

A PROJECTED LAGRANGIAN ALGORITHM  
FOR SEMI-INFINITE PROGRAMMING

by

I. D. Coope                      and                      G. A. Watson \*

No. 32

September, 1983

Abstract    A globally convergent algorithm is presented for the solution of a wide class of semi-infinite programming problems. The method is based on the solution of a sequence of equality constrained quadratic programming problems, and usually has a second order convergence rate. Numerical results illustrating the effectiveness of the method are given.

Keywords    Semi-infinite programming, descent algorithm, penalty function, Lagrangian function, global convergence, second order convergence.

\* Department of Mathematical Sciences,  
University of Dundee, Scotland.

## 1. Introduction

There is currently increasing interest in that class of optimization problems known as semi-infinite programming problems, which are characterized by a finite number of variables and an infinite number of constraints. Such problems arise, for example, in air pollution control, in the solution of weakly singular integral equations, in probability distributions, etc.: details of these and other applications may be found in the conference proceedings [ 3 ], [ 8 ] and in papers referenced therein. In particular, [ 3 ] is useful as a state-of-the-art treatment of the subject area. It is clear that a large body of theory exists for semi-infinite programming problems, although with the exception of some special cases, such as continuous linear Chebyshev approximation, the algorithmic development is less far advanced. This is not to say that the provision of algorithms for general problems has been entirely neglected. A number of locally convergent methods have been published, mainly based on the application of Newton's method to first order necessary conditions for a solution (for example [10] [11]). Suitable initial approximations can often be obtained through the solution of a discretization of the original problem, and this has led to the formulation of two (or even three) phase methods (see, for example, [ 6 ] and the review paper [ 9 ]).

To our knowledge, the only methods developed so far which claim to be globally convergent are those given in [13], [14] (although a conceptual globally convergent method based on continuation is suggested in [ 5 ]). The algorithm presented in [14] adapts to the present situation a well-established technique for the globalization of methods for finite problems, involving the minimization of an exact penalty function, and as the numerical results in [14] show, can perform well.

However, it has two main disadvantages: firstly each iteration requires the solution of an inequality constrained quadratic programming problem and secondly, fast ultimate convergence depends on conditions which may well not be satisfied. The purpose of this paper is to present a modification of that method which, in particular, overcomes these difficulties.

We begin by introducing the problem to be solved and some notation. Let  $X \subset \mathbb{R}^N$  be a Cartesian product of closed intervals and let  $g: X \times \mathbb{R}^n \rightarrow \mathbb{R}$ , with  $g$  twice continuously differentiable as a function of its parameters. Let  $f: \mathbb{R}^n \rightarrow \mathbb{R}$  be a twice continuously differentiable function, and consider the problem:

$$\text{find } \underline{a} \in \mathbb{R}^n \text{ to minimize } f(\underline{a}) \quad (1.1)$$

$$\text{subject to } g(\underline{x}, \underline{a}) \leq 0, \forall \underline{x} \in X.$$

The algorithm to be developed will be capable of finding a stationary point of (1.1), defined as follows. We let  $\phi_j(\underline{a})$  and  $\psi_j(\underline{x}, \underline{a})$  respectively denote the partial derivatives of  $f$  and  $g$  with respect to  $a_j$ ,  $j = 1, 2, \dots, n$ , although the explicit dependence on  $\underline{x}$  and  $\underline{a}$  will sometimes be suppressed when no confusion can arise.

Definition 1. Let  $\underline{a}^* \in \mathbb{R}^n$  be feasible in (1.1) and let there exist points  $\underline{x}_i^*$  with  $g(\underline{x}_i^*, \underline{a}^*) = 0$ ,  $i = 1, 2, \dots, t$  and non-negative multipliers  $\lambda_i^*$ ,  $i = 1, 2, \dots, t$  such that

$$\phi_j(\underline{a}^*) + \sum_{i=1}^t \lambda_i^* \psi_j(\underline{x}_i^*, \underline{a}^*) = 0, \quad j = 1, 2, \dots, n. \quad (1.2)$$

Then  $\underline{a}^*$  is a stationary point of (1.1).

The Fritz-John first order necessary conditions for  $\underline{a}^*$  to solve (1.1) correspond to (1.2) with  $\phi_j(\underline{a}^*)$  replaced by  $\lambda_0 \phi_j(\underline{a}^*)$ , where  $\lambda_0^* \geq 0$  (for example [1]). If a first order constraint qualification holds at  $\underline{a}^*$  then  $\lambda_0^* \neq 0$  and (1.2) becomes the usual (Kuhn-Tucker) necessary conditions.

