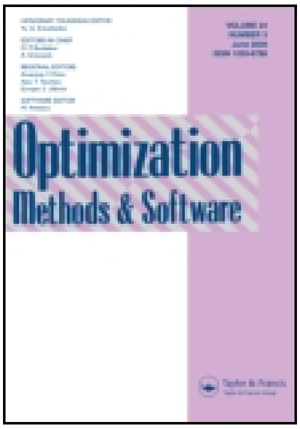

Optimization Methods and Software
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/goms20

An interior point algorithm for continuous minimax: implementation and computation

B. Rustem [a], S. Žaković [a] & P. Parpas [a]
[a] Department of Computing, Imperial College, London, UK

Published online: 13 Oct 2008.

To cite this article: B. Rustem, S. Žaković & P. Parpas (2008) An interior point algorithm for continuous minimax: implementation and computation, Optimization Methods and Software, 23:6, 911-928, DOI: 10.1080/10556780802079891

To link to this article: http://dx.doi.org/10.1080/10556780802079891

Optimization Methods & Software
Vol. 23, No. 6, December 2008, 911–928

An interior point algorithm for continuous minimax: implementation and computation

B. Rustem*, S. Žaković and P. Parpas

Department of Computing, Imperial College, London, UK

(*Received 17 August 2006; final version received 19 March 2008*)

In this paper, we propose an algorithm for the constrained continuous minimax problem. The algorithm, motivation, and numerical experience are reported in this paper. Theoretical properties and the convergence of the proposed method are discussed in a separate paper [B. Rustem, S. Zakovic, and P. Parpas, *Convergence of an interior point algorithm for continuous minimax*, J. Optim. Theory Appl. (2007), in press]. The algorithm uses quasi-Newton search direction, based on sub-gradient information, conditional on maximizers. The initial problem is transformed to an equivalent equality constrained problem, where the logarithmic barrier function is used to ensure feasibility. In the case of multiple maximizers, the algorithm adopts semi-infinite programming iterations towards epi-convergence. Satisfaction of the equality constraints is ensured by an adaptive quadratic penalty function. The algorithm is augmented by a discrete minimax procedure to compute the semi-infinite programming steps and ensure overall progress when required by the adaptive penalty procedure. Progress towards the solution is maintained using merit functions. Computational results are included to illustrate the efficient performance of the algorithm.

Keywords: continuous minimax; interior point algorithms; semi-infinite programming; epi-convergence

1. Formulation of the problem

Consider the following problem:

$$\min_{x} \max_{y \in Y} \{ f(x, y) | g(x) = 0, x \geq 0 \}, \tag{1}$$

where Y is a compact subset of $\mathcal{R}^m$, $x \in \mathcal{R}^n$, $f(x, y)$ is continuous in x and y, twice continuously differentiable in x, and $g\colon \mathcal{R}^n \to \mathcal{R}^k$ is continuous and twice differentiable in x. We also denote the feasible region with $X_f = \{x \in \mathcal{R}^n | g(x) = 0, \quad x \geq 0\}$.

When the maximizer y is defined on a discrete set, Equation (1) becomes a discrete minimax problem and algorithms for solving such problems have been considered by a number of authors, including Womersley and Fletcher [33], Polak [22], Rustem and Nguyen [29], Obasanjo and Rustem [18].

The continuous problem (1) arises in numerous disciplines, including n–person games [27], finance [28], economics [37], and policy optimization [3]. In general, they are used by the decision

*Corresponding author. Email: br@doc.ic.ac.uk

ISSN 1055-6788 print/ISSN 1029-4937 online
© 2008 Taylor & Francis
DOI: 10.1080/10556780802079891
http://www.informaworld.com

maker to assess the worst-case strategy of the opponent and compute the optimal response. The opponent can also be interpreted as nature choosing the worst-case value of the uncertainty, and the solution would be the strategy which ensures the optimal response to the worst-case. Neither the robust decision maker nor the opponent would benefit by deviating unilaterally from this strategy. The solution of Equation (1) can be characterized as a saddle point when f is convex in x and f is concave in y. A survey of algorithms for computing saddle points can be found in [8,28].

We consider an algorithm in which $f(x, y)$ is not necessarily convex in x and concave in y. Previous attempts to solve this problem have mostly focused on unconstrained problems (Kiwiel [14], Demyanov and Pevnyi [8], and Rustem and Howe [28]), with notable exceptions such as Breton and El Achem [5] who use projected sub-gradient and bundle methods. An earlier attempt to solve a similar problem, with *decoupled* constraints (i.e. constraints that contain only either x or y), was made by Sasai [31]. Among recent work is that of Polak *et al.* [23], where a smoothing technique (using exponential penalty functions) is applied to a finite (discrete) minimax problem. This work has been extended in [21] where the smoothing technique is applied to solving finite and semi-infinite min–max–min problems.

The approach in this paper uses an interior point methodology and an adaptive penalty merit function to encourage feasibility and descent. It differs from the above in the general framework as well as the choice of the barrier and merit functions. It uses semi-infinite programming steps that: (i) involve the solution of a discrete minimax problem; (ii) are initiated to complement the interior point algorithm descent procedure and to prevent the penalty parameter increasing indefinitely; and (iii) have an additional role in the epi-convergent procedure to glean information about maximizers and construct an effective algorithm robust to the existence of single or multiple maximizers.

The algorithm is thus intended to combine the desirable aspects of semi-infinite programming with descent steps for the max function based on sub-gradients and a merit function. Semi-infinite programming performs well when the set of maximizers $Y(x_k)$, at some x_k, is stable in that the maximizers do not change at every iteration. Additionally, in this case, multiple maximizers are used to generate sub-gradients. Helpful descent steps can then be performed, provided that sufficient number of maximizers are known, so that the merit function can be reduced further. If $Y(x_k)$ changes radically at every iteration, as in the case when $f(x, y)$ is concave in y and there usually is a different unique maximizer corresponding to every x, a gradient-based descent direction can perform significantly better. As also mentioned in [40], there are clearly computational difficulties regarding a step size strategy to be overcome and these are discussed in Section 5. However, semi-infinite programming steps and a sub-gradient-based approach can complement each other rather well in accumulating maximizer information and in expediting progress through descent directions, respectively.

