Multiplier methods for engineering optimization — **Source link**

Jasbir S. Arora, A. I. Chahande, J. K. Paeng

Institutions: University of Iowa

Related papers:

- An augmented lagrangian treatment of contact problems involving friction

- A mixed formulation for frictional contact problems prone to Newton like solution methods

- Multiplier and gradient methods

- Contact Problems in Elasticity: A Study of Variational Inequalities and Finite Element Methods

- Optimization of large structures subjected to dynamic loads with the multiplier method

Multiplier methods for engineering optimization

J. S. Arora, A. I. Chahande, J. K. Paeng

MULTIPLIER METHODS FOR ENGINEERING OPTIMIZATION

J. S. ARORA, A. I. CHAHANDE AND J. K. PAENG

Optimal Design Laboratory, College of Engineering, The University of Iowa, Iowa City, IA 52242, U.S.A.

SUMMARY

Multiplier methods used to solve the constrained engineering optimization problem are described. These methods solve the problem by minimizing a sequence of unconstrained problems defined using the cost and constraint functions. The methods, proposed in 1969, have been determined to be quite robust, although not as efficient as other algorithms. They can be more effective for some engineering applications, such as optimum design and control of large scale dynamic systems. Since 1969 several modifications and extensions of the methods have been developed. Therefore, it is important to review the theory and computational procedures of these methods so that more efficient and effective ones can be developed for engineering applications. Recent methods that are similar to the multiplier methods are also discussed. These are continuous multiplier update, exact penalty and exponential penalty methods.

1. INTRODUCTION AND MOTIVATION

This paper presents a review of the multiplier methods for optimum design of engineering systems. These are also called the *augmented Lagrangian methods*, so the two names will be used interchangeably. The basic idea of the methods is to transform the given constrained problem into a sequence of unconstrained problems. The functional for the unconstrained problem is defined using the cost and constraint functions of the original constrained problem and certain multipliers for the constraints. The sequence of solution points for the unconstrained problems converges to the solution of the original problem. This idea is very attractive for several engineering applications, especially for dynamic response and optimal control problems.[69] However, the methods need to be carefully studied so that efficient computational procedures for such applications can be developed. This then is the basic purpose of the present paper—to study various multiplier methods with respect to their computational procedures and rate of convergence, and discuss their applicability to engineering problems.

A general framework for efficient use of multiplier methods for optimal design of structural and mechanical systems under static loads was presented by Belegundu and Arora.[8-10] It was shown that the adjoint variable method[4,47] for calculating the gradient of the transformed functional does not require gradients of the individual functions. As a result, the unconstrained minimization steps could be performed efficiently. This advantage becomes much more effective for many engineering applications where most constraints are implicit functions of the design variables. For example, many constraints in dynamic response and control optimization problems are functions of the nodal displacements that are in turn dependent on design variables and time.

1

This implicit nature of the constraints makes computation of the functions and their gradients very expensive. In primal methods, the gradient of each pointwise dynamic constraint at all the local maximum points is required in calculating the search direction.[49-52] This requires large computational effort. But in the multiplier methods, all pointwise implicit constraints are summed up and integrated over the time interval. That is, they are collapsed into one equivalent constraint functional.[69] Therefore, the gradient of only one functional is needed in search direction calculation.

The original multiplier method requires exact unconstrained minimization at each step. This is numerically impossible, as it may require many iterations and, therefore, many function evaluations. However, there are variations of the methods that work quite well with inexact unconstrained minimization at each step. Even then it can result in too much computational effort for large scale problems. Therefore, it may be desirable to quit unconstrained minimization after a fixed number of iterations, e.g. 1 to $2n$, where n is the number of design variables. The effect of inexact unconstrained minimization on the multiplier methods needs to be analysed and understood. This is done in Section 3.

The performance of the multiplier methods also depends quite heavily on the rate of convergence of the unconstrained minimization algorithm used. Therefore, it is important to use a good unconstrained minimization method. Considerable advances have recently been made in unconstrained minimization techniques. These advances will be described in a separate paper.

An overview of the multiplier methods is presented in Section 2. The steps of a general algorithm that need to be investigated to improve its performance are identified. Fundamentals of the methods are described in Section 3. Augmented functionals are defined, necessary and sufficiency conditions are given, duality is discussed, and an implementable algorithm is stated. In Section 4, the continuous update methods that allow the multipliers to be updated more often are described. These methods have potential for engineering applications that needs to be exploited. The exact penalty methods are discussed in Section 5. The exponential penalty methods are described in Section 6. They also need to be investigated for engineering applications.

2. OVERVIEW OF MULTIPLIER METHODS

Multiplier methods were originally proposed by Hestenes[48] and Powell.[74] They were also proposed a year later by Haarhoff and Buys.[41] A review of the literature on these methods can be found in the survey papers and books by Rockafellar,[76] Fletcher,[33] Pierre and Lowe,[70] Bertsekas[16] and Powell.[75] Local convergence rate with both increasing and finite penalty, and the effect of approximate unconstrained minimization are discussed by Buys.[20] Polak and Sangiovanni-Vincentelli,[71] Poljak and Tretyakov[73] and Bertsekas.[19]

To discuss the methods and their basic ideas we shall consider the following general equality–inequality constrained problem:

Problem P. Find $x \in R^n$ to minimize a cost function $f(x)$ subject to

equality constraints:
$$g_i(x) = 0; \quad i = 1, l \tag{1}$$

inequality constraints:
$$g_i(x) \leqslant 0; \quad i = l + 1, m \tag{2}$$

Many concepts and procedures can be discussed using only equality constraints. This will be called the equality constrained problem (ECP). When only inequality constraints are considered, l is zero in (1) and the corresponding problem will be called ICP.

It has been shown clearly in many journal articles and books that Problem P is quite general and can be used to model a wide variety of engineering optimization problems (Haug and

Arora,[47] Arora and Thanedar,[5] Arora,[2,3] and many references cited therein). One difficulty with engineering applications is that many of the functions of the problem depend implicitly on the design variables. This makes their gradient evaluation computationally tedious and expensive. Some of the multiplier methods can help alleviate this difficulty as they require the gradient of only one augmented functional. This gradient can be evaluated efficiently using the adjoint variable method[4] without requiring gradients of individual constraints.[8,69] This is an important aspect of these methods for engineering applications that will be emphasized throughout the paper. It will be seen that the multiplier methods requiring gradients of individual constraints are not suitable for engineering applications.

Since the *Lagrangian* for the Problem P will be referred to and used throughout, we define it as

$$L(\mathbf{x}, \mathbf{u}) = f(\mathbf{x}) + \sum_{i=1}^{m} u_i g_i(\mathbf{x}) \tag{3}$$

where $\mathbf{u}$ is a vector of Lagrange multipliers. The multiplier methods employ augmented Lagrangians in which some penalty terms involving constraints are added to the ordinary Lagrangian in (3). The methods can also be viewed as an extension of the primal–dual methods and penalty function methods. This will be explained in the next section. The Lagrange multipliers and certain penalty parameters for each constraint are used in constructing a *transformed functional* that, in general, can be written as

$$\Phi(\mathbf{x}, \mathbf{u}, \mathbf{r}) = f(\mathbf{x}) + P(\mathbf{g}(\mathbf{x}), \mathbf{u}, \mathbf{r}) \tag{4}$$

where $P(\mathbf{g}(\mathbf{x}), \mathbf{u}, \mathbf{r})$ is a generalized penalty functional and $\mathbf{r}$ contains penalty parameters. It will be seen later that $\Phi(\mathbf{x}, \mathbf{u}, \mathbf{r})$ can be formed by adding penalty terms to the ordinary Lagrangian in (3); therefore it is called the *augmented Lagrangian*. Values of $\mathbf{u}$ and $\mathbf{r}$ are chosen at the beginning of each unconstrained minimization and then the functional $\Phi(\mathbf{x}, \mathbf{u}, \mathbf{r})$ is minimized with respect to $\mathbf{x}$. At the end of this minimization, $\mathbf{u}$ and $\mathbf{r}$ are updated and the process repeated until convergence.

Effectiveness of the multiplier methods for optimal engineering design had not been realized until the adjoint variable procedure of design sensitivity analysis for the augmented Lagrangian was developed.[8] However, a more comprehensive numerical study showed the methods still to be inefficient compared to the primal methods for static response structural design problems.[10] The primary reason for this inefficiency was attributed to the unconstrained minimization and update procedures for $\mathbf{u}$ and $\mathbf{r}$. But encouragingly, better performance than the primal methods was observed when the method was applied to dynamic response optimization problems.[68,69] It is this behaviour that has motivated the present study in the hope of developing better methods for optimum design of large and complex systems.

The basic motivation for the methods is to avoid the ill-conditioning associated with the usual penalty function methods.[30] In contrast to the penalty function methods, the penalty parameters $\mathbf{r}$ need not go to infinity to achieve convergence of the multiplier methods. As a consequence, the augmented Lagrangian $\Phi(\mathbf{x}, \mathbf{u}, \mathbf{r})$ has good conditioning.

To show basic steps of a multiplier method, we state the following general algorithm:

Algorithm A: Basic steps of a multiplier method

Step 1. Set $k = 0$. Estimate $\mathbf{u}^{(0)}$ and penalty parameters $\mathbf{r}^{(0)}$.
Step 2. Minimize $\Phi(\mathbf{x}, \mathbf{u}^{(k)}, \mathbf{r}^{(k)})$. Let $\mathbf{x}^{(k)}$ be the solution.
Step 3. If convergence criteria are satisfied, stop the iteration process.
Step 4. Update $\mathbf{u}^{(k)}$ and increase $\mathbf{r}^{(k)}$, if necessary.
Step 5. Set $k = k + 1$ and go to Step 2.

Conceptually a multiplier method is quite simple and its essence is contained in Steps 2 and 4 only. Therefore, the performance of the method depends on how well these two steps are executed. In Step 2, the unconstrained minimization method used and the accuracy requirement for the minimum of Φ determine the behaviour and efficiency of the method. Procedures for updating **u** and **r** also govern robustness and efficiency of the method. The foregoing two aspects are critical for performance of the method and will be discussed throughout.

In Step 2, unconstrained minimization is usually terminated when

$$\| \nabla\Phi(\mathbf{x}^{(k)}, \mathbf{u}^{(k)}, \mathbf{r}^{(k)}) \| \leqslant \varepsilon_k \tag{5}$$

is satisfied, where $\varepsilon_k \to 0$ as $k \to \infty$. (Note that ∇ and ∇^2 will represent gradient and Hessian operators with respect to the design variables **x** throughout, unless noted otherwise.) One way to increase efficiency of the method is to terminate the unconstrained minimization with a crude approximation to the solution. This enables the Lagrange multipliers **u** to be updated more frequently, which is important for large scale engineering optimization problems where an exact minimization is impossible and highly inefficient. To accomplish this, it is better to terminate the unconstrained minimization when the gradient of the augmented Lagrangian is less than some measure of the infeasibility of the constraints as[16]

$$\| \nabla\Phi(\mathbf{x}^{(k)}, \mathbf{u}^{(k)}, \mathbf{r}^{(k)} \| \leqslant \zeta \| \mathbf{g}(\mathbf{x}^{(k)}) \| \tag{6}$$

where ζ is some fixed positive parameter and the norm on the right hand side is over the violated constraints only. Some authors[24] have suggested to reduce ζ in certain situations, resulting in the following convergence criterion:

$$\| \nabla\Phi(\mathbf{x}^{(k)}, \mathbf{u}^{(k)}, \mathbf{r}^{(k)}) \| \leqslant \zeta_k \| \mathbf{g}(\mathbf{x}^{(k)}) \| \tag{7}$$

or

$$\| \nabla\Phi(\mathbf{x}^{(k)}, \mathbf{u}^{(k)}, \mathbf{r}^{(k)}) \| \leqslant \min\{\varepsilon_k, \zeta_k \| \mathbf{g}(\mathbf{x}^{(k)}) \| \} \tag{8}$$

where $\zeta_k \to 0$ and $\varepsilon_k \to 0$ as $k \to \infty$. In addition to the above termination criterion, a *limit* on the number of unconstrained minimization iterations may be imposed. In this regard, continuous update methods (discussed later in the paper), where only few unconstrained minimization iterations are performed in Step 2, have been successfully used. Computational experience thus far indicates that considerable savings are realized by accepting an *inexact* unconstrained minimium.

A simple Lagrange multiplier update procedure needed in Step 4 of Algorithm A is derived for the ith equality constraint as[48, 74]

$$\theta_i^{(k+1)} = \theta_i^{(k)} + g_i(\mathbf{x}^{(k)}) \tag{9}$$

where $\theta_i^{(k)}$ define the Lagrange multipliers as $u_i^{(k)} = r_i \theta_i^{(k)}$ and k denotes the unconstrained minimization stage. It is important to note that (9) does not require gradients of individual constraints, making it quite suitable for engineering applications. This is generally called the Hestenes–Powell formula. The following multiplier approximation for equality constraints, called the projection formula, has been also used:[31, 41, 63]

$$\mathbf{u}(\mathbf{x}) = -(\nabla\mathbf{g}(\mathbf{x})^{\mathrm{T}}\nabla\mathbf{g}(\mathbf{x}))^{-1}\nabla\mathbf{g}(\mathbf{x})^{\mathrm{T}}\nabla f(\mathbf{x}) \tag{10}$$

where $\nabla\mathbf{g}(\mathbf{x})$ is an $n \times l$ matrix whose ith column is the gradient of $g_i(\mathbf{x})$. Even though (10) is used, the method is equivalent to the Hestenes–Powell multiplier method (Proposition 4.5, Reference 89). A drawback of (10) is that gradients of individual constraints are needed, making it unsuitable for engineering applications.

When inequality constraints are present, the multipliers can be updated using the constraint function values as[77,82]

$$\theta_i^{(k+1)} = \theta_i^{(k)} + \max(g_i(\mathbf{x}^{(k)}), -\theta_i^{(k)}); \quad \theta_i^{(k+1)} \geqslant 0; \quad i = l + 1, m \tag{11}$$

It has been shown that the update procedures in (9) and (11) are merely the steepest ascent methods for an equivalent dual of the Problem P.[20] A Newton-like update procedure which requires curvature information has been suggested to establish a faster rate of convergence.[20,34] Convergence properties of the multiplier methods using various update procedures have been analysed and a more sophisticated update procedure to obtain faster convergence rate has been suggested.[89] However, these procedures require gradients of individual constraints, making them unsuitable for engineering applications. Other update formulas are discussed later in the paper.

Miele and co-workers[61,62,64,65] presented many computational results for equality constrained problems using (9) and several other update procedures. In one algorithm the Lagrange multipliers **u** were determined to minimize the error in the optimality conditions for a given **x**. In another algorithm, they were determined so that the constraints were satisfied to the first order. In yet another algorithm,[65] a modified Powell–Hestenes procedure was used as

$$\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} + s\mathbf{g}(\mathbf{x}^{(k)})$$

where the scalar parameter s was determined so that the error in the optimum conditions is minimized. Two variations of all these algorithms were studied. In one, penalty parameters were held constant, and in another one they were varied. The algorithm using the multiplier update procedure with varying penalty parameters performed better. It was also suggested to update multipliers more frequently for better convergence. However, all these procedures require gradients of individual constraints. Therefore, they are not suitable for engineering applications.

In many engineering applications, if the design variables do not remain within certain bounds, singularities can occur in the problem formulation. Therefore, the explicit *design variable bound* constraints must be imposed during unconstrained minimization in Step 2. This can be done quite easily by modifying the unconstrained minimization routine.[9,79] Proper scaling of the cost and constraint functions can improve performance of the multiplier methods. This aspect has also been studied.[79]

3. FUNDAMENTALS OF MULTIPLIER METHODS

In this section, we will discuss some fundamental concepts and procedures associated with the multiplier methods.

3.1. Augmented Lagrangian

The augmented Lagrangian can be defined in several ways. The most popular and commonly used functional includes quadratic penalty terms that are defined for the Problem P as

$$\Phi(\mathbf{x}, \boldsymbol{\theta}, \mathbf{r}) = f(\mathbf{x}) + \frac{1}{2} \sum_{i=1}^{l} r_i [(g_i + \theta_i)^2 - \theta_i^2] + \frac{1}{2} \sum_{i=l+1}^{m} r_i [(g_i + \theta_i)_+^2 - \theta_i^2] \tag{12}$$

where $r_i \theta_i (\equiv u_i)$ are the Lagrange multipliers and $(h)_+ = \max(0, h)$. This functional, suggested by Rockafellar, is closely related to the following one suggested by Fletcher:[34]

$$\bar{\Phi}(\mathbf{x}, \boldsymbol{\theta}, \mathbf{r}) = f(\mathbf{x}) + \frac{1}{2} \sum_{i=1}^{l} r_i (g_i + \theta_i)^2 + \frac{1}{2} \sum_{i=l+1}^{m} r_i (g_i + \theta_i)_+^2 \tag{13}$$

The relationship between the two functionals is given as

$$\Phi(x, \theta, r) = \bar{\Phi}(x, \theta, r) - \frac{1}{2} \sum_{i=1}^{m} r_i \theta_i^2 \tag{14}$$

The term $\dfrac{1}{2} \sum_{i=1}^{m} r_i \theta_i^2$ is independent of x. Therefore, the minimum solution x* of both the functionals for given r and θ (or u) is the same, though the functional values are different. The functional in (12) has a suitable structure for duality that is discussed in Section 3.4. In a discussion of duality, it is simpler to work with u_i's instead of θ_i's. Therefore, the augmented Lagrangian in (12) is written in terms of u_i's as follows:

$$\Phi(x, u, r) = \begin{cases} f(x) + \sum_{i=1}^{l} \left(\frac{1}{2} r_i g_i^2 + u_i g_i \right) + \sum_{i=l+1}^{m} \left(\frac{1}{2} r_i g_i^2 + u_i g_i \right); \text{ if} \left(g_i + \frac{u_i}{r_i} \right) \geqslant 0 \text{ for } i > l \\ f(x) + \sum_{i=1}^{l} \left(\frac{1}{2} r_i g_i^2 + u_i g_i \right) - \sum_{i=l+1}^{m} \frac{u_i^2}{2r_i}; \text{ if} \left(g_i + \frac{u_i}{r_i} \right) < 0 \text{ for } i > l \end{cases} \tag{15}$$

It can be clearly seen from the above expression that $\Phi(x, u, r)$ has certain penalty terms that are added to the ordinary Lagrangian in (3); hence it is called the *augmented Lagrangian.*