A key assumption in the algorithm developed in [14] is that in a neighbourhood of  $\underline{a}^*$ , the variables  $\underline{x}_i$  representing local maxima of the function  $g(\underline{x}, \underline{a})$  can be eliminated from (1.2) by expressing them as functions of  $\underline{a}$ . For this, we require, in particular that the implicit function theorem be applicable to any zero-derivative conditions characterizing these local maxima. The algorithm in fact requires the more general assumption that an analogous elimination be possible at all points encountered in the solution process, and this forces restriction of points  $\underline{a}$  being considered to a subset  $B$ , say, of  $R^n$  at which appropriate conditions hold. We now define such a subset and require some further notation. In particular, for given  $\underline{a} \in R^n$ , let  $E(\underline{a})$  denote the set of local maxima of  $g(\underline{x}, \underline{a})$  in  $X$  which satisfy  $g(\underline{x}, \underline{a}) \geq -\eta$ , where  $\eta > 0$  is a prescribed constant. (The role of  $\eta$  will become clear subsequently; however it is a minor one and we will not explicitly show dependence upon it). Let  $\underline{x} \in E(\underline{a})$ , with  $\sigma_1, \sigma_2, \dots, \sigma_\ell$  the indices of the components of  $\underline{x}$  on the boundary of  $X$ . Also let  $\nabla_1 g$  denote the vector in  $R^\ell$  whose components are the partial derivatives of  $g$  with respect to the components of  $\underline{x}$  in these positions (evaluated at  $\underline{x}, \underline{a}$ ).

Let  $\nabla_2 g$  denote the vector in  $R^{N-\ell}$  whose components are the partial derivatives of  $g$  with respect to the remaining components of  $\underline{x}$  and let  $\nabla_2^2 g$  denote the corresponding  $(N-\ell) \times (N-\ell)$  matrix of second partial derivatives of  $g$ . Then we define  $B \subset R^n$  as the open set of points (assumed non-empty) such that

- (1) there are a finite number of points  $\underline{x} \in X$  such that
 
$$\nabla_2 g = 0,$$
- (2) at each point of (1) corresponding to a local maximizer of  $g$ ,  $\nabla_2^2 g$  is negative definite and each component of  $\nabla_1 g$  is non-zero.

The significance of these assumptions is that for  $\underline{a} \in B$ , appropriate components of each of the  $p$  local maximizers  $\underline{x}_i$  of  $g(\cdot, \underline{a})$  may be regarded locally as differentiable functions of the parameters  $\underline{a}$  through the use of the implicit function theorem applied to

$$\nabla_2 g(\underline{x}_i, \underline{a}) = 0, \quad i = 1, 2, \dots, p. \quad (1.3)$$

For any  $\underline{a} \in B$ , we may therefore define the Lagrangian function

$$\mathcal{L}(\underline{a}, \underline{\lambda}) = f(\underline{a}) + \sum_{i=1}^p \lambda_i h_i(\underline{a}), \quad (1.4)$$

where

$$h_i(\underline{a}) = g(\underline{x}_i(\underline{a}), \underline{a}), \quad i = 1, 2, \dots, p \quad (1.5)$$

and the functions  $\underline{x}_i(\underline{a})$ ,  $i = 1, 2, \dots, p$ , are defined by (1.3).

In particular, if  $\underline{a}^* \in B$ , then it is easy to see the relationship between (1.2) and the usual first order necessary conditions, as the required result carries over directly from the finite case: we may write (1.2) as

$$\nabla_{\underline{a}} \mathcal{L}(\underline{a}^*, \underline{\lambda}^*) = \underline{0}, \quad (1.6)$$

where without loss of generality we may assume that  $t = p(\underline{a}^*)$ .

In addition we may obtain the corresponding second order conditions in an analogous manner. The relevant results are as follows.

Theorem 1. Let  $\underline{a}^* \in B$  solve (1.1) and let there exist  $\underline{\lambda}^*$  such that (1.6) holds. Then if a certain regularity assumption holds

$$\underline{s}^T [\nabla_{\underline{a}}^2 \mathcal{L}(\underline{a}^*, \underline{\lambda}^*)] \underline{s} \geq 0$$

for all  $\underline{s} : \underline{s}^T \nabla h_i(\underline{a}^*) = 0, i = 1, 2, \dots, t$ .

Theorem 2. Let  $\underline{a}^* \in B$  be a feasible point satisfying (1.6) and let

$$\underline{s}^T [\nabla_{\underline{a}}^2 \mathcal{L}(\underline{a}^*, \underline{\lambda}^*)] \underline{s} > 0$$

for all  $\underline{s} : \underline{s}^T \nabla h_i(\underline{a}^*) = 0, i = 1, 2, \dots, t$ . Then  $\underline{a}^*$  is a local solution of (1.1).

The basis of the algorithm developed here is the systematic provision of descent directions for an exact penalty function appropriate to (1.1). Given  $\theta > 0$  such a function may be defined for  $\underline{a} \in B$  by

$$P(\underline{a}) = f(\underline{a}) + \theta \sum_{i=1}^p [h_i(\underline{a})]_+, \quad (1.7)$$

where  $h_i(\underline{a}), i = 1, 2, \dots, p$  is given by (1.5) and (1.3). It may be shown by the standard arguments available in the finite case that provided  $\theta > \lambda_i^*, i = 1, 2, \dots, t$  then  $\underline{a} \in B$  minimizing  $P(\underline{a})$  is a stationary point of (1.1) in the sense that  $\underline{a}$  is feasible in (1.1) and satisfies (1.6) [2].

In the next section we show how descent directions for  $P$  may be calculated at any approximation  $\underline{a} \in B$ , and in section 3 we describe a suitable active set strategy. This leads to the detailed statement of the proposed algorithm which is given in section 4. Numerical results for an implementation of the algorithm, applied to a number of semi-infinite programming problems, are presented in section 5.

