Proceedings of 1972 IEEE Conference on Decision and Control
New Orleans, Louisiana, December 1972.

CDC

PAPER No. WP4-3

A NEW PENALTY FUNCTION METHOD FOR CONSTRAINED MINIMIZATION

Barry W. Kort
and
Dimitri P. Bertsekas
Engineering-Economic Systems Department
Stanford University, Stanford, California

Abstract

During recent years it has been shown that the performance of penalty function methods for constrained minimization can be improved significantly by introducing gradient type iterations for solving the dual problem. In this paper we present a new penalty function algorithm of this type which offers significant advantages over existing schemes for the case of the convex programming problem. The algorithm treats inequality constraints explicitly and can also be used for the solution of general mathematical programming problems.

I. Introduction

During the last decade penalty function methods have gained recognition as one of the most effective class of methods for solving nonlinear programming problems. They require very simple programming, they avoid the necessity of staying within the feasible region and they are quite reliable particularly when used in conjunction with effective unconstrained minimization algorithms. Their chief disadvantage stems from the fact that they require the solution of a sequence of unconstrained minimization problems with increasingly unfavorable structure. This ill-conditioning is due to increasingly high penalty terms. It is well known that penalty methods yield as a by-product of the computation a sequence of vectors converging to the Lagrange multiplier vector of the problem. This latter vector is a solution of the dual problem (defined under suitable local convexity assumptions)[1], a fact which has led to the interpretation that penalty methods seek to solve the dual problem.

During recent years a number of researchers, [2], [3], [4], were led to using penalty methods in order to attack the dual problem in a more direct way, i.e. by making use of a recursive iteration of the gradient type. Their approach can be interpreted as follows. Consider the problem

$$\text{minimize} \quad f(x) \qquad x \in R^n$$
$$\text{subject to} \quad h_i(x)=0 \quad i=1,2,..,p \tag{1}$$

Now if (x^*, λ^*) is a local solution and Lagrange multiplier pair of problem (1) then (x^*, λ^*) is also a local solution and Lagrange multiplier pair of the "penalized" problem

$$\text{minimize} \quad f(x) + 1/2\, c\sum_{i=1}^{p}[h_i(x)]^2$$
$$\text{subject to} \quad h_i(x) = 0 \qquad i=1,2,..,p \tag{2}$$

Furthermore under general assumptions it follows [1] that for c greater than come constant $\bar{c}$ problem (2) has locally convex structure and λ^* is a local solution of the dual problem

$$\max_{\lambda \varepsilon R^p} \psi(\lambda)$$

where the dual functional ψ is defined by

$$\psi(\lambda)= \min_{x \varepsilon R^n} \{f(x)+\sum_{i=1}^{p}\lambda_i h_i(x)+\frac{1}{2}c\sum_{i=1}^{p}[h_i(x)]^2\} \tag{3}$$

The method proposed in [2],[3],[4],[5], consists of solving the unconstrained minimization problem indicated in (3) for fixed values of λ and c, say λ^k and c^k, to obtain a minimizing point x^k and then update λ by means of the iteration

$$\lambda_i^{k+1} = \lambda_i^k + c^k h_i(x^k) \quad i=1,2,..,p \tag{4}$$

Since $h_i(x^k)$ is the partial derivative of ψ with respect to λ_i^k the iteration (4) can be interpreted as an iteration of the gradient type for solving the dual problem. Furthermore it can be shown [1] that as c increases iteration (4) approximates a Newton step for solving the dual. Thus as c^k increases the iteration (4) is expected to converge to the solution λ^* and converge faster--albeit at the expense of making the unconstrained problem in (3) more ill-conditioned. The choice of c^k presents considerable difficulty since c^k must be greater than $\bar{c}$, an unknown constant, and also since it is not clear how to update c^k in order to keep the total number of function evaluations for solving the problem at a low level. One possibility is to use a sequence $c^k \to \infty$ and then the method converges since it behaves like a penalty function method. Hopefully the iteration (4) speeds up convergence. While it appears that in many problems such acceleration is realized [6], this method of updating c^k partially defeats the purpose of the algorithm which is clearly to avoid the ill-conditioning associated with high penalty terms in the problem of (3). Another possibility is to update c^k on a trial and error basis, i.e. to increase c^k if the algorithm does not make satisfactory progress. Such schemes have been proposed in the literature and some computational experience has been reported [2],[5].

The algorithm discussed in the previous paragraph is mainly applicable to problems with equality constraints. Inequality constraints can be treated in two ways: either by converting them to equality constraints by using slack nonnegative variables or by incorporating into the algorithm features which enable it to "separate" during the iterations the active from the inactive inequality constraints and subsequently treat the active constraints essentially as equality constraints [5]. Aside from the obvious computational disadvantages, an important drawback of such schemes is that they fail to take advantage of special structural properties of the problem such as convexity. Thus a convex program is treated in the same way as a nonconvex program.