The augmented Lagrangian functional in (13) suggests an interesting *interpretation* for the θ_i's. First consider *inequality constraints* only. At the beginning, $\theta_i = 0$ $(i = 1, m)$ is generally chosen that corresponds to the exterior penalty method.[30] At this stage only g_i's with positive values are penalized. For the next stage the corresponding θ_i's may become positive (note that θ_i's for inequalities remain non-negative). This not only increases the penalty for g_i's with a positive value, but also penalizes g_i even if it has a negative value that is greater than the corresponding $-\theta_i$; i.e. if $g_i + \theta_i \geqslant 0$. In subsequent stages, further increase or decrease of θ_i is made according to the sign of the corresponding g_i. In this way, θ_i is seen as a parameter that shifts the threshold level for the penalty term corresponding to g_i, as shown in Figure 1. In view of this it can be seen that the violated constraints are driven monotonically to zero in successive stages with smaller
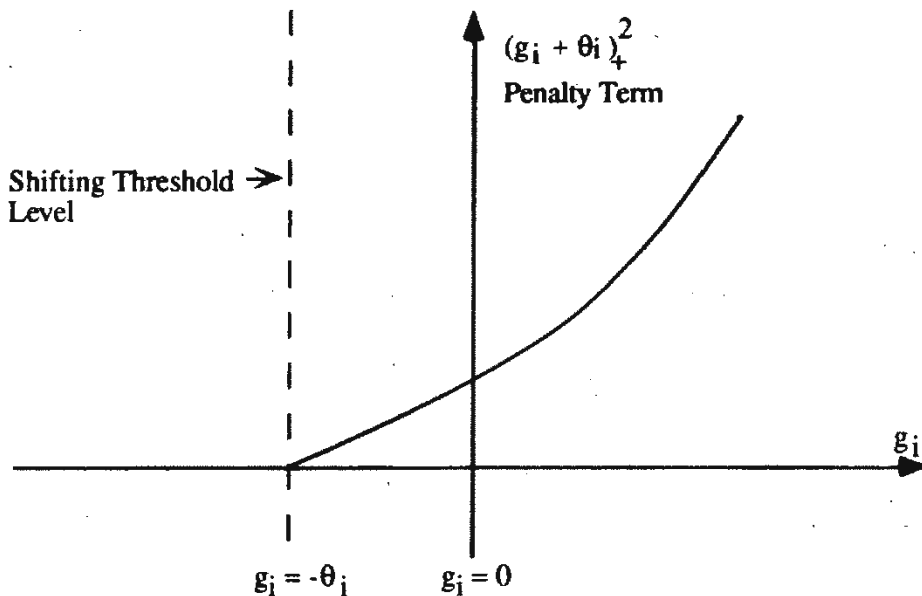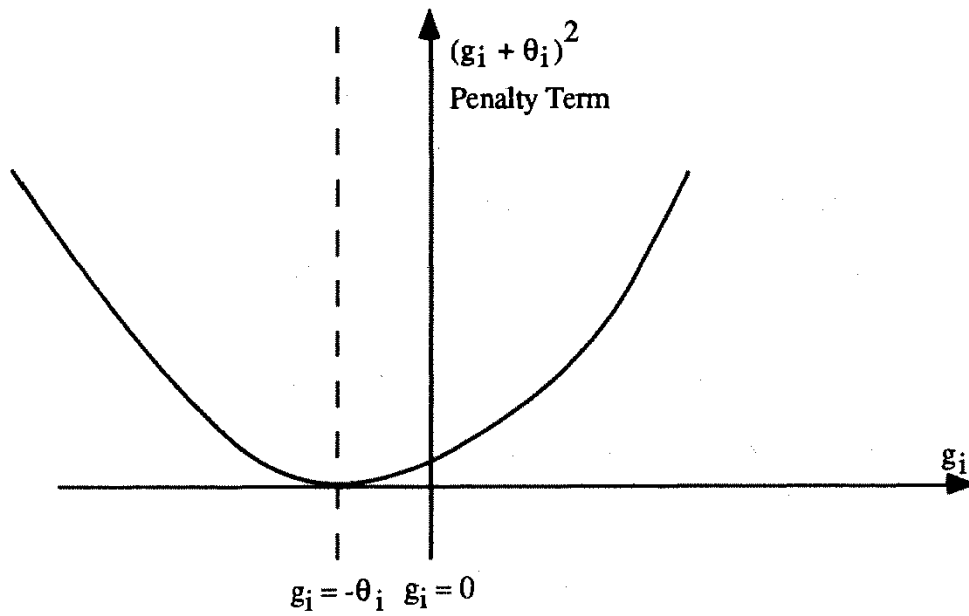

Figure 1. Interpretation of θ_i for an inequality constraint
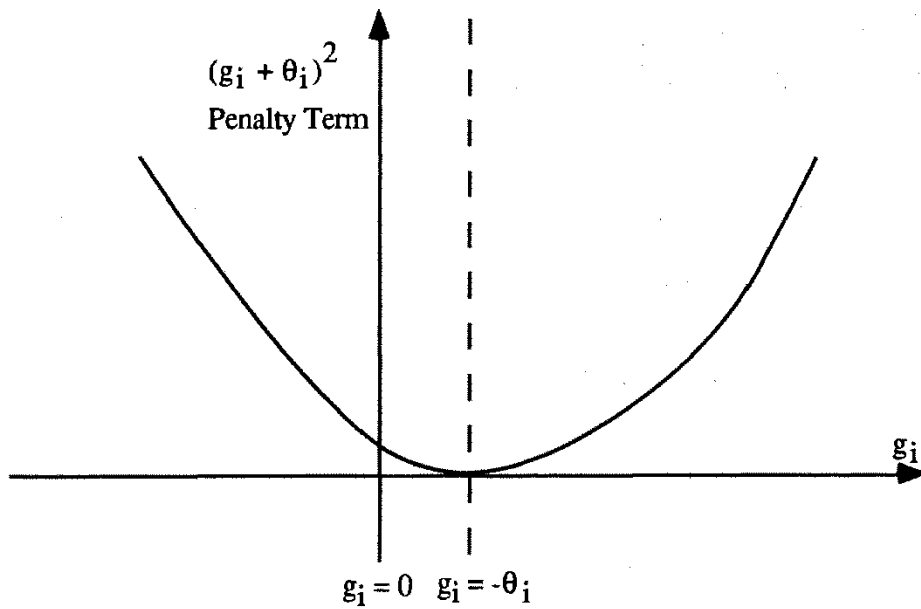
Figure 2. Interpretation of θ_i for an equality constraint

positive increments added to the corresponding θ_i's each time. However, this is not always the case. In earlier stages, a constraint g_i may be violated but strictly satisfied at the solution, i.e. $g_i(x^*) < 0$. The algorithm adjusts to this by causing corresponding θ_i to relax to zero in the later stages as the value of g_i becomes negative.

In case of *equality constraints*, the role of the θ_i's becomes clear from Figure 2. In this case, initially when $\theta_i = 0$ ($i = 1, l$), all non-zero constraints are penalized. In the next stage, the sign of θ_i depends on the corresponding sign of g_i. In the subsequent stages the value of θ_i increases if g_i is positive and decreases if g_i is negative. However, the penalty decreases only if $g_i \to - \theta_i$ from either side. Also, for $\theta_i \neq 0$, the penalty is non-zero even for $g_i = 0$. Thus, all the equality constraints are penalized all the time.

3.2. An analysis of the method

In order to understand the basic ideas of the multiplier method and see its differences from the penalty methods, consider a one-dimensional problem:

$$\text{minimize } f(x) = (x - 2)^2 \text{ subject to } g(x) = (x - 1) \leqslant 0$$

The minimum for the problem is at $x^* = 1$. The exterior penalty function for the problem is given as

$$\Phi(x) = (x - 2)^2 + \tfrac{1}{2}r(x - 1)_+^2$$

where r is a given positive scalar. The minimum of this penalty function is at $x = (r + 4)/(r + 2)$. The only way this approaches the solution of the original problem is when $r \to \infty$. Even then, $x = 1$ cannot be obtained exactly.

The second-order derivative of the penalty function is

$$\frac{d^2\Phi}{dx^2} = \begin{cases} 2 + r; & \text{for } x \geqslant 1 \\ 2; & \text{otherwise} \end{cases}$$

As r becomes larger the curvature of the penalty function to the right of $x = 1$ becomes larger (see Figure 3). Thus, the penalty function becomes more and more ill-conditioned. This affects the ability of a numerical procedure to locate the exact minimum of the penalty function, although it gets quite close to $x = 1$.

To see how difficulties encountered by the exterior penalty function can be alleviated by the multiplier method, consider the augmented Lagrangian functional in (13) for the problem as

$$\Phi(x) = (x - 2)^2 + \tfrac{1}{2}r(x - 1 - \theta)_+^2$$

The minimum for this functional is at $x = [r(1 - \theta) + 4]/[r + 2]$. In this case also, x approaches the solution of the original problem for finite θ with $r \to \infty$. However, exact solution of the
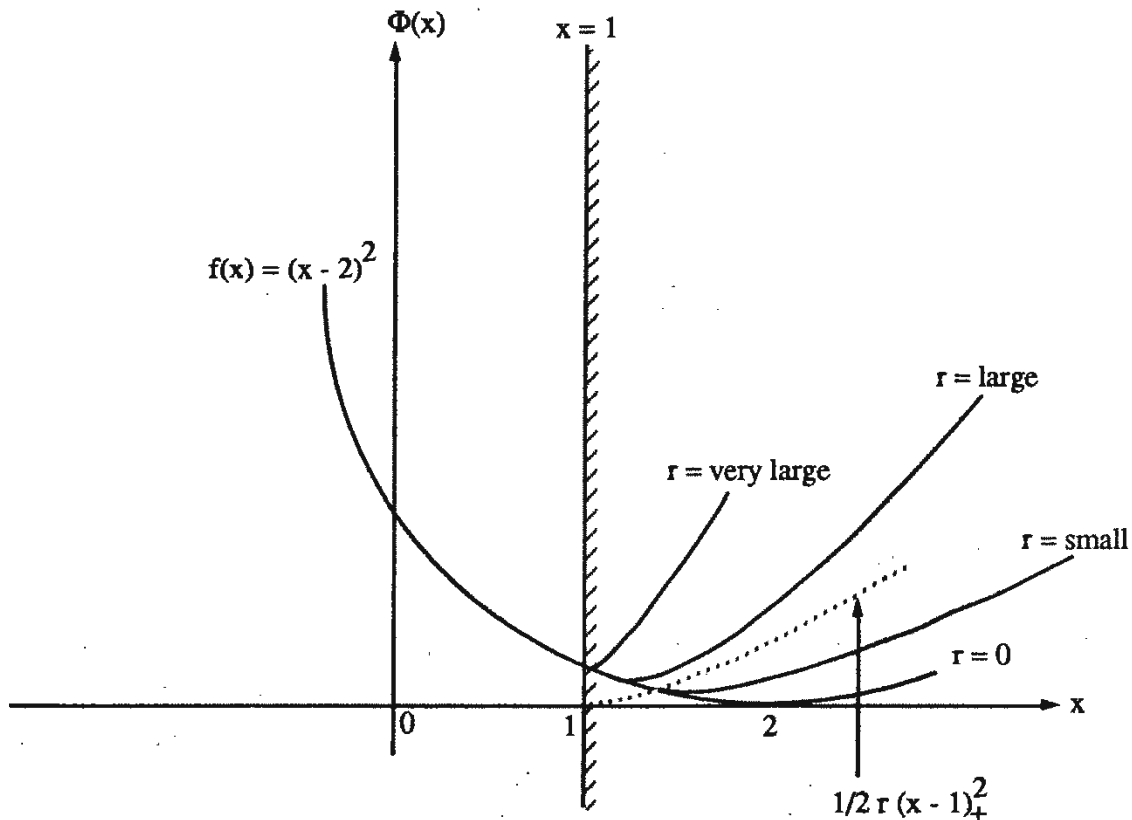
Figure 3. Exterior penalty function

8

original problem can be obtained for a finite value of r if an appropriate value of θ is chosen. To get $x = 1$, the values of θ and r should have the relation $\theta = 2/r$. For example, if $r = 10$ and $\theta = 0\cdot2$, the exact solution $x = 1$ is obtained. Thus, exact solution of the original problem can be obtained if proper values of θ and r are known. Since the value of r need not become very large, the curvature of the augmented functional need not be large. The functional is not ill-conditioned.

Let us see the effect of choosing different values of θ for $r = 10$ on the solution of the augmented Lagrangian functional. The optimum values for x with different values for θ are given in Table I.

It can be seen that, as the value of θ becomes larger, the minimum of the functional shifts to the left of $x = 1$ and vice versa (see Figure 4). As the value of θ becomes closer to $0\cdot2$ from either side, the minimum of the functional comes closer to the solution of the original problem. This then should be the goal of any multiplier update procedure; i.e. to change θ_i at every iteration so as to bring the corresponding u_i closer to its true value. It can be seen that this is achieved by procedure in (11) for arbitrary initial choice of θ.

Table I

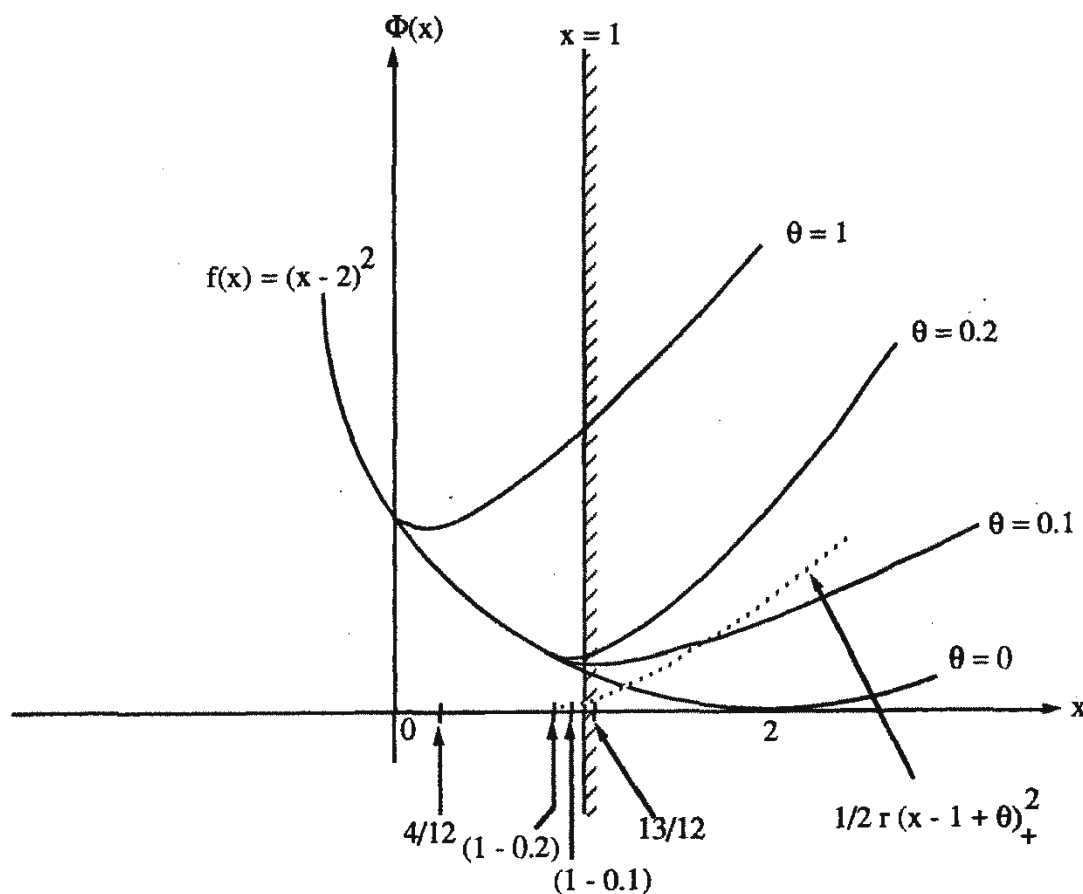| θ	2	1	0·2	0·1	0·05
x	$\dfrac{-6}{12}$	$\dfrac{4}{12}$	1	$\dfrac{13}{12}$	$\dfrac{13\cdot5}{12}$

Figure 4. Augmented Lagrangian functional ($r = 10$)

3.3. Necessary and sufficient conditions

In this section we compare the necessary and sufficient conditions for the Problem P and those for the unconstrained problem of minimizing the functional given in (12) for sufficiently large k, ie. after sufficient number of iterations.

Let x^* be the required solution and $\nabla g_i(x^*)$; $i \in E \cup I$ be *linearly independent* at x^*, where E is the index set for equality constraints defined as $E = \{i: i = 1, l\}$ and I is the index set for active inequalities defined as $I = \{i: u_i^* > 0, i = l + 1, m\}$. Then there exist unique Lagrange multipliers u^* such that the gradient of the Lagrangian in (3) vanishes, i.e.

$$\nabla f(x^*) + \sum_{i=1}^{m} u_i^* \nabla g_i(x^*) = 0 \tag{16}$$

Let $x^{(k)}$ be the minimum of $\Phi(x, u^{(k)}, r)$, where $r > 0$ and $u^{(k)}$ are the current values of the parameters at the kth iteration. First-order optimality conditions for the augmented Lagrangian in (12) give

$$\nabla \Phi(x^{(k)}, u^{(k)}, r) = \nabla f(x^{(k)}) + \sum_{i=1}^{l} r_i(g_i(x^{(k)}) + \theta_i^{(k)}) \nabla g_i(x^{(k)})$$

$$+ \sum_{i=l+1}^{m} r_i(g_i(x^{(k)}) + \theta_i^{(k)})_+ \nabla g_i(x^{(k)}) = 0 \tag{17}$$

Here we have assumed that

$$\nabla(g_i(x^{(k)}) + \theta_i^{(k)})_+^2 = 2(g_i(x^{(k)}) + \theta_i^{(k)})_+ \nabla g_i(x^{(k)}); \quad i = l + 1, m$$

This makes $\nabla \Phi$ continuous.[59]

Since $(g_i + \theta_i)_+ = \max(g_i + \theta_i, 0) = \theta_i + \max(-\theta_i, g_i)$, (17) can be written as

$$\nabla \Phi(x^{(k)}, u^{(k)}, r) = \nabla f(x^{(k)}) + \sum_{i=1}^{l} r_i[\theta_i^{(k)} + g_i(x^{(k)})] \nabla g_i(x^{(k)})$$

$$+ \sum_{i=l+1}^{m} r_i[\theta_i^{(k)} + \max(-\theta_i^{(k)}, g_i(x^{(k)}))] \nabla g_i(x^{(k)}) = 0 \tag{18}$$

Assume that $\theta_i^{(k)}$ are chosen to satisfy (9) for equality constraints and (11) for inequality constraints. Also assume that, for some large k, $\theta_i^{(k)} \to \theta_i^*$ ($i = 1, m$), which implies that

$$g_i(x^{(k)}) = 0; \quad i = 1, l \tag{19}$$

$$\max(-\theta_i^{(k)}, g_i(x^{(k)})) = 0; \quad i = l + 1, m \tag{20}$$

and (18) is reduced to (16). Therefore, we may conclude that $x^{(k)} = x^*$ and $r_i \theta_i^{(k)} = r_i \theta_i^* = u_i^*$. Note that (20) can be satisfied only when $\theta_i^{(k)} = 0$ or $g_i(x^{(k)}) = 0$ or both are zero. These conditions can also be written as $\theta_i^{(k)} g_i(x^{(k)}) = 0$; $i = l + 1, m$. Thus (18) to (20) are equivalent to requiring the following Kuhn–Tucker necessary conditions for the Problem P:

$$\nabla f(x^{(k)}) + \sum_{i=1}^{m} u_i^{(k)} \nabla g_i(x^{(k)}) = 0 \tag{21}$$

$$g_i(x^{(k)}) = 0; \quad i = 1, l \tag{22}$$

$$u_i^{(k)} \geqslant 0, \quad g_i(x^k) \leqslant 0, \quad \text{and} \quad u_i^{(k)} g_i(x^{(k)}) = 0; \quad i = l + 1, m \tag{23}$$

From the foregoing discussion it is evident that the aim of choosing u_i (or θ_i) in (9) and (11) is

such that $g_i(\mathbf{x}^{(k)}) \to 0$; $i = 1, l$ and $\max(-\theta_i^{(k)}, g_i(\mathbf{x}^{(k)})) \to 0$; $i = l + 1, m$. This results in $\mathbf{x}^{(k)} \to \mathbf{x}^*$. It has been proved that $\mathbf{x}^*$ is an isolated local minimum point of $\Phi(\mathbf{x}, \mathbf{u}^*, \mathbf{r})$ provided that r_i are sufficiently large but finite.[34]

It has been assumed that $\nabla g_i(\mathbf{x}^*)$; $i \in E \cup I$ are *linearly independent*. This is equivalent to saying that $\mathbf{x}^*$ is a *regular point*. If $\mathbf{x}^*$ is a regular point, then the Lagrange multipliers u_i^*, $i = 1, m$ are unique. The assumption that $\mathbf{x}^*$ is a *regular point* should be made with caution as it is not valid for the following cases.