## 2. The calculation of descent directions

In the algorithm of [14], a descent direction  $\underline{d}$  for  $P(\underline{a})$  at a point  $\underline{a} \in B$  is obtained through the solution of the following quadratic programming problem.

$$\text{Find } \underline{d} \in \mathbb{R}^n \text{ to minimize } \underline{d}^T \phi(\underline{a}) + \frac{1}{2} \underline{d}^T H \underline{d} \quad (2.1)$$

subject to  $h_i(\underline{a}) + \underline{d}^T \nabla h_i(\underline{a}) \leq 0, i = 1, 2, \dots, p,$

where  $H$  is a given positive definite matrix. It is shown in [13] that a solution  $\underline{d}$  to (2.1) is a descent direction for  $P$  at  $\underline{a}$  provided that there is a Lagrange multiplier vector associated with (2.1) which has no component larger than  $\theta$ . On the other hand, if  $\underline{d} = \underline{0}$  solves (2.1) then examination of the Kuhn-Tucker conditions for (2.1) shows that  $\underline{a}$  is a stationary point of (1.1).

A key property possessed by (2.1) is that if  $H$  is chosen to be  $\nabla^2 \mathcal{L}(\underline{a}, \underline{\lambda})$ , with  $\underline{a}$  and  $\underline{\lambda}$  approximations to the optimal values, then in a neighbourhood of a stationary point  $\underline{a}^*$ ,  $\underline{d}$  solving (2.1) is, under mild conditions, the increment in  $\underline{a}$  given by

the application of Newton's method to the system of equations (1.6) together with  $h_i(\underline{a}) = 0$ ,  $i = 1, 2, \dots, p$ . A potential difficulty however, is that  $\nabla^2 \mathcal{L}(\underline{a}^*, \underline{\lambda}^*)$  may have negative eigenvalues, so that if H is forced to be positive definite, then the desirable connection with Newton's method is lost. The intention here is to maintain that connection by working with an appropriate equality constraint quadratic subproblem instead of with (2.1).

For given  $\underline{a} \in B$ , let C be the  $n \times p$  matrix whose  $j^{\text{th}}$  column is  $\nabla h_j(\underline{a})$ ,  $j = 1, 2, \dots, p$  and consider the quadratic programming problem:

$$\begin{aligned} \text{find } \underline{d} \in \mathbb{R}^n \text{ to minimize } & \underline{d}^T \underline{\phi} + \frac{1}{2} \underline{d}^T H \underline{d} \\ \text{subject to } & C^T \underline{d} + \underline{h} = \underline{0}, \end{aligned} \quad (2.2)$$

where  $\underline{h} \in \mathbb{R}^p$  has  $j^{\text{th}}$  component  $h_j(\underline{a})$  and where H is a given  $(n \times n)$  symmetric matrix. We assume for the moment that  $p \leq n$ .

Let the QR factorization of C (which we will assume to have full rank) be given by

$$C = [W:Z] \begin{bmatrix} U \\ \underline{0} \end{bmatrix} \quad (2.3)$$

where  $W: \mathbb{R}^p \rightarrow \mathbb{R}^n$ ,  $Z: \mathbb{R}^{n-p} \rightarrow \mathbb{R}^n$  with  $[W:Z]$  orthogonal, and

$U: \mathbb{R}^p \rightarrow \mathbb{R}^p$  is upper triangular. Let

$$\underline{d} = W \underline{d}_1 + Z \underline{d}_2.$$

Then if  $\underline{d}_1$  is chosen to satisfy

$$U^T \underline{d}_1 = -\underline{h}, \quad (2.4)$$

it is clear that the constraints of (2.2) are satisfied and we are left with an unconstrained problem in the vector  $\underline{d}_2$ .

Setting the derivative with respect to  $\underline{d}_2$  to zero then gives

$$Z^T H Z \underline{d}_2 = -Z^T (\underline{q} + H W \underline{d}_1), \quad (2.5)$$

which has a unique solution provided that the reduced matrix  $Z^T H Z$  is non-singular. If the conditions of Theorem 2 are satisfied at a stationary point  $\underline{a}^*$ , then  $Z^T [\nabla^2 \mathcal{L}(\underline{a}, \underline{\lambda})] Z$  is positive definite in a neighbourhood of this point and it is clear that the choice  $H = \nabla^2 \mathcal{L}(\underline{a}, \underline{\lambda})$  is appropriate (for some suitably chosen  $\underline{\lambda}$ ).

If  $\nabla^2 \mathcal{L}$  is not positive definite (and there is no reason why it should be) at a stationary point, then the choice  $H = \nabla^2 \mathcal{L} + \mu I$  can be used with  $\mu$  chosen to force positive definiteness of  $H$ . This is the strategy used in [14] but it has the disadvantage of slowing the rate of convergence on some problems. Clearly the matrix  $Z^T H Z$  is the one which should be forced to be positive definite since then the choice  $H = \nabla^2 \mathcal{L} + \mu I$  will allow  $\mu = 0$  to be chosen in a neighbourhood of the solution where the conditions of Theorem 2 hold. On the other hand, forcing only  $Z^T H Z$  to be positive definite may cause the solutions to the quadratic programming subproblems (2.1) or (2.2) to be non-descent directions for the penalty function (1.7) as the following example illustrates.

$$\min\{a_1 + a_2\} \quad (2.6)$$

$$\text{subject to } x - a_1 a_2 \leq 0, \quad x \in [0, 1].$$

$$a_1, a_2 > 0.$$

The extra conditions  $a_1, a_2 > 0$  have been added to exclude the lower branch of the hyperbolic constraint; they are not active at the solution which is clearly given by  $\underline{a}^* = (1, 1)^T$ .