The semi-infinite programming formulation (1) is given by:

$$\min_{x \in X_f, \tau} \{\tau \mid f(x, y) - \tau \leq 0, \quad \forall y \in Y\}, \tag{2}$$

which has an infinite number of constraints corresponding to the elements in Y. We compare our results with an algorithm based on semi-infinite programming [4,11,36]. Other recent contributions in this area are by Lawrence and Tits [16] and Stein and Still [32] in which sequential quadratic programming and interior point techniques are considered for generalized semi–infinite problems.

One strategy for problems with an infinite number of constraints is through discretization. Under this framework, the original problem is replaced by a problem with a finite number of constrains. There are a number of discretization schemes, exchange methods, or methods based on local reduction [13]. Polak in [20] discusses discretization and finer consistent approximations of the original semi–infinite problem through epi-graphs and epi-convergence that we also introduce later in this paper. Convergence is established through a sequence of the solutions to the discretized problems, assuming that such sequences have an accumulation point. The present paper augments the discretization scheme by a procedure to generate a descent direction and a merit function to stabilize progress towards the solution. This allows a direct convergence result for the algorithm.

Let x^i denote the ith element of vector x. The penalty formulation to ensure feasibility regarding the inequality constraints on the slack variable is realized using a barrier function such as $-\log(x^i)$. The framework for solving the problem is closely related to the interior point literature (e.g. [1,9]). The transformed minimax problem, for $x > 0$, is given by:

$$\min_x \max_{y\in Y} \left\{ f(x,y) - \mu \sum_{i=1}^{n} \log(x^i) | g(x) = 0 \right\}. \tag{3}$$

We define the augmented objective function as:

$$P(x,y;c,\mu) = f(x,y) + \frac{c}{2}\|g(x)\|_2^2 - \mu \sum_{i=1}^{n} \log(x^i), \tag{4}$$

and the merit function Ψ as the maximum of the augmented objective (4):

$$\Psi(x;c,\mu) = \max_{y\in Y} P(x,y;c,\mu). \tag{5}$$

The algorithm discussed below solves Equation (1) and is based on a sequence of optimization problems characterized by the penalty parameter $c \geq 0$ and barrier parameter $\mu \geq 0$. The max function $\Phi(x)$ is defined as follows:

$$\Phi(x) = \max_{y\in Y} f(x,y) \tag{6}$$

and the following set of maximizers at x:

$$\widehat{Y}(x) = \{y \in Y | f(x,y) = \Phi(x)\}.$$

Given the maximizers $y \in \widehat{Y}(x)$ and using the fact that the maximum over a set of scalars is equal to the maximum over their convex combination, Equation (3) can be equivalently expressed as:

$$\min_x \max_{\beta\in\mathcal{B}} \left\{ \sum_{y\in\widehat{Y}(x)} \beta^y f(x,y) - \mu \sum_{i=1}^{n} \log(x^i) | g(x) = 0 \right\},$$

where β^y is the element of β corresponding to maximizer y and

$$\mathcal{B} = \{\beta | \sum_y \beta^y = 1, \quad \beta^y \geq 0\}. \tag{7}$$

A similar reasoning can be applied to the merit function $\Psi(x;c,\mu)$:

$$\Psi(x;c,\mu) = \sum_{y\in\widehat{Y}(x)} \beta^y P(x,y;c,\mu).$$

We summarize the three definitions related to the objective function:

$$\Phi(x) = \max_{y\in Y} f(x,y),$$

$$P(x,y;c,\mu) = f(x,y) + \frac{c}{2}\|g(x)\|_2^2 - \mu \sum_{i=1}^{n} \log(x^i),$$

$$\Psi(x;c,\mu) = \max_{y\in Y} P(x,y;c,\mu).$$

2. Sub–gradient of $\Phi(x)$ and basic iteration

Problem (6) poses several difficulties: firstly, $\Phi(x)$ is in general continuous but may have kinks, so it might not be differentiable. At a kink, the maximizer is not unique and the choice of sub-gradient to generate a search direction is not simple. Secondly, $\Phi(x)$ may not be computed accurately as it would require infinitely many iterations of an algorithm to maximize $f(x, y)$. Finally, in Equation (5) global maxima are required in view of possible multiple solutions. The use of local, or insufficient number of global, maxima cannot ensure a monotonic decrease in $\Phi(x)$.

The sub-differential of the max function $\Phi(x)$ can be expressed as:

$$\partial\Phi(x) = \operatorname{conv}\{\nabla_x f(x, y) | y \in \widehat{Y}(x)\}.$$

Similarly, the sub-differential of the merit function:

$$\partial\Psi(x; c, \mu) = \operatorname{conv}\{\nabla_x P(x, y; c, \mu) | y \in \widehat{Y}(x)\}.$$

For non–unique maximizers $y \in \widehat{Y}(x)$, by Caratheodory's theorem [25]: for every vector $\nabla\Psi(x; c, \mu) \in \partial\Psi(x; c, \mu)$, there exist $n + 1$ points $y_j \in \widehat{Y}(x)$, $j = 1, \dots, n + 1$, and multipliers β_j, $j = 1, \dots, n + 1$, with $\sum_j^{n+1} \beta_j = 1$, $\beta_j \geq 0$, such that:

$$\nabla\Psi(x; c, \mu) = \sum_{y\in\widehat{Y}(x), \beta\in\mathcal{B}} \beta^y \nabla_x f(x, y) + \frac{c}{2}\nabla g(x)^t g(x) - \mu X^{-1} e, \tag{8}$$

where X^{-1} is the diagonal matrix, defined as $X^{-1} = \operatorname{diag}(1/x^1, 1/x^2, \dots, 1/x^n)$. Similarly, we have:

$$\nabla\Phi(x) = \max_{\beta\in\mathcal{B}} \sum_{y\in\widehat{Y}(x), \beta\in\mathcal{B}} \beta^y \nabla_x f(x, y).$$