1. If any equality constraint $g_i(\mathbf{x}) = 0$ is expressed as a pair of inequalities as $g_i(\mathbf{x}) \leqslant 0$ and $g_i(\mathbf{x}) \geqslant 0$, then $\mathbf{x}^*$ cannot be a regular point.
2. If the inequality constraints $g_i(\mathbf{x}) \leqslant 0$, $i = l + 1, m$ are expressed as an equivalent constraint
$$\sum_{i=l+1}^{m} (g_i)_+^2 = 0,$$
then $\mathbf{x}^*$ cannot be a regular point.
3. Consider a continuum constraint $g_i(\mathbf{x}, t) \leqslant 0$ for $t \in [t_0, t_f]$. This constraint may be expressed in an equivalent form that eliminates the parameters t as

$$\int_{t_0}^{t_f} (g_i(\mathbf{x}, t))_+^2 \, dt = 0 \tag{24}$$

then $\mathbf{x}^*$ is not a regular point.

If the problem is so formulated that $\mathbf{x}^*$ is not a regular point, then a method that works with the Lagrange multipliers may not be the best choice. The Lagrange multipliers are not necessarily unique in that case.

So far, the discussion has centred on necessary conditions for $\mathbf{x}^*$ to be a local minimum of $\Phi(\mathbf{x}, \mathbf{u}, \mathbf{r})$. Now *sufficiency conditions* are stated. Assume that $f(\mathbf{x})$ and $g_i(\mathbf{x})$, $i = 1, m$ are twice continuously differentiable. Then as a sufficient condition for $\mathbf{x}^*$ to be an isolated local minimum point of the Problem P, the following inequality must hold:

$$(\mathbf{x}, \nabla^2 L(\mathbf{x}^*, \mathbf{u}^*)\mathbf{x}) > 0 \tag{25}$$

for all $\mathbf{x} \neq \mathbf{0}$ and satisfying the conditions

$$(\nabla g_i(\mathbf{x}^*), \mathbf{x}) = 0; \quad i \in E \cup I \tag{26}$$

$$(\nabla g_i(\mathbf{x}^*), \mathbf{x}) \leqslant 0; \quad i \notin E \cup I \tag{27}$$

where $\nabla^2 L$ is the Hessian of the Lagrangian in (3) and $(\mathbf{a}, \mathbf{b})$ implies $\mathbf{a}^T\mathbf{b}$. If $\nabla^2 L(\mathbf{x}^*, \mathbf{u}^*)$ is positive definite then we have a *stronger sufficiency condition* for $\mathbf{x}^*$ to be an isolated local minimum. Note that, if the foregoing conditions are not satisfied, then $\mathbf{x}^*$ is not an isolated local minimum, but it may still be a local minimum.[2] Let $\mathbf{x}^{(k)}$ be such that, for sufficiently large k, it satisfies the necessary conditions in (18) to (20). As the sufficiency condition for $\mathbf{x}^{(k)}$ to be a local minimum of $\Phi(\mathbf{x}, \mathbf{u}^{(k)}, \mathbf{r})$, $\nabla^2 \Phi(\mathbf{x}^{(k)}, \mathbf{u}^{(k)}, \mathbf{r})$ must be positive definite. For $\mathbf{x}^{(k)} = \mathbf{x}^*$ the conditions in (25) and (27) do not guarantee positive definiteness of $\nabla^2 \Phi(\mathbf{x}^*, \mathbf{u}^*, \mathbf{r})$. However, the penalty parameters $\mathbf{r}$ can be increased such that $r_i \geqslant \bar{r}_i$ (for some $\bar{r}_i > 0$; $i = 1, m$) make $\nabla^2 \Phi(\mathbf{x}^*, \mathbf{u}^*, \mathbf{r})$ positive definite. Also, if $\nabla^2 L(\mathbf{x}^*, \mathbf{u}^*)$ is positive definite, so is $\nabla^2 \Phi(\mathbf{x}^*, \mathbf{u}^*, \mathbf{r})$.

3.4. Duality theory

The duality theory will show that the optimum Lagrange multipliers $\mathbf{u}$ are determined by a *maximization problem*. This problem is unconstrained even if there are inequality constraints. However, to make sure that Lagrange multipliers are non-negative for inequality constraints,

simple constraints must be imposed on them in numerical computations. Here we discuss some important results of duality for *general non-convex programming* problems.

Duality theory for the augmented Lagrangian is developed based on the duality theory originally developed for the Lagrangian defined in (3). It turns out that the point (x^*, u^*) corresponds to a local minimum of the Problem P in (1) and (2) if it is also a saddle point of $L(x, u)$. A point (x^*, u^*) is called the *saddle point* of $L(x, u)$ if it satisfies the following inequalities:

$$L(x^*, u) \leqslant L(x^*, u^*) \leqslant L(x, u^*) \qquad (28)$$

with $L(x^*, u^*)$ bounded below and for $x \in S$ and $u \in T$, where S and T are subsets of the domain for $L(x, u)$ in the neighbourhood of the point (x^*, u^*). If a saddle point exists, then the problem of finding a local minimum of Problem P can be viewed as the problem of finding a saddle point of the function $L(x, u)$. These saddle points exist for $x \in S$ only if the cost function is convex, every equality constraint is linear and every inequality is convex; i.e. if the problem is *convex*. However, a saddle point for the augmented Lagrangian $\Phi(x, u, r)$ exists if penalty parameters are sufficiently large, i.e. r_i are greater than some threshold value $\bar{r}_i$.[37]

Methods based on a technique that searches for saddle points of the function to find a solution to the Problem P are referred to as *primal–dual methods*. Problem P is called the primal and the corresponding *dual* is defined as[12,34]

Problem D. For some constant r

$$\text{maximize } \phi(u) \text{ for } u \in T \qquad (29)$$

where the dual function $\phi(u)$ is defined as

$$\phi(u) = \min_{x \in S} \Phi(x, u, r) \qquad (30)$$

It is well known that the optimum solutions of Problem P and Problem D satisfy

$$\inf(P) \geqslant \sup(D) \qquad (31)$$

In (30), if $\Phi(x, u, r)$ is replaced by $L(x, u)$, then in (31) equality may not hold, unless the set S and the cost function are convex. The difference between $\inf(P)$ and $\sup(D)$ in (31) is termed the *duality gap*. However, if certain assumptions are made on the convexity of the original problem, the duality gap can be eliminated using the augmented Lagrangian.[78] This is called strong duality. In general, for an engineering problem the assumptions required for the strong duality cannot be guaranteed to satisfy for any set S. However, if $S^* \subset S$ is considered as a very small neighbourhood of a local minimum then the necessary and sufficient conditions for the local minimum are enough to show duality. This is known as the *local duality theory*. Fletcher[34] reviewed and extended local duality theory for the equality–inequality constrained problem corresponding to the augmented Lagrangian. Local duality can also be found in Luenberger[59] for the ordinary Lagrangian and in Buys[20] for the augmented Lagrangian.

Here we present major results of the duality theory in a simplified form; more details can be found in Mangasarian,[60] Rockafellar,[76] Fletcher,[34] Tapia[89] and Bertsekas.[19] Consider a point (x^*, u^*) satisfying first-order necessary conditions and second-order sufficiency conditions for a local minimum of Problem P.

Theorem. There exist $\bar{r}_i > 0$ $(i = 1, m)$, such that, for any $r_i \geqslant \bar{r}_i$ $(i = 1, m)$, x^* is a local minimum of $\Phi(x, u^*, r)$ and $\nabla^2 \Phi(x^*, u^*, r)$ is positive definite.

This is an important result that implies that, if penalty parameters are sufficiently large and if u^* is

known, then x^* can be obtained by just one unconstrained minimization of $\Phi(x, u^*, r)$. This suggests that the Lagrange multipliers u should be estimated more accurately.

For further discussion, assume that the penalty parameters are sufficiently large, as required by the foregoing theorem. Also consider the Problem P with equality constraints only; inequalities will be discussed later on. Consider $x(u)$ as a function that is implicitly determined by solving the non-linear necessary conditions for minimization:

$$\nabla \Phi(x, u) = 0 \tag{32}$$

From the foregoing theorem, we known $\nabla^2 \Phi(x^*, u^*)$ is positive definite for sufficiently large r_i ($i = 1, m$). Then according to the implicit functions theorem, there exists an open neighbourhood $T^* \subset T$ about u^* and $S^* \subset S$ about x^* such that for any $u \in T^*$ there exists a unique x in S^* satisfying (32). Furthermore, $x(u)$ is continuous and continuously differentiable, and $\nabla^2 \Phi(x, u)$ is positive definite for all $u \in T^*$. Further, the following results can be stated:

$$x(u^*) = x^* \tag{33}$$

$$\nabla_u x(u) = - \nabla g(x)^T [\nabla^2 \Phi(x, u)]^{-1} \tag{34}$$

$$\nabla_u \phi(u) = g(x(u)) \tag{35}$$

$$\nabla_u \phi(u^*) = g(x^*) = 0 \tag{36}$$

$$\nabla_{uu}^2 \phi(u) = - \nabla g(x)^T \nabla^2 \Phi(x, u)^{-1} \nabla g(x) \tag{37}$$

where $\phi(u)$ is the dual function defined in (30) with T replaced by T^* and S replaced by S^*. Equation (36) shows that the dual function is stationary with respect to the Lagrange multipliers at u^*. Since $\nabla^2 \Phi(x^*, u^*)$ is positive definite, (37) shows that $\nabla_{uu}^2 \phi(u)$ is negative definite. Therefore it can be concluded that, if x^* solves the primal problem, its associated Lagrange multiplier u^* solves the dual problem since necessary and sufficient conditions of optimality are satisfied. This observation and (35) and (9) show that the *Hestenes–Powell multiplier update procedure is the steepest ascent method to solve the dual problem* with step size as one. This motivates development of different multiplier update procedures[20, 34, 89] that are reviewed in a later section.

For the general equality–inequality Problem P, results similar to the foregoing can be derived:[34]

$$\frac{\partial \phi}{\partial u_i} = g_i; \quad i = 1, l \tag{38}$$

$$= \max(-\theta_i, g_i); \quad i = l + 1, m \tag{39}$$

$$\nabla_{uu}^2 \phi = \begin{bmatrix} - \nabla g^T B^{-1} \nabla g & 0 \\ 0 & - \text{diag}(1/r_i) \end{bmatrix} \begin{matrix} i \in J \\ i \notin J \end{matrix} \tag{40}$$

where $J = \{i : i = 1, l \text{ and } i = l + 1, m \text{ for } (g_i + \theta_i) > 0\}$, $\text{diag}(1/r_i)$ is a diagonal matrix with $1/r_i$ as the diagonal elements, and $B = \nabla^2 \Phi(x, u)$. It can be shown that the matrix in (40) is non-singular.[34] Therefore, to get an approximation to the Lagrange multipliers u^*, we can either solve the dual Problem D by taking a step based on gradient-like methods or Newton-like methods. For very *large penalty parameters* it is shown that the *Newton step is approximately the same as the steepest ascent step* (with unit step length). But for smaller values of the penalty parameters, the Newton-like step can still be advantageous (only when gradients of individuals constraints are easier to compute).

It is evident from (38) and (39) that the conditions in (19) and (20) are simply the necessary condition for $\phi(u)$ to have a maximum (or stationary) point. In this event, duality theory provides

an excellent way of proving convergence properties of the multiplier algorithms. Local duality gives only local convergence results for some fixed values of the penalty parameters. The role of penalty parameters in order to achieve global convergence is discussed in later sections.

Finally, a stronger result can be stated: if $\mathbf{x}(\mathbf{u})$ can be guaranteed to be a global minimizer of $\Phi(\mathbf{x}, \mathbf{u})$ for the Problem P, then $\mathbf{u}^*$ is a global maximizer of the dual function $\phi(\mathbf{u})$.

3.5. Convergence properties

Global convergence of multiplier methods was established by introducing automatic procedures for revising penalty parameters $r_i, i = 1, m.$[15,34,74] An automatic penalty limitation procedure has also been introduced with a proof of global convergence.[72]

Global convergence of the multiplier methods can be discussed by imbedding them into the general penalty function methods.[15] This viewpoint yields global convergence results. By *global convergence* we mean convergence to a local minimum starting from an arbitrary point $\mathbf{x}$; it does not imply a global minimum point. For a general penalty function method, there exist non-negative $\mathbf{r}^{(0)}$ and $M \in (0, \infty)$ such that

$$\| \mathbf{x}^{(k)} - \mathbf{x}^* \| \leqslant \frac{M}{\| \mathbf{r}^{(k)} \|} \| \mathbf{u}^{(k)} - \mathbf{u}^* \| \tag{41}$$

and

$$\| \mathbf{u}^{(k+1)} - \mathbf{u}^* \| \leqslant \frac{M}{\| \mathbf{r}^{(k)} \|} \| \mathbf{u}^{(k)} - \mathbf{u}^* \| \tag{42}$$

for all $r_i^{(k)} > r_i^{(0)} > 0$ and $\mathbf{u}^{(k)} \in T$, where T is an open sphere centred at $\mathbf{u}^*$. The results shown above can be used to establish global convergence of the multiplier method. The only assumption here is that $\| \mathbf{r}^{(k)} \| > M$ and $r_i^{(k)} > r_i^{(0)} > 0$ for all k. Furthermore, the result shows that the sequence $\| \mathbf{u}^{(k)} - \mathbf{u}^* \|$ converges at least *linearly* if $\| \mathbf{r}^{(k)} \|$ is bounded above and *superlinearly* if $\| \mathbf{r}^{(k)} \|$ goes to infinity.

The *rate of convergence* of multiplier methods is critically dependent on the Lagrange multiplier update procedures. The simplest procedure given in (9) gives rise to linear convergence. If superlinear convergence is required for bounded $\| \mathbf{r} \|$, it becomes necessary to use a more accurate procedure.[24]

In Powell's algorithm[74] (given in Section 3.8), a parameter $\bar{K}$ is used to enforce global convergence. This non-negative parameter represents the maximum constraint violation as

$$\bar{K} = \max \left\{ \max_{1 \leqslant i \leqslant l} |g_i|; \quad \max_{l+1 \leqslant i \leqslant m} |\max(g_i, -\theta_i)| \right\} \tag{43}$$

To enforce global convergence, $\bar{K}$ is required to reduce at every iteration of the unconstrained minimization algorithm, and the entire iterative process is stopped if the following criteria are satisfied:

$$\bar{K} \leqslant \varepsilon, \quad \text{and} \quad \| \nabla\Phi(\mathbf{x}^{(k)}, \mathbf{u}^{(k)}, \mathbf{r}) \| \leqslant \varepsilon \tag{44}$$

where ε is an error tolerance. If $\bar{K}$ is not reduced at a particular iteration, then the value of r_i is increased by some automatic procedure.[34] Therefore the role of the penalty parameters $\mathbf{r}$ is important for obtaining global convergence of the multiplier methods. We aim to force convergence to a Kuhn–Tucker point $(\mathbf{x}^*, \mathbf{u}^*)$ by adjusting $\mathbf{x}$, $\mathbf{u}$ and $\mathbf{r}$ iteratively.

The foregoing scheme is not guaranteed to work, although it appears to be well conceived. A different scheme has been proposed that is proved (theoretically) to be globally convergent.[72] It

has been shown that, when conditions in (41) and (42) hold, $\|\mathbf{u}^{(k+1)} - \mathbf{u}^{(k)}\| \to 0$ linearly, with the rate constant being proportional to $1/\|\mathbf{r}^{(k)}\|$. It is suggested that the penalty be increased on the basis of a test on the norm of the difference of successive multipliers:

$$\|\mathbf{u}^{(k+1)} - \mathbf{u}^{(k)}\| \leqslant M\varepsilon_k \qquad (45)$$

with $M > 0$, $\varepsilon_k \in (0, 1)$. The test will eventually be satisfied for a large enough penalty r. Such a test is used to detect when the penalty parameters r are large enough. This algorithm needs an *a priori* estimate of the maximum magnitude of the multipliers. For a general non-convex problem this is not possible. Therefore, further modification of the algorithm is needed for practical applications. Apart from this, to achieve global convergence, the maximum step length to be used in the unconstrained minimization algorithm needs to be decreased when the multipliers are updated (multipliers are updated using (9) and (11)). Therefore, for best results it is necessary to use an unconstrained minimization algorithm that converges precisely to isolated minimum points.

To conclude, the multiplier methods are globally convergent and can be implemented in a robust manner. The problems associated with step size calculations in the primal methods to force global convergence are not present; i.e. a separate descent function is not needed to calculate the step size. The methods can be expected to be more reliable as a result.

3.6. Forms of augmented Lagrangians

The quadratic form of the augmented Lagrangian given in (12) has been most widely used in practical implementations of the multiplier methods. However, for the following reasons, it may be occasionally advantageous to use other forms.[19]

1. While the cost function may be bounded below in the constraint set, the augmented Lagrangian need not be (over the entire space) for all values of the penalty parameters.
2. The methods most likely to be used for unconstrained minimization of the augmented Lagrangian rely conceptually on continuity of second derivatives. Under extreme circumstances, discontinuity in the augmented Lagrangian can considerably slow down the rate of convergence of these methods and can be the cause of algorithmic failure. Exponential penalty functions discussed in Section 6 have better continuity than the augmented Lagrangian in (12).
3. Multiplier methods corresponding to different types of penalty functions, used to construct the augmented Lagrangians, can exhibit drastically different rates of convergence.

A few other forms of the augmented Lagrangians have been suggested by some researchers during the 1970s.[19,68]

3.7. Other aspects of multiplier methods

3.7.1. Geometric interpretation. This section provides a simple geometric interpretation of multiplier methods that also motivates the convergence analysis. The iterative process for problems with only inequality constraints is shown in Figure 5.[10] This geometrical interpretation is based on the result proved by Powell.[74] He showed that minimizing $\Phi(\mathbf{x}, \mathbf{u}, \mathbf{r})$ to obtain $\mathbf{x}(\mathbf{u})$ is equivalent to solving the problem

$$\text{minimize} \quad f(\mathbf{x})$$

$$\text{subject to} \quad g_i(\mathbf{x}) \leqslant \max(-\theta_i, g_i(\mathbf{x}(\mathbf{u}))); \quad i = 1, m$$
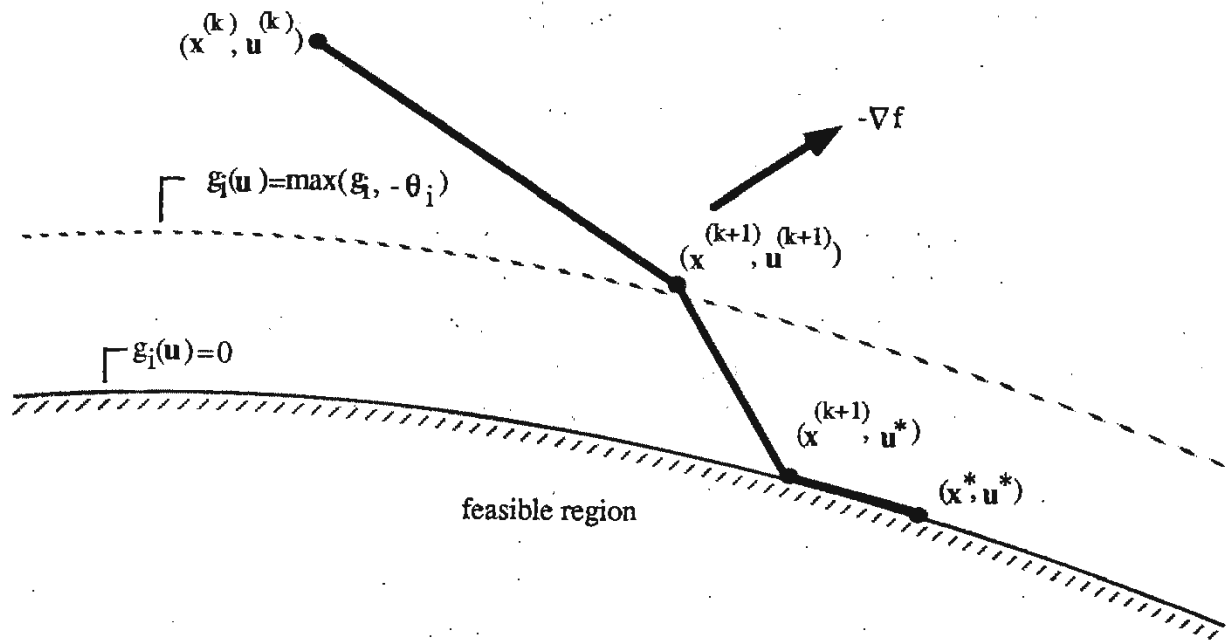
Figure 5. Direction vector in multiplier methods

The constraints in the above problem define a modified feasible domain for given **u**, as shown in Figure 5. Once the minimum of the foregoing problem is obtained, the constraints are modified and the process repeated. The equality constraints can be included in the definition of the foregoing problem. By requiring the parameter $\bar{K}$ in (43) to go to zero at the optimum, the artificial and the real constraint boundaries merge.