The set  $E(\underline{a}) = \{1\}$  is independent of  $\underline{a}$  in this simple example and  $t = p(\underline{a}^*) = p(\underline{a}) = 1$  with the optimal Lagrange multiplier easily calculated to be  $\lambda^* = 1$ . The definitions (1.4), (1.5) then give

$$\nabla^2 \mathcal{L}(\underline{a}, \lambda) = \begin{bmatrix} 0 & -\lambda \\ -\lambda & 0 \end{bmatrix}$$

which is clearly indefinite.

Suppose now that  $\underline{a} = (1 - \varepsilon, 1 - \varepsilon)^T$  and  $\lambda = \frac{1}{2}$  is the current approximation to  $(\underline{a}^*, \lambda^*)$  and choose  $H = \nabla^2 \mathcal{L}(\underline{a}, \lambda)$  so that  $Z^T H Z = \frac{1}{2}$ .

Then the solution to (2.1) or (2.2) gives

$$\underline{d} = \beta(1, 1)^T, \quad \beta = \varepsilon + \frac{1}{2}\varepsilon^2/(1 - \varepsilon) \quad (2.7)$$

which is clearly an excellent direction for  $0 < \varepsilon < 1$ . The Lagrange multiplier associated with (2.1) or (2.2) is

$$\hat{\lambda} = \frac{1 - \beta/2}{1 - \varepsilon} > 1 = \lambda^*.$$

However, the direction  $\underline{d}$ , given by (2.7) is not a descent direction for the penalty function (1.7) when  $\theta = \hat{\lambda}$  because the directional derivative,  $P'(\underline{a}; \underline{d}) = \beta^2 > 0$ . Fortunately it is always possible (and appropriate) to make  $\underline{d}$  a descent direction in such cases by increasing the penalty parameter  $\theta$  and the following results allow a threshold value to be determined.

Theorem 3. If  $Z^T H Z$  is positive definite and  $C^T Z = [0]$  where  $[C:Z]$  has full rank, then  $\exists \bar{\sigma}$  such that  $[H + \sigma C C^T]$  is positive definite for all  $\sigma > \bar{\sigma}$ .

This is essentially a restatement of the result in [4; p132].

Theorem 4. If  $\hat{\underline{d}}$  solves the quadratic programming problem (2.2) then  $\hat{\underline{d}}$  also solves:

$$\begin{aligned} \min \{ \hat{\underline{d}}^T \phi + \frac{1}{2} \hat{\underline{d}}^T [H + \sigma C C^T] \hat{\underline{d}} \} \\ \text{subject to } C^T \hat{\underline{d}} + \underline{h} = \underline{0}, \end{aligned} \quad (2.8)$$

for any value of  $\sigma$  and the Lagrange multipliers of problem (2.8)  $\hat{\lambda}(\sigma)$  say, are related to those of problem (2.2) by the equation

$$\hat{\lambda}(\sigma) = \hat{\lambda}(0) + \sigma \underline{h}. \quad (2.9)$$

This result is a straightforward application of the Kuhn-Tucker conditions.

Using Theorems 3 and 4 and Han's result [7], we deduce that the solution to problem (2.1) will be a descent direction for the penalty function (1.7) provided that the parameter  $\theta$  satisfies

$$\theta \geq \hat{\lambda}_i(\bar{\sigma}) \quad , \quad i = 1, 2, \dots, p. \quad (2.10)$$

At points remote from  $\underline{a}^*$  it may be necessary to increase  $\theta$  in order to ensure the satisfaction of the inequalities (2.10). Of course  $\bar{\sigma}$  will not generally be known so in the algorithm described in section 4 if  $\underline{d}$ , calculated to solve problem (2.1), is not a descent direction then the current value of  $\theta$  must be too small and so it is repeatedly doubled until the directional derivative changes sign. (For a more detailed description of the overall strategy for determining a suitable value of  $\theta$  see section 4).

### 3. Active set strategy

In order to obtain a suitable descent direction for the penalty function (1.7) we have shown that, if the correct active set of constraints has been identified, it is appropriate to solve problem (2.2) with the matrix

$$H = \nabla^2 \mathcal{L}(\underline{a}, \underline{\lambda}) + \mu I, \quad (3.1)$$

where  $\mu (\geq 0)$  is chosen so that  $Z^T H Z$  is positive definite. Now the matrix  $Z$  depends on the matrix  $C$  and this complicates matters when some of the constraints of problem (2.2) would not be active at the solution to problem (2.1). This situation is easily identified