The algorithm proposed in this paper uses an approach similar to the algorithm described above. However, inequality constraints are handled by a new technique which does not attempt to convert them into equality constraints. For non-convex programming

*This work was carried out at the Dept. of Engineering-Economic Systems, Stanford University, Stanford, Calif. and supported by the National Science Foundation under Grant GK 29237 and by Bell Telephone Laboratories.

The authors are with the Dept. of Engineering-Economic Systems, Stanford University, Stanford, Calif.

problems, our algorithm offers no spectacular advantages over the earlier described method. (In fact for the case of equality constraints only, our algorithm reduces to the earlier one.) The main advantage arises when our algorithm is applied to a convex programming problem. Unlike the parameter c of problem (3) our parameter can be chosen arbitrarily: there is no critical value analogous to $\bar{c}$. This threshold-free feature has three important ramifications. First, the algorithm converges from any starting point. Second, the algorithm does not require a parameter adjusting scheme. Third, the sequence of unconstrained problems does not get increasingly ill-conditioned. Another important feature is that the algorithm doesn't increase the dimensionality-as in the case of slack variables, nor does it require a separation mechanism to determine which of the inequality constraints are active.

In the next section we describe our algorithm for the case of the convex programming problem, and we prove its convergence. In Section 3 we give the extension to the general non-linear programming problem. Finally in the last section we report some computational experience.

II. The Basic Algorithm

Consider the convex programming problem

$$\text{minimize } f_o(x)$$
$$\text{subject to } f_i(x) \leq 0 \qquad i=1,2,..,m \tag{5}$$

where f_o, f_i are real valued convex and differentiable functions on R^n (n-dimensional Euclidean space). We assume that there exists a point $\bar{x} \in R^n$ such that $f_i(\bar{x}) < 0$, i=1,2,..,m and that the set of optimal solutions to problem (5) is nonempty and bounded. Under these assumptions it is well known [1] that if x^* is an optimal solution of problem (5) then there exists a Lagrange multiplier vector $\lambda^* = (\lambda_1^*,..,\lambda_m^*)$ such that x^* is an optimal solution of the problem

$$\text{minimize } f_o(x) + \sum_{i=1}^{m} \lambda_i^* f_i(x)$$

and $\qquad \lambda_i^* \geq 0, \qquad \lambda_i^* f_i(x_i^*) = 0.$

Furthermore λ^* maximizes the dual functional

$$\psi(\lambda) = \min_x \{f_o(x) + \sum_{i=1}^{m} \lambda_i f_i(x)\} \tag{6}$$

over all vectors $\lambda = (\lambda_i,..,\lambda_m)$ with $\lambda_i \geq 0, i=1,..,m,$

and if f^* is the optimal value of problem (5) we have

$$f^* = \max_{\lambda \geq 0} \psi(\lambda)$$

The algorithm that we propose for maximization of the dual functional has the following form:

Given $\lambda^k = (\lambda_1^k,...,\lambda_m^k)$ with $\lambda_i^k > 0, i=1,..,m,$ solve an unconstrained minimization problem of the form

$$\text{minimize} \qquad f_o(x) + \sum_{i=1}^{m} \lambda_i^k \phi_\varepsilon [f_i(x)] \tag{7}$$

where ϕ_ε is a real valued function of a single variable, depending on a positive parameter ε, the form of which will be specified in what follows. Let x^k be a solution of this problem. Then update λ^k by means of the iteration

$$\lambda_i^{k+1} = \lambda_i^k \phi_\varepsilon' [f_i(x^k)] \qquad i=1,2,..,m \tag{8}$$

where ϕ' denotes the derivative of ϕ.

We shall show in the remainder of this section that the iteration (8) converges to a solution of the dual problem and that convergence is accelerated as ε is chosen small.

To define the class of functions ϕ_ε which we shall use, let us consider first the class of functions $\phi: R \to R$ characterized by the following properties

(P1) $\quad \phi''(t) = \frac{d^2}{dt^2} \phi(t) > 0 \qquad \forall t \in R$

(P2) $\quad \phi(0) = 0$

(P3) $\quad \phi'(0) = \frac{d}{dt} \phi(0) = 1$

(P4) $\quad \lim_{t \to -\infty} \phi(t) = -1$

(P5) $\quad \lim_{t \to \infty} \phi'(t) = \infty$

Examples of functions ϕ satisfying properties (P1)-(P5) above are

$$\phi(t) = \begin{cases} t+t^2 & t \geq 0 \\ \frac{t}{1-t} & t < 0 \end{cases} \tag{9}$$

$$\phi(t) = e^t - 1 \tag{10}$$

$$\phi(t) = \begin{cases} 3t^2/4 + t & t \geq 0 \\ \frac{1}{(1-t/2)^2} - 1 & t < 0 \end{cases} \tag{11}$$