Another geometrical interpretation of the multiplier methods can be given by considering the *equality* constrained problem ECP. It can be shown that minimizing $\Phi(\mathbf{x}, \mathbf{u}, \mathbf{r})$ to obtain $\mathbf{x}(\mathbf{u})$ is equivalent to solving the constrained problem:[15]

$$\text{minimize } f(\mathbf{x}) \text{ subject to } g_i(\mathbf{x}) = e_i; \quad i = 1, l \tag{46}$$

with e_i as a perturbation of $g_i(\mathbf{x}) = 0$.

Let $p(\mathbf{e})$ be the solution of the perturbed problem defined in (46); i.e.

$$p(\mathbf{e}) = \min_{g(\mathbf{x}) = \mathbf{e}} f(\mathbf{x}) \tag{47}$$

It is known that

$$p(0) = f(\mathbf{x}^*) = \text{optimal solution of (46) with } \mathbf{e} = \mathbf{0} \tag{48}$$

$$\frac{\partial p(0)}{\partial e_i} = - u_i^* \tag{49}$$

For a scalar r, we can write

$$\min_{\mathbf{x}} \Phi(\mathbf{x}, \mathbf{u}, r) = \min_{\mathbf{e}} \left[\min_{g(\mathbf{x}) = \mathbf{e}} \left\{ f(\mathbf{x}) + (\mathbf{u}, \mathbf{g}) + \tfrac{1}{2} r(\mathbf{g}, \mathbf{g}) \right\} \right] \tag{50}$$

From the definition in (47), (50) can be rewritten as

$$\min_{\mathbf{x}} \Phi(\mathbf{x}, \mathbf{u}, r) = \min_{\mathbf{e}} \left\{ p(\mathbf{e}) + (\mathbf{u}, \mathbf{e}) + \tfrac{1}{2} r(\mathbf{e}, \mathbf{e}) \right\} \tag{51}$$

The above equation can be interpreted geometrically as shown in Figure 6. In the figure, e and r are treated as scalars for simplicity. The parabolic curve $p_c(\mathbf{e})$ stands for the augmented
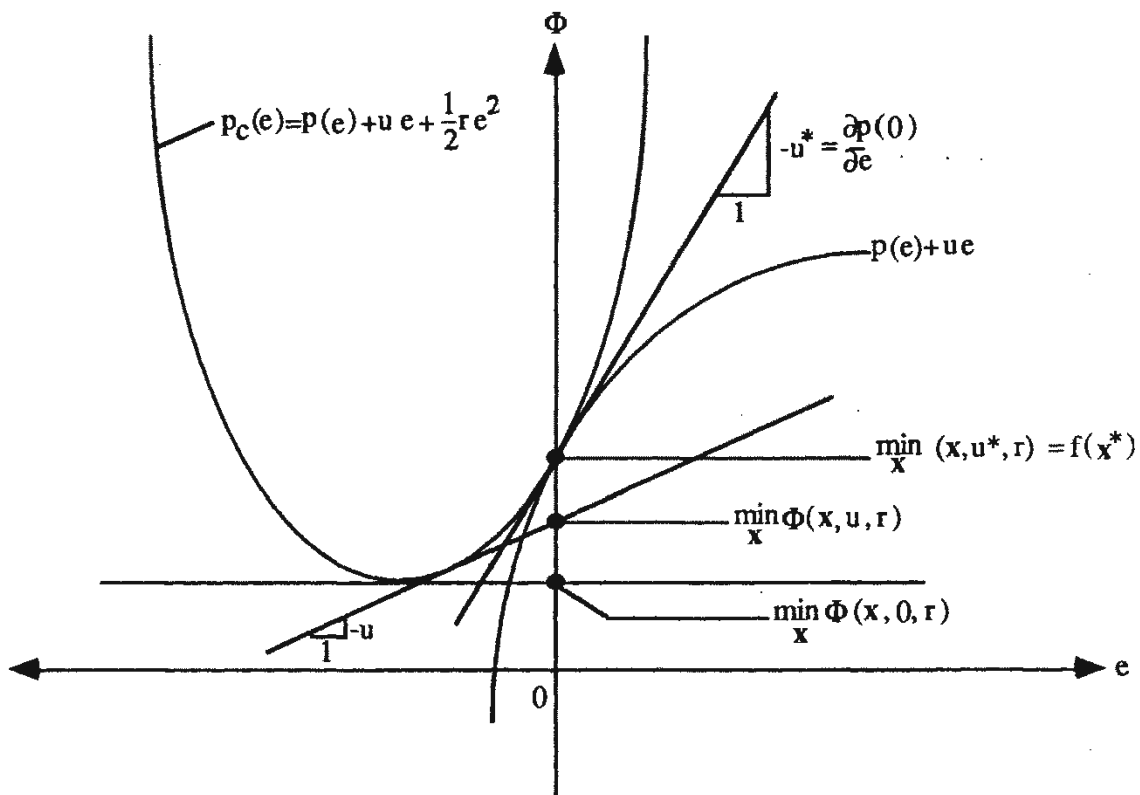
Figure 6. Geometric interpretation of augmented Lagrangian

Lagrangian of the perturbed constrained problem. Note that the addition of $1/2\ r(e, e)$ to $p(e)$ has an important convexification effect. We see that, as u gets closer to the Lagrange multiplier u^*, the corresponding value of $\Phi(x, u, r)$ gets closer to the optimum for the problem. This leads to the conclusion that convergence of the multiplier algorithms can be accelerated if an appropriate Lagrange multiplier update procedure that forces $u^{(k)} \to u^*$ is used.

3.7.2. Potential constraint strategy. Because of the structure of the augmented Lagrangian, only violated, active or nearly active constraints are considered in each iteration. The inequality constraints that are satisfied (excluding active and nearly active) are not required in constructing the augmented Lagrangian function and its derivative (see Figure 1). This is called the *potential constraint strategy.*

For a general inequality–equality problem, a *potential constraint set* is defined as

$$I_p = \{i: i = 1, l \text{ and } i = l + 1, m \text{ for } (g_i + \theta_i) \geq 0\}$$

where θ_i are small positive numbers determined from u_i and r_i as $\theta_i = u_i/r_i$. If we have more than 500 inequality constraints, then, depending on the number of design variables, the set I_p may consist of only 10 to 15 inequality constraints near the solution. The potential constraint strategy is very important in engineering applications as it reduces the computational effort. One of the advantages of using the multiplier method is that the potential constraint strategy is automatically incorporated.

3.7.3. Lagrange multiplier update procedures. The convergence properties of multiplier methods are highly dependent on the Lagrange multiplier update procedures, as noted previously. The usual way of implementing multiplier methods is to minimize the augmented

17

Lagrangian for a fixed value of the multipliers and then update them. However, they can be updated more frequently, as in continuous multiplier update methods that are discussed in Section 4.

A multiplier update procedure can be generally defined as[89,90]

$$\mathbf{u}^{(k+1)} = \mathbf{U}(\mathbf{x}^{(k)}, \mathbf{u}^{(k)}, \mathbf{r}^{(k)}) \tag{52}$$

with the property that

$$\mathbf{u}^* = \mathbf{U}(\mathbf{x}^*, \mathbf{u}^*, r^*) \tag{53}$$

where $(\mathbf{x}^*, \mathbf{u}^*)$ is a solution of the problem. If $\mathbf{U}$ does not depend explicitly on $\mathbf{u}$, i.e.

$$\frac{\partial \mathbf{U}}{\partial u_i}(\mathbf{x}, \mathbf{u}, \mathbf{r}) = 0, \quad i = 1, m \tag{54}$$

then $\mathbf{U}$ is said to be a *Lagrange multiplier approximation procedure*. Several *multiplier update procedures* for the equality constrained problem that have appeared in the literature are given as follows:

$$\mathbf{U}_{HP}(\mathbf{x}, \mathbf{u}, r) = \mathbf{u} + \mathbf{Rg} \tag{55}$$

$$\mathbf{U}_F(\mathbf{x}, \mathbf{u}, r) = - [\nabla \mathbf{g}^T \nabla \mathbf{g}]^{-1} \nabla \mathbf{g}^T \nabla f \tag{56}$$

$$\mathbf{U}_M(\mathbf{x}, \mathbf{u}, r) = [\nabla \mathbf{g}^T \nabla \mathbf{g}]^{-1} [\mathbf{g} - \nabla \mathbf{g}^T \nabla f] \tag{57}$$

$$\mathbf{U}_B(\mathbf{x}, \mathbf{u}, r) = \mathbf{u} + [\nabla \mathbf{g}^T \mathbf{B}^{-1} \nabla \mathbf{g}]^{-1} \mathbf{g} \tag{58}$$

$$\mathbf{U}_T(\mathbf{x}, \mathbf{u}, r) = [\nabla \mathbf{g}^T \mathbf{D}^{-1} \nabla \mathbf{g}]^{-1} [\mathbf{g} - \nabla \mathbf{g}^T \mathbf{D}^{-1} \nabla f] - \mathbf{Rg} \tag{59}$$

$$\mathbf{U}_G(\mathbf{x}, \mathbf{u}, r) = \mathbf{u} + [\nabla \mathbf{g}^T \mathbf{D}^{-1} \nabla \mathbf{g} + \mathbf{A}]^{-1} [\mathbf{g} - \nabla \mathbf{g}^T \mathbf{D}^{-1} \nabla \Phi] \tag{60}$$

The matrix $\mathbf{R}$ is diagonal with the penalty parameter r_i as its ith diagonal element. $\mathbf{D}^{-1}$ in (59) and (60) is an $n \times n$ matrix (see discussion following (79) for choices of this matrix), and $\mathbf{A}$ in (60) is an $m \times m$ matrix (these can be taken as identity matrices). $\mathbf{B}^{-1}$ in (58) is the inverse of the Hessian of the augmented Lagrangian or its approximation.

It is of interest to observe that (55) to (57) use only first-order information, whereas (58) to (60) use second-order information. It is impossible to accurately credit those responsible for each of these multiplier update formulas. However, in the literature they are often credited as follows. Formula (55) was introduced independently by Hestenes[48] and Powell[74] and is, therefore, known as the Hestenes–Powell ($\mathbf{U}_{HP}$) multiplier update procedure. Rockafellar[77] and Schuldt[82] both independently suggested modification to this formula to include inequality constraints. Rosen[80] was probably first to introduce the formula in (56). Later it was introduced and used by several authors.[31,41,61−65] Note that the formula (56) is the least-square solution for $\mathbf{u}$ of the overdetermined linear system

$$\nabla f + \nabla \mathbf{gu} = 0 \tag{61}$$

Equivalently, (56) also arises when ∇f is projected onto the tangent subspace of the constraints, so it is called a projection formula.[89] Glad and Polak[39] suggest slight modification to the formula to include inequality constraints. Formula (57) is a special case of a class of formulas used by Miele et al.[64] Formula (58) was introduced by Buys.[20] The multiplier method in conjunction with Buys update formula has been extended to inequality constraints by Fletcher.[34] The multiplier update formula in (59) was introduced by Tapia.[88] A special case of this formula could be traced back to

Bard and Greenstadt.[6] A similar formula from a different direction was also independently introduced by Han.[46] Tapia[89,90] suggested extension of this formula to include inequality constraints using squared slack variables. Tapia[89] also introduced formula (60). All these update formulas correspond to the augmented Lagrangian given in (12). Update formulas corresponding to other augmented Lagrangians can be derived using the theory used to derive (55)–(60); see Bertsekas.[19]

The following theorem is worth mentioning here.[90]

Theorem. Formulas (57), (58) and (59) are all special cases of the formula (60). Moreover, formulas (55) and (56) are special cases of (60) if and only if $r_i > 0$ $(i = 1, l)$ in (12).

The update formulas U_{HP} and U_F are equivalent, as are U_B and U_T with exact unconstrained minimization.[89] This is not true for the continuous version of the multiplier method discussed later. It is important to note that only the formula in (55) does not require gradients of individual constraints, so this is the most appropriate formula for engineering applications.

3.7.4. Penalty parameters. The behaviour of multiplier methods depends on *initial penalty parameters* $r^{(0)}$ and the rate at which they are changed. Instead of using the vector r containing a different penalty parameter for each constraint, we can use the same penalty parameter r (scalar) for all of them. But using a different penalty parameter for each constraint is beneficial for problems with poorly scaled constraint functions. Also, we can develop a scheme in which each penalty parameter is changed depending on the behaviour of the corresponding constraint.

The *main considerations* in selecting the penalty parameter sequence are as follows.[19]

1. The parameters r should eventually become larger than a threshold level necessary to bring to bear the positive features of the multiplier iterations.
2. The initial parameters $r^{(0)}$ should not be so large that ill-conditioning is forced upon the unconstrained minimization routine too early.
3. The parameters r are not increased too slowly, at least in the early minimizations, to the extent that the multiplier iteration has a poor convergence rate.

These requirements are to some extent contradictory. In addition, for non-convex problems, it is difficult to know *a priori* the corresponding threshold level for the penalty parameters. The augmented Lagrangian attains a very important feature if the penalty parameters are increased beyond their threshold levels. This important feature is the existence of a saddle point[37] which does not require convexity of cost function or constraint functions.

The penalty parameters play an important role in achieving better convergence properties. This is illustrated from the result: if penalty parameters $r_i \to \infty$ $(i = 1, m)$ and θ_i $(i = 1, m)$ are fixed, then $g_i \to 0$ $(i = 1, l)$, $\max(g_i, -\theta_i) \to 0$ $(i = l + 1, m)$, and they behave asymptotically as

$$g_i \doteq \frac{\text{constant}}{r_i}; \quad i = 1, l \tag{62}$$

$$\max(g_i, -\theta_i) \doteq \frac{\text{constant}}{r_i}; \quad i = l + 1, m \tag{63}$$

Thus it is important to increase penalty parameters so as to force iterates of Lagrange multipliers u into a region about u* in which local convergence is guaranteed by local duality theory.[34] Once in this region, the penalty parameters could be held fixed and only the Lagrange multipliers u varied such that $u \to u*$ at the appropriate rate.

3.8. A multiplier algorithm

Based on all the foregoing considerations, the following multiplier algorithm due to Powell is stated:

Algorithm B: Powell's multiplier algorithm

Step 1. Set $k = 0$, $K = \infty$; estimate vectors $\mathbf{x}^{(0)}$, $\boldsymbol{\theta}^{(0)}$, $\mathbf{r}$ and scalars $\alpha > 1$, $\beta > 1$, $\varepsilon > 0$, where ε is the desired accuracy; α is used to enforce sufficient decrease in the constraint violations, and β is used to increase the penalty parameters.

Step 2. Set $k = k + 1$.

Step 3. Minimize $\Phi(\mathbf{x}, \boldsymbol{\theta}^{(k)}, \mathbf{r})$ of (12) or (13) with respect to $\mathbf{x}$. Let $\mathbf{x}^{(k)}$ be the best point obtained in this step.

Step 4. Evaluate $g_i(\mathbf{x}^{(k)})$; $i = 1, m$. Set $\bar{K}$ of (43) and check for convergence criteria in (44). If these criteria are satisfied, then stop. Otherwise, establish the following sets of equality and inequality constraints whose violation did not improve by the factor α:

$$I_E = \{i: |g_i| > K/\alpha, i = 1, l\} \qquad \text{(equalities)} \qquad (64)$$

$$I_I = \{i: |\max(g_i, -\theta_i)| > K/\alpha, i = l + 1, m\} \qquad \text{(inequalities)} \qquad (65)$$

Step 5. If $\bar{K} \geqslant K$ (i.e. constraint violation did not improve), set $r_i = \beta r_i$ and $\theta_i^{(k+1)} = \theta_i^{(k)}/\beta$ for all $i \in I_E \cup I_I$, and go to Step 2. That is, increase the penalty parameters by the factor β and reduce the corresponding θ_i by the same factor, thus keeping the multipliers unchanged.

Step 6. Update $\theta_i^{(k)}$ by setting (note that this step is executed only when the constraint violations have improved)

$$\theta_i^{(k+1)} = \theta_i^{(k)} + g_i(\mathbf{x}^{(k)}); \quad i = 1, l \qquad (66)$$

$$\theta_i^{(k+1)} = \theta_i^{(k)} + \max(g_i(\mathbf{x}^{(k)}), -\theta_i^{(k)}); \quad i = l + 1, m \qquad (67)$$

If $\bar{K} \leqslant K/\alpha$ (constraint violation has improved by the factor α), set $K = \bar{K}$ and go to Step 2.

Step 7. Set $r_i = \beta r_i$ and $\theta_i^{(k+1)} = \theta_i^{(k+1)}/\beta$ for each $i \in I_E \cup I_I$ (note that this step is executed only when the constraint violations do not improve by the factor α). Set $K = \bar{K}$ and go to Step 2.

For engineering design problems, it is necessary to impose the design variable bounds during unconstrained minimization in Step 3 to prevent absurd designs. This needs to be treated in the unconstrained minimization routine.

The initial choices of θ, $\mathbf{r}$, α and β are very important. Usually initial θ_i ($i = 1, m$) are set to zero. The initial choices of penalty parameters r_i ($i = 1, m$) are important from the point of view of the rate of convergence, as discussed before. Penalty parameters are usually chosen to satisfy $1/2 r_i g_i^2(\mathbf{x}^{(0)}) = |\Delta f(\mathbf{x}^{(0)})|$, where $|\Delta f(\mathbf{x}^{(0)})|$ is the expected decrease in the cost function at the initial design point $\mathbf{x}^{(0)}$. This requires some prior knowledge about the problem. Another way to choose the initial penalty parameters is to require the cost function and the generalized penalty functional (see (4)) or the norm of their gradients to be equal at the initial point $\mathbf{x}^{(0)}$. $\alpha = 1\cdot5$ and $\beta = 10$ were used by Belegundu and Arora[10] and Paeng and Arora.[68] On the other hand, $\alpha = 4$ and $\beta = 10$ were chosen by Powell.[74] The initial design $\mathbf{x}^{(0)}$ can be feasible or infeasible. To prevent excessive iterations, a limit on the maximum number of function and gradient evaluations may also be imposed.

3.9. Effect of inexact minimization

It is tempting to terminate the unconstrained minimization process in Step 3 of the algorithm with an inexact solution. This way the number of function and gradient evaluations may be reduced. Here we study the effect of inexact minimizations by considering the same example as considered in Section 3.2.