since it implies that at least one of the Lagrange multipliers of problem (2.2) is negative, indicating that a constraint should be dropped. A further difficulty now arises in that dropping a constraint causes a column to be deleted from the matrix  $C$ , which in turn results in the matrix  $Z$  defined by equation (2.3) having an extra column. In this case the matrix  $Z^T H Z$  is increased in size by one row and column and it is possible that the current value of  $\mu$  may not be large enough to guarantee positive definiteness of this new  $Z^T H Z$  matrix. It is, therefore, necessary to consider increasing  $\mu$  when a constraint is deleted. Of course, increasing  $\mu$  may mean that the deleted constraint should be reinserted and this is accounted for in the strategy adopted which essentially solves a sequence of problems of the form (2.2) but adjusts the number of equality constraints and the value of  $\mu$  automatically until a solution to (2.1) is obtained. Specifically, the active set method described in [4, pp88-90] is used except that initially, and whenever a constraint is dropped from the active set, the matrix  $H$  is revised if necessary to make  $Z^T H Z$  positive definite by increasing  $\mu$  according to the formula

$$\mu := 4\mu + \mu_{\min}, \quad (3.2)$$

and by repeatedly applying (3.2) until positive definiteness is obtained. Initially  $\mu = 0$  is always tried, which explains the need for the extra parameter  $\mu_{\min} (>0)$ .

In a neighbourhood of  $\underline{a}^*$ , the solution to equations (2.3) - (2.5) will also usually solve problem (2.1) and the active set strategy is not invoked. If this is not the case, then the solution to problem (2.2) provides an initial feasible point for the active

set strategy. The only potential difficulty occurs when  $p > n$  in which case problem (2.2) has no solution, whereas we assume problem (2.1) always has a solution. This difficulty is also easily resolved using the active set strategy. All that is required is that some subset of the constraints be used to define an initial feasible point for problem (2.1). In the algorithm described in the next section we have assumed, for simplicity, that satisfaction of the  $n$  most violated constraints (i.e. those with larger values of  $h_i$ ) always gives rise to a feasible point. This is not foolproof but almost always  $p \leq n$  was satisfied for the problems solved in section 5. An exception is in the case of problem 9 which has infinitely many local maxima in the set  $E(\underline{a}^*)$  and this caused failure of the algorithm. (Note, however, that both assumptions (1) and (2) of section 1 are not satisfied in this case).

#### 4. A projected Lagrangian algorithm

The algorithm described in this section models very closely that described in [14]; the major difference is due to the method of calculating the descent direction  $\underline{d}$  for the penalty function (1.7) and this has already been discussed in sections 2 and 3. In order to guarantee descent of the penalty function  $P$  a suitable step length  $\gamma$  in the direction  $\underline{d}$  must be determined. As in [14]  $\gamma$  is chosen as the largest member of the sequence  $\{1, \frac{1}{2}, \frac{1}{4}, \dots\}$  satisfying

$$T(\gamma, \underline{a}) \geq \rho, \quad (4.1)$$

where

$$T(\gamma, \underline{a}) = \frac{P(\underline{a} + \gamma \underline{d}) - P(\underline{a})}{\gamma P'(\underline{a}, \underline{d})} \quad (4.2)$$

and  $P'(\underline{a}; \underline{d}) < 0$  is the directional derivative of the penalty function (1.7) at  $\underline{a}$  in the direction  $\underline{d}$ . The value  $\rho = .0001$  was used as in [14]<sup>1</sup>.

In order to calculate H, defined by equations (3.1) and (1.4), at the start of each iteration it is necessary to obtain approximations to the Lagrange multipliers  $\lambda_i$ ,  $i = 1, 2, \dots, p$ . Because successive values of p may change at each iteration, (particularly far from a stationary point) it is not appropriate to use the Lagrange multipliers of the problem (2.1) as estimates. Instead,  $\underline{\lambda}$  was chosen at the start of each iteration to minimize the least squares norm of the vector

$$\nabla \mathcal{L}(\underline{a}, \underline{\lambda}) = \phi(\underline{a}) + \sum_{i=1}^p \lambda_i \psi(\underline{x}_i, \underline{a}) = \phi + C\underline{\lambda}$$

with any negative components replaced by zeros. This least squares problem is easily solved, using the QR factorization of the constraints of problem (2.2), by back substitution in the triangular system

$$U\underline{\lambda} = -W^T \phi .$$

To complete the definition of H it is necessary to fix a suitable value for  $\mu_{\min}$  to be used in (3.2). In early numerical trials the value  $\mu_{\min} = 1$  was used and this usually gave good results. However, on some problems when  $Z^T \nabla^2 \mathcal{L}(\underline{a}, \underline{\lambda}) Z$  was singular it became clear that a smaller value was sometimes more appropriate particularly when the current approximation  $\underline{a}$  was far from the solution. Also when  $Z^T \nabla^2 \mathcal{L}(\underline{a}, \underline{\lambda}) Z$  has large negative eigenvalues it makes sense to consider increasing  $\mu_{\min}$ . Therefore the following rules were used to adjust  $\mu_{\min}$  (and hence  $\mu$  through (3.2)) from iteration to iteration.

<sup>1</sup> In [14], p. 199 the values quoted for  $\rho$  (there called  $\sigma$ ) and the limiting directional derivative should have the number of zeros interchanged.

(1) If  $\mu = \mu_{\min}$  and  $T > .5$  and  $\gamma = 1$  set  $\mu_{\min} = \mu_{\min}/4$

(2) If  $\mu = \mu_{\min}$  and  $\gamma \leq \frac{1}{4}$  set  $\mu_{\min} = \mu_{\min} * 4$ .