Let us consider now for any $\varepsilon > 0$ the function

$$\phi_\varepsilon(t) = \varepsilon \phi(\tfrac{t}{\varepsilon}) \tag{12}$$

where ϕ satisfies (P1) - (P5). Notice that as ε is decreased $\phi_\varepsilon(t)$ approaches ∞ for $t > 0$ and zero for $t \leq 0$ (see Fig. 1). Hence sequential minimization of

$$f_o(x) + \sum_{i=1}^{m} \lambda_i \phi_{\varepsilon_k} [f_i(x)]$$

for a fixed set of values $\lambda_i > 0$ i=1,..,m and a sequence of values $\varepsilon_k \to 0$ yields a penalty function method of the standard type.

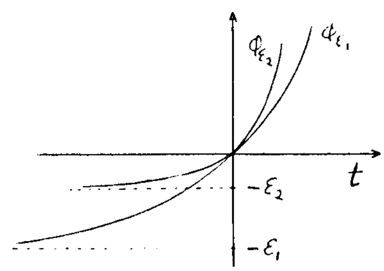

Figure 1.

Consider now for any $\varepsilon > 0$ the convex programming problem

$$\text{minimize} \quad f_o(x) \tag{13}$$

subject to $\phi_\varepsilon[f_i(x)] \leq 0$ i=1,2,..,m and denote by $\psi_\varepsilon(\lambda)$ the corresponding dual functional

$$\psi_\varepsilon(\lambda) = \min_x \{f_o(x) + \sum_{i=1}^{m} \lambda_i \phi_\varepsilon[f_i(x)]\} \tag{14}$$

Let x^* and λ^* be an optimal solution and a corresponding Lagrange multiplier of the above problem. Then

$$\nabla f_o(x^*) + \sum_{i=1}^{m} \lambda_i^* \phi_\varepsilon'[f_i(x^*)] \nabla f_i(x^*) = 0$$

Since $\phi_\varepsilon'(0) = 1$ and $\lambda_i^* = 0$ if $f_i(x^*) < 0$ the above equation yields

$$\nabla f_o(x^*) + \sum_{i=1}^{m} \lambda_i^* \nabla f_i(x^*) = 0$$

which in turn implies that x^* and λ^* are an optimal solution and a Lagrange multiplier for the original convex programming problem (5). Clearly a similar result holds if the roles of problems (5) and (13) are reversed and we have:

Proposition 1: Problems (5) and (13) are equivalent in that they have the same optimal solutions and corresponding Lagrange multipliers.

Now let $\lambda_i^k > 0$, $i=1,..,m$, $\varepsilon > 0$ be given scalars and let x^k be an optimal solution of the problem

$$\text{minimize} \quad f_c(x) + \sum_{i=1}^{m} \lambda_i^k \phi_\varepsilon[f_i(x)] \tag{15}$$

Using equation (14) we have

$$\psi_\varepsilon(\lambda^k) = f_o(x^k) + \sum_{i=1}^{m} \lambda_i^k \phi_\varepsilon[f_i(x^k)] \tag{16}$$

Furthermore since

$$\nabla f_o(x^k) + \sum_{i=1}^{m} \lambda_i^k \phi_\varepsilon'[f_i(x^k)] \nabla f_i(x^k) = 0$$

it follows that x^k also solves the problem

$$\text{minimize} \quad f_o(x) + \sum_{i=1}^{m} \lambda_i^{k+1} f_i(x)$$

where we have set

$$\lambda_i^{k+1} = \lambda_i^k \phi_\varepsilon'[f_i(x^k)] \quad i=1,2,..,m \tag{17}$$

This implies, using the defining equation (6) for the dual functional, that

$$\psi(\lambda^{k+1}) = f_o(x^k) + \sum_{i=1}^{m} \lambda_i^{k+1} f_i(x^k)$$

Thus by solving the unconstrained minimization problem (15) one obtains the value of the dual functional at λ^{k+1}. The above discussion motivates the following algorithm.

Algorithm 1:

Step 1: Select a vector $\lambda^o = (\lambda_1^o, .., \lambda_m^o)$ with $\lambda_i^o > 0$, $i=1,..,m$ and a scalar $\varepsilon > 0$.

Step 2: Given λ^k solve the unconstrained minimization problem

$$\text{minimize} \quad f_o(x) + \sum_{i=1}^{m} \lambda_i^k \phi_\varepsilon[f_i(x)]$$

where ϕ_ε is defined by (12) and ϕ satisfies properties (P1)-(P5). Let x^k be a solution of this problem.

Step 3: Set $\lambda_i^{k+1} = \lambda_i^k \phi_\varepsilon'[f_i(x^k)]$ $i=1,..,m$. If $\lambda^{k+1} = \lambda^k$ stop. Else return to Step 2.

