno) 알고리즘, 그리고 원시법에 속하는 GRP (gradient projection) 알고리즘을 식 (2)의 원래 QP 또는 식 (4)의 쌍대 QP에 대한 비교연구의 대상해법으로 선정하였다. 심플렉스 알고리즘은 LP의 해법으로서 절대적인 알고리즘이며, ALM-BFGS 알고리즘은 변환법 중에서 가장 우수한 알고리즘임이 알려져 있고,<sup>(13-16)</sup> GRP 알고리즘은 원시법 중에서 축차적인 부분제를 구성하지 않고도 탐색방향을 결정할 수 있다는 특성을 가지기 때문이다.<sup>(17)</sup> FDM (feasible direction method)이나 GRG(generalized reduced gradient method) 등은 NLP의 해법으로는 그 유용성이 인정되지만, 그 자체에 부분제를 푸는 과정이 또 다시 필요하게 되므로 QP부문제의 해법으로는 비효율적이라 판단되어 비교연구의 대상에서 제외하였다.

### 3.1 심플렉스 알고리즘

심플렉스 알고리즘은 LP의 거의 절대적인 해법이지만, QP의 쿤-터커조건이 선형연립방정식과 유사한 형태로 표현되므로 심플렉스 알고리즘을 변형하여 QP의 해를 구하는데 이용한다. 또 일반적인 LP에서 심플렉스 알고리즘은 2-단계로 구성되지만, QP부문제를 심플렉스 알고리즘으로 풀 수 있는 형태로 변환하면, 최소화할 목적함수가 없기 때문에 1-단계 알고리즘만으로 QP의 해를 구할 수 있다.<sup>(11)</sup>

심플렉스 알고리즘을 사용하여 QP부문제의 해를 구하기 위해서 표준 LP형태로 변환하는 과정을 거친다. 즉, 식 (2b)의 부등호제약조건에 음이 아닌 부가변수벡터  $r$ 를 더하여 등호제약조건만을 갖는 QP부문제로 변환하고 쿤-터커조건을 이용한다. 또 부호제약이 없는 설계변수  $x$ 는 음수가 아닌 값을 가지는 두개의 변수로 분해하여야 하며( $x=e-p$ ,  $e, p \geq 0$ ) 등호제약조건에 대응하는 라그랑지승수  $v$ 도 두개의 벡터로 각각 분해한다( $v=t-z$ ,  $t, z \geq 0$ ). 이러한 준비작정이 완료되면 표준 심플렉스 알고리즘에서 취급할 수 있는 화표(tableau)가 구성된다.

$$\begin{bmatrix} A & -A & B & O & C & -C \\ B^T & -B^T & O & I & O & O \\ C^T & -C^T & O & O & O & O \end{bmatrix} \begin{bmatrix} e \\ p \\ u \\ r \\ t \\ z \end{bmatrix} = \begin{bmatrix} -a \\ -b \\ -c \end{bmatrix} \quad (5)$$

위의 과정에서 부가변수벡터  $r$ 에 대응하는 상보부 가조건  $u^T r = 0$ 은 특별히 취급하여야 하며 이를 위해서는 심플렉스 알고리즘의 피벗요소를 결정할 때 상보쌍(complementary pair)이 둘 다 기저변수가 되지 않게 하된다.<sup>(18)</sup>

쌍대 QP에 심플렉스 알고리즘을 이용할 때, 원래의 QP로부터 쌍대 QP를 만들 필요는 없다. 즉, 원래 QP의 쿤-터커조건으로부터 직접 쌍대 QP의 화표를 구성할 수 있게 된다.<sup>(11)</sup>

$$\begin{bmatrix} -B^T A^{-1} B I - B^T A^{-1} C B^T A^{-1} C \\ -C^T A^{-1} B O - C^T A^{-1} C C^T A^{-1} C \end{bmatrix} \begin{bmatrix} u \\ r \\ t \\ z \end{bmatrix} = \begin{bmatrix} B^T A^{-1} a - b \\ C^T A^{-1} a - c \end{bmatrix} \quad (6)$$

식 (6)의 화표를 상보부가조건을 처리하도록 변형된 1-단계 심플렉스 알고리즘을 사용하여 쌍대변수를 계산하고 이로부터 다시 원래변수를 계산한다. 식 (5)의 원래 QP에 대한 계수행렬에 비하여 식 (6)의 쌍대 QP에 대한 계수행렬의 크기가 항상 작기 때문에, 심플렉스 알고리즘을 사용할 경우에 쌍대 QP가 화표구성 후의 수치계산에서 항상유리함을 쉽게 알 수 있다.

### 3.2 GRP 알고리즘

일반적인 NLP에서 목적함수의 최속강하(steepest descent)벡터를 선형화된 제약함수의 접평면(tangent plane)에 투영시켜서 탐색방향을 구하고 그 탐색방향을 따라 이동거리를 계산하는 과정을 반복함으로써 최적해를 구하는 기법이 GRP 알고리즘이다. 이는 비선형 목적함수와 선형 제약함수를 갖는 최적화문제에 이용될 수 있도록 Rosen에 의해 고안되었으나, Arora와 Haug에 의하여 개량되어 일반적인 NLP를 풀 때 1차원탐색을 하지 않고 이동거리를 결정하는 방법이 제안되었다.<sup>(17)</sup> 또한 GRP 알고리즘은 투영행렬의 계산과정에서 활동제약조건만이 이용되는 잠재제약조건방책(potential constraint strategy)을 사용한다. 따라서 당초의 제약함수의 갯수보다 적은 갯수의 제약함수에 대한 경사도벡터가 최적해의 결정과정에서 계산되므로 다른 최적화기법에 비하여 상대적으로 계산량이 적어진 다.