The motivation for rule (1) is that if the test is satisfied it indicates that the current value of  $\mu$  (necessarily greater than zero) may be causing too small a step to be taken in the direction defined by the quadratic programming subproblem. Because  $T > .5$  there is reasonable agreement with a linear model and decreasing  $\mu_{\min}$  by a factor of 4 should cause a corresponding increase in the size of  $\tilde{d}$  used for the next iteration. In rule (2) the situation suggests that the current step size,  $\gamma$ , has had to be decreased from 1 because the current value of  $\mu$  is too small and  $\mu_{\min}$  is therefore increased. These rules were found to work very well on the range of problems tested and were the ones used to obtain the numerical results of the next section.

Finally, we give the rules used for adjusting the penalty parameter  $\theta$ . If the initial choice is big enough then there is no need to consider adjustment in theory. In practice, though, too large a value of  $\theta$  can retard the rate of convergence by placing too much emphasis on reducing constraint violations. The initial value of  $\theta$  was 1 with  $\theta$  reset to the maximum value of the Lagrange multipliers at the solution to problem (2.1) whenever this value exceeded  $\theta$ . This guarantees that the direction calculated, as described in section 2, is a descent direction when  $H$  is positive definite, but if  $H$  has negative eigenvalues and the current  $\theta$  does not give rise to a descent direction then  $\theta$  is temporarily increased further as described in section 2. Equation (2.9) shows that this latter device will not usually be required close

to the solution because  $\hat{\lambda}(\sigma) \rightarrow \hat{\lambda}(0)$  as the solution is approached. This is borne out by the numerical results for problem 14 presented in the next section.

#### Algorithm summary

- (1) Given an approximation  $\underline{a}$  to  $\underline{a}^*$  determine the set  $E(\underline{a})$  defined in section 1.
- (2) Determine a direction  $\underline{d}$  as described in sections 2,3 and increase  $\theta$  if necessary so that  $\underline{d}$  is a descent direction for  $P(\underline{a},\theta)$ .
- (3) Set  $\underline{a} = \underline{a} + \gamma \underline{d}$  with  $\gamma$  chosen to satisfy (4.1).
- (4) If insufficient accuracy go to step (1).

Step (1) was implemented as in [14] by superimposing a uniform grid on the set  $X$  and identifying local maxima on the discrete set of points thus obtained. These discrete local maxima were then refined using a Newton-like iteration. The value  $\eta = 0.5$  was used in the definition of the set  $E(\underline{a})$  given in section 1, in order to obtain a valid comparison with the results presented in [14] and summarized in Table 1.

#### 5. Numerical Results

The algorithm was coded in ALGOL on the Burroughs B6930 computer of the University of Canterbury, which gives about 11 decimal places for single precision. For comparison purposes the algorithm was applied to the thirteen test problems listed in [14] and the results are summarized in Table 1. The column headed  $k$  gives the number of iterations required to increase the (negative) directional

derivative to a value greater than  $-.00001$ , that headed P' gives the final value of the directional derivative and t gives the final number of active points. Numbers in brackets correspond to values obtained by the algorithm in [14] and it can be seen that the present algorithm compares favourably. There is no significant difference in the two algorithms on problems 1,2,7,10,12, and 13; of course, it is to be expected that the two algorithms will give similar results whenever  $\nabla^2 \mathcal{L}$  remains positive definite throughout the calculation. Significant improvements were, however, obtained on problems 3,4(n = 6,8), 8 and 11. Problem 4 is interesting because  $\nabla^2 \mathcal{L}^*$  is singular and it is worth considering this problem in detail:

Problem 4

$$x = [0,1]$$

$$f = \sum_{i=1}^n \frac{a_i}{i}$$

$$g = \tan x - \sum_{i=1}^n a_i x^{i-1}.$$

n = 3 Starting point  $(0,0,0)^T$ ;  $\tilde{a}^* = (0.08910, 0.42305, 1.04526)^T$ ,  
 $f^* = 0.649042$

n = 6 Starting point  $(0,0,0,0,0,0)^T$ ;  $\tilde{a}^* = (0.0, 1.02326, -0.24060,$   
 $1.22168, -1.38826, 0.94133)$ ,  $f^* = 0.616085$ .

n = 8 Starting point:  $\tilde{a}^*$  for n = 6, to 3 decimal places, with zeros  
in the last two components;  $\tilde{a}^* = (0.0, 1.00342, -0.06095,$   
 $0.75112, -1.40994, 2.65270, -2.311602, 0.03267)$ ,  $f^* = 0.615653$ .

In each case  $\tilde{a}^*$  is the point reached by the algorithm when terminated and  $f^*$  is the corresponding function value. This problem arises from the one sided  $L_1$  approximation of  $\tan x$  by a polynomial [12] and the

poor choice of basis functions makes the problem severely ill-conditioned for quite moderate values of  $n$ . When  $n = 6$ , Table 1 shows that the earlier algorithm of [14] terminated after 25 iterations with a directional derivative greater than  $-.00001$ , and with 3 active points for the constraint function  $g$ . The present algorithm required only 20 iterations and found a solution with 4 active points. Because the solutions to the two algorithms gave quite different results, the projected Lagrangian algorithm was rerun using the solution to the  $n=6$  problem, obtained by the algorithm of [14], as the starting point. The new algorithm did not accept this point as a solution but converged once more to the solution given above. Similar remarks apply to the  $n=8$  case but because this problem is so ill-conditioned it is unlikely that the solution given above is accurate to more than 3 or 4 significant figures.