Notice that if the algorithm terminates in a finite number of steps the resulting vector λ^k is a Lagrange multiplier vector for problem (5) and hence in this case it yields a solution to the dual problem.

We now prove that the minimization in step 2 of the above algorithm can always be carried out, and therefore the algorithm is well defined.

Proposition 2: The unconstrained minimization problem in step 2 of Algorithm 1 has a solution.

Proof: From the properties of $\phi(t)$ it follows that the equation $\phi'(t) = a$ has a solution for every $a > 0$. Hence the concave function $at - \phi(t)$ has a maximum value $\sigma(a)$. If we define also $\sigma(0) = \lim_{a \to 0} \sigma(a) = 1$, we have

$$\phi(t) \geq at - \sigma(a) \qquad a \geq 0 \tag{19}$$

from which $\phi_\varepsilon(t) \geq at - \varepsilon\sigma(a)$ $\quad a \geq 0$.

Hence for all x and for all $\tilde{\lambda}_i \geq 0, i=1,2,..m$ we have

$$f_o(x) + \sum_{i=1}^{m} \lambda_i^k \phi_\varepsilon[f_i(x)] \geq f_o(x) + \sum_{i=1}^{m} \tilde{\lambda}_i f_i(x) - \sum_{i=1}^{m} \varepsilon \lambda_i^k \sigma(\tilde{\lambda}_i/\lambda_i^k).$$

We shall prove that the function on the left has no direction of recession ([7],p.69) and hence the set of its minimizing points is nonempty and compact ([7],Th. 27.1.d). Let $y \neq 0, y \in R^n$ be any direction. If y is not a direction of recession of f_o take $\tilde{\lambda}_i = 0, i=1,..,m$ in the above inequality and it follows that y is not a direction of recession of the function on the left. Assume y is a direction of recession of f_o. Then ([7], Cor.8.3.3,Th.27.3,Cor.27.3.3) from the assumption that the set of all solutions of problem (5) is nonempty and compact it follows that there exists a function $f_i, i \in \{1,..,m\}$ for which y is not a direction of recession. Take $\tilde{\lambda}_j = 0, j \neq i$ and $\tilde{\lambda}_i$ sufficiently large in the above inequality to guarantee that y is not a direction of recession of the function on the right ([7]Th.9.3). Then the inequality yields that y is not a direction of recession of the function on the left. Q.E.D.

Note that the above proposition implies that the dual functional ψ has finite values at the points λ^k $k=1,2,...$ even if it has an infinite value at λ^o. We next prove that at each iteration the value of the dual functional is increased. This fact is of basic importance in proving the convergence of Algorithm 1.

Proposition 3: For any $\lambda^k = (\lambda_1^k, .., \lambda_m^k)$ defined via Algorithm 1 and such that $\lambda^{k+1} \neq \lambda^k$ we have

$$\psi(\lambda^k) < \psi_\varepsilon(\lambda^k) < \psi(\lambda^{k+1})$$

where ψ is the dual functional of the problem (5) and ψ_ε is the dual functional of problem (13).

Proof: Since $\lambda^k \neq \lambda^{k+1}$ it follows that $\phi_\varepsilon'[f_i(x^k)] \neq 1$ or equivalently $\phi_\varepsilon[f_i(x^k)] \neq 0$ for some $i \in \{1,2,..,m\}$. From (19) with $a=1$, we have

$$\phi_\varepsilon(t) > t \quad t \neq 0 \quad \text{and} \quad \phi_\varepsilon(0) = 0.$$

Also $\phi_\varepsilon'(t) > 0$ $\forall t \in R$, so from iteration (17) we see $\lambda_i^k > 0$, $i=1,..,m$. Hence

$$\psi(\lambda^k) = \min_x \{f_o(x) + \sum_{i=1}^{m} \lambda_i^k f_i(x)\} \leq$$
$$\leq f_o(x^k) + \sum_{i=1}^{m} \lambda_i^k f_i(x^k) <$$
$$< f_o(x^k) + \sum_{i=1}^{m} \lambda_i^k \phi_\varepsilon[f_i(x^k)] = \psi_\varepsilon(\lambda^k)$$

and hence the first inequality is proved. To prove the second inequality we first note that

$$\phi_\varepsilon(t) < t\,\phi_\varepsilon'(t) \qquad \forall\, t \neq 0 \tag{20}$$

The above inequality can be seen to hold by the fact that

$$\frac{d}{dt}[t\,\phi_\varepsilon'(t) - \phi_\varepsilon(t)] = t\phi_\varepsilon''(t)$$

Since $\phi_\varepsilon''(t) > 0, \forall t$, $[t\phi_\varepsilon'(t) - \phi_\varepsilon(t)]$ is decreasing for $t < 0$, increasing for $t > 0$ and is zero for $t=0$. Hence (20) holds. Now using (20) and the definition of λ^{k+1} (c.f.(17))

$$\psi_\varepsilon(\lambda^k) = f_o(x^k) + \sum_{i=1}^{m} \lambda_i^k \phi_\varepsilon[f_i(x^k)] <$$
$$< f_o(x^k) + \sum_{i=1}^{m} \lambda_i^k \phi_\varepsilon'[f_i(x^k)] f_i(x^k) =$$
$$= f_o(x^k) + \sum_{i=1}^{m} \lambda_i^{k+1} f_i(x^k) = \psi(\lambda^{k+1}) \quad \text{Q.E.D.}$$

Having established the fact that the sequence $\psi(\lambda^k)$ is strictly increasing we are ready to establish the convergence of Algorithm 1.