원래 QP 및 쌍대 QP부문제는 선형 제약함수만을 가지므로 여기에서는 Rosen의 GRP 알고리즘을 선택한다.

먼저 QP목적함수의 최속강하벡터  $f$ 를 활동제약함수의 접평면에 투영시켜 탐색방향  $s^0$ 를 결정한다.

$$s^0 = -[I - N(N^T N)^{-1} N^T] \nabla f(x^0) \quad (7)$$

여기서,  $I$ =단위행렬,  $N=[\nabla h_i \ \nabla g_j]$ ,  $i=1, l$  및  $g_j$ 는 활동제약함수이다. QP에서는 제약함수들이 선형이므로  $N$ 은 상수행렬이 되며, 활동제약함수의 갯수에 따라 그 크기는 변할 수 있다. 한편 쌍대 QP의 경우는 탐색방향의 결정에  $f^0$ 를 사용하며 행렬  $N$ 도 한층 더 쉽게 얻을 수 있다.

탐색방향 \$s^q\$ 상에서 이동거리는 현재 활동이 아닌 새로운 부등호제약함수와 만나도록 결정한다. 탐색 방향이 활동제약함수들의 접평면에 있으므로 등호제약함수는 항상 만족된다. 만약 새로운 부등호 제약함수와 만나기 전에 목적함수를 최소화하는 설계가 존재하면 그곳까지를 이동거리로 하며, 이때는 비제약문제를 푸는 것과 같다. 탐색방향 \$s^q\$ 를 따라 구해질 새로운 설계 \$x^{q+1} = x^q + \alpha s^q\$ 에서 활동제약함수가 아닌 \$g\$ 와 만난다면 이는 이동거리 \$\alpha\$ 에 관한 다음 식을 풀면 된다.

$$g(\alpha) = g(x^q) + \alpha_i \nabla g_i s^q = 0 \quad (8)$$

현재 활동제약함수가 아닌 모든 \$g\_i\$ 에 대하여 각각의 이동거리 \$\alpha\_i\$ 를 식 (8)에서 계산하고 그중 양수이면서 가장 작은 값을 제약함수에 대한 이동거리로 한다.

$$\alpha_M = \min \left\{ \alpha_i > 0 \mid \alpha_i = -\frac{g_i(x^q)}{\nabla g_i(x^q) S^q} \right\} \quad (9)$$

또 \$\nabla f(\alpha\_M) S^q\$ 의 값이 양수이면 \$\alpha = 0\$ 과 \$\alpha = \alpha\_M\$ 사이에 목적함수의 최소값이 존재하므로 비제약문제를 풀어서 \$0 < \alpha^\* < \alpha\_M\$ 인 이동거리 \$\alpha^\*\$ 를 구한다.<sup>(19)</sup>

### 3.3 ALM-BFGS 알고리즘

제약 최적화문제를 비제약 최적화문제로 변환하기 위하여 ALM 알고리즘을 사용하고 변환된 비제약 문제에 BFGS 알고리즘을 이용하여 해를 구하는 것이 ALM-BFGS 알고리즘이다.<sup>(16)</sup> ALM 알고리즘은 라그랑지승수를 반칙계수(penalty parameter)로 확대 사용함으로써 변환법의 또 다른 부류인 SUMT (sequential unconstrained minimization techniques) 알고리즘의 단점과 문제점을 극복하고자 개발되었다. 이 알고리즘의 장점은 반칙계수가 무한대로 되지 않으면서 최적해를 구할 수 있기 때문에 수치적 약조건(numerical ill-conditioning)을 극복할 수 있고, SUMT 알고리즘보다 수렴속도가 빠르다. 그리고 전역적 수렴성(global convergence)을 가지며, 초기치를 유용영역(feasible region)이나, 불용영역(infeasible region)의 제한을 받지 않고 임의로 지정할 수 있는 장점을 가진다.<sup>(22)</sup>

BFGS 알고리즘은 비제약 최적화문제의 목적함수의 헷세행렬의 역행렬을 이전 실행에서 계산된

경사도벡터의 정보만을 이용하여 근사화시켜 비제약 최적화문제의 최적해를 구하는 방법이다. 비제약 목적함수의 경사도벡터만을 이용하여 헷세행렬의 역행렬이 양정성과 대칭성을 만족하도록 근사계산하기 때문에 Newton 알고리즘보다 계산량이 적으면서 Newton 알고리즘과 같이 2차적으로 수렴(quadratically convergent)하는 특성을 갖는다.