In problem 6, the presence of exponential terms in the constraint function  $g(\underline{a}, \underline{x})$  caused large negative eigenvalues to be present in the projected Lagrangian Hessian in the early iterations. Initially the value  $\mu = 1365$  was required to make  $Z^T \nabla^2 \mathcal{L} Z + \mu I$  positive definite. This value was reduced to 341 on the second iteration and  $\mu = 1$  on the third; thereafter  $\mu = 0$  was acceptable. The detailed progress of the algorithm on this problem is given in Table 2 and clearly demonstrates the second order rate of convergence once the correct number of active points is identified and the projected Lagrangian Hessian becomes positive definite.

The only problem which caused difficulties for the algorithm was problem 9. Here the assumptions of section 1 are not satisfied and this caused the algorithm to fail.

Problem (See [14])	n	k	t	P'	Comments
1	2	16 (16)	1 (1)	-5.7 -6 (-1.1 -5)	$\lambda_1^* = 5461(\rightarrow\infty)$
2	2	7 (7)	1 (1)	-2.5 -10 (-3.4 -7)	
3	3	10 (14)	1 (1)	-6.2 -12 (-6.7 -6)	
4	3	5 (5)	2 (2)	-5.4 -8 (-5.3 -8)	} $\nabla^2 \mathcal{L}^*$ singular but } $[Z^T \nabla^2 \mathcal{L} Z]^*$ positive } definite - n=8 case } very ill-conditioned.
	6	20 (25)	4 (3)	-6.4 -6 (-5.9 -6)	
	8	16 (14)	5 (3)	-7.4 -6 (-9.6 -6)	
5	3	4 (5)	2 (2)	-6.9 -6 (-7.5 -6)	
6	2	9 (8)	1 (1)	-1.1 -8 (-5.3 -6)	
7	3	3 (3)	1 (1)	0.0 (0.0)	
8	6	9 (19)	4 (4)	-1.1 -8 (-7.5 -7)	
9	6	18 (9)	? (1)	-4.8 -2 (-3.5 -3)	$\mu^{(18)} = 256,  E(a^*)  = \infty$ irregularities caused failure.
10	3	3 (3)	1 (1)	-2.8 -7 (-3.9 -9)	
11	3	12 (19)*	2 (2)	-2.2 -7 (-3.0 -8)*	} Values with asterisks } have been corrected } from those appearing in } [14]
12	3	4 (4)	1 (1)	-1.7 -11 (-2.2 -10)	
13	3	4 (4)*	1 (1)	-3.5 -7 (-3.6 -7)*	

Table 1

Summary for all problems

Problem 9

$$X = [-1,1] \times [-1,1]$$

$$f = -4a_1 - \frac{2}{3}(a_4 + a_6)$$

$$g = a_1 + a_2x + a_3y + a_4x^2 + a_5xy + a_6y^2 - 3 - (x^2 - y^2)^2.$$

Starting point  $(5,1,1,1,1,1)^T$ ;  $\underline{a}^* = (3,0,0,0,0,0)^T$ ,  $f^* = -12$ .

This example has an infinite number of points in  $E(\underline{a}^*)$ , corresponding to the line segments of  $y = \pm x$  within  $X$ . Symmetry in the

components of the starting point caused the algorithm to always generate approximations of the form  $\underline{a} = (\alpha, 0, 0, \beta, 0, \beta)^T$ . At such points the

constraint function  $g$  has one maximizing point at the centre of the region  $X$  when  $\beta < 0$  and 4 local maximizing points, corresponding to the

four corners of the region  $X$ , when  $\beta > 0$ . Thus the set  $E(\underline{a})$  has either

one, four or infinitely many members accordingly as  $\beta < 0$ ,  $\beta > 0$  or

$\beta = 0$ . For  $\beta < 0$  the solution to the subproblem 2.1 always gives a

direction which makes  $a_1 = 3$  if  $\gamma = 1$ . If the corner constraints

were added to this subproblem then the exact solution would be obtained

in one iteration but unfortunately this can never be the case (unless

$\beta = 0$  when the added complication of  $|E(\underline{a})| = \infty$  arises). In practice

the algorithms switched between using 1 or 4 constraints and in either

case the projected Lagrangian Hessian matrix is singular, and the

choice of  $\mu$  critically affects the size of the correction at each

iteration. Despite these difficulties the algorithm still made progress

towards the solution and after 18 iterations the approximation

$\underline{a}^{(18)} = (3.002, 0.0, 0.0, -.000079, 0.0, -.000079)^T$  was obtained. An

error was flagged at iteration 19 because the local maximizer of  $g$  had

a numerically singular second derivative matrix, violating assumption 2

of section 1. This was the only problem for which the choice  $\mu = 0$

was not made in the final iterations.

k	p	t	$\mu$	$\theta$	P'	$\gamma$	P
1	2	1	1365	81.65	-1625.2	1.0	3013.8
2	2	1	341	201.49	-2576.8	1.0	1100.3
3	1	0	1	201.49	-356.69	1.0	178.97
4	0	0	0	201.49	-165.44	.125	159.54
5	1	1	0	201.49	-108.70	1.0	99.410
6	1	1	0	201.49	-3.8144	1.0	98.354
7	1	1	0	201.49	-1.1971	1.0	97.161
8	1	1	0	201.49	-0.0026	1.0	97.158
9	1	1	0	201.49	$-1.1 \times 10^{-8}$		