Let us first consider that accurate minimum of the augmented functional is obtained in Step 3 of the algorithm. Let $r = 10$ for the entire procedure and $\theta = 0$ initially. Then after the first unconstrained minimization the solution is $x = 14/12$. The update procedure (11) yields $\theta = 0 + \max(0.19, 0) \approx 0.19$. Thus θ is increased from zero, indicating that minimum of the augmented functional will shift to the left. The other observation is that θ is almost equal to 0.2 and, after the next step, the exact minimum of the augmented functional is $12.1/12$, which is very close to the required solution.

Now consider an inexact minimum of the augmented functional in Step 3 as $x = 18/12$. The update procedure (11) yields $\theta = 0 + \max(0.5, 0) = 0.5$. In this case also θ is increased and the minimum of the augmented functional will be shifted to the left in the next iteration. However, the shift is too much in this case. The exact minimum in Step 3 with $\theta = 0.5$ is $x = 9/12$, and (11) yields $\theta = 0.5 + \max(-0.25, -0.5) = 0.25$. This will shift the exact minimum of the augmented functional to the right and closer to the required solution. Let us assume again that the exact solution $x = 9/12$ is not obtained; instead an inaccurate neighbouring solution $x = 10/12$ is used. Then, (11) yields $\theta = 0.5 + \max(-0.167, -0.5) \approx 0.433$, and this will also shift the minimum to the right, but not as close to the required solution as when the exact minimum of the functional is used. Now let us consider another inaccurate solution as $x = 13/12$, which is closer to the required solution; however, it is far away from the exact solution $x = 9/12$ of the augmented functional. In this case, (11) yields $\theta = 0.5 + \max(0.084, -0.5) \approx 0.584$, which indicates that the minimum of the functional will shift farther to the left rather than to the right.

In conclusion, it can be seen that an exact minimum of the augmented functional is necessary for each unconstrained minimization to ensure proper shift of the minimum towards the required solution. With an inaccurate minimum, the move towards the final solution may not be appropriate, resulting in convergence difficulty. Since an exact minimum of the augmented functional for engineering applications is inefficient, the inexact minimum may be used in some iterations as long as in some others an accurate minimum is used.

4. CONTINUOUS MULTIPLIER UPDATE METHODS

In the multiplier methods discussed in the previous section, the augmented Lagrangian is minimized for a fixed value of the Lagrange multipliers and penalty parameters. At the end of unconstrained minimization, the multipliers and penalty parameters are updated. Such algorithms requiring exact unconstrained minimization can be inefficient. This is particularly true for large scale engineering problems. Therefore, it may be more efficient to update the Lagrange multipliers more often; e.g. after a few steps of unconstrained minimization.[61,62,64,96] An algorithm that requires only one step of a quasi-Newton method before the multipliers and the penalty parameters are updated has been proposed.[89] Methods based on this idea have been characterized as diagonalization of the multiplier methods;[21,89] however, we shall call them *continuous multiplier update methods*. For better results, the update procedures for x and u should be compatible, i.e. they should both use either first-order information only, or first- and second-order information. The purpose of this section is to study fundamentals of this class of methods.

Results on global convergence and rate of convergence of the continuous update methods can be found in several references.[36,38] Analysis of different procedures with continuous updating has been presented.[38] Convergence with different multiplier update procedures and secant update procedures for the Hessian has been investigated for equality constrained problems.[36] It has been shown that at least a linear rate of convergence is guaranteed by all the update procedures. Automatic increase of the penalty parameters after each iteration of unconstrained minimization, depending on the value of a test function, has been proposed.[39,66] Global convergence of this algorithm is shown with bounded penalty.

For inequality constraints, usually a potential constraint strategy is used. Use of squared slack variables also results in a similar strategy. A formulation for the general equality–inequality problem has been presented using squared slack variables that does not result in increased dimension or numerical instability.[91]

The basic idea of continuous multiplier update methods can be understood by relating them to *sequential quadratic programming*. This will be done in the sequel. It will be shown that one iteration of sequential quadratic programming corresponds to one iteration of the unconstrained minimization algorithm for the augmented Lagrangian, followed by an update of the multipliers. We first consider equality constraints only.

Consider first only the equality constrained problem. The first-order necessary conditions give a system of $n + l$ non-linear equations in $n + l$ unknowns as follows:

$$\begin{bmatrix} \nabla L(\mathbf{x}, \mathbf{u}) \\ \mathbf{g}(\mathbf{x}) \end{bmatrix} = \mathbf{0} \tag{68}$$

The methods that solve the system of equations (68) to solve Problem P are called *Lagrange methods*.[19,59] A Newton step to solve the non-linear system in (68) gives

$$\begin{bmatrix} \mathbf{B}^{(k)} & \nabla \mathbf{g}(\mathbf{x}^{(k)}) \\ \nabla \mathbf{g}(\mathbf{x}^{(k)})^{\mathrm{T}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x}^{(k+1)} \\ \Delta \mathbf{u}^{(k+1)} \end{bmatrix} = \begin{bmatrix} - \nabla L(\mathbf{x}^{(k)}, \mathbf{u}^{(k)}) \\ - \mathbf{g}(\mathbf{x}^{(k)}) \end{bmatrix} \tag{69}$$

where $\Delta \mathbf{x}^{(k+1)} = \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}$ and $\Delta \mathbf{u}^{(k+1)} = \mathbf{u}^{(k+1)} - \mathbf{u}^{(k)}$. If $\mathbf{B}^{(k)} = \nabla^2 L(\mathbf{x}^{(k)}, \mathbf{u}^{(k)})$ we have exact Newton method. However, for iterations in (69) to work, we need matrix $\mathbf{B}^{(k)}$ to be positive definite and matrix $\nabla \mathbf{g}(\mathbf{x}^{(k)})$ to have full column rank. The latter requirement is satisfied if regularity of $\mathbf{x}^{(k)}$ is assumed. To satisfy the first requirement, we can replace L by Φ given in (12), and take $\mathbf{B} = \nabla^2 \Phi$, since $\nabla^2 \Phi$ can be made positive definite by suitably increasing the penalty parameters. In order to avoid explicit computation of $\nabla^2 \Phi$, a positive definite secant approximation for it can also be used.[59]

Observe that, if we add $\nabla \mathbf{g}(\mathbf{x}^{(k)}) \mathbf{u}^{(k)}$ to the top part of (69), we get

$$\begin{bmatrix} \mathbf{B}^{(k)} & \nabla \mathbf{g}(\mathbf{x}^{(k)}) \\ \nabla \mathbf{g}(\mathbf{x}^{(k)})^{\mathrm{T}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x}^{(k+1)} \\ \mathbf{u}^{(k+1)} \end{bmatrix} = \begin{bmatrix} - \nabla f(\mathbf{x}^{(k)}) \\ - \mathbf{g}(\mathbf{x}^{(k)}) \end{bmatrix} \tag{70}$$

Now we can show how some optimization methods are related to (70). Consider the standard quadratic subproblem defined in the sequential quadratic programming (SQP) methods, as

Problem QP

$$\text{Minimize} \quad \nabla f(\mathbf{x}^{(k)})^{\mathrm{T}} \mathbf{d}^{(k)} + \tfrac{1}{2} \mathbf{d}^{(k)\mathrm{T}} \mathbf{B}^{(k)} \mathbf{d}^{(k)} \tag{71}$$

$$\text{subject to} \quad \nabla \mathbf{g}(\mathbf{x}^{(k)})^{\mathrm{T}} \mathbf{d}^{(k)} + \mathbf{g}(\mathbf{x}^{(k)}) = 0 \tag{72}$$

where $\mathbf{d}^{(k)}$ is the search direction, and $\mathbf{B}^{(k)}$ is some approximation to $\nabla^2 L(\mathbf{x}^{(k)}, \mathbf{u})$ with the vector $\mathbf{u}$ corresponding to the multipliers obtained during solution of the Problem QP at the previous step.

The first-order necessary conditions for the Problem QP are

$$\mathbf{B}^{(k)}\mathbf{d}^{(k)} + \nabla g(\mathbf{x}^{(k)})\mathbf{u} = -\nabla f(\mathbf{x}^{(k)}) \tag{73}$$

$$\nabla g(\mathbf{x}^{(k)})^{\mathrm{T}}\mathbf{d}^{(k)} = -g(\mathbf{x}^{(k)}) \tag{74}$$

Observe that the system of equations in (70) is the same as (73) and (74) with $\mathbf{d}^{(k)} = \Delta \mathbf{x}^{(k+1)}$. Also the multipliers $\mathbf{u}$ obtained by solving the Problem QP corresponding to the constraints in (72) are nothing but $\mathbf{u}^{(k+1)}$ in (70). So solution of the QP at each step of the SQP methods represents a Newton-like step in (70). With this observation, we see that one iteration of the SQP method updates $\mathbf{x}$ and $\mathbf{u}$ simultaneously. Once $\mathbf{d}$ and $\mathbf{u}$ have been determined, $\mathbf{u}$ is kept fixed and a step size is computed in the direction $\mathbf{d}$ to minimize a descent function which can be taken as an augmented Lagrangian functional. Only inaccurate minimization is usually done in this step to improve efficiency.

The SQP methods for the general inequality–equality constrained problem are derived from this connection of the Problem QP with the Newton method for solving the non-linear set of necessary conditions.[59] In this case, the quadratic programming subproblem is defined as

$$\text{Minimize} \quad \nabla f(\mathbf{x}^{(k)})^{\mathrm{T}} \mathbf{d}^{(k)} + \tfrac{1}{2} \mathbf{d}^{(k)\mathrm{T}} \mathbf{B}^{(k)} \mathbf{d}^{(k)}$$

$$\text{subject to} \quad \nabla g_i(\mathbf{x}^{(k)})^{\mathrm{T}} \mathbf{d}^{(k)} + g_i(\mathbf{x}^{(k)}) = 0; \quad i = 1, l$$

$$\nabla g_i(\mathbf{x}^{(k)})^{\mathrm{T}} \mathbf{d}^{(k)} + g_i(\mathbf{x}^{(k)}) \leqslant 0; \quad i = l+1, m$$

An attractive feature of these SQP methods is that it applies directly to inequality as well as equality constraints without the use of a potential constraint strategy, although such a strategy might be used to define the quadratic subproblem.[2]

With the foregoing analysis, the relationship between the SQP and the multiplier methods emerges. It can be seen that, if the Lagrange multipliers are updated at the end of each unconstrained minimization iteration using solution of the QP subproblem, then the multiplier method iteration is the same as the SQP iteration. This can lead to better understanding of the multiplier methods, and a theoretical basis for the continuous multiplier update methods.

The idea of a multiplier update method is to *avoid complete solution* of the system of equations in (70) by directly approximating $\mathbf{u}^{(k+1)}$. Thus, by continuous multiplier update methods we mean an iterative procedure such as

$$\mathbf{u}^{(k+1)} = \mathbf{U}(\mathbf{x}^{(k)}, \mathbf{u}^{(k)}) \tag{75}$$

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{d}^{(k)} \tag{76}$$

where $\mathbf{U}$ is one of the multiplier update procedures discussed in Section 3.7.3, and $\mathbf{d}^{(k)}$ is some suitable direction. From the top part of the system of equations in (70), we get

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \mathbf{B}^{(k)^{-1}}[\nabla f^{(k)} + \nabla g(\mathbf{x}^{(k)})\mathbf{u}^{(k+1)}]$$

Using the definition of the Lagrangian in (3), the foregoing equation becomes

$$\mathbf{d}^{(k)} = -\mathbf{B}^{(k)^{-1}} \nabla L(\mathbf{x}^{(k)}, \mathbf{u}^{(k+1)})$$

In general we can take

$$\mathbf{d}^{(k)} = -\mathbf{B}^{(k)^{-1}} \nabla \Phi(\mathbf{x}^{(k)}, \mathbf{u}^{(k+1)}) \tag{77}$$

where $\mathbf{B}^{(k)}$ is an approximation to $\nabla^2 \Phi(\mathbf{x}^*, \mathbf{u}^*)$. Using (77) in the lower part of the system of equations in (70), we get

$$\nabla g(\mathbf{x}^{(k)})^T \mathbf{B}^{(k)^{-1}} \nabla \Phi(\mathbf{x}^{(k)}, \mathbf{u}^{(k+1)}) = g(\mathbf{x}^{(k)}) \tag{78}$$

Solving for $\mathbf{u}^{(k+1)}$ we get

$$\mathbf{u}^{(k+1)} = [\nabla g(\mathbf{x}^{(k)})^T \mathbf{B}^{(k)^{-1}} \nabla g(\mathbf{x}^{(k)})]^{-1} [g(\mathbf{x}^{(k)}) - \nabla g(\mathbf{x}^{(k)})^T \mathbf{B}^{(k)^{-1}} \nabla f(\mathbf{x}^{(k)})] - \mathbf{R}g(\mathbf{x}^{(k)}) \tag{79}$$

If matrix $\mathbf{D}$ is taken as $\mathbf{B}^{(k)}$ in (59), then the expression for $\mathbf{U}_T$ is the same as that for $\mathbf{u}^{(k+1)}$ in (79). Therefore, we can take $\mathbf{U} = \mathbf{U}_T$ in (75). This way the continuous multiplier update method is the same as the SQP method.[59,89,90]

Now variations of the iterative procedure in (75) and (76) are discussed. Instead of taking $\mathbf{U} = \mathbf{U}_T$ any other multiplier update procedure discussed in Section 3.7.3 can be used. Instead of one step of (76) and then one step of (75), we may use more than one step of (76) and then one step of (75). Consider the case where several steps of (76) are taken with constant $\mathbf{u}$ until convergence, i.e. until $\nabla L(\mathbf{x}, \mathbf{u}) = \mathbf{0}$ is satisfied. This corresponds to the multiplier methods discussed before.[59]

An algorithm based on continuous multiplier update methods can be used to solve engineering problems such that computation of the gradient of individual constraints is not needed. For example, Algorithm B in Section 3.8 could be used as a continuous multiplier update algorithm with the Step 3 modified as

$$\mathbf{d}^{(k-1)} = -\nabla \Phi(\mathbf{x}^{(k-1)}, \boldsymbol{\theta}^{(k)}, \mathbf{r}^{(k)}) + \beta_c \mathbf{d}^{(k-2)} \tag{80}$$

$\mathbf{x}^{(k)}$ is determined from

$$\Phi(\mathbf{x}^{(k)}) = \min_{\alpha_s} \Phi(\mathbf{x}^{(k-1)} + \alpha_s \mathbf{d}^{(k-1)})$$

and β_c is a parameter related to the conjugate gradient method. The resulting algorithm will be a gradient or conjugate gradient algorithm with the updating of the multipliers $\mathbf{u}$ (or $\boldsymbol{\theta}$) and the penalty parameters $\mathbf{r}$ imbedded into each iteration of the descent method. Note that since $\boldsymbol{\theta}$ and $\mathbf{r}$ are changing, the augmented Lagrangian is changing at each iteration. This technique of changing $\mathbf{u}$ at the end of each one-dimensional search has been used successfully.[65,96]

In (80), $\beta_c = 0$ corresponds to the steepest descent method. However, β_c corresponding to some conjugate gradient method may be chosen, such as the Fletcher–Reeves method:

$$\beta_c = \frac{\|\nabla \Phi(\mathbf{x}^{(k-1)}, \boldsymbol{\theta}^{(k)}, \mathbf{r}^{(k)})\|^2}{\|\nabla \Phi(\mathbf{x}^{(k-2)}, \boldsymbol{\theta}^{(k-1)}, \mathbf{r}^{(k-1)})\|^2} \tag{81}$$

This choice of β_c was used by Tripathi and Narendra.[96] However, it appears to be more appropriate to use $\|\nabla \Phi(\mathbf{x}^{(k-2)}, \boldsymbol{\theta}^{(k)}, \mathbf{r}^{(k)})\|^2$ in the denominator of (81). Since it is not intended to calculate gradients of individual constraints, it is not easy to calculate $\nabla \Phi(\mathbf{x}^{(k-2)}, \boldsymbol{\theta}^{(k)}, \mathbf{r}^{(k)})$ using $\boldsymbol{\theta}^{(k)}, \mathbf{r}^{(k)}$ and other available information. One needs to calculate $\nabla \Phi(\mathbf{x}^{(k-1)}, \boldsymbol{\theta}^{(k)}, \mathbf{r}^{(k)})$ and $\nabla \Phi(\mathbf{x}^{(k-2)}, \boldsymbol{\theta}^{(k)}, \mathbf{r}^{(k)})$ at each iteration, starting from scratch. This increases the computational effort. To overcome this difficulty, (81) has been used to calculate β_c, even though it produces discontinuity in the augmented Lagrangian function. It has also been suggested to execute Step 3 of the algorithm n times if conjugate gradient steps are taken before any multiplier or penalty parameter is updated.[65]

Further, a modification to the Hestenes–Powell update formula in (55) has been suggested for equality constraints as[65]

$$\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} + sg(\mathbf{x}^{(k)}) \tag{82}$$

where s is a scalar obtained by minimizing (with respect to s) $\|\nabla L(\mathbf{x}, \mathbf{u}^{(k+1)})\|^2$. However, this modification needs the gradient of individual constraints. It is also suggested to calculate the penalty parameters such that the order of magnitude of the penalty term with the multipliers $\left(\text{i.e.}\right.$

$\left.\sum_{i=1}^{m} u_i g_i\right)$ and the penalty term with the penalty parameters $\left(\text{i.e. } \dfrac{1}{2} \sum_{i=1}^{m} r_i g_i^2\right)$ are equal (see Section 3.1).

In (80), $\mathbf{d}^{(k-1)}$ could also be calculated as

$$\mathbf{B}^{(k-1)}\mathbf{d}^{(k-1)} = -\nabla\Phi(\mathbf{x}^{(k-1)}, \boldsymbol{\theta}^{(k)}, \mathbf{r}^{(k)}) \tag{83}$$

where $\mathbf{B}^{(k-1)}$ is an approximation to the Hessian matrix $\nabla^2\Phi(\mathbf{x}, \boldsymbol{\theta}, \mathbf{r})$. Note that all three arguments of Φ are changing at each iteration. The matrix $\mathbf{B}$ is updated at every iteration using some secant update procedure such that it satisfies the quasi-Newton condition[59]

$$\mathbf{B}^{(k)}\mathbf{s}^{(k-1)} = \mathbf{y}^{(k-1)}$$

where
$$\mathbf{s}^{(k-1)} = \mathbf{x}^{(k)} - \mathbf{x}^{(k-1)} \quad \text{and}$$

$$\mathbf{y}^{(k-1)} = \nabla\Phi(\mathbf{x}^{(k)}, \boldsymbol{\theta}^{(k)}, \mathbf{r}^{(k)}) - \nabla\Phi(\mathbf{x}^{(k-1)}, \boldsymbol{\theta}^{(k)}, \mathbf{r}^{(k)}) \tag{84}$$

for consistent secant update.[89] Again, since it is not intended to calculate gradients of individual constraints at each iteration, we need to compute $\nabla\Phi(\mathbf{x}^{(k)}, \boldsymbol{\theta}^{(k)}, \mathbf{r}^{(k)})$ and $\nabla\Phi(\mathbf{x}^{(k-1)}, \boldsymbol{\theta}^{(k)}, \mathbf{r}^{(k)})$ from scratch. If gradients of individual constraints are available, we just need to store them at the previous iteration. Then $\nabla\Phi(\mathbf{x}^{(k-1)}, \boldsymbol{\theta}^{(k)}, \mathbf{r}^{(k)})$ can be calculated using the available information with very little effort.