The determination of β^y is discussed in the next section. A problem which the set $\widehat{Y}(x)$ poses is that it can be of infinite cardinality. For example, the problem

$$\min_{x\in R} \max_{-2\leq y\leq 2} (x - 1)^2 y$$

has an infinite number of solutions, as every pair $(x^*, y^*) = (1, y)$ for $y \in \widehat{Y} = \{y | -2 \leq y \leq 2\}$ is a solution. To avoid such problems, we use the discretization described below.

Instead of using the whole set of maximizers at x, $\widehat{Y}(x)$, starting with $y_0 \in \arg\max_{y\in Y} f(x_0, y)$, we define the sets $Y_i \subset Y$ so that:

$$Y_0 = \{y_0\};\ y_i \in \arg\max_{y\in Y} f(x_i, y);\ Y_i = Y_{i-1} \cup \{y_i\}, \quad i = 1, 2, \dots$$

Now, it is possible to define a finite set of maximizers at the current point $x_k : Y(x_k) = \{y \in Y_k | f(x_k, y) = \Phi(x_k)\}$. Hence, we note that $Y(x_k) \subset Y_k$. Table 1 summarizes the various maximizer concepts introduced in this paper. The cardinality of $Y(x_k)$ is denoted by $|Y(x_k)|$. For any compact set $Y_i \subset Y$ let $\Phi_{Y_i}(x) \equiv \max_{y\in Y_i} f(x, y)$. Then for any $x \in \mathcal{R}^n$ we have

Table 1. Maximizer concepts.

Set	Definition
$\widehat{Y}(x)$	$\hat{Y}(x) = \{y \in Y \| f(x, y) = \Phi(x)\}$
Y_k	Set of potential maximizers accumulated from x_0 to x_k
$Y(x_k)$	$Y(x_k) = \{y \in Y_k \| f(x_k, y) = \Phi(x_k)\}$

$\Phi_{Y_i}(x) \le \Phi(x) = \Phi_Y(x)$. The original problem (1) is approximated with a finite number of maximizers:

$$\min_{x \in X_f} \Phi_{Y_i}(x), \quad i = 0, 1, 2, \dots \tag{9}$$

To clarify the relationship between the original problem (1) and the approximation (9), we restate these using the following epi-graphs:

$$\begin{aligned} \mathcal{E} &= \{(x^0, x) | x \in X_f, \quad x^0 \ge \Phi(x)\}, \\ \mathcal{E}_i &= \{(x^0, x) | x \in X_f, \quad x^0 \ge \Phi_{Y_i}(x)\}. \end{aligned} \tag{10}$$

Given the epi-graphs $\mathcal{E}$ and $\mathcal{E}_i$ in Equation (10), the problems (1) and (9) can be reformulated as follows:

$$\min_{(x^0,x) \in \mathcal{E}} x^0, \tag{11}$$

$$\min_{(x^0,x) \in \mathcal{E}_i} x^0. \tag{12}$$

It can be seen, from Equations (11) and (12), that the two problems differ only in the constraint set. For the sequence of problems (12) to approximate (11) well, $\mathcal{E}_i$ must converge to $\mathcal{E}$.

Definition *The sequence of approximating problems* (12) *epi-converges to the problem* (11) *if the epi-graphs $\mathcal{E}_i$ converge to the epigraph $\mathcal{E}$ in the following sense*

$$\underline{\lim}\mathcal{E}_i = \overline{\lim}\mathcal{E}_i = \mathcal{E}.$$

Clearly, approximate problems in optimization are necessary in order to facilitate computation. The practical importance of epi-convergence stems from the fact that an approximation that epi-converges to the original problem will also (under some conditions) have the same minima. Epi-convergence and its use in algorithm design is discussed in [20] and [26]. In the algorithm described in this paper, epi-graphs $\mathcal{E}_i$ are generated by:

$$\begin{aligned} &x_{i+1} \in \arg\min_{x \in X_f} \Phi_{Y_i}(x); \; y_{i+1} \in \arg\max_{y \in Y} f(x_{i+1}, y), \\ &Y_{i+1} = Y_i \bigcup \{y_{i+1}\}; \; \mathcal{E}_{i+1} = \{(x^0, x) | x \in X_f, \quad x^0 \ge \Phi_{Y_{i+1}}(x)\}. \end{aligned}$$

The difficulties associated with $\Phi(x)$ and possible large numbers of maximizers are addressed through a number of methods based on the characterization of maximizers of $f(x, y)$. Another related difficulty is the potentially costly evaluation of $\Phi(x)$ during line search, required to ensure global convergence. A class of algorithms, including Demyanov and Malozemov [7], Hager and Presler [12] and Zhou and Tits [40] are based on approximating $\Phi(x)$ by progressively increasing the number of discretization points. The algorithm presented in this paper invokes Caratheodory's theorem [25], to characterize the search direction with a small number of maximizers that would ensure convergence. Rustem and Howe [28] consider an algorithm that defines a direction of progress for $\Phi(x)$ based on the maximizers at x corresponding to the minimum norm sub-gradient of $\nabla_x f(x, \cdot)$ and a Hessian approximation. The algorithm extends the first order approach of Panin [19] and Kiwiel [14] to quasi-Newton descent directions, conditional on the maximizer, and also attempts to deal with the problem of multiple maximizers. The algorithm presented in [28], is in some respects similar to the algorithm presented here. In [28] it assumed that sufficient maximizers are available to compute a descent direction. In this paper, we complement this approach with semi-infinite steps when a descent direction cannot be easily computed.