QP부문제에 대한 ALM 변환함수(transformation function)는 다음과 같다.

$$T(x, u, v, \gamma_q) = f(x) + \sum_{j=1}^m [u_j \phi_j + \gamma_q \phi_j^2] + \sum_{k=1}^l [v_k h_k(x) + \gamma_q h_k^2(x)] \quad (10)$$

여기서 \$\phi\_j = \max \{g\_j(x), -u\_j/(2\gamma\_q)\}\$, \$\gamma\_q\$ = 반칙계수이며, 라그랑지승수의 수정에는 다음 식을 사용한다.

$$u_j^{q+1} = u_j^q + 2\gamma_q \phi_j^q \quad j=1, m \quad (11a)$$

$$v_k^{q+1} = v_k^q + 2\gamma_q h_k(x) \quad k=1, l \quad (11b)$$

식 (10)의 비제약 목적함수에 대한 BFGS 알고리즘의 \$q\$ 번째 탐색방향벡터는 다음 식으로 결정한다.

$$s^q = -H^q \nabla T(x^q) \quad (12)$$

헷세행렬의 근사역행렬 \$H^q\$ 는 첫번째 실행에서는 단위행렬로 두고(\$H^0 = I\$) 그 다음 실행부터는 다음과 같이 정의한다.

$$H^{q+1} = H^q + E^q \quad (13)$$

여기서 \$E^q\$ 는 대칭수정행렬(symmetric update matrix)로 다음과 같다.

$$E^q = (\delta^T y + y^T H^q y) \delta \delta^T / (\delta^T y)^2 - [H^q y \delta^T + \delta (H^q y)^T] / \delta^T y \quad (14)$$

$$\delta = x^q - x^{q-1}$$

$$y = \nabla T(x^q) - \nabla T(x^{q-1})$$

QP의 경우 ALM 변환함수 \$T\$ 의 헷세행렬을 쉽게 얻을 수 있으나, ALM-BFGS 알고리즘에서는 먼저 변환함수를 구성하고 반복해법에 의해 BFGS 알고리즘을 수행하므로 식 (12), (13)을 이용하는 것이

효율적이다. 또 원래 QP 및 쌍대 QP에 공통적으로 ALM-BFGS 알고리즘을 이용가능하다.

### 3.4 해법의 이론적 비교 고찰

공학적 최적설계문제에서 사용되는 부문제로서의 QP는 원래의 NLP에 비하여 규모가 작고 볼록계획 문제로 만들어지며 반드시 전역적 최적해가 존재하므로 해를 구하기는 쉽다. 그러나 이는 최적화 과정에서 여러번 반복하여 계산되어야 하고, 그 해가 원래의 NLP의 탐색방향벡터로 사용되므로 효율적이며 정밀한 계산이 요구된다. 본 연구에서 비교 대상이 되는 QP부문제의 해법 알고리즘은 각기 다른 이론적 특성을 갖는다. 심플렉스 알고리즘은 피벗팅과 소거법을 반복하여 최적해를 구하며, GRP와 ALM-BFGS는 직접법으로서 초기치를 이용하여 반복적으로 탐색방향과 이동거리를 결정함으로써 최적해를 구한다. 또 ALM-BFGS 알고리즘은 제약조건을 갖는 최적화문제를 비제약 최적화문제로 변환하여 순차적으로 최적해를 구하는 변환법에 속하며 GRP 알고리즘은 주어진 최적화문제로부터 직접 최적해를 구하는 원시법에 속한다.

일반적으로 최적화 알고리즘은 그 이론적 특성이 컴퓨터 프로그램에 의한 수치적 거동에서 그대로 재현되지 않을 수 있다.<sup>(6)</sup> 예를 들어 이론적으로는 전역적 수렴성이 보장된 알고리즘이라 하더라도 수치적 계산과정에서는 최적해에 수렴하지 않을 수 있다는 것이다. 따라서 이상의 각 알고리즘에 대한 이론적 고찰을 기초로 하여 수치적 실험예와 함께 각 QP부문제의 해법의 효율성을 비교판단하여야 하며, 그 결과를 토대로 하여 알고리즘의 개발 및 기존 알고리즘의 개선과 조합 알고리즘의 개발방향이 제시되어야 한다.

알고리즘의 특성비교를 위해 고려된 항목들은 전역적 수렴성, 잠재제약조건방책의 사용여부, 비제약 최적화의 도입여부, 헷세행렬의 상태수(condition number), 해의 수렴여부를 판단하는 허용오차(allowable tolerance), 초기치의 필요성 등이다.

(1) 전역적 수렴성 : 전역적 수렴성이 보장된 알고리즘이면 항상 최적해를 구할 수 있다. 따라서 알고리즘이 전역적 수렴성을 가지는가 하는 것은 대단히 중요한 요인이다. 특히 QP부문제가 볼록계획문제이므로 임의의 국부적 최적해는 곧 전역적 최적해라고 할 수 있기 때문에 알고리즘의 전역적 수렴성은 더욱 바람직하다.

(2) 잠재제약조건방책 : 알고리즘에서 잠재제약조건방책을 사용하면 일반적으로 최적화 과정에 관계하는 부등호제약함수의 수가 적어지므로 계산량이 감소된다.

(3) 비제약 최적화 : 알고리즘이 비제약 최적화문제를 취급할 수 있어야 한다. 이것이 QP의 해법과정에 포함되지 않으면 QP부문제를 풀기 전에 비제약 최적화문제를 별도로 처리해야 한다.