Proposition 4: Assume that Algorithm 1 generates an infinite bounded sequence of vectors $\{(x^k, \lambda^k)\}$. Then any limit point (x, λ) of the sequence such that x is a feasible point is an optimal solution and Lagrange multiplier pair for problem (5).

Proof: Let $\{(x^k, \lambda^k)\}_{k \in K}$ be a subsequence converging to $(\bar{x}, \bar{\lambda})$ which satisfies the assumptions of the proposition. Consider also the subsequence $\{(x^{k+1}, \lambda^{k+1})\}_{k \in K}$. It contains a convergent subsequence $\{(x^{k+1}, \lambda^{k+1})\}_{k \in \tilde{K}} \to (\tilde{x}, \tilde{\lambda})$. Now for every $k \in \tilde{K}$ we have

$$\psi_\varepsilon(\lambda^k) = f_o(x^k) + \sum_{i=1}^{m} \lambda_i^k \phi_\varepsilon[f_i(x^k)] \leq f_o(x) + \sum_{i=1}^{m} \lambda_i^k \phi_\varepsilon[f_i(x)] \quad \forall\, x \in R^n$$

Hence taking limits

$$f_o(\bar{x})+\sum_{i=1}^{m}\bar{\lambda}_i\ \phi_\varepsilon[f_i(\bar{x})]\leq f_o(x)+\sum_{i=1}^{m}\bar{\lambda}_i\ \phi_\varepsilon[f_i(x)]\ \forall x\varepsilon R^n \quad (21)$$

and $\{\psi_\varepsilon(\lambda^k)\}_{k\varepsilon\tilde{K}}\to f_o(\bar{x})+\sum_{i=1}^{m}\bar{\lambda}_i\phi_\varepsilon[f_i(\bar{x})] = \psi_\varepsilon(\bar{\lambda})$. A similar argument shows that $\{\psi_\varepsilon(\lambda^{k+1})\}_{k\varepsilon\tilde{K}}\to\psi_\varepsilon(\tilde{\lambda})$ and since the sequence $\{\psi_\varepsilon(\lambda^{k+1})\}$ is monotonically increasing we have $\psi_\varepsilon(\tilde{\lambda})=\psi_\varepsilon(\bar{\lambda})$. Since for every $k\varepsilon K$

$$\lambda_i^{k+1}=\lambda_i^k\ \phi_\varepsilon'[f_i(x^k)] \quad i=1,..,m$$

it follows by taking limits that

$$\tilde{\lambda}_i=\bar{\lambda}_i\phi_\varepsilon'\ [f_i(\bar{x})] \quad i=1,..,m \qquad (22)$$

If $\tilde{\lambda}=\bar{\lambda}$ then it follows from (21) and (22) that $\bar{\lambda}_i$ is a Lagrange multiplier vector and the proposition is proved. Assume now that $\tilde{\lambda}\neq\bar{\lambda}$. We shall come to a contradiction. If $\bar{\lambda}_i>0$, $i=1,..,m$ the contradiction would be immediate since (21), (22) and Proposition 3 show that $\psi_\varepsilon(\tilde{\lambda})>\psi_\varepsilon(\bar{\lambda})$ while we have $\psi_\varepsilon(\tilde{\lambda})=\psi_\varepsilon(\bar{\lambda})$. If $\bar{\lambda}_i=0$ for some $i\varepsilon\{1,..,m\}$ then $\tilde{\lambda}_i=\bar{\lambda}_i=0$. Hence if $\tilde{\lambda}\neq\bar{\lambda}$ there must exist some index $j\varepsilon\{1,..,m\}$ such that $\tilde{\lambda}_j>0$, $\bar{\lambda}_j>0$ and $f_j(\bar{x})\neq 0$. These relations are sufficient to show that $\psi_\varepsilon(\tilde{\lambda})>\psi_\varepsilon(\bar{\lambda})$ using an entirely similar proof as in Proposition 3. Hence the contradiction is established in this case as well. Q.E.D.