The Lagrangian associated with Equation (3) is

$$L(x, y, \lambda; \mu) = \max_{y \in Y}\{f(x, y) - \mu \sum_{i=1}^{n} \log(x^i) - \lambda^t g(x)\}.$$

Consider the diagonal matrix $X = \mathrm{diag}(x_1, x_2, \ldots, x_n)$ and the column vector $e = [1, 1, \ldots, 1]^t \in R^n$. The first-order optimality condition for the problem (3) is given by:

$$\nabla\Phi(x) - \mu X^{-1}e - \nabla g^t(x)\lambda = 0; \quad g(x) = 0,$$

where $\nabla_x \Phi(x)$ is a sub-gradient of $\Phi(x)$ and its evaluation is considered in the next section. Optimality conditions for this class of problems are discussed in [28]. Invoking the nonlinear transformation $z = \mu X^{-1}e$ yields:

$$F = \left[(\nabla\Phi(x))^{\mathrm{T}}, (g(x))^{\mathrm{T}}, (XZe - \mu e)^{\mathrm{T}}\right]^{\mathrm{T}} = 0, \tag{13}$$

where $F = F(x, \lambda, z; \mu)$ and Z is the diagonal matrix $Z = \mathrm{diag}(z_1, z_2, \ldots, z_k)$. These equations represent the perturbed optimality conditions for problem (1). We note that $F = F(x, \lambda, z; 0)$, $x \geq 0$, $z \geq 0$ represent the optimality conditions. Returning to the original problem with equality and non-negativity constraints, Equation (1), we define the Lagrangian:

$$L_{ec}(x, z, \lambda) = \Phi(x) - z^t x - \lambda^t g(x), \tag{14}$$

where $z \in \{v \in R^n | v \geq 0\}$ is the multiplier vector corresponding to $x \geq 0$. The Lagrangian (14) is defined over a set of maximizers. Its sub-gradient is defined as the convex combination of the gradients corresponding to the set of maximizers. A consistent Hessian definition, mentioned below, is based on the convex combination of the Hessians corresponding to the maximizers[1].

The primal-dual interior point algorithm is essentially the Newton or quasi-Newton method for solving approximately the perturbed conditions (13) for a fixed value of μ. The Newton system for Equation (13) above is

$$\begin{bmatrix} H_k & -\nabla g_k^t & -I \\ \nabla g_k & 0 & 0 \\ Z_k & 0 & X_k \end{bmatrix} \begin{bmatrix} \Delta x_k \\ \Delta\lambda_k \\ \Delta z_k \end{bmatrix} = -\begin{bmatrix} \nabla\Phi(x_k) - z_k - \nabla g_k^t \lambda_k \\ g_k \\ X_k Z_k e - \mu e \end{bmatrix}, \tag{15}$$

where H_k is a positive definite approximation of the Hessian of the Lagrangian (14). The Hessian H_k is approximated using the updating formula suggested by Powell [24]. Using matrix-vector notation, Equation (15) can be expressed as $\nabla F_k \Delta w_k = -F_k$ where $w_k = (x_k, \lambda_k, z_k)$, $\Delta w_k = (\Delta x_k, \Delta\lambda_k, \Delta z_k)$ and $F_k = F(x_k, \lambda_k, z_k; \mu)$. The solution of Equation (15) is given by:

$$\begin{aligned} \Delta x_k &= \Omega_k^{-1}\nabla g_k^t \Delta\lambda_k - \Omega_k^{-1} h_k \\ \Delta\lambda_k &= -[\nabla g_k \Omega_k^{-1} \nabla g_k^t]^{-1}(g_k - \nabla g_k \Omega_k^{-1} h_k) \\ \Delta z_k &= -z_k + \mu X_k^{-1} e - X_k^{-1} Z_k \Delta x_k, \end{aligned} \tag{16}$$

where $\Omega_k = H_k + X_k^{-1} Z_k$ and $h_k = \nabla\Phi(x_k) - \mu X_k^{-1} e - \nabla g_k^t \lambda_k$. Let us introduce two new matrices M_k and $\mathcal{P}_k$ given by: $M_k = \nabla g_k \Omega_k^{-1} \nabla g_k^t$, $\mathcal{P}_k = (I - \Omega_k^{-1} \nabla g_k^t M_k^{-1} \nabla g_k)$. Then the first two equations of the system (16) can be written as:

$$\begin{aligned}\Delta x_k &= -\mathcal{P}_k \Omega_k^{-1}(\nabla \Phi(x_k) - \mu X_k^{-1} e) - \Omega_k^{-1} \nabla g_k^t M_k^{-1} g_k \\ \Delta \lambda_k &= -M_k^{-1}(g_k - \nabla g_k \Omega_k^{-1} h_k).\end{aligned} \tag{17}$$

Starting from an initial point w_0, the algorithm generates a sequence $\{w_k\}$:

$$w_{k+1} = w_k + \alpha_k \Delta w_k, \quad k = 0, 1, 2, \ldots$$

In order to maintain feasibility of w_{k+1}, the algorithm needs to ensure $x_{k+1}, z_{k+1} > 0$. In [38,39] a single step size is used for all variables. In [35], different step size strategies are suggested. The algorithm in this paper uses a merit function with an adaptive penalty parameter and $\alpha_k = (\alpha_{x_k}, \alpha_{z_k}, \alpha_{z_k})$, where α_{x_k} and α_{z_k} are different step sizes for the primal x variables and dual pair λ, z, respectively.