(4) 헷세행렬의 상태수 : 목적함수의 헷세행렬의 상태수에 따라 알고리즘의 수렴성이 가급적 영향을 받지 않아야 한다. 탐색적인 기법을 사용할 경우, QP부문제는 제약함수 보다는 목적함수의 형태에 따라 계산량이 증가할 수 있기 때문이다.

(5) 허용오차 : 알고리즘이 허용오차내에서 수렴여부를 판단한다면 최적해에 포함된 오차를 알 수 있다.

(6) 초기치 : 일반적인 공학적 최적설계문제에서 수치알고리즘의 반복회수는 초기치에 따라 크게 영향을 받는다. 그러나 QP와 같이 특수한 형태의 문제에 대해서는 수치적 실험이 필요하다. 이상의 이

표 1. Theoretical aspect of algorithms

Algorithm	Simplex	GRP	ALM-BFGS
Global Convergency	yes	no	yes
Potential Constraint Strategy	no	yes	no
Solution of Unconstrained Problem	no	yes	yes
Influence of Condition Number	no	yes	yes
Use of Allowable Tolerance	no	yes	yes
Use of Initial Guess	no	yes	yes

표 2. Computer programs of QP solver

Solution	Computer program	
	For Primal QP	For Dual QP
ALM-BFGS	QPALM	DQPALM
GRP	QPGRP	DQPGRP
Simplex	QPSPX	DQPSPX

론적 특성을 비교대상의 알고리즘에 대해 요약하면 표 1과 같다.

4. 수치예의 적용 및 고찰

심플렉스, GRP 및 ALM-BFGS 알고리즘을 원래 QP 및 쌍대 QP부문제를 풀기 위한 6개의 프로그램으로 각각 작성하였으며 이들은 표 2에 표시되었다. 작성된 프로그램을 사용하여 각기 다른 특성을 가진 예제를 풀어 봄으로써 각 알고리즘의 수치적 수행성과 원래 QP와 쌍대 QP에 대한 각 알고리즘의 유용성을 비교하고자 한다.

4.1 수치예

예제의 선택에서 고려된 사항은 수치적 수행성에 주된 영향을 줄 수 있는 헷세행렬 A의 상태수, 설계변수의 갯수, 부등호제약조건의 갯수(B<sup>T</sup> 및 b), 등호제약조건(C<sup>T</sup> 및 c)의 유무 등이다. 예제 1~3은 소규모의 QP로서 목적함수의 헷세행렬의 상태수와 부등호 및 등호제약조건의 갯수와 유무에 따른 프로그램의 수행성을 점검하기 위한 것이다. 여기서 x\*는 해석적 최적해(원래변수)이며, u\* 및 v\*는 해석적으로 구한 라그랑지 승수벡터(쌍대변수)로서 부등호 및 등호제약조건에 각각 대응한다.

예제 1 :

$$A = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}, a = \begin{bmatrix} -2 \\ -2 \end{bmatrix}$$

$$B^T = \begin{bmatrix} 1 & 1 \\ -1 & 0 \end{bmatrix}, b = \begin{bmatrix} -4 \\ 2 \end{bmatrix}$$

$$x^* = [2 \ 1]^T, u^* = [0 \ 2]^T$$

예제 2 :

$$A = \begin{bmatrix} 4 & -6 \\ -6 & 18 \end{bmatrix}, a = \begin{bmatrix} -18 \\ 9 \end{bmatrix}$$

$$B^T = \begin{bmatrix} 1 & -2 \\ 4 & -3 \end{bmatrix}, b = \begin{bmatrix} -10 \\ -20 \end{bmatrix}$$

$$x^* = [6.3 \ 1.73333]^T, u^* = [0 \ 0.8]^T$$

예제 3 :

$$A = \begin{bmatrix} 4 & 2 & -1 \\ 2 & 4 & -0.8 \\ -1 & -0.8 & 2 \end{bmatrix}, a = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$B^T = \begin{bmatrix} -1.3 & -1.2 & -1.1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, b = \begin{bmatrix} 1.15 \\ -0.7 \\ -0.7 \\ -0.7 \end{bmatrix}$$

$$C^T = [1 \ 1 \ 1 \ 1], c = -1$$

$$x^* = [0.24151 \ 0.18408 \ 0.57432]^T, u^* = [0 \ 0 \ 0 \ 0]^T,$$

$$v^* = -0.75994$$

예제 4 : Rosen-Suzuki 문제를<sup>(10)</sup> NLP의 최적해에서 선형화하고 이미 알고 있는 라그 란지승수로부터 양정성 가중행렬을 구성하여 QP부문제를 만든다. 이 때 QP부문제의 해로써 구해진 탐색방향벡터가 영벡터가 되므로 x\*=0이며, 그 때의 QP제약 함수의 값은 NLP의 최적해에서의 제약함수의 값과 같아야 한다. 이 예제는 초기치가 필요없는 심플렉스 알고리즘과 초기치를 필요로 하는 GRP 및 ALM-BFGS의 초기치에 따른 수치해석적 특성을 비교하고자 선정하였다.

$$A = \begin{bmatrix} 12 & 0 & 0 & 0 \\ 0 & 8 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 4 \end{bmatrix}, a = \begin{bmatrix} -5 \\ -3 \\ -13 \\ 5 \end{bmatrix}$$

$$B^T = \begin{bmatrix} 1 & 1 & 5 & -3 \\ -1 & 4 & 4 & -5 \\ 2 & 1 & 4 & -1 \end{bmatrix}, b = \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix}$$