Notice that in Proposition 4 the point $\bar{x}$ is required to be a feasible point. Now if $(\bar{x},\bar{\lambda})$ is the limit of a subsequence $\{(x^k,\lambda^k)\}_{k\varepsilon K}$ and $\bar{x}$ is such that some constraint is violated i.e. $f_i(\bar{x})>0$ for some i, then the corresponding Lagrange multiplier $\bar{\lambda}_i$ must be zero while we have $\lambda_i^{k+1}>\lambda_i^k>0$ whenever $f_i(x^k)>0$. Thus the algorithm tends to yield feasible points. In particular if the whole sequence $\{(x^k,\lambda^k)\}$ is convergent to $(\bar{x},\bar{\lambda})$ then $\bar{x}$ is a feasible point and an optimal solution for problem (5). Further conditions which guarantee that the algorithm yields feasible points $\bar{x}$ will be given in a future publication.

We mention that a number of variations of the basic Algorithm 1 are possible by allowing ε to change from one iteration to the next. We state one such variation as an algorithm.

Algorithm 2:

Consider Algorithm 1 where instead of keeping ε at a fixed predetermined value, a convergent sequence $\{\varepsilon_k\}$ with $\varepsilon_k>0$ is used, where ε_k corresponds to the kth unconstrained minimization problem. The sequence $\{\varepsilon_k\}$ may be either predetermined or computed during the course of the iteration.

An example of a scheme which fits within the framework of Algorithm 2 is when ε is initially fairly large and is gradually reduced to its limit. Since, as we shall see later, a smaller ε accelerates the convergence of the dual iteration but results in a more ill-conditioned unconstrained minimization, there is ample reason to believe that such schemes work well. Another scheme would be to start with a moderate value of ε and decrease it as necessary if the algorithm does not coverge fast enough. The extreme case of Algorithm 2 is when $\varepsilon_k\to 0$. Under these circumstances the algorithm works as a combined primal dual and penalty method with the penalty term tending to infinity. While it would appear that the resulting extreme ill-conditioning of the unconstrained problem should result in increased computational requirements, in experiments that we performed the algorithm with $\varepsilon_k\to 0$ worked very satisfactorily and gave a 50 to 70% reduction in computation time over the corresponding pure penalty method ($\varepsilon_k\to 0,\lambda^k$ fixed). This reduction must be attributed to the dual iteration. We shall have more to say about computational experience in the last section.

The convergence of Algorithm 2 can be easily established by using a proof identical to the one of Proposition 4 when $\{\varepsilon_k\}$ has a nonzero limit. For the case where $\varepsilon_k\to 0$ a proof similar to corresponding proofs for usual penalty function methods can be used.

Another variation of both Algorithms 1 and 2 is obtained when one uses a different parameter ε (or a different sequence $\{\varepsilon_k\}$) for each of the constraints. One rationale for such a strategy would be to compensate for constraints which are badly scaled.

In order to terminate the algorithm one may employ any of several error bounds. A very simple bound can be written when x^k happens to be a feasible point: $\psi_\varepsilon(\lambda^k)<f^*<f_o(x^k)$. If $\tilde{x}$ is any known interior point of the feasible region, an alternative upper bound can be given. Recall from the proof of Proposition 2 the inequality

$$f_o(x)+\sum_{i=1}^{m}\lambda_i^* f_i(x)\leq f_o(x)+\sum_{i=1}^{m}\lambda_i^k\phi_\varepsilon[f_i(x)]+\sum_{i=1}^{m}\varepsilon\lambda_i^k\sigma(\lambda_i^*/\lambda_i^k)$$

Hence $f^*\leq\psi_\varepsilon(\lambda^k)+\sum_{i=1}^{m}\varepsilon\lambda_i^k\sigma(\lambda_i^*/\lambda_i^k)$ Next suppose $\hat{\lambda}$ is an upper bound for λ^* i.e. $\lambda_i^*\leq\hat{\lambda}_i$ $i=1,2,..m$. Then $\psi_\varepsilon(\lambda^k)\leq f^*\leq\psi_\varepsilon(\lambda^k)+\sum\varepsilon\lambda_i^k\max[1,\sigma(\hat{\lambda}_i/\lambda_i^k)]$ It can be easily shown that an upper bound $\hat{\lambda}$ can be obtained from the interior point formula

$$\hat{\lambda}_i=\frac{\mu-f_o(\tilde{x})}{f_i(\tilde{x})}\geq\lambda_i^*$$

where $\tilde{x}$ is an interior point and $\mu\leq f^*$. In particular $\psi(\lambda^{k+1})$ can be used for μ.