The algorithm generates a descent direction based on a sub-gradient of $\Phi(x)$ and an approximate Hessian, in the presence of possible multiple maximizers of Equation (5) or Equation (6). It uses a switching scheme between a continuous minimax-based interior point algorithm incorporating a minimum-norm sub-gradient and a discrete minimax formulation appropriately incorporating epi-graphs to determine potential multiple maximizers. This is discussed further in Section 4 (see Equation (25)). Each approach uses a different step size strategy to ensure sufficient progress.

The overall iterative process is in two stages: first, Equation (3) is solved for μ fixed. This is the inner iteration. Once Equation (3) is solved, μ is reduced, convergence criteria is checked (outer iteration) and, if necessary, another inner iteration is performed.

3. Computing sub-gradients of $\Phi(x)$

$\Psi(x; c, \mu)$ is in general continuous but may have kinks, so it might not be differentiable. At a kink, the maximizer is not unique and the choice of sub-gradient to generate a search direction is not simple.The quasi-Newton direction in this paper is based on the combination of the gradients corresponding to the multiple maximizers that ensures descent. The quadratic approximation to $P(x, y; c, \mu)$ in Equation (4) at x_k is given by:

$$P_k(\Delta x_k, y; c, \mu) = P(x_k, y; c, \mu) + \Delta x^t \nabla_x P(x_k, y; c, \mu) + \frac{1}{2}\|\Delta x_k\|_{H_k}^2. \tag{18}$$

We consider two approaches for determining β. The first is motivated by the selection of the worst-case descent direction based on a combination of possible maximizers $y \in Y(x_k)$. In the presence of multiple maximizers, the new direction $\Delta w_k(\beta_k)$ is computed such that: $\Delta w_k(\beta_k) = -(\nabla F_k^{\beta_k})^{-1} F_k^{\beta_k}$ where:

$$\beta_k = \arg\max_{\beta \in \mathcal{B}} \left\{ \Delta x_k(\beta)^t \left(\sum_{y \in Y(x_k)} \beta^y \nabla_x f(x_k, y) + c_k \nabla g_k^t g_k - \mu X_k^{-1} e \right) + \frac{1}{2}\|\Delta x_k(\beta)\|_{H_k}^2 \right\}, \tag{19}$$

which is a concave maximization problem in β. The direction $\Delta x_k(\beta_k)$, which is conditional on the given β_k, is given by:

$$\Delta x_k(\beta_k) = -\mathcal{P}_k \Omega_k^{-1} \left(\sum_{y \in Y(x_k)} \beta_k^y \nabla_x f(x_k, y) - \mu X_k^{-1} e \right) - \Omega_k^{-1} \nabla_x g_k M_k^{-1} g_k, \tag{20}$$

where F_k^{β} is defined as

$$F_k^{\beta_k} = \left[\sum_{y \in Y(x_k)} \beta_k^y \nabla_x f(x_k, y) - z_k - \nabla_x g_k^t \lambda_k, g_k, X_k Z_k e - \mu e \right]^{\mathrm{T}}. \tag{21}$$

The solution to problem (19) is unique when the vectors: $\nabla_x f(x_k, y)$, $y \in Y(x_k)$ are linearly independent. Otherwise, a minimum norm β_k is determined.

The second approach to determining β is given by:

$$\beta_k^y = 1 \text{ for some } y = y_{k+1} \in Y(x_k)$$
$$\beta_k^y = 0, \quad \forall y \neq y_{k+1}, \quad y \in Y(x_k).$$

Thus, $\Delta x_k(1)$ corresponds to one element of β_k being unity and the rest null. The choice of the corresponding maximizer is:

$$\tilde{y}_{k+1} \in \arg \max_{y \in Y(x_k)} \{\Delta x_k(1)^t (\nabla f(x_k, y) + c \nabla g_k^t g_k - \mu X_k^{-1} e) \tag{22}$$
$$+ \frac{1}{2} \|\Delta x_k(1)\|_{H_k}^2\}.$$

The motivation for this choice is the selection of the worst-case descent direction among the maximizers. Hence, for such $\tilde{y}_{k+1}$ we have $F_k^{\beta=1}$:

$$F_k^{\beta=1} = \left[\nabla_x f(x_k, \tilde{y}_{k+1}) - z_k - \nabla_x g_k^t \lambda_k, g_k, X_k Z_k e - \mu e\right]^{\mathrm{T}}.$$

Consider, therefore, two possible directions $\Delta w(1)$ and $\Delta w(\beta_k)$, depending on the sub-gradient $\nabla \Phi(x)$ used. The direction $\Delta w(1)$ is easier to compute, as it does not entail the solution of the quadratic programming problem (19). We define two characterizations of $\nabla P(x_k, ., c_k, \mu)$. One corresponds to the maximizer y_{k+1} responsible for the worst-case descent direction and the other corresponds to multiple maximizers, $y \in Y(x_k)$. These are given by:

$$\nabla P(x_k, y_{k+1}; c_k, \mu) = \nabla_x f(x_k, y_{k+1}) + \frac{c_k}{2} \nabla g(x_k)^t g(x_k) - \mu X_k^{-1} e$$
$$\sum_{y \in Y(x_k)} \beta_k^y \nabla P(x_k, y; c_k, \mu) = \sum_{y \in Y(x_k)} \beta_k^y \nabla_x f(x_k, y) + \frac{c_k}{2} \nabla g(x_k)^t g(x_k) - \mu X_k^{-1} e.$$

These two characterizations of $\nabla P(x_k, ., c_k, \mu)$ lead to two possible sub–gradient choices for $\nabla \Phi(x_k)$ and $\nabla \Psi(x_k; c_k, \mu)$. These are summarized in Table 2 together with the corresponding Δw_k. In the rest of this paper, we ignore the argument β and use Δw_k, except when distinguishing between $\Delta w_k(1)$ and $\Delta w_k(\beta_k)$. The new maximizer y_{k+1} is chosen as a maximizer of the following

Table 2. Choices for $\nabla_x \Phi(x_k)$, $\nabla_x \Psi(x_k; c_k, \mu)$ with $y \in Y(x_k)$.