If gradients of individual constraints are available, then a more accurate multiplier update procedure, such as in (56) to (60), can be used. Fontecilla et al.[36] referred to these methods as a class of quasi-Newton methods for equality constrained optimization. Under certain assumptions, the following important results were proved: Let $\mathbf{U}$ be some multiplier update procedure and matrix $\mathbf{B}$ be updated using a secant update procedure in the continuous multiplier update methods. Then

1. The continuous multiplier update method is locally linearly convergent in $\mathbf{x}$.
2. If $\mathbf{U}$ is the multiplier update procedure in (59) with matrix $\mathbf{D} = \mathbf{B}$, then the continuous multiplier method is locally superlinearly convergent in $\mathbf{x}$.

These results are a generalization and extension of the results obtained by Glad.[38]

Only limited numerical experimentation has been done in the literature with these methods.[11] Problems with only equality constraints have been considered. For some simple mathematical problems the methods have performed better than the original multiplier methods in terms of reliability and efficiency. Recently, a continuous update method was used for launch vehicle trajectory optimization with equality constraints.[1] A modified form of the update formula (59) was used. This modification is similar to the one in (82) for the Hestenes–Powell formula. Here, $\mathbf{u}^{(k+1)}$ is also found such that

$$\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} + s\Delta\mathbf{u}^{(k)} \tag{85}$$

minimizes (with respect to s) $\|\nabla L(\mathbf{x}, \mathbf{u}^{(k+1)})\|^2$, where $\Delta\mathbf{u}^{(k)}$ is a change in the multipliers given by (59). This continuous multiplier update algorithm is based on the method suggested by Polak and Tits.[72] Specifically, the penalty parameter modification procedure is similar. Better performance of the method over other methods has been reported.

5. EXACT PENALTY METHODS

Multiplier methods require solution of several unconstrained problems. After every unconstrained minimum several parameters are suitably adjusted to start the next unconstrained minimization. However, in the case of exact penalty methods, solution of only one unconstrained problem is required.[19] At the beginning, if proper values of the parameters are chosen, then (x^*, u^*) corresponding to the unconstrained minimum satisfies Kuhn–Tucker necessary conditions for the Problem P. The method to achieve this property uses exact penalty functionals. There are *two classes* of exact penalty functionals: *non-differentiable and differentiable*. Generally, non-differentiable functionals have been used as descent functionals in linearization algorithms for minmax and non-linear programming problems (e.g. sequential quadratic programming).

There are two types of *differentiable exact penalty* functionals: the first one depends only on x, and the second one depends on x and u. A *functional depending on x only* can be formed by explicitly *substituting an approximation formula* for u into the augmented Lagrangian. Then the form of the augmented Lagrangian given in (12) for only equality constraints becomes

$$\Phi(x) = f(x) + \frac{1}{2} \sum_{i=1}^{l} r_i g_i(x)^2 + \sum_{i=1}^{l} g_i(x) u_i(x) \tag{86}$$

Any of the multiplier approximation formulas discussed in Section 3.7.3 can be used. Note that all multiplier approximation formulas use gradients of individual constraints (i.e. ∇g). The exact penalty functional with approximation formula U_F in (56) was introduced by Fletcher.[31] Similar penalty functionals have been discussed by Fletcher and Lill,[35] Fletcher,[32] Mukai and Polak,[66] Tapia[89,90] and Luenberger.[59] Glad and Polak[39] introduced a similar penalty functional for equality–inequality constrained problems. That functional was shown to be exact and continuously differentiable for only inequality constraints by Di Pillo and Grippo.[27] It was modified to include additional barrier terms on the boundary of the compact constraint set.[28]

A continuously *differentiable exact penalty functional depending on both x and u* for equality constrained problems was introduced by Di Pillo and Grippo.[25] It was later extended to inequality constraints.[26] A similar functional for inequality constraints was proposed by Lucidi.[58] An implicit potential set strategy for inequality constraints has also been developed.[18]

The augmented Lagrangian function introduced by Di Pillo and Grippo[25] is given as (only equality constraints are considered)

$$\Phi(x, u, r_a, r_c) = L(x, u) + \frac{r_a}{2} \sum_{i=1}^{l} [g_i(x)]^2 + \frac{r_c}{2} |\nabla L(x, u)|^2 \tag{87}$$

where $L(x, u)$ is the Lagrangian defined in (3), and $r_a > 0$ and $r_c > 0$ are penalty parameters. The square of the first-order necessary conditions is added to the Lagrangian to form an exact penalty functional. In this case, (x^*, u^*) is related to a local minimum of an exact penalty functional, rather than to a saddle point as in the multiplier methods. As a consequence, theoretical results are different from those for multiplier methods. From a theoretical standpoint, saddle point and duality theories no longer hold. The approach with the functional in (87) increases the dimensionality of the problem as both x and u are simultaneously treated as unknowns in the minimization process.

The presence of first-order derivatives of the constraints in the penalty functionals in (86) and (87) makes the problem more complicated, even if it can avoid the sequence of unconstrained minimizations. Computation of the gradient of individual constraints cannot be avoided.

The choice of an unconstrained minimization algorithm is an important problem with these augmented Lagrangians. A quasi-Newton algorithm may be a good choice. But the gradient of

Φ in (87) requires the Hessian of the Lagrangian, and the gradient of (86) requires second-order derivatives of individual constraints (since Lagrange multiplier expressions contain gradients of constraints). The computational aspects associated with unconstrained minimization have been discussed by Di Pillo et al.[29] In general, the exact penalty methods are difficult to use. They have not been applied to practical applications.

6. EXPONENTIAL PENALTY METHODS

The purpose of this section is to describe some optimization techniques that are similar to the Hestenes–Powell multiplier method. These methods involve exponential functions in updating the multipliers and in defining the augmented functionals. The methods do not require gradients of individual constraints, and have been discussed for only the inequality constrained problem (ICP) in the literature. They have been presented from three entirely different points of view. We shall first present the basic ideas of these three derivations, and then propose a procedure to extend the methods for the general Problem P.

Templeman and Li[94] transformed the ICP to an *equivalent surrogate form* (SICP) as

$$\underset{\mathbf{x}}{\text{minimize}}\, f(\mathbf{x}) \text{ subject to}$$

$$\sum_{i=1}^{m} v_i g_i(\mathbf{x}) = 0; \quad \sum_{i=1}^{m} v_i = 1; \quad \text{and} \quad v_i \geqslant 0; \quad i = 1, m \tag{88}$$

The parameters v_i in (88) are called *surrogate multipliers*. These give relative values for the Lagrange multipliers, as will be seen later. However, in this section, we will refer to them as multipliers.

The desired solution $\mathbf{x}^*$ of the ICP is considered to be sought indirectly through a sequence of solutions of the SICP. This approach assumes that the ICP and SICP are equivalent at the solution point. The set of multipliers $\mathbf{v}^*$ exists and can be found such that $\mathbf{x}^*$ that solves the SICP with $\mathbf{v}^*$, also solves the ICP. The solution procedure is to select initial values of the multipliers and solve the problem in (88). The multipliers are then updated using the following formula:

$$v_i^{(k+1)} = \frac{\exp(\beta_e g_i(\mathbf{x}^{(k)}))}{\sum_{j=1}^{m} \exp(\beta_e g_j(\mathbf{x}^{(k)}))}; \quad i = 1, m \tag{89}$$

where β_e is required to be an increasing positive number.

The basis for the formula (89) is found in Shannon (informational) entropy measure and Jaynes' maximum entropy formalism. The measure and formalism can be explained by considering a random process. In this process, a discrete random variable z can have any value from a set $\{z_1, \ldots, z_N\}$. Let p_i be the probability for z to have the value z_i, $i = 1, N$. The maximum entropy formalism is concerned with assigning the least biased values to the probabilities p_i using only information that can be inferred from the random process itself.

The informational entropy function, called the Shannon entropy, is given as[83]

$$S_e = -K_e \sum_{i=1}^{N} p_i \ln(p_i)$$

where K_e is a positive constant. According to the maximum entropy formalism,[53] to get proper values of probability p_i, the function S_e should be maximized over the variables p_i, $i = 1, N$ subject to constraints that depend on any available information for the random process. One

additional constraint

$$\sum_{i=1}^{N} p_i = 1$$

that represents the normality condition is also used. If the available information on the random process is in the form of m_e ($m_e < N - 1$) equality constraints, the problem can be posed as an optimization problem. A solution to the problem can be obtained using the necessary conditions.[95] The solution involves m_e Lagrange multipliers corresponding to the m_e constraints representing available information.

In deriving the expression in (89) multipliers v_i play the role of the discrete probabilities p_i. Owing to constraints in (88), these multipliers need to satisfy the following constraints:

$$\sum_{i=1}^{m} v_i^{(k+1)} = 1 \tag{90}$$

and

$$\sum_{i=1}^{m} v_i^{(k+1)} g_i(\mathbf{x}^{(k+1)}) = 0 \tag{91}$$

Since $\mathbf{x}^{(k+1)}$ is not known at this stage, (91) is modified to

$$\sum_{i=1}^{m} v_i^{(k+1)} g_i(\mathbf{x}^{(k)}) = \varepsilon \tag{92}$$

where ε represents the error introduced by using $\mathbf{g}(\mathbf{x}^{(k)})$ in place of $\mathbf{g}(\mathbf{x}^{(k+1)})$. The updated multipliers $\mathbf{v}^{(k+1)}$ should be such that

$$v_i^{(k+1)} \begin{cases} > v_i^{(k)} & \text{if } g_i(\mathbf{x}^{(k)}) > 0 \\ = v_i^{(k)} & \text{if } g_i(\mathbf{x}^{(k)}) = 0 \\ < v_i^{(k)} & \text{if } g_i(\mathbf{x}^{(k)}) < 0 \end{cases} \tag{93}$$

Therefore, the value of ε should be always positive. Also, it is expected that ε will go to zero as we approach the solution. To satisfy these conditions, β_e in (89) is taken to be an increasing positive number.

To assign proper values to $\mathbf{v}^{(k+1)}$ subjected to constraints in (90) and (92), a criterion is taken from the maximum entropy formalism.[94] This results in an optimization problem where the entropy of the multipliers $\mathbf{v}^{(k+1)}$ is maximized subject to the constraints in (90) and (92). The equality constraint in (92) is a single constraint ($m_e = 1$) representing the available information. Equation (89) is a solution of the resulting optimization problem obtained using the necessary conditions. The parameter β_e is a Lagrange multiplier corresponding to the constraint in (92). Templeman and Li[94] gave justification for the use of these methods based on probabilistic and information-theoretic terms in the solution process for the inequality constrained problem that is deterministic.

As an alternate derivation for the Lagrange multiplier formula (89), an augmented Lagrangian for the SICP is constructed as

$$\Phi = f(\mathbf{x}) + \alpha_e \sum_{i=1}^{m} v_i g_i + \lambda_e \left(\sum_{i=1}^{m} v_i - 1 \right) - \frac{1}{\rho} \sum_{i=1}^{m} v_i \ln v_i \tag{94}$$

where α_e and λ_e are the Lagrange multipliers for the corresponding constraints in (88). The last term in (94) represents the entropy with ρ as an arbitrary positive constant. The necessary condition with respect to the multipliers v and its simplification using (88) gives

$$v_i = \frac{\exp(\rho\alpha_e g_i(\mathbf{x}))}{\sum\limits_{j=1}^{m} \exp(\rho\alpha_e g_j(\mathbf{x}))}; \quad i = 1, m \tag{95}$$

This is similar to the previous update formula in (89) with $\beta_e = \rho\alpha_e$. Substituting this v into the augmented Lagrangian in (94), and simplifying, we get

$$\Phi = f(\mathbf{x}) + \frac{1}{\rho} \ln\left[\sum_{j=1}^{m} \exp(\rho\alpha_e g_j(\mathbf{x})) \right] \tag{96}$$

This functional is sequentially minimized with increasing values of α_e and some fixed positive ρ to get a solution for the ICP.[56,94]

It is interesting to note that Kreisselmeier and Steinhauser[55] also introduced a *functional* (*without derivation*) *in the context of control design problems* as

$$KS(\mathbf{x}) = \frac{1}{\rho} \ln\left[\sum_{j=1}^{m} \exp(\rho\alpha_e g_j(\mathbf{x})) \right] \tag{97}$$

which is precisely the penalty term in (96). In these problems, a *vector performance index* is optimized where $g_j(\mathbf{x})$, $j = 1, m$ rate different design specifications. In other applications[43-45,84-86] the functional $KS(\mathbf{x})$ represents a *cumulative constraint* for the m inequality constraints $g_j(\mathbf{x}) \leqslant 0, j = 1, m$. $g_j(\mathbf{x})$ may also represent the value at the jth discrete point of a continuum constraint. In that case $KS(\mathbf{x})$ can be used to replace the continuum constraint. $g_j(\mathbf{x})$ may also be a *goal constraint* corresponding to a cost function in *multicriteria optimization problems*.[99] The function $KS(\mathbf{x})$ represents a *composite functional* combining all the inequality constraints and the cost functions. It is important to note that the goal vector corresponding to which the composite functional is formed must be dimensionally homogeneous.[93] It has also been demonstrated that the general minimax problems are closely related to vector optimization problems. As a consequence, a general continuous scalar optimization formulation is derived that uses the functional in (97) corresponding to the minimax problems.

The functional in (97) satisfies the following inequalities:

$$\max_{i=1,m} (g_i) \leqslant KS(\mathbf{x}) \leqslant \max_{i=1,m} (g_i) + \frac{\ln(m)}{\rho} \tag{98}$$

This property is important because it gives some physical insight about the behaviour of the functional. It shows that the KS functional is a conservative envelope function that follows the maximum constraint function among g_i. The larger the value of ρ, the closer the KS functional follows the maximum constraint. For a very large value of ρ, the gradient of the KS functional will follow the gradient of the maximum constraint. This can result in a discontinuous gradient because the maximum constraint can shift from one to another at each iteration. This can cause difficulties in gradient-based optimization procedures. Typical values of the scalar ρ are between 5 and 200. The choice of ρ must be a compromise between the desire to closely follow the maximum constraint on the one hand and the concern for discontinuous and abrupt change in the gradient on the other hand.[7] It is suggested that ρ be chosen as follows: if the constraint

tolerance at optimum is ε (i.e. if $|g_i| \leqslant \varepsilon$ indicates that the constraint is active), then choose ρ as

$$\rho = \frac{\ln(m)}{\varepsilon} \tag{99}$$

In general, a small value of ρ is selected at the start and increased near the solution.

An alternate definition of the KS functional that reduces the numerical difficulties caused by computing the exponential of large numbers is given as

$$KS(\mathbf{x}) = g_{\max} + \frac{1}{\rho} \ln \left[\sum_{i=1}^{m} \exp(\rho g_i(\mathbf{x}) - g_{\max}) \right] \tag{100}$$

where α_e is taken as one and $g_{\max} = \max_{i=1,m} g_i(\mathbf{x})$ is evaluated at $\mathbf{x}$ and taken to be a constant. Both forms of the KS functional produce the same results, subject to roundoff errors. The first derivatives with respect to the design variables $\mathbf{x}$ for both formulations are identical, again subject to roundoff errors. The KS functional in (100) has been successfully used in multicriteria and multilevel structural optimization problems.[7,85,99]

In the following paragraphs, two augmented functionals having exponential penalty terms are given. Following that, a derivation of an exponential functional that is the same as the KS functional is given. ,

In place of the augmented Lagrangian functional in (15), a functional having an exponential penalty term[54] can be used to solve the ICP. This functional is given as

$$\Phi(\mathbf{x}) = f(\mathbf{x}) + \frac{1}{r} \left[\sum_{i=1}^{m} v_i \{\exp(rg_i(\mathbf{x})) - 1\} \right] \tag{101}$$

where r is a scalar penalty parameter. The necessary condition for minimization of $\Phi(\mathbf{x})$ in (101) gives

$$\nabla\Phi(\mathbf{x}) = \nabla f(\mathbf{x}) + \sum_{i=1}^{m} (v_i \exp(rg_i(\mathbf{x}))) \nabla g_i(\mathbf{x}) = 0 \tag{102}$$

Comparison of (102) with (16) suggests the multipliers be updated as

$$v_i^{(k+1)} = v_i^{(k)} \exp(rg_i(\mathbf{x}^{(k)})); \quad i = 1, m \tag{103}$$

Note that the map of the penalty term in (101) has a form similar to the curve in Figure 1. However, in this case the curve passes through the origin and has a smooth curvature.

The foregoing augmented functional in (101) has been used in a few applications, e.g. optimization of multicommodity network flows[98] and electric diode network analysis.[14] In these applications there are phenomena that are described by sharply rising exponential functions. Therefore, considerable difficulty is encountered while solving these problems with other methods. Using the functional in (101) the problems are solved with relative ease. Convergence analysis for non-convex problems relating to this functional and the associated method of multipliers has also been discussed.[67]

An augmented functional with a slightly different exponential penalty term is defined for an inequality constrained problem as

$$\Phi(\mathbf{x}) = f(\mathbf{x}) + \frac{1}{r} \left[\sum_{i=1}^{m} \{\exp(rg_i(\mathbf{x})) - 1\} \right] \tag{104}$$

where r is scalar penalty parameter. The ICP problem is solved by minimizing this functional. The exponential penalty term in (104) has also been minimized to find a feasible point for the system of

inequalities $g_j(\mathbf{x}) \leqslant 0, j = 1, m.$[81] The necessary condition for the functional gives

$$\nabla\Phi(\mathbf{x}) = \nabla f(\mathbf{x}) + \sum_{i=1}^{m} (\exp(rg_i(\mathbf{x})))\nabla g_i(\mathbf{x}) = 0 \tag{105}$$

Comparison of (105) with (16) suggests approximating the multipliers as

$$v_i^{(k+1)} = \exp(rg_i(\mathbf{x}^{(k)})); \quad i = 1, m \tag{106}$$

In this method the penalty parameter r is increased after every unconstrained minimization. The multiplier approximation in (106) is not needed for the optimization process when the exponential penalty term in (104) is used. However, (106) is given here since it will be used later to show how (89) can be derived deterministically.

A functional similar to the one in (97) has also been defined in context of the following problem:[19]

$$\operatorname*{minimize}_{\mathbf{x}} \gamma(\mathbf{g}(\mathbf{x})) \tag{107}$$

where $\mathbf{x}$ belongs to some compact set and γ is some functional defined using the constraints. This problem is a subclass of the general non-differentiable and ill-conditioned optimization problems.[17]

The problem in (107) can be equivalently stated by introducing an additional vector $\mathbf{w} \in R^m$ as

$$\text{minimize} \quad \gamma(\mathbf{g}(\mathbf{x}) - \mathbf{w})$$

$$\text{subject to} \quad \mathbf{w} \leqslant 0$$

Using the exponential penalty term in (101), an augmented functional for this problem can be defined as

$$\Phi(\mathbf{x}) = \gamma(\mathbf{g}(\mathbf{x}) - \mathbf{w}) + \frac{1}{r}\left[\sum_{i=1}^{m} v_i\{\exp(rw_i) - 1\} \right] \tag{108}$$

The corresponding multipliers can be updated according to (103) as

$$v_i^{(k+1)} = v_i^{(k)} \exp(rw_i); \quad i = 1, m \tag{109}$$

The additional variables w_i can be expressed in terms of the variables $\mathbf{x}$.[17] Therefore w_i in (108) and (109) can be eliminated. After elimination, the functional in (108) forms a good approximation to the functional in (107). The augmented Lagrangian in (15) and the corresponding multiplier update procedure in (11) can also be used to get expressions similar to (108) and (109). Inequalities similar to (98) can also be derived for these approximating functions.[13]

Using the exponential penalty term in (108) in connection with the functional

$$\gamma(\mathbf{g}(\mathbf{x})) = \max_{i=1,m} (g_i(\mathbf{x})) \tag{110}$$

yields the twice differentiable functional that approximates (110)[17,19] as

$$\tilde{\gamma} = \frac{1}{r}\ln\left[\sum_{i=1}^{m} v_i \exp(rg_i(\mathbf{x})) \right] \tag{111}$$

The corresponding multiplier update procedure is derived using (109) as

$$v_i^{(k+1)} = \frac{v_i^{(k)} \exp(rg_i(\mathbf{x}))}{\sum_{j=1}^{m} v_j^{(k)} \exp(rg_j(\mathbf{x}))}; \quad i = 1, m \tag{112}$$

where r is a scalar penalty parameter. The multipliers v_i given in (112) satisfy the constraint in (90) and the non-negativity condition. Note that (111) and (112) are derived without using any probabilistic ideas. Inequalities similar to (98) can also be derived for the functional in (111).