예제 5-8 : 실제적인 구조최적설계문제로서 3부재 트러스의 설계최적화문제(참고문헌 11, pp.41-45, pp. 484-489)의 탐색방향을 결정하는 예제들이다.<sup>(11)</sup>  
3부재 트러스 최적화를 위한 NLP는 부재의 단

표 3. Constraint coefficients of examples 5-8

Example No.	$B^T$		b
Example 5 ( $\epsilon=0.1$ )	$-1.30316 \times 10^{-1}$	$-4.61873 \times 10^{-2}$	$-1.35620 \times 10^{-2}$
Example 6 ( $\epsilon=0.7$ )	$-3.46405 \times 10^{-1}$ $-3.46405 \times 10^{-2}$ $-6.51578 \times 10^{-2}$	$-4.89890 \times 10^{-1}$ $-4.89890 \times 10^{-1}$ $-2.30937 \times 10^{-2}$	$-6.16637 \times 10^{-1}$ $-6.16637 \times 10^{-1}$ $-5.06781 \times 10^{-1}$
Example 7 ( $\epsilon=1.0$ )	$-1.63297 \times 10^{-2}$ $-1.73202 \times 10^{-2}$ $-8.16484 \times 10^{-3}$ $1.00000 \times 10^{-2}$ 0	$-2.30937 \times 10^{-2}$ $-2.44945 \times 10^{-2}$ $-1.15468 \times 10^{-2}$ 0 $1.00000 \times 10^{-2}$	$-8.19281 \times 10^{-1}$ $-8.08319 \times 10^{-1}$ $-9.09640 \times 10^{-2}$ $-9.36000 \times 10^{-1}$ $-9.67000 \times 10^{-1}$
Example 8 ( $\epsilon=100$ )	$6.49967 \times 10^{-2}$ $3.24984 \times 10^{-2}$ $-1.00000 \times 10^1$ 0	$-4.61873 \times 10^{-2}$ $-2.30937 \times 10^{-2}$ 0 $-1.00000 \times 10^1$	$-1.26356 \times 10^0$ $-1.13178 \times 10^0$ $-6.30000 \times 10^{-1}$ $-3.20000 \times 10^1$

표 4. State of example problems

Problem	NDV	NE	NI	CD
Example 1	2	0	2	1.0
Example 2	2	0	2	11.4
Example 3	3	1	4	4.0
Example 4	4	0	3	3.0
Example 5	2	0	1	1.0
Example 6	2	0	4	1.0
Example 7	2	0	9	1.0
Example 8	2	0	13	1.0

표 5. Results of example 1

Program	CPU time (sec)	$\ x^* - x\ $	$\ w^* - w\ $
QPALM	24.76	$2.236 \times 10^{-5}$	$7.420 \times 10^{-3}$
QPGRP	3.35	0	0
QPSPX	3.12	0	0
DQPALM	29.22	$8.062 \times 10^{-5}$	$3.614 \times 10^{-4}$
DQPGRP	55.64	$1.650 \times 10^{-3}$	$6.580 \times 10^{-3}$
DQPSPX	0.94	0	0

표 6. Results of example 2

Program	CPU time (sec)	$\ x^* - x\ $	$\ w^* - w\ $
QPALM	58.99	0	$7.380 \times 10^{-3}$
QPGRP	4.34	0	0
QPSPX	3.89	0	0
DQPALM	43.77	$2.463 \times 10^{-1}$	$3.206 \times 10^0$
DQPGRP	495.20	$6.014 \times 10^{-3}$	$7.823 \times 10^{-2}$
DQPSPX	0.94	0	0

면적을 설계변수, 구조물의 중량을 목적함수로 하고 각 부재의 응력에 대한 제약조건과 설계변수 제약 조건으로 구성된다. 구조의 대칭성을 가정하여 설계변수는 2개로 한다. 또 3개의 하중조건에 대해 3개의 부재에서 9개의 응력제약조건과 설계변수의 상한 및 하한에 관한 4개의 설계변수 제약조건이 있다. NLP 알고리즘에서 잠재제약조건방책이 사용되면  $\epsilon$ -활동계수의 크기에 따라 QP부문제의 제약함수의 갯수가 달라진다.

여기서는 3부재 트러스 설계문제를 NLP의 최적해 근방 ( $\bar{x}_1=6.4, \bar{x}_2=3.3$ )에서 선형화하고 활동제약조건의 범위  $\epsilon=0.1, 0.7, 1.0, 100$ 에 대하여 각각 1개, 4개, 9개, 그리고 13개의 부등호제약함수를 가지는

4개의 QP부문제를 구성한다. 이 예제는 설계변수의 수와 QP 목적함수가 일정할 때 제약함수의 수에 따른 각 알고리즘의 수치적 수행성을 비교하기 위한 것이다. 예제 5-8에 사용된 QP 목적함수에서  $A=I$ ,