$\tilde{a}^* = (0.719961, -1.450487)^T$ ,  $f^* = 97.158852$

Table 2  
Details for Problem 6

k	p	t	$\mu$	$\theta$	P'	$\gamma$	P
1	0	0	0	1.0	-5.1	1.0	2.15
2	1	1	0	1.05 2.10	+0.05 -0.22	1.0	2.26
3	1	1	0	1.10	-0.055	1.0	2.2013
4	1	1	0	1.10	-0.0013	1.0	2.200000
5	1	1	0	1.10	$-8.2 \times 10^{-7}$		

$\tilde{a}^* = (-.095310, .095310)^T$ ;  $f^* = 2.2$

Table 3  
Details for Problem 14

Finally, we introduce a new problem to demonstrate the effect of a persistent negative eigenvalue in the second derivative matrix of the Lagrangian function (1.4).

Problem 14

$$x = [0, 1]$$

$$f = c^2 e^{a_1} + e^{a_2}$$

$$g = x - e^{a_1 + a_2}$$

Starting point  $(0.8, 0.9)^T$ ;  $a^* = (\ln|c|, \ln|c|)^T$ ,  $f^* = 2|c|$ .

This has essentially the same properties as problem (2.6) but without the need to include extra positivity constraints. The results of applying the algorithm to this problem with  $c = 1.1$  are given in Table 3. Note that on the second iteration the solution to subproblem (2.1) gives a non-descent direction for the penalty function, indicated by the positive directional derivative. However, the matrix  $Z^T[\nabla^2 \mathcal{L}]Z$  is positive definite so that increasing the penalty parameter  $\theta$  temporarily by a factor of two causes the directional derivative to change sign. The steplength has the value  $\gamma = 1$  on every iteration and  $\mu = 0$  throughout even though  $\nabla^2 \mathcal{L}^* = \begin{bmatrix} 0 & -|c| \\ -|c| & 0 \end{bmatrix}$  has one negative eigenvalue. A second order rate of convergence is observed.

## 6. Concluding remarks

The algorithm presented in this paper is capable of the effective solution of a wide class of semi-infinite programming problems. It is globally convergent under mild assumptions on the problem, and typically has a second order convergence rate, with the solution of an equality constrained quadratic programming problem required per iteration. Perhaps the most awkward part of the method is the computation of the set  $E(\underline{a})$ , which is required at least once on each iteration. This is of course not a finite calculation, and must always be a compromise between theory and practice. It should be emphasised, however, that this is an essential calculation with any method which aims to provide an accurate solution to a problem of this semi-infinite type.

It is possible that far from a stationary point, better progress can be made for some problems by incorporating a procedure for finding the solution of a discretization of the original problem. This remains to be seen, but whether as a method in its own right, or as a safe and effective second phase for a method of the two-phase variety, we believe that an algorithm such as the one described in this paper has an important role to play in the numerical treatment of semi-infinite programming problems.

## Acknowledgement

This research was carried out while the second author was a Visiting Erskine Fellow at the University of Canterbury.

References

- [ 1] Clarke, F. H., A new approach to Lagrange multipliers, *Mathematics of Operations Research* 1, 165-174 (1976).
- [ 2] Charalambous, C., A lower bound for the controlling parameter of the exact penalty function, *Math. Programming*, 17, pp 278-290 (1978).
- [ 3] Fiacco, A. V. and K. O. Kortanek (eds), *Semi-infinite programming and applications, Proceedings of an international symposium, Springer-Verlag, Berlin* (1983).
- [ 4] Fletcher, R., *Practical methods of optimization, vol. II. Constrained optimization, John Wiley and Sons, Chichester* (1981).
- [ 5] Gfrerer, H., Guddat, J., Wacker, H., and W. Zulehner, Globalization of locally convergent algorithms for nonlinear optimization problems with constraints, in [3], pp 128-137 (1983).
- [ 6] Gustafson, S. -Å., A three-phase algorithm for semi-infinite programs, in [3], pp 138-157 (1983).
- [ 7] Han, S. P., A globally convergent method for non-linear programming, *J. Opt. Theo. Applns*, 22, pp 297-309 (1977).
- [ 8] Hettich, R. (ed), *Semi-infinite programming. Proceedings of a workshop, Springer-Verlag, Berlin* (1979).
- [ 9] Hettich, R., A review of numerical methods for semi-infinite optimization, in [3], pp 158-178 (1983).
- [ 10] Hettich, R. and W. van Honstede, On quadratically convergent methods for semi-infinite programming, in [8], pp 97-111 (1979).
- [ 11] van Honstede, W., An approximation method for semi-infinite problems, in [8], pp 126-136 (1979).
- [ 12] Roleff, K., A stable multiple exchange algorithm for linear semi-infinite programming, in [8], pp 83-96 (1979).
- [ 13] Watson, G. A., Globally convergent methods for semi-infinite programming, *BIT* 21 pp 362-373 (1981).
- [ 14] Watson, G. A., Numerical experiments with globally convergent methods for semi-infinite programming problems, in [3] pp 193-205 (1983).