An interesting geometric interpretation of Algorithm 1 can be given in terms of the primal functional [8] corresponding to the convex programming problems (5) and (13). This functional (otherwise referred to as optimal response function, perturbation function, minimum value function) is defined respectively as

$$\omega(z) = \inf f_o(x)$$
$$\text{s.t.}\quad f_i(x)\leq z_i \quad i=1,2,..m$$

$$\omega_\varepsilon(z) = \inf f_o(x)$$
$$\text{s.t.}\quad \phi_\varepsilon[f_i(x)]\leq z_i \quad i=1,2,..m$$

It is well known that the primal functional corresponding to a convex programming problem summarizes all the important characteristics of the problem and furthermore it determines completely the dual functional via the conjugacy relation [8],

$$\psi(\lambda) = \inf_z\{\omega(z)+\langle\lambda,z\rangle\}$$

$$\psi_\varepsilon(\lambda) = \inf_z\{\omega_\varepsilon(z)+\langle\lambda,z\rangle\}$$

A typical form of the primal functionals ω and ω_ε is given in Fig. 2. Now in terms of this figure one can interpret Algorithm 1 as follows:

Given λ^o, the unconstrained minimization of step 2 determines a point where λ^o is a support hyperplane to $\omega_\varepsilon(z)$. Then the vector λ^1, which is determined by the iteration of step 3, is a support hyperplane to the primal functional ω as shown in the figure. The next unconstrained minimization determines a point of support to $\omega_\varepsilon(z)$ for the hyperplane λ^1. This point is "projected" back on $\omega(z)$ to determine λ^2 and so on. The corresponding values of the dual functional are the points where these hyperplanes intersect the vertical axis. Proposition 3 guarantees that these intersection points go up towards the optimal value of the problem.

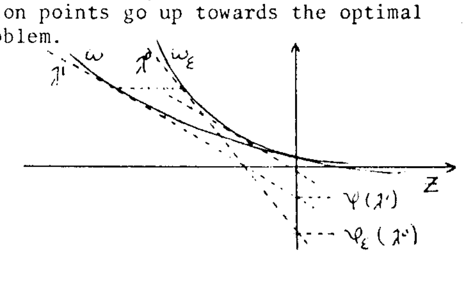

Figure 2.

For smaller $\varepsilon, \omega_\varepsilon$ has a sharper "knee" which increases the rate of convergence. This fact has been substantiated by an analysis that due to space limitation will be presented in a future publication.

In Fig. 3 we show the geometric interpretation of the pure penalty function method which results when λ^k is fixed and $\varepsilon_k \to 0$. Figure 4 shows the geometric interpretation of Algorithm 2 where λ^k is updated via the dual iteration and $\{\varepsilon_k\}$ is the same sequence tending to zero. It is apparent from the comparison of the figures that the addition of the dual iteration accelerates considerably the convergence. Effectively this means that the algorithm will converge before ε is decreased to extremely low values.

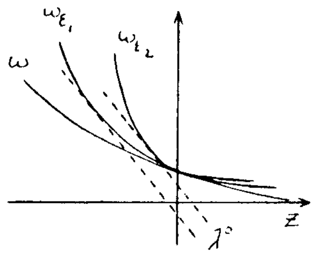

Figure 3.

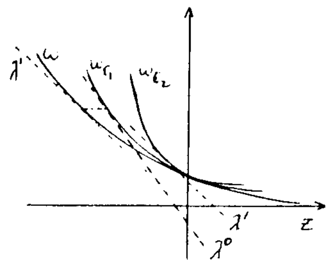

Figure 4.

III. Extension to the Nonconvex Case

Consider now the general nonlinear programming problem

$$\text{minimize } f(x) \tag{23}$$
$$\text{subject to } g_i(x) \leq 0 \quad i=1,2,..m$$
$$h_j(x) = 0 \quad j=1,2,..,p$$

where f, g_i, h_j are continuously differentiable functions. The proposed algorithm has the same form as in the convex programming problem. In Section 1 we saw that the equality constraints can be handled by a quadratic penalty term. Just as the ϕ_ε penalty was selected from a general class satisfying P1-P5, we can introduce a class of penalties for equality constraints. The quadratic choice happens to be one of the more obvious examples. Consider a real valued twice differentiable function $\theta: R \to R$ satisfying the following six properties.

(Pa) $\theta(0)=0$

(Pb) $\theta'(0)=0$

(Pc) $\lim_{t\to+\infty} \theta'(t)=+\infty$

(Pd) $\lim_{t\to-\infty} \theta'(t)= -\infty$

(Pe) $\theta''(t) > 0 \ \forall t \in R$

(Pf) $\theta''(0) = 1$

Examples:

1) $\theta(t) = 1/2\ t^2$ 2) $\theta(t) = \cosh t - 1$

For $\varepsilon > 0$ define $\theta_\varepsilon(t) = \varepsilon\ \theta(t/\varepsilon)$.