$\nabla_x \Phi(x_k)$	Δw_k	$\nabla_x \Psi(x_k; c_k, \mu)$
$\nabla_x f(x_k, y_{k+1})$	$\Delta w_k(1) = -(\nabla F_k^{\beta_k=1})^{-1} F_k^{\beta_k=1}$	$\nabla P(x_k, y_{k+1}; c_k, \mu)$
$\sum_y \beta_k^y \nabla_x f(x_k, y)$	$\Delta w_k(\beta_k) = -(\nabla F_k^{\beta_k})^{-1} F_k^{\beta_k}$	$\sum_y \beta_k^y \nabla P(x_k, y; c_k, \mu)$

augmented quadratic approximation to $\Psi(x; c, \mu)$:

$$y_{k+1} \in \arg\max_{y\in Y}\{P_k(\Delta x_k(1), y; c, \mu) - \mathcal{C}[\Phi(x_k) - f(x_k, y)]^2\}.$$

The problem above can be interpreted as follows: among all the maximizers we wish to choose the one that is also a maximizer $Y(x_k)$. We formulate this problem as a penalty function to enforce the choice of y to a maximizer at x_k. If $\mathcal{C} \geq 0$ is large enough, then we can ensure that $\Phi - f = 0$.

4. Primal-dual step size selection

To determine the new iterate x_{k+1}, we adopt Armijo's rule [2] to ensure sufficient decrease in the merit function. The maximum allowable step size is determined by the boundary of the feasible region. The pseudo-code for choosing the sub-gradient of $\Phi(x)$, the penalty parameter c and the merit function is presented in Algorithm 3.

Step size determination for the dual variables z uses the information provided by the new primal iterate x_{k+1}, in order to find the new iterate z_{k+1}. This is a modification of the strategy suggested by Yamashita [34] Yamashita and Yabe [35], and Akrotirianakis and Rustem [1].

Although the barrier parameter μ is fixed, we determine a step α^i_{zk} along the direction Δz^i_k, for each dual variable z^i_k, $i = 1, 2, \dots, n$, such that the box constraints are satisfied.

$$\alpha^i_{z_k} = \max\{\alpha > 0\colon \mathrm{LB}^i_k \leq (x^i_k + \alpha_{x_k}\Delta x^i_k)(z^i_k + \alpha\Delta z^i_k) \leq \mathrm{UB}^i_k\}. \tag{23}$$

The lower bounds LB^i_k and upper bounds UB^i_k, $i = 1, 2, \dots, n$ are defined as:

$$\mathrm{LB}^i_k = \min\{\frac{1}{2}m\mu, (x^i_k + \alpha_{x_k}\Delta x^i_k)z^i_k\}, \mathrm{UB}^i_k = \max\{2M\mu, (x^i_k + \alpha_{x_k}\Delta x^i_k)z^i_k\},$$

where the parameters m and M are chosen such that

$$0 < m \leq \min\left\{1, \frac{(1-\gamma)(1-(\gamma/(M_0)^{\mu}))\chi_l}{\mu}\right\} \text{ and } M \geq \max\left\{1, \frac{\hat{\chi}_l}{\mu}\right\} > 0, \tag{24}$$

where χ_l and $\hat{\chi}_l$ are set in Algorithm 3, $\gamma \in (0, 1)$, and M_0 is a positive large number. These two parameters are always fixed to constants which satisfy Equation (24) while μ is fixed. The values of m and M change when the barrier parameter μ is decreased.

The common dual step length α_{z_k} is the minimum of all individual step lengths $\alpha^i_{z_k}$ with the restriction of being always not more than one. The step size for the dual variables y can be either $\alpha_{y_k} = 1$ or $\alpha_{y_k} = \alpha_{z_k}$.

Merit function (5) penalizes infeasibility and measures progress towards the optimal solution. There are, however, two issues regarding (5). Firstly, penalty parameter c might grow unacceptably, potentially leading to numerical instability. Secondly, the algorithm may not have the required number of maximizers to generate a descent direction.

The two problems are detected through a single observation based on the descent property of the direction $\Delta x(\beta)$. If $\Delta x(\beta)$ cannot ensure descent it is necessary to either increase c or generate further maximizers. The latter is achieved by performing individual steps of semi-infinite programming. This requires the solution of the following discrete minimax problem, at the kth iteration, defined as:

$$\min_{x\in X_f}\ \max_{y\in Y(x_k)}\{f(x, y)\}. \tag{25}$$

5. Choice of *c* and μ

The set $\{(x_*(\mu), \lambda_*(\mu), z_*(\mu)) : \mu > 0\}$, which contains solutions of the perturbed optimality conditions (13) for different values of the barrier parameter, is called the *central path*. Since the perturbed optimality conditions approximate the optimality conditions of the original problem as $\mu \to 0$, the points in the central path converge to a solution of the initial problem. The central path can thus be thought of as the route, followed by primal-dual interior point algorithms, towards the solution of a nonlinear optimization problem. The distance of the current point from the central path is measured by the Euclidean norm of the perturbed optimality conditions.

The barrier parameter μ plays an important role in interior point methods. By decreasing the parameter to zero, the algorithm converges to the optimal solution. Again, there are a number of different strategies for reducing the barrier parameter [9,10]. We adopt the strategy originally developed for nonlinear programming in [1,10,15]. This barrier selection rule has performed very effectively in [1], and is presented in Algorithm 1.