If the exponential penalty term in (104) is used instead of that in (101), and the procedure used to derive approximating functional in (111) is followed, we get

$$\bar{\gamma} = \frac{1}{r} \ln \left[\sum_{i=1}^{m} \exp(rg_i(\mathbf{x})) \right] \tag{113}$$

If (106) is used, instead of (103), we get a multiplier update procedure similar to (112) as

$$v_i^{(k+1)} = \frac{\exp(rg_i(\mathbf{x}))}{\displaystyle\sum_{j=1}^{m} \exp(rg_j(\mathbf{x}))}; \quad i = 1, m \tag{114}$$

where r is a scalar penalty parameter. No probabilistic ideas are used to derive (113) and (114). Inequalities similar to (98) can also be derived for the functional in (113).

It is interesting to note that the functional in (113) is the same as the penalty term in (96) and the KS function in (97) with $r = \rho$ ($\alpha_e = 1$). Also, the update procedure in (114) is the same as in (89) and (95) with $r = \beta_e = \rho$ ($\alpha_e = 1$). As noted earlier, derivation of (89), (95) and (96) is based on probabilistic ideas. However, derivation of (113) and (114) is based on only deterministic concepts. The functionals given in (113), (111) and (97) are related to the same functional defined in (110). In conclusion, functionals given in (97) and (113), and the penalty term for the functional in (96), are closely linked. There must be some relationship between the probabilistic ideas used to derive the expressions in (89), (95) and (96) and the deterministic methods used to derive (113) and (114). Thus, a different interpretation of the surrogate problem (SICP) may be possible.

During calculation of $\bar{\gamma}$ in (111), it is possible that computer overflow (or underflow) will occur if $rg_i(\mathbf{x})$ is too large (small). To eliminate this difficulty, an alternate form has been suggested as[19]

$$\tilde{\gamma} = \frac{1}{r} \ln \left\{ \sum_{i=1}^{m} v_i c_i \right\} + \gamma(\mathbf{g}(\mathbf{x})) - \frac{a}{r} \tag{115}$$

with

$$c_i = \begin{cases} \exp\{a - r[\gamma(\mathbf{g}(\mathbf{x})) - g_i(\mathbf{x})]\}; & \text{if } a - r[\gamma(\mathbf{g}(\mathbf{x})) - g_i(\mathbf{x})] > -a \\ 0; & \text{otherwise}; \quad i = 1, m \end{cases} \tag{116}$$

where $\gamma(\mathbf{g}(\mathbf{x}))$ is given in (110) and $a > 0$ is a large scalar such that both e^{-a} and e^{a} lie within the computer range. The alternate form for the multiplier update procedure in (112) becomes

$$v_i^{(k+1)} = \frac{v_i^{(k)} c_i}{\displaystyle\sum_{j=1}^{m} v_j^{(k)} c_j}; \quad i = 1, m \tag{117}$$

where c_i is given in (116).

In the following paragraph, several ways to solve the ICP using exponential functionals in (97), (111) and (113) are suggested.

Converting a constrained optimization problem to an unconstrained problem requires that the cost function be combined with the constraints in some manner as is done with penalty function methods. Using this philosophy, all the cost functions appearing in the multicriteria optimization problem are transformed to equivalent goal constraints at the beginning of each unconstrained minimization.[99] Then the KS functional is used to form a composite functional corresponding to all the resulting constraints. This can also be done for problems with a single cost function.

Another way is to use the KS functional (or the one in (113)) to form a composite functional for the constraints only and add it to the cost function, much like penalty functions. This will give the functional Φ in (96). Still another way would be to use the $\bar{\gamma}$ functional in (111) to form the penalty term and define the augmented functional as

$$\Phi = f(\mathbf{x}) + \frac{1}{r} \ln \sum_{i=1}^{m} v_i \exp(rg_i(\mathbf{x})) \tag{118}$$

Then (112) can be used to update the multipliers v_i at the beginning of each unconstrained minimization.

An advantage of the foregoing forms of augmented functionals over the one in (15) is that they have better *differentiability* properties (e.g. see the point $g_i = -\theta_i$ in Figure 1 corresponding to the functional in (13)). In addition they are defined on feasible as well as infeasible domains unlike some usual penalty functions such as the interior penalty function. One drawback is that they are defined for inequality constraints only. They need to be generalized for equalities as well, which is discussed in the sequel.

For the general Problem P, a new functional is proposed as follows:

$$\Phi(\mathbf{x}, \mathbf{r}) = f(\mathbf{x}) + \sum_{i=1}^{l} (\tfrac{1}{2}r_i g_i(\mathbf{x})^2 + u_i g_i(\mathbf{x}))$$

$$+ \frac{\beta_b}{r_{\text{norm}}} \ln \sum_{i=l+1}^{m} v_i \exp(r_i g_i(\mathbf{x})) \tag{119}$$

where r_{norm} may be selected as

$$r_{\text{norm}} = \left(\sum_{i=l+1}^{m} r_i^2 \right)^{1/2} \tag{120}$$

and β_b is a parameter chosen to balance the penalties due to the terms corresponding to equality and inequality constraints. β_b can be chosen such that at the initial design point $\mathbf{x}^{(0)}$

$$\sum_{i=1}^{l} (\tfrac{1}{2}r_i g_i(\mathbf{x}^{(0)})^2 + u_i g_i(\mathbf{x}^{(0)})) \doteq \frac{\beta_b}{r_{\text{norm}}} \ln \sum_{i=l+1}^{m} v_i \exp(r_i g_i(\mathbf{x}^{(0)})) \tag{121}$$

The multipliers can be updated by the following procedure:

$$u_i^{(k+1)} = u_i^{(k)} + r_i g_i(\mathbf{x}^{(k)}); \quad i = 1, l \tag{122}$$

$$v_i^{(k+1)} = \frac{v_i^{(k)} \exp(r_i g_i(\mathbf{x}^{(k)}))}{\sum_{j=l+1}^{m} v_j^{(k)} \exp(r_j g_j(\mathbf{x}^{(k)}))}; \quad i = l+1, m \tag{123}$$

Initially take $u_i = 0$; $i = 1, l$, $v_i = 1/(m - l)$; $i = l + 1, m$ and some suitable positive values for the penalty parameters.

It can be observed that the term corresponding to the equality constraints in the functional (119) is the same as in the functional (15). The term corresponding to the inequality constraints is similar to the penalty term of the functional (118). The multiplier update procedure in (122) for the equality constraints is the same as suggested by Hestenes.[48] The update procedure in (123) for the inequality constraints is similar to (112). Using this functional and the multiplier update procedures, an algorithm similar to the multiplier methods or continuous multiplier update methods can be developed. Motivation for suggesting the functional in (123) is as follows.

33

1. For engineering problems, evaluation of gradients of individual constraints is avoided, as in Powell's algorithm of Section 3.8.
2. The term corresponding to inequality constraints has a better differentiability property, as noted earlier. Therefore, the algorithm is expected to perform better with this functional than with the one in (15).
3. The functional is defined in the feasible as well as infeasible region.
4. We expect convergence of an algorithm with this functional and multiplier update procedure with bounded penalty parameters, therefore avoiding ill-conditioning. This can be proved by considering equality constraints and inequality constraints separately.

7. DISCUSSION AND CONCLUSIONS

In this paper, multiplier methods for solving general equality–inequality constrained engineering optimization problems are investigated. In these methods, the cost and constraint functions, and certain multipliers, are used to define an unconstrained functional. This functional is then minimized keeping the multipliers fixed. At the end of unconstrained minimization, the multipliers are updated and the procedure repeated. Several important features of these methods are described and discussed. Local convergence based on duality theory is described that gives additional insight into the multiplier update procedures. Global convergence from the penalty functions point of view is discussed.

The sequential unconstrained minimization technique (SUMT) with interior penalty functions forms a simple mathematical programming technique. However, it has some drawbacks, one being that it needs a feasible starting point. Some of these drawbacks can be overcome by using the extended interior penalty functions.[42] Exterior penalty functions can also be used. For all these penalty functions, penalty parameters need to go to infinity for convergence. They cannot handle equality constraints very well.

The multiplier methods were proposed in 1969 to alleviate the theoretical and numerical difficulties of the SUMT. They can also result in substantial computational savings in optimal design of structural and mechanical systems, especially under dynamic loads, since they collapse all constraints into one equivalent functional. The dynamic response and control optimization problems have continuum constraints that are difficult to treat. Substantial computational effort is expended in the treatment of such constraints.[49–52,97] Thus, it is desirable to reduce their number, and that is precisely what the augmented Lagrangian methods do. By summing up all the dynamic inequality constraints and by integrating them over the time interval the augmented functional is defined as[68,69]

$$\Phi(\mathbf{x}, \boldsymbol{\theta}, \mathbf{r}) = f(\mathbf{x}) + \frac{1}{2} \int_{t_0}^{t_f} \sum_{i=1}^{m} r_i [(g_i(\mathbf{x}, t) + \theta_i(t))_+^2 - \theta_i^2(t)] \, dt$$

Equality constraints can also be treated similarly. With this approach, one needs to treat only one transformed functional during the optimization process. Sensitivity of the functional Φ does not need sensitivity of the individual constraints.

All the multiplier update procedures, except for the formulas given in (9) and (11), use gradients of individual constraints. Therefore these formulas are most suitable for engineering applications. However, they require an accurate unconstrained minimum at each step for proper convergence. For engineering applications, the search for an accurate minimum can be inefficient. Therefore, the question of inaccurate unconstrained minimization with the use of formulas (9) and (11) for updating multipliers is investigated. It is shown that, for convergence of the method, accurate minimizations must be used in some iterations.

In engineering optimization we want to avoid calculation of gradients of individual constraints for the following two reasons.

1. The number of constraints is very large. One way of reducing the number of constraints is to use some kind of a potential constraint strategy; the other is to use the multiplier methods.
2. Constraints are generally implicit functions of the design variables. In dynamic response optimization, constraints are also functions of time. There is no efficient and simple way to treat these continuum constraints. One possibility is to approximate them by constraints at several discrete time points. However, the problem of computing gradients of the constraint at each of these points still remains. In the multiplier methods, we need to calculate gradient of one implicit functional only.

Using the adjoint variable method of design sensitivity analysis,[8] $\nabla\Phi$ can be computed without computing ∇g_i assuming that ∇g_i are not used in construction of the Φ itself. It is important to note that for updating the design variable vector $\mathbf{x}$ with gradient or quasi-Newton methods we need $\nabla\Phi$ but not ∇g_i separately. In the Newton method, we need $\nabla^2\Phi$ that can also be calculated without calculating ∇g_i. If we use a multiplier update formula that does not use ∇g_i, we can completely avoid calculation of ∇g_i. Only the formula given in (55) has this feature, but that gives only a linear rate of convergence. It is a challenge to develop multiplier update procedures that use only $\mathbf{g}$, $\nabla\Phi$ and $\nabla^2\Phi$ (i.e. never use ∇g_i) and result in superlinear convergence.

Now the question is how to get a better rate of convergence with all these desirable features. To investigate this, recent methods that are similar to the multiplier methods are also discussed. These are continuous multiplier update methods, exact penalty methods and exponential penalty methods. The *continuous multiplier update methods* are similar to the multiplier methods, but they update the multipliers more often. With proper update procedures, these methods can be more stable and reliable.[11]

The *exact penalty methods* have one desirable feature; the solution is reached in one unconstrained minimization. However, one needs gradients of individual constraints to construct a continuously differentiable exact augmented Lagrangian. Therefore, even with the use of the adjoint variable method of sensitivity analysis, we cannot avoid computation of gradients of individual constraints. Since the construction of a differentiable exact augmented Lagrangian without the use of gradients of individual constraints seems impossible, these methods cannot be recommended for engineering applications.

The *exponential penalty functionals* have been developed for problems with inequality constraints. They have been used for multicriteria and multilevel engineering optimization problems, and a few other applications. However, they can also be extended to solve the general Problem P. For this, a new functional incorporating the exponential penalty term for inequality constraints and regular penalty term for equality constraints is proposed in (119). This can result in an algorithm similar to the multiplier methods. It has an advantage similar to the other multiplier method: gradients of individual constraints are not needed, as seen in the update procedures given in (122) and (123).

There are several decomposition methods in the literature based on some augmented Lagrangians.[40,87,92] These methods are applicable to *separable problems*. They are similar to the methods discussed in this paper in the sense that they also use multiplier update procedures and are based on some augmented Lagrangian as in (12). In the literature these methods are used to solve some important optimal control problems in other fields, such as in economics[22,23] and large water management systems.[92] These optimal control problems have been formulated as separable problems by discretizing the time. In our view, the problems can be solved more efficiently by the approach proposed in this paper without discretizing the time.

In conclusion, this comprehensive review of the multiplier methods has revealed the following basic concepts and ideas that are useful in further development of the methods.

1. The multiplier methods are globally convergent; i.e. they can be implemented in such a way that, starting from any initial estimate, they converge to a local minimum point.

2. The study of duality has revealed that the multiplier update procedures can be viewed as ascent steps in the dual space that maximize the augmented Lagrangian with respect to the Lagrange multipliers. This interpretation has led to the development of several multiplier update procedures.

3. The multiplier update procedures that require gradients of individual constraints defeat the basic idea of the multiplier methods for engineering applications. The study of the continuous multiplier update method reveals that, if gradients of individual constraints are calculated, it is better to use the sequential quadratic programming method. In that case, the two methods are essentially the same. The sequential quadratic programming directly treats equality as well as inequality constraints without using any potential constraint strategy (although it can be also incorporated).

4. The study of continuous multiplier update methods also reveals that it is possible to update multipliers after only an approximate minimum of the augmented Lagrangian has been obtained. It is useful to update the multipliers even more often, i.e. every unconstrained minimization step. These methods need to be further investigated for practical applications.

5. The study of exact penalty methods reveals that they are difficult to implement and use for practical applications. However, it is of interest to note that some nondifferentiable exact penalty functions are used as descent functions in some primal methods (to determine a step size).

6. The exponential penalty methods have been developed recently. They have been successfully applied to a few applications. They appear to have promise for exploitation for engineering applications.

7. Numerical implementation of the multiplier methods (in general all transformation methods) is easier compared to other methods. All that one needs is a good unconstrained minimization subroutine which can be used to develop a special purpose software for an application. This is especially true for transient dynamic response and control problems where time-dependent constraints must be treated. In these methods, there is no need to keep track of the extreme response points for each constraint, as is needed in the primal methods. Also, there is no need to implement computational procedures to compute the gradient of individual constraints. Such procedures can be very complicated for many engineering applications.

APPENDIX

List of symbols

$(h)_+$ $\max(0, h)$
A $m \times m$ matrix
a a large scalar

α	parameter used in Powell's multiplier algorithm
α_e	Lagrange multiplier corresponding to the first equality in (88)
α_s	step length in the direction $\mathbf{d}^{(k-1)}$
$\mathbf{B}$	approximation to the Hessian matrix (with respect to $\mathbf{x}$) of the augmented Lagrangian functional or Lagrangian function
β	parameter used in Powell's multiplier algorithm
β_b	parameter chosen to balance the penalties due to terms corresponding to equality and inequality constraints in (119)
β_c	parameter related to a conjugate gradient method
β_e	Lagrange multiplier corresponding to the constraint in (92)
c_i	defined in (116)
$\mathbf{D}$	$n \times n$ matrix
$\mathbf{d}^{(k)}$	search direction at the kth step
$\Delta\mathbf{u}^{(k)}$	change in $\mathbf{u}$ at the $(k+1)$th step
$\Delta\mathbf{x}^{(k)}$	change in $\mathbf{x}$ at the $(k+1)$th step
E	index set for equality constraints
e_i	perturbation of the constraint $g_i(\mathbf{x}) = 0$
$\exp(.)$	exponential function
$\phi(\mathbf{u})$	dual function of the augmented Lagrangian functional $\Phi(\mathbf{x}, \mathbf{u})$ or Lagrangian functional $L(\mathbf{x}, \mathbf{u})$
$f(\mathbf{x})$	cost function
$\Phi(\mathbf{x}, \mathbf{u}, \mathbf{r})$	augmented Lagrangian functional (augmented Lagrangian)
$\bar{\Phi}(\mathbf{x}, \mathbf{u}, \mathbf{r})$	alternate form of augmented Lagrangian functional defined in (13)
$\tilde{\gamma}(.)$	approximation functional defined in (111)
$\gamma(.)$	a functional defined using $(m - l)$ inequality constraints
$\mathbf{g}(\mathbf{x})$	constraint vector of dimension $m \times 1$
$\bar{\gamma}(.)$	approximation functional defined in (113)
$g_i(\mathbf{x})$	ith constraint
g_{max}	maximum of $(m - l)$ inequality constraints
I	index set for active inequality constraints
I_E	index set defined in (64)
I_I	index set defined in (65)
I_p	potential constraint set
J	index set used in (40)
K	parameter used in Powell's multiplier algorithm
k	multiplier method step number
$\bar{K}$	non-negative parameter defined in (43) representing the maximum constraint violation
K_e	positive constant
$KS(\mathbf{x})$	Kreisselmeier Steinhauser functional
l	number of equality constraints
$L(\mathbf{x}, \mathbf{u})$	Lagrangian functional (Lagrangian)
λ_e	Lagrange multiplier corresponding to the constraint in (90)
$\ln(.)$	natural logarithm
M	non-negative constant in (41) and (42)
m	total number of constraints
$m - l$	number of inequality constraints
m_e	number of constraints based on available information on a random process
N	number of probable values of the discrete variable z

n	number of design variables
$p(\mathbf{e})$	solution of perturbed problem given in (47)
$P(\mathbf{g}(\mathbf{x}), \mathbf{u}, \mathbf{r})$	generalized penalty function
p_i	probability of a discrete variable z to have the value z_i
$\theta_i^{(k)}$	kth approximation to the ith parameter, that is equal to $u_i^{(k)}/r_i^{(k)}$
$\mathbf{R}$	diagonal matrix with the penalty parameter r_i as its ith diagonal element
ρ	arbitrary positive constant
$\mathbf{r}$	penalty parameter vector of dimension $m \times 1$
$\mathbf{r}^{(k)}$	kth penalty parameter vector
r_a	penalty parameter (scalar)
r_c	penalty parameter (scalar)
r_{norm}	parameter defined in (120)
S	subset of the domain for $L(\mathbf{x}, \mathbf{u})$ in the neighbourhood of the point $(\mathbf{x}^*, \mathbf{u}^*)$ in the $\mathbf{x}$-space
s	scalar parameter
$\mathbf{s}^{(k-1)}$	change in $\mathbf{x}$, i.e. $\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}$
S^*	small neighbourhood of the point $(\mathbf{x}^*, \mathbf{u}^*)$ and subset of S
S_e	Shannon entropy
T	subset of the domain for $L(\mathbf{x}, \mathbf{u})$ in the neighbourhood of the point $(\mathbf{x}^*, \mathbf{u}^*)$ in the $\mathbf{u}$-space
t	parameter used in (24)
T^*	small neighbourhood of the point $(\mathbf{x}^*, \mathbf{u}^*)$ and subset of T
t_0	initial value of the parameter t
t_f	final value of the parameter t
$\mathbf{u}$	Lagrange multiplier vector of dimension $m \times 1$
$U(.)$	multiplier update procedure
$\mathbf{u}^{(k)}$	kth approximation to Lagrange multiplier vector
$\mathbf{u}^*$	Lagrange multiplier vector at the solution point
U_B	Buys multiplier update procedure
U_F	projection formula or Fletcher multiplier update procedure
U_G	general multiplier update procedure
U_{HP}	Hestenes–Powell multiplier update procedure
U_T	Tapia multiplier update procedure
u_i	ith Lagrange multiplier
$u_i(\mathbf{x})$	ith multiplier approximation function
U_M	Miele multiplier update procedure
$\mathbf{v}$	surrogate multiplier vector of dimension $m \times 1$
$\mathbf{w}$	additional variable vector of dimension $m \times 1$
$\mathbf{x}$	design variable vector of dimension $n \times 1$
$\mathbf{x}^{(k)}$	kth approximation to the solution
$\mathbf{x}^*$	solution point
$\mathbf{y}^{(k)}$	vector defined in (84)
$\| . \|$	Euclidean norm
∇	gradient operator with respect to the design variable vector $\mathbf{x}$
∇^2	Hessian operator with respect to the design variable vector $\mathbf{x}$
$\nabla_{\mathbf{uu}}^2$	Hessian operator with respect to the multiplier vector $\mathbf{u}$
$\nabla \mathbf{g}(\mathbf{x})$	$n \times l$ matrix whose ith column is the gradient of ith equality constraint (the matrix is of dimension $n \times m$ if all the constraints are considered)
$\nabla_{\mathbf{u}}$	gradient operator with respect to the multiplier vector $\mathbf{u}$