표 7. Results of example 3

Program	CPU time (sec)	$\ x^* - x\ $	$\ w^* - w\ $
QPALM	80.31	$1.158 \times 10^{-4}$	$1.200 \times 10^{-5}$
QPGRP	18.50	$5.477 \times 10^{-5}$	0
QPSPX	21.04	$5.477 \times 10^{-5}$	0
DQPALM	88.98	$1.939 \times 10^{-1}$	$9.699 \times 10^{-1}$
DQPGRP	6.64	$5.477 \times 10^{-5}$	0
DQPSPX	6.97	$5.477 \times 10^{-5}$	0

표 8. Results of example 4

Program	CPU time (sec)	$\ x^* - x\ $	$\ w^* - w\ $
QPALM	-	-	-
QPGRP	5.04	0	0
QPSPX	15.64	0	0
DQPALM	-	-	-
DQPGRP	-	-	-
DQPSPX	2.84	0	0

표 9. CPU time of examples 5 8

Program	Example 5	Example 6	Example 7	Example 8
QPALM	10.20	17.25	28.89	38.12
QPGRP	3.07	3.50	4.17	4.77
QPSPX	1.80	8.83	40.80	92.39
DQPALM	3.51	73.42	194.25	610.76
DQPGRP	2.25	8.02	37.90	92.70
DQPSPX	0.37	3.67	25.32	67.55

$a = [2.82843 \ 1]^T$ 이며 제약함수계수  $B^T$  및  $b$ 는 표 3과 같다.

#### 4.2 수치예의 특성

예제들의 상이한 특성을 요약하면 표 4와 같다. 여기서 NDV=설계변수의 수, NE=등호제약조건의 수, NI=부등호제약조건의 수, CD=행렬 A의 상태수이다. 예제 1과 2는 상태수에서 많은 차가 있으며 예제 2가 비교적 약조건인 경우이다. 예제 3은 등호제약조건을 포함하며 NI>NDV이고, 예제 4에서는 NDV>NI이다. 예제 5-8은 상태수가 1.0으로 대단히 양호하며 부등호제약조건의 수에 있어서만 다양한 차가 있는 문제들이다.

#### 4.3 결과 및 고찰

예제문제에 대한 각 프로그램의 수행 결과를 표 5-9에 나타내었다. 예제 1-4에 대해서는 각 알고리즘의 효율성과 해의 정확성을 비교하기 위해 IBM PC/XT에서의 실행시간, 해석해  $x^*$ 와 수치해  $x$ 의 차의 norm, 그리고 라그랑지승수의 해석해  $w^* = [u^* \ v^*]$ 와 수치해  $w = [u \ v]$ 의 차의 norm을 제시하였다. 예제 5-8의 경우에는 6개의 프로그램으로부터 모두 동일한 최적해와 라그랑지승수가 구해졌기 때문에, 부등호제약함수의 갯수의 변화에 따른 각 알고리즘의 실행시간만을 비교하였다. 원래 QP의 프로그램(QPALM, QPGRP, QPSPX)에서는 최적해와 라그랑지승수를 계산할 때까지를 실행시간으로, 쌍대 QP의 프로그램(DQPALM, DQPGRP, DQPSPX)에서는 쌍대문제로의 변환구성에서 쌍대변수의 최적해 계산 및 원래변수의 산정까지를 실행시간으로 하였다.

GRP 및 ALM-BFGS의 초기치는 모든 예제에서 영벡터로 두고 실행하였다. 특히 QPGRP에서는 등호제약조건이 있는 경우 그것이 만족되도록 프로그램의 내부에서 다시 초기치를 계산한다. 그리고 ALM-BFGS의 반칙계수는  $\gamma_0 = 2^n$ 로 두었다.

모든 프로그램이 예제 1-3의 해를 구했으나, 예제 4에서는 QPGRP, QPSPX 그리고 DQPSPX만이 해를 구할 수 있었다. 예제 4는 알고리즘이 초기치의 영향을 받는지의 여부를 비교하기 위하여 최적해를 초기치로 두고 수행한 것이다. 이론적으로는 전역적 수렴성을 가진 ALM-BFGS 알고리즘으로 작성된 QPALM과 DQPALM에서 수렴해를 구하지 못했음은 특기할 만한 사실이다. 설계변수의 갯수와 목적함수가 동일하고 제약함수의 갯수를 변화시킨 예제 5-8에서는 모든 알고리즘으로부터 동일한 해를 구할 수 있었다. 또 모든 예제에 대하여 QPGRP, QPSPX 및 DQPSPX의 실행시간이 일반적으로 적음을 알 수 있고 따라서 이들의 수치수행상의 신뢰성을 확인할 수 있었다.

예제 1과 2에 대한 수치계산 결과를 보면, 일반적으로 헷세행렬의 상태수가 커지면 프로그램의 실행시간이 증가함을 알 수 있으며 이는 이론적 예측과 일치한다. 실행시간에 있어서 DQPGRP가, 그리고 해의 정확도에 있어서 DQPALM이 상태수의 변화에

민감하며, QPGRP와 QPSPX 및 DQPSPX는 상태수가 증가하여도 해의 정확성 및 실행시간에서 큰 영향을 받지 않았다. 등호제약조건이 있는 예제 3의 경우, 심플렉스와 GRP 알고리즘에서는 쌍대 QP (DQPGRP, DQPSPX)가 유리하지만, ALM-BFGS 알고리즘에서는 쌍대 QP(DQPALM)가 약간 불리하다. 최적해를 초기치로 사용한 예제 4에서는 QPGRP, QPSPX 그리고 DQPSPX가 초기치의 영향을 받지 않고 해를 구할 수 있었다.