ALGORITHM 1 *Barrier parameter update rule*

```
Step 0: Initialize $x = x_k, \lambda = \lambda_k, z = z_k, c = c_k, \mu = \mu^l, r_1 = 0.85$
Step 1: If $\|F(x, \lambda, z; c, \mu)\|_2 > 0.1\eta\mu^l$ go to Step 4
Step 2: If $\mu^l \leq 10^{-4}\mu^{l+1} = \min\{r_1\mu^l, 0.01r_1^{k+2\sigma}\|F(x, \lambda, z; c, 0)\|_2\}$ go to Step 5
Step 3: $\mu^{l+1} = \min\{r_1\mu^l, 0.01r_1^{k+\sigma}\|F(x, \lambda, z; c, 0)\|_2\}$ go to Step 5
Step 4: $\mu^{l+1} = \min\{0.95\mu^l, 0.01(0.95)^k\|F(x, \lambda, z; c, 0)\|_2\}$
Step 5: Return (to Algorithm 3)
```

It has been shown in [9,10] that the direction Δw_k defined by Equation (16) is a descent direction for $\|F(x, \lambda, z; \mu)\|_2^2$ as long as the matrix ∇F_k is non-singular.

The penalty parameter *c* plays an important role in the descent of the merit function $\Psi(x; c, \mu)$. At each iteration, its value is determined such that a descent property is ensured. This is simple when all the maximizers are known at the current point, since Caratheodory's theorem can be used to select an appropriate sub-set of maximizers to ensure descent. If sufficient number of maximizers are not known, an epi-convergent procedure is initiated with individual steps of semi-infinite programming adding new maximizers.

The sub-gradient of $\Psi(x_k; c, \mu)$ at the *k*th iteration is:

$$\nabla\Psi(x_k; c_k, \mu) = \nabla\Phi(x_k) + c_k\nabla g_k^t g_k - \mu X_k^{-1}e.$$

The direction Δx_k is a descent direction for Ψ, at the current point x_k, if

$$(\nabla\Phi(x_k) + c_k\nabla g_k^t g_k - \mu X_k^{-1}e, \Delta x_k) + \frac{1}{2}\|\Delta x_k\|_{H_k}^2 \leq 0. \tag{26}$$

By considering the second equation of the Newton system (15), the directional derivative $\Delta x_k^t\nabla\Psi(x_k; c_k, \mu)$ can be written as

$$\Delta x_k^t\nabla\Psi(x_k; c_k, \mu) = \Delta x_k^t\nabla\Phi(x_k) - c_k\|g_k\|^2 - \mu\Delta x_k^t X_k^{-1}e, \tag{27}$$

where c_k is the value of the penalty parameter at the beginning of the *k*th iteration. Since the barrier parameter μ is fixed throughout the inner iterations, we can deduce from Equation (27) that the sign of $\Delta x_k^t\nabla\Psi(x_k; c_k, \mu)$ depends on the value of the penalty parameter. If c_k is not large enough then the descent property (26) may not be satisfied. Thus, a new value $c_{k+1} > c_k$ must be determined to guarantee the satisfaction of the descent property.

The descent of the merit function depends not only on the value of c but also on the maximizers $y \in Y(x)$. In general, the presence of multiple maximizers requires for the sub-gradient of the max function to be computed as in Equation (19), in order to ensure the descent. In [28], this is simplified to a direction-dependent only on one y_{k+1} as long as it ensures descent. In this paper, when only one maximizer is known, descent is assured with an appropriate choice of c. In the case of multiple potential maximizers, the direction based on one maximizer only is used as long as it ensures descent without an increase in c. This avoids unnecessarily computation of β in Equation (19). When a single maximizer cannot ensure descent, problem (19) is solved and a new sub-gradient, which depends on all known maximizers, is computed. If this new direction is still not descent, the value of c is increased. Thus, the direction Δw_k is given by:

$$\Delta w_k = \begin{cases} -(\nabla F_k^{\beta=1})^{-1} F_k^{\beta=1} & \text{if } n_{\max} = 1 \text{ or } \Delta x_k(1)^t \nabla \Psi(x_k, y_{k+1}; c_k, \mu) \le 0 \\ -(\nabla F_k^{\beta})^{-1} F_k^{\beta}, & \text{otherwise.} \end{cases}$$

We note that $\Delta x_k(1)$ corresponds to the direction for $\beta = 1$. When this direction does not lead to descent and there are other known potential maximizers, the algorithm proceeds along the direction $-(\nabla F_k^{\beta})^{-1} F_k^{\beta}$.

The choice for c and $\nabla \Phi(x)$ is shown in Algorithm 2. The merit function is based on Equation (5) and ensures progress towards a minimum. We note that the algorithm may exhibit the Maratos effect [17] with this merit function. Checks on the growth of c are incorporated to address this problem.

Let $\epsilon_g > 0$ be a finite precision to which the equality constraints are satisfied. Thus, for merit function (5), we have a worst-case feasibility precision:

$$\|g(x)\|^2 > \epsilon_g. \tag{28}$$

Assume that at some inner iteration k, $0 < \|g(x_k)\|^2 \le \epsilon_g$ and the descent condition (26) are not satisfied. In this case, Equation (26) is initiated as an epi-convergent procedure to generate further maximizers. This is the semi-infinite programming step discussed in Step 2 of Algorithm 2. If a further maximizer is thereby identified, Algorithm 2 returns to minimizing merit function (5). As mentioned earlier, this is a variation of the 'watchdog' technique [6]. In the context of interior point methods, it was also used by Gay et al. in [10].

In Step 2 of Algorithm 2, when Δx_k is not a descent direction for the merit function (5) and $0 < \|g_k\|_2^2 < \epsilon_g$, then x_{k+1} and v are given by the solution of Equation (25). The value v is used in Step 2(a) of Algorithm 2. At x_{k+1}, the new maximizer is computed as:

$$\hat{y}_{k+1} = \arg\max_{y \in Y} f(x_{k+1}, y). \tag{29}$$

The algorithm terminates if

$$f(x_{k+1}, \hat{y}_{k+1}) \le v, \tag{30}$$

otherwise the new maximizer is added to the set of maximizers

$$\mathrm{Y}_{k+1} = Y_k \cup \hat{y}_{k+1} \tag{31}$$

and a new iteration is performed. Where v is the optimal value of the problem given by Equation (25).