REFERENCES

1. V. Adimurthy, 'Launch vehicle trajectory optimization', *Acta Astronaut.*, **15**, 845–850 (1987).
2. J. S. Arora, *Introduction to Optimum Design*, McGraw-Hill, New York, 1989.
3. J. S. Arora, 'Computational design optimization: A review and future directions', *Struct. Safety*, **7**, 131–148 (1990).
4. J. S. Arora and E. J. Haug, 'Methods of design sensitivity analysis in structural optimization', *AIAA J.*, **17**, 970–973 (1979).
5. J. S. Arora and P. B. Thanedar, 'Computational methods for optimum design of large complex systems', *Comp. Mech.*, **1**, 221–242 (1986).
6. Y. Bard and J. Greenstadt, 'A modified Newton method for optimization with equality constraints', in R. Fletcher (ed.), *Optimization*, Academic Press, London, 1969, pp. 299–306.
7. J.-M. Barthelemy and M. F. Riley, 'An improved multilevel optimization approach for the design of complex engineering systems', *Proc. AIAA/ASME/ASCE/AHS 27th Structures, Structural Dynamics and Materials Conference* San Antonio, TX, *Paper 86-0950*, 1986.
8. A. D. Belegundu and J. S. Arora, 'Potential of transformation methods in optimal design', *AIAA J.*, **19**, 1372–1374 (1981).
9. A. D. Belegundu and J. S. Arora, 'A computational study of transformation methods for optimal design', *AIAA J.*, **22**, 535–542 (1984).
10. A. D. Belegundu and J. S. Arora, 'A study of mathematical programming methods for structural optimization, Part I: Theory, Part II: Numerical aspects', *Int. j. numer. methods eng.*, **21**, 1583–1624 (1985).
11. M. Bertocchi, E. Cavalli and E. Spedicato, 'Computational performance of diagonalized multiplier quasi-Newton methods for nonlinear optimization with equality constraints', in L. C. W. Dixon and G. P. Szegö (eds.), *Numerical Optimization of Dynamic Systems*, North-Holland, Amsterdam, 1980, pp. 247–268.
12. D. P. Bertsekas, 'Combined primal–dual and penalty methods for constrained minimization', *SIAM J. Control Optimization*, **13**, 521–544 (1975).
13. D. P. Bertsekas, 'Minmax methods based on approximation', *Proc. 1976 Johns Hopkins Conf. on Information Science Systems*, Baltimore, Md., 1976, pp. 363–365.
14. D. P. Bertsekas, 'A new algorithm for solution of resistive networks involving diodes', *IEEE Trans. Circuits Syst.*, **23**, 599–608 (1976).
15. D. P. Bertsekas, 'On penalty and multiplier methods for constrained minimization', *SIAM J. Control Optimization*, **14**, 216–235 (1976).
16. D. P. Bertsekas, 'Multiplier methods: A survey', *Automatica*, **12**, 133–145 (1976).
17. D. P. Bertsekas, 'Approximation procedures based on the methods of multipliers', *J. Optimization Theory Applic.*, **23**, 487–510 (1977).
18. D. P. Bertsekas, 'Enlarging the region of convergence of Newton's method for constrained optimization', *J. Optimization Theory Applic.*, **36**, 221–252 (1982).
19. D. P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*, Academic Press, New York, 1982.
20. J. D. Buys, 'Dual algorithms for constrained optimization problems', *Doctorate Dissertation*, University of Leiden, Netherland, 1972.
21. R. H. Byrd, 'Local convergence of the diagonalized method of multipliers', *J. Optimization Theory Applic.*, **26**, 485–500 (1978).
22. K. Cichocki, 'Application of augmented Lagrangians in dynamic planning models', in L. C. W. Dixon and G. P. Szegö (eds.), *Numerical Optimization of Dynamic Systems*, North-Holland, Amsterdam, 1980, pp. 247–268.
23. K. Cichocki and J. S. Sosnowski, 'Solutions of dynamic nonlinear planning model via augmented Lagrangian', in K. Cichocki and A. Strauszak (eds.), *Systems Analysis Applications to Complex Programs*, Pergamon Press, Oxford, 1978, pp. 57–61.
24. I. D. Coope and R. Fletcher, 'Some numerical experience with a globally convergent algorithm for nonlinearly constrained optimization', *J. Optimization Theory Applic.*, **32**, 1–16 (1980).
25. G. Di Pillo and L. Grippo, 'A new class of augmented Lagrangians in nonlinear programming', *SIAM J. Control Optimization*, **17**, 618–628 (1979).
26. G. Di Pillo and L. Grippo, 'A new augmented Lagrangian function for inequality constraints in nonlinear programming problems', *J. Optimization Theory Applic.*, **32**, 495–519 (1982).
27. G. Di Pillo and L. Grippo, 'A continuously differentiable exact penalty function for nonlinear programming problems with inequality constraints', *SIAM J. Control Optimization*, **23**, 72–84 (1985).
28. G. Di Pillo and L. Grippo, 'An exact penalty method with global convergence properties for nonlinear programming problems', *Math. Programming*, **36**, 1–18 (1986).
29. G. Di Pillo, L. Grippo and S. Lucidi, 'Globally convergent exact penalty algorithms for constrained optimization', in A. Prekopa *et al.* (eds.), *12th IFIP Conference on System Modelling and Optimization*, Springer-Verlag, Berlin, 1986.
30. A. V. Fiacco and G. P. McCormick, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, Wiley, New York, 1968.
31. R. Fletcher, 'A class of methods for nonlinear programming with termination and convergence properties', in J. Abadie (ed.), *Integer and Nonlinear Programming*, North-Holland, Amsterdam, 1970, pp. 157–175.
32. R. Fletcher, 'A class of methods for nonlinear programming III: Rate of convergence', in F. A. Lootsma (ed.), *Numerical Methods for Nonlinear Optimization*, Academic Press, New York, 1972, pp. 371–382.
33. R. Fletcher, 'Methods related to Lagrangian functions', in P. E. Gill and W. Murray (eds.), *Numerical Methods for*

Constrained Optimization, Academic Press, London, 1974, pp. 219–239.

34. R. Fletcher, 'An ideal penalty function for constrained optimization', J. Inst. Math. Optimization, 15, 319–342 (1975).
35. R. Fletcher and S. Lill, 'A class of methods for nonlinear programming II: Computational experience', in J. B. Rosen et al. (eds.), Nonlinear Programming, Academic Press, London, 1971, pp. 67–92.
36. R. Fontecilla, T. Steihaug and R. A. Tapia, 'A convergence theory for a class of quasi-Newton methods for constrained optimization', SIAM J. Numer. Anal., 24, 1133–1151 (1987).
37. S. Fujii and H. Fujimoto, 'A transformation method with two parameters for constrained optimal control problems', J. Dyn. Syst. Meas. Control ASME, 100, 124–131 (1978).
38. S. T. Glad, 'Properties of updating methods for the multipliers in augmented Lagrangians', J. Optimization Theory Applic., 28, 135–156 (1979).
39. T. Glad and E. Polak, 'A multiplier method with automatic limitation of penalty growth', Math. Programming, 17, 140–155 (1979).
40. E. A. Gunn, M. Thorburn and A. Rai, 'A decomposition method based on the augmented Lagrangian', INFOR, 26, 91–113 (1988).
41. P. C. Haarhoff and J. D. Buys, 'A new method for the optimization of nonlinear function subject to nonlinear constraints', Comp. J., 13, 178–184 (1970).
42. R. T. Haftka and M. P. Kamat, Elements of Structural Optimization, Martinus Nijhoff, Dordrecht, Netherland, 1985.
43. P. Hajela, 'Further developments in the controlled growth approach for optimal structural synthesis', ASME Paper 82-DET-62, 1982.
44. P. Hajela, 'Techniques in optimum structural synthesis with static and dynamic constraints', Ph.D. Thesis, Stanford University, 1982.
45. P. Hajela, 'A look at two underutilized methods for optimum structural design', Eng. Optimization, 11, 21–30 (1987).
46. S. P. Han, 'Dual variable metric algorithms for constrained optimization', SIAM J. Control Optimization, 15, 546–565 (1977).
47. E. J. Haug and J. S. Arora, Applied Optimal Design, Wiley-Interscience, New York, 1979.
48. M. R. Hestenes, 'Multiplier and gradient methods', J. Optimization Theory Applic., 4, 303–320 (1969).
49. C. C. Hsieh and J. S. Arora, 'Treatment of general boundary conditions and point-wise state variable constraints in optimum design for static and dynamic response', Technical Report CAD-SS-84.1, College of Engineering, The University of Iowa, 1984.
50. C. C. Hsieh and J. S. Arora, 'Design sensitivity analysis and optimization of dynamic response', Comp. Methods Appl. Mech. Eng., 43, 195–219 (1984).
51. C. C. Hsieh and J. S. Arora, 'Structural design sensitivity with general boundary conditions: Dynamic problem', Int. j. numer. methods eng., 21, 267–283 (1985).
52. C. C. Hsieh and J. S. Arora, 'A hybrid formulation for treatment of point-wise state variable constraints in dynamic response optimization', Comp. Methods Appl. Mech. Eng., 48, 171–189 (1985).
53. E. T. Jaynes, 'Information theory and statistical mechanics', Phys. Rev., 106, 620–630 (1957): 108, 171–190 (1957).
54. B. W. Kort and D. P. Bertsekas, 'A new penalty function algorithm for constrained minimization', Proc. 1972 IEEE Conf. on Decision and Control, New Orleans, Louisiana, 1972, pp. 162–166.
55. G. Kreisselmeier and R. Steinhauser, 'Systematic control design by optimizing a vector performance index', Proc. IFAC Symp. on Computer Aided Design of Control Systems, Zurich, Switzerland, Aug. 1979.
56. X. S. Li, 'Entropy and optimization', Ph.D. Thesis, University of Liverpool, 1987.
57. X. S. Li and A. B. Templeman, 'Entropy-based optimum sizing of trusses', Civil Eng. Syst., 5, 121–128 (1988).
58. S. Lucidi, 'New results on a class of exact augmented Lagrangians', J. Optimization Theory Applic., 58, 259–282 (1988).
59. D. G. Luenberger, Linear and Nonlinear Programming, Addison-Wesley, Reading, Massachusetts, 1984.
60. O. L. Mangasarian, 'Unconstrained Lagrangians in nonlinear programming', SIAM J. Control Optimization, 13, 772–791 (1975).
61. A. Miele, E. E. Cragg, R. R. Iyer and A. V. Levy, 'Use of the augmented penalty function in mathematical programming problems, Part 1', J. Optimization Theory Applic., 8, 115–130 (1971).
62. A. Miele, E. E. Cragg and A. V. Levy, 'Use of the augmented penalty function in mathematical programming problems, Part 2', J. Optimization Theory Applic., 8, 131–153 (1971).
63. A. Miele and A. V. Levy, 'Modified quasilinearization and optimal initial choice of the multipliers, Part 1, Mathematical programming problems', J. Optimization Theory Applic., 6, 364–380 (1970).
64. A. Miele, A. V. Levy and E. E. Cragg, 'Modifications and extensions of the conjugate gradient-restoration algorithm for mathematical programming problems', J. Optimization Theory Applic., 6, 450–472 (1971).
65. A. Miele, P. E. Moseley, A. V. Levy and G. M. Coggins, 'On the method of multipliers for mathematical programming problems', J. Optimization Theory Applic., 10, 1–32 (1972).
66. H. Mukai and E. Polak, 'A quadratically convergent primal-dual algorithm with global convergence properties for solving optimization problems with equality constraints', Math. Programming, 9, 336–349 (1975).
67. V. H. Nguyen and J. J. Strodiot, 'On the convergence rate of penalty function method of exponential type', J. Optimization Theory Applic., 27, 495–508 (1979).
68. J. K. Paeng and J. S. Arora, 'Dynamic response optimization of mechanical and structural systems with state space multiplier methods', Technical Report No. ODL-85.18, Optimal Design Laboratory, College of Engineering, The University of Iowa, 1985.
69. J. K. Paeng and J. S. Arora, 'Dynamic response optimization of mechanical systems with multiplier methods',

J. Mechanisms, Transmission Automation Design, **111**, 73–80 (1989).

70. D. A. Pierré and M. J. Lowe, *Mathematical Programming via Augmented Lagrangians: An Introduction with Computer Programs*, Addison-Wesley, Massachusetts, 1975.

71. E. Polak and A. Sangiovanni-Vincentelli, 'Theoretical and computational aspects of the optimal design centering, tolerancing and tuning problem', *IEEE Trans. Circuits Syst.*, **26**, 795–813 (1979).

72. E. Polak and A. Tits, 'A globally convergent, implementable, multiplier method with automatic penalty limitation', *Appl. Math. Optimization*, **6**, 335–360 (1980).

73. B. T. Poljak and N. V. Tret'yakov, 'The method of penalty estimates for conditional extremum problems', *USSR Compt. Math. Mathematical Phys.*, **13**, 42–58 (1974).

74. M. J. D. Powell, 'A method for nonlinear constraints in minimization problems', in R. Fletcher (ed.), *Optimization* Academic Press, New York, 1969.

75. M. J. D. Powell, 'Algorithms for nonlinear constraints that use Lagrangian functions', *Math. Programming*, **14**, 224–248 (1978).

76. R. T. Rockafellar, 'A dual approach to solving nonlinear programming problems by unconstrained optimization', *Math. Programming*, **5**, 354–373 (1973).

77. R. T. Rockafellar, 'The multiplier method of Hestenes and Powell applied to convex programming', *J. Optimization Theory Applic.*, **12**, 555–562 (1973).

78. R. T. Rockafellar, 'Augmented Lagrange multiplier functions and duality in nonconvex programming', *SIAM J. Control Optimization*, **12**, 268–285 (1974).

79. R. R. Root and K. M. Ragsdell, 'Computational enhancement to the method of multipliers', *J. Mech. Design ASME*, **102**, 517–522 (1980).

80. J. B. Rosen, 'The gradient projection method for nonlinear programming, I. Linear constraints', *J. Soc. Indust. Appl. Math.*, **8**, 181–217 (1960).

81. R. B. Schnabel, 'Determining feasibility of a set of nonlinear inequality constraints', *Math. Programming Study*, **16**, 137–148 (1980).

82. S. B. Schuldt, 'A method of multipliers for mathematical programming problems with equality and inequality constraints', *J. Optimization Theory Applic.*, **17**, 155–161 (1975).

83. C. E. Shannon, 'A mathematical theory of communication', *Bell Syst. Tech. J.*, **27**, 379–428 (1948).

84. J. Sobieszczanski-Sobieski, C. L. Bloebaum and P. Hajela, 'Sensitivity of control-augmented structure obtained by a system decomposition method', *Proc. AIAA/ASME/ASCE/AHS 29th Structures, Structural Dynamics and Materials Conference*, Williamsburg, Virginia, *Paper No. 88-2205*, Apr. 1988.

85. J. Sobieszczanski-Sobieski, A. R. Dovi and G. A. Wrenn, 'A new algorithm for general multiobjective optimization', *Proc. AIAA/ASME/ASCE/AHS 29th Structures, Structural Dynamics and Materials Conference*, Williamsburg, Virginia, *Paper 88-2434*, Apr. 1988.

86. J. Sobieszczanski-Sobieski, B. B. James and A. R. Dovi, 'Structural optimization by multilevel decomposition', *AIAA J.*, **23**, 1775–1782 (1985).

87. A. Tanikawa and H. Mukai, 'New Lagrangian function for nonconvex primal-dual decomposition', *Comp. Math. Applic.*, **13**, 661–676 (1987).

88. R. A. Tapia, 'Newton's method for optimization problems with equality constraints', *SIAM J. Numer. Anal.*, **11**, 874–886 (1974).

89. R. A. Tapia, 'Diagonalized multiplier methods and quasi-Newton methods for constrained optimization', *J. Optimization Theory Applic.*, **22**, 135–194 (1977).

90. R. A. Tapia, 'Quasi-Newton methods for equality constrained optimization: Equivalence of existing methods and a new implementation', in O. L. Mangasarian *et al.* (eds.), *NonlinearProgramming 3*, Academic Press, New York, 1978, pp. 125–164.

91. R. A. Tapia, 'On the role of slack variables in quasi-Newton methods for constrained optimization', in L. C. W. Dixon and G. P. Szegö (eds.), *Numerical Optimization of Dynamic Systems*, North-Holland, Amsterdam, 1980, pp. 235–246.

92. P. Tatjewski, 'New dual-type decomposition algorithm for nonconvex separable optimization problems', *Automatica*, **25**, 233–242 (1989).

93. A. B. Templeman, 'Entropy-based minimax applications in shape-optimal design', in H. A. Eschenauer and G. Thierauf (eds.), *Discretization Methods and Structural Optimization—Procedures and Applications, Proc. GAMM Seminar*, Siegen, FRG, Oct. 1988, Springer-Verlag, Berlin, 1989, pp. 335–342.

94. A. B. Templeman and X. S. Li, 'A maximum entropy approach to constrained non-linear programming', *Eng. Optimization*, **12**, 191–205 (1987).

95. M. Tribus, 'Information theory as the basis for thermostatics and thermodynamics', *J. Appl. Mech. ASME*, **28**, 1–8 (1961).

96. S. S. Tripathi and K. S. Narendra, 'Constrained optimization problem: Using multiplier methods', *J. Optimization Theory Applic.*, **9**, 59–70 (1972).

97. C. H. Tseng and J. S. Arora, 'Optimum design of systems for dynamics and controls using sequential quadratic programming', *AIAA J.*, **27**, 1793–1800 (1989).

98. K. S. Vastola, 'A numerical study of two measures of delay for network routing', *M.S. Thesis*, Department of Electrical Engineering, University of Illinois, Champaign-Urbana, 1979.

99. G. A. Wrenn, 'An indirect method for numerical optimization using the Kreisselmier–Steinhauser function', *Final Report*, NASA-CR-4220, NAS 1.26:4220, NTIS HC A05/MF A01, N89-16779, 1989.