Given $\lambda^k \in R^m, \mu^k \in R^p, \varepsilon > 0$ with $\lambda_i^k > 0, i=1,2,..m$, solve the unconstrained problem

$$\text{minimize } f(x) + \sum_{i=1}^{m} \lambda_i^k \phi_\varepsilon[g_i(x)] + \sum_{i=1}^{p} \mu_j^k h_j(x) + \sum_{i=1}^{p} \theta_\varepsilon[h_j(x)] \tag{24}$$

where ϕ_ε is as in Section 2 and θ_ε is as defined above. If x^k is a solution point of (24) set

$$\lambda_i^{k+1} = \lambda_i^k \ \phi_\varepsilon'[g_i(x^k)] \quad i=1,2,..m \tag{25}$$
$$\mu_j^{k+1} = \mu_j^k + \theta_\varepsilon'[h_j(x^k)] \quad j=1,2,..p \tag{26}$$

and repeat the iteration.

Unlike the convex case, convergence cannot be guaranteed for arbitrary choices of ϕ_ε and θ_ε. The unconstrained problem (24) may have only local minima and convergence may depend on choosing ε smaller than some critical threshold. Conditions under which the algorithm is convergent for non-convex problems will be given in a future publication.

A geometrical interpretation based on the primal functional can be made for the generalized algorithm. In particular we can illustrate the case for a problem with equality constraints only. We define the primal functionals

$$\omega(z) = \inf f(x) \quad \text{s.t. } h_j(x)=z_j,\ j=1,2,..,p$$

$$\omega_\varepsilon(z) = \inf f(x) + \sum_{j=1}^{p} \theta_\varepsilon[h_j(x)] \quad \text{s.t. } h_j(x)=z_j\ j=1,2,..,p$$

Clearly $\omega_\varepsilon(z) = \omega(z) + \sum_{j=1}^{p} \theta_\varepsilon(z_j)$

If μ^k defines a support hyperplane to $\omega_\varepsilon(z)$ at $z=z^k$, then μ^{k+1} defines a tangent hyperplane to $\omega(z)$ at $z=z^k$. Figure 5 shows this relation.

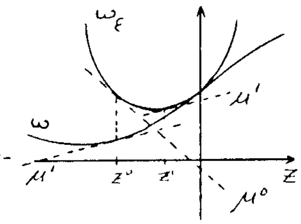

Figure 5.

IV. Computational Experience

The algorithm of this paper was used to solve some standard test problems and performed very satisfactorily. It generally performed considerably better than its equivalent penalty function version ($\varepsilon \to 0, \lambda$ fixed). Extensive experiments have been performed on only one convex problem, the Rosen-Suzuki problem [9]. The function ϕ chosen was the "quadratic-reciprocal" function (9) and the unconstrained minimization routine was the Fletcher-Powell method (available on the IBM-360 as the "FMFP" Scientific Subroutine). Two different methods of updating ε were used. In the first method ε was given an initial moderate value ε_1 which was reduced to a fixed value ε_2 for the second and subsequent minimizations. For a wide range of values of ε_2 the number of function evaluations for solving the problem was from 130 to 200 for six significant digit accuracy. The second method of updating was by preselecting a sequence $\varepsilon_k \to 0$. For various sequences and starting points the number of function evaluations required were around 150. When a sequence $\varepsilon_k \to 0$ was used but λ was not updated, i.e. the method was operated as a usual penalty function method the number of function evaluations was around 380. The reduction in number of function evaluations when λ was updated was in the order of 50%-70% for a variety of runs. Generally the performance of the algorithm was relatively insensitive to the value and method of updating of the parameter ε. We note finally that from a limited number of experiments it was indicated that some savings in computation time can be obtained when only moderate accuracy is demanded in the initial unconstrained minimizations.

V. References

1. Luenberger, D.G., Introduction to Linear and Nonlinear Programming, Addison-Wesley (in press).
2. Powell, M.J.D., "A Method for Nonlinear Constraints in Minimization Problems," in Optimization, R. Fletcher (ed.), Academic Press, 1969. pp 283-298.
3. Hestenes, M.R., "Multiplier and Gradient Methods," J. Optimization Theory Appl. No. 4, 1969, pp 303-320.
4. Haarhoff, P.C. and Buys, J.D., "A New Method for the Optimization of a Nonlinear Function Subject to Nonlinear Constraints," Comput.J.Vol.13, 1970, pp 178-184.
5. Buys, J.D., "Dual Algorithms for Constrained Optimization," Ph.D. Thesis, Rijksuniversiteit te Leiden, June 1972.
6. Miele, A., Moseley, P.E., Levy, A.U. and Coggins, G. M., "On the Method of Multipliers for Mathematical Programming Problems," J.Optimization Theory, Appl. No.1, 1972, pp.1-33.
7. Rockafellar, R.T., Convex Analysis, Princeton Univ. Press, 1970.
8. Luenberger, D.G., Optimization by Vector Space Methods, J. Wiley, 1969.
9. Kowalik, J., Osborne, M.R. and Ryan D.M., "A New Method for Constrained Optimization," Op.Res. No.17 1969, p. 973.