예제 5-8의 실행시간을 살펴보면 일반적으로 제약함수의 증가에 따라 동일 최적해를 얻는데 필요한 실행시간이 증가함을 알 수 있다. 특히 부등호제약함수의 갯수가 증가하면 쌍대변수의 갯수도 증가하므로 쌍대 QP의 실행시간이 한층 더 큰 비율로 증가하였다. 또 심플렉스 알고리즘에서는 잠재제약조건방책을 사용하지 않으므로 제약함수의 증가에 따라 잠재제약조건방책을 사용하는 QPGRP에 비하여 높은 비율로 실행시간이 증가하였으며, QPSPX가 DQPSPX의 실행시간보다 항상 많이 소요된다.

수치예의 실행결과로부터 얻을 수 있는 결론은 QPGRP와 DQPSPX가 QP의 해법으로서 효율적이며 신뢰할만하다는 것이다. 또 QP부문제의 제약함수의 갯수가 설계변수의 갯수보다 크거나 같은 경우 QPGRP가 DQPSPX보다 효율적이다.

## 5. 조합 알고리즘의 제안

이론적 특성의 고찰과 수치적 수행성을 비교해 보면, 6개의 프로그램중에서 QPALM 및 DQPALM이 가장 비효율적이었으며 초기치의 영향을 받고, DQPGRP는 수렴특성의 변동이 대단히 크게 나타났다. 또 문제의 규모 및 형태에 관계없이 모든 예제에서 QPSPX가 DQPSPX보다 많은 계산시간이 소요된다. QPGRP는 모든 예제에서 안정적으로 수렴하며 잠재제약조건방책을 사용하므로, 다른 프로그램에 비하여 제약조건의 갯수의 변화에 가장 적게 영향을 받는다. 설계변수의 갯수가 일정하고 제약조건의 갯수가 증가할 때 제약조건의 갯수가 설계변수의 갯수보다 작은 경우에는 DQPSPX가, 반대의 경우에는 QPGRP가 효율적이었다.

공학적 최적설계문제에 SQP와 같이 QP부문제를 풀어서 탐색방향을 결정하는 알고리즘을 사용할 때,

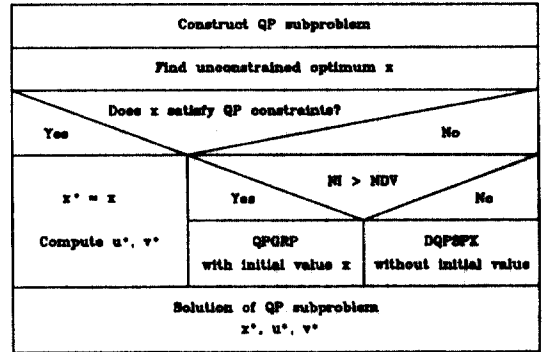


그림 1. Flow Diagram of Combined QP Algorithm

순차적으로 구성된 QP부문제의 일반적인 상태는 NLP의 해가 초기설계와 최적해 근방에 있는 두가지 경우로 분류할 수 있다. 설계초기에는 보통 활동제약조건의 갯수가 설계변수의 갯수보다 많으며 QP부문제의 양정성 가중행렬의 상태수의 변동이 심하다. 설계가 최적해 근방으로 진행함에 따라 활동제약조건의 갯수가 설계변수의 갯수보다 적어지며 상태수도 안정되는 경향이 있다. 이때에는 QP부문제에서 구해질 탐색방향벡터가 영벡터에 가까워지므로 QP의 초기치와 최적해가 거의 일치하게 된다.

이와 같은 최적설계문제의 특성과 QP해법의 이론적 및 수치적 비교고찰에 근거하여, QP부문제에 효율적으로 이용될 수 있는 조합알고리즘을 구성하면 그림 1과 같다.

## 6. 요약 및 결론

공학 및 구조최적설계 과정에서 반복적으로 이용되는 QP부문제의 효율적인 해법을 개발하기 위하여 기존의 최적화기법 중 QP해법에 사용가능한 ALM-BFGS 알고리즘, GRP 알고리즘 및 심플렉스 알고리즘에 대한 비교연구를 수행하였다. QP부문제의 구성과정에서는 원래 QP와 쌍대 QP를 아울러 고려하였다. 알고리즘의 이론적 특성으로는 전역적 수렴성, 잠재제약조건방책의 사용여부, 비제약 최적화의 여부, 목적함수의 헷세행렬의 형태, 허용오차, 초기치의 필요성 여부등을 비교하였다. 수치적 수행성의 비교를 위해 각 알고리즘을 원래 QP 및 쌍대 QP 해석용 프로그램(QPALM, DQPALM, QPGRP, DQPGRP, QPSPX, DQPSPX)으로 작성하고, 다양한

특성을 갖는 예제를 이들을 이용하여 풀었다. 또 이론적 고찰 및 수치예에 대한 수행성을 토대로 하여 강건한 조합 알고리즘도 아울러 제시하였다. 본 연구의 결론을 요약하면 다음과 같다.

(1) 공학 및 구조최적설계과정에서 이용되는 QP 부문제의 해법으로는 원래 QP에 대한 GRP 알고리즘과 쌍대 QP에 대한 심플렉스 알고리즘이 효율적이며, 다른 기법에 비해 신뢰성이 높다.

(2) 제약함수의 갯수가 설계변수의 갯수보다 많은 QP부문제의 해법에 ALM-BFGS 및 GRP 알고리즘을 사용하려면 원래 QP를 직접 푸는 것이 효율적이며, 특히 잠재제약조건방책을 사용하는 GRP 알고리즘이 더욱 유용하다.

(3) 심플렉스 알고리즘을 QP부문제에 사용할 때는 원래 QP보다 쌍대 QP로 변환하는 것이 항상 효율적이다.

(4) 원래 QP 및 쌍대 QP의 모든 경우에서 QP 목적함수의 비제약 최적해를 먼저 구하는 것이 바람직하다. 심플렉스 알고리즘이 가용영역 내부의 해를 구하지 못하는 단점과 비제약 최적화문제에 대한 GRP 알고리즘의 비효율성이 보완될 수 있기 때문이다.

(5) 원래 QP의 GRP 알고리즘과 쌍대 QP의 심플렉스 알고리즘이 가장 유망한 QP부문제의 해법임을 확인하고, 이들의 장점을 최대한 활용할 수 있도록 효율적인 조합 알고리즘을 구성하였다.

## 감사의 글

이 연구는 한국과학재단의 1990-91년도 일반기초 연구과제(901-1301-010-2) 연구비지원에 의해 수행되었으며, 이에 재단당국에 감사한다.

## 참 고 문 헌

1. Arora, J.S., Lim, O.K., *Computational Enhancements of Pshenichny's Algorithm for Constrained Optimization*, The University of Iowa, 1984.
2. Belegundu, A.D. and Arora, J.S., "A Recursive Quadratic Programming Method with Active Set Strategy for Optimal Design", *Int. J. for Num. Meth. in Engrg.*, Vol.20, 1984, pp.803-816.

3. Gabriele, G.A. and Beltracchi, T.J., "An Investigation of Pshenichny's Recursive Quadratic Programming Method for Engineering Optimization", *J. of Mechanisms, Transmissions, and Automation in Design*, Vol.29, 1987, pp. 248-256.
4. Lim, O.K. and Arora, J.S., "An Active Set RQP Algorithm for Engineering Design Optimization", *Computer Methods in Applied Mechanics and Engineering*, Vol.57, 1986, pp.51-65.
5. Thanedar, P.B., Arora, J.S., and Tseng, C.H., "A Hybrid Optimization Method And Its Role in Computer-Aided Design", *Computers and Structures*, Vol.23, No.3, 1986, pp.305-314.
6. Thanedar, P.B., Arora, J.S., Tseng, C.H., Lim, O. K., and Park, G.J., "Performance of Some SQP Algorithms on Structural Design Problems", *Int. J. for Num. Meth. in Engrg.*, Vol.23, 1986, pp.2187-2203.
7. Tseng, C.H. and Arora, J.S., "On Implementation of Computational Algorithms for Optimal Design 1: Preliminary Investigation", *Int. J. for Num. Meth. in Engrg.*, Vol.26, 1988, pp.1365-1382.
8. Tseng, C.H. and Arora, J.S., "On Implementation of Computational Algorithms for Optimal Design 2: Extensive Numerical Investigation", *Int. J. for Num. Meth. in Engrg.*, Vol.26, 1988, pp.1383-1402.
9. Vanderplaats, G.N., "An Efficient Feasible Directions Algorithm for Design Synthesis", *AIAA J.*, Vol.22, No.11, 1984, pp.1633-1640.
10. Luenberger, D.G., *Linear and Nonlinear Programming*, Addison-Wesley, 1984.
11. Arora, J.S., *Introduction to Optimum Design*, McGraw-Hill, 1989.
12. Vanderplaats, G.N., *Numerical Optimization Techniques for Engineering Design*, McGraw Hill, 1984.
13. Belegundu, A.D. and Arora, J.S., "A Computational Study of Transformation Methods for Optimum Design", *AIAA J.* Vol.22, No.4, 1984, pp.535-542.
14. Belegundu, A.D. and Arora, J.S., "A Study of Mathematical Programming Methods for Structural Optimization. Part I: Theory", *Int. J. for Num. Meth. in Engrg.*, Vol.21, 1984, pp.1583-1599.
15. Belegundu, A.D. and Arora, J.S., "A Study of Mathematical Programming Methods for Structural Optimization. Part II: Numerical Results", *Int. J. for Num. Meth. in Engrg.*, Vol.21, 1984, pp.1601-1623.

16. 서경민, 류연선, 류청로, "ALM-BFGS 알고리즘을 이용한 혼성방파제의 최적단면설계에 관한 연구", 대한토목학회 논문집, 1991(투고중).
17. Haug, E.J. and Arora, J.S., *Applied Optimal Design*, John-Wiley & Sons, 1979.
18. Reklaitis, G.V., Ravindran, A., and Ragsdell, K.M., *Engineering Optimization*, John-Wiley & Sons, 1983.
19. Fox, R.L., *Optimization Methods for Engineering Design*, Addison-Wesley, 1971.

(接受 : 1991. 12. 2)