The descent condition (26) for merit function (5) may not always be satisfied for the following reasons: firstly, the direction $-(\nabla F_k^{\beta=1})^{-1} F_k^{\beta=1}$ is computed using only one maximizer, but there are other known potential maximizers. This is addressed by computing the direction

$-(\nabla F_k^\beta)^{-1} F_k^\beta$ and restarting the iteration without altering the penalty parameter c. Secondly, the direction $-(\nabla F_k^\beta)^{-1} F_k^\beta$ and the corresponding $\Delta x_k(\beta)$ are justified in view of the multiple maximizers employed but the penalty parameter c is not sufficiently large to ensure descent. An increase in c is required. Thirdly, insufficient number of maximizers are included in the computation of the direction $-(\nabla F_k^\beta)^{-1} F_k^\beta$. A semi-infinite programming step is employed in Step 2(a), by solving a discrete minimax problem. The corresponding maximizer is added to the potential maximizer set Y_{k+1} and the new iteration is started without increasing c. Even when a new maximizer has just been added to Y_{k+1}, there is a possibility that the number of known maximizers are not sufficient to generate a descent direction. To address this possibility, the algorithm searches for a step size that ensures satisfaction of the Armijo inequality in Step 2c until a relative tolerance is reached. Step sizes less than this value are disallowed and the algorithm explores the possibility of further maxima using semi-infinite programming steps. The algorithm may alternate between semi-infinite programming and descent steps. However, as the set of potential maximizers Y_k grows, the epi-graph of the max function $\Psi(x_k; c_*, \mu)$ is characterized with increasing accuracy and the maximizers generate descent steps. At that stage, Theorem 1 in [30], ensures monotonic decrease of $\Psi(x_{k+1}; c_*, \mu)$.

Semi-infinite programming ensures that:

$$\min_x \max_{y \in Y(x_k)} \{P(x, y; c, \mu) | g(x) = 0\} \leq \max_{y \in Y} \{P(x_k, y; c, \mu) | g(x) = 0\} \equiv \Psi(x_k; c_*, \mu).$$

Given $x_{k+1}, v, \hat{y}_{k+1}$ computed using Equation (29), if Equation (30) is satisfied, then x_{k+1} is the solution of the inner iteration (Algorithm 2), otherwise, using Equation (31), the algorithm proceeds further with the multiple maximizers in Y_{k+1}. In such cases, convergence is due to compactness with semi-infinite programming aided further by the decrease in $\Psi(x_k; c_*, \mu)$ ensured by subsequent accumulation of sufficient number of maximizers. In cases where sufficient number of maximizers are attained $\forall k \geq k_*$, for some k_*, the monotonic decrease of the sequence $\{\Psi(x_k; c_*, \mu)\}$ is established in Theorem 1 in the second part of this paper.

A detailed description of the algorithm for solving the inner iteration problem (the barrier parameter μ is fixed) is given below.

Algorithm 2 *Inner iteration*

Step 0: Initialization set $\beta = 1$, $\Delta x_k(1)$ is used when there is a single maximizer or descent is assured even in the presence of multiple maximizers.

Step 1(a): If $\|F(x_k, y_k, \lambda_k, z_k; \mu)\|_2 \leq \eta\mu$ exit to Algorithm 3. Compute the descent direction if $\beta = 1$ then

$$y_{k+1} = \arg \max_{y \in Y(x_k)} \left\{ \Delta x_k^t(1) \nabla_x P(x_k, y; c_k, \mu) + \frac{1}{2} \|\Delta x_k(1)\|_{H_k}^2 \right\}$$

$\Delta x_k = \Delta x_k(1)$ and $\nabla \Phi(x_k) = \nabla_x f(x_k, y_{k+1})$ go to *Step 1(c)*

Step 1(b): If a single maximizer is not sufficient for progress, compute $\Delta x_k(\beta)$

$$\beta_k = \arg \max_{\beta \in \mathcal{B}} \left\{ \Delta x_k(\beta)^t \sum_{y \in Y(x_k)} \beta^y \nabla_x P(x_k, y; c_k, \mu) + \frac{1}{2} \|\Delta x_k(\beta)\|_{H_k}^2 \right\}$$

This implies the values $\nabla \Phi(x_k) = \beta_k^t \nabla_x f(x_k, y)$ and $\Delta \boldsymbol{x}_k = \Delta\, \boldsymbol{x}_k(\beta_k)$ which are actually computed in *Step 1(c)*

Step 1(c): Interior point step

$$\Delta w_k(\beta_k) = -(\nabla F_k^{\beta_k})^{-1} F_k^{\beta_k}, \quad \alpha_{x_k}^{\max} = \min_{1\le i\le n}\left\{\frac{-x_k^i}{\Delta x_k^i}:\Delta x_k^i < 0\right\} \quad \hat{\alpha}_{x_k} = \min\{\gamma\alpha_{x_k}^{\max}, 1\}.$$

Step 2(a)(i): Test for descent of the merit function $\mathcal{M}_{\text{num}} = \Delta x_k^t \nabla\Phi_k - c_k\|g_k\|_2^2 - \mu^l \Delta x_k^t X_k^{-1} e + \|\Delta x_k\|_{H_k}^2$ if $((\mathcal{M}_{\text{num}} \ge 0)$ and $(0 \le \|g_k\|_2^2 \le \epsilon_g))$ then If descent condition is not satisfied, and c cannot be increased due to small $\| g_k \|_2^2$, generate new maximizer and x_{k+1} using semi-infinite programming step.

Step 2(a)(ii): Semi-infinite programming step

$$x_{k+1} = \arg\min_{x\in X_f},\ \max_{y\in Y(x_k)}\{f(x,y)\}, \quad v = \min_{x\in X_f}\max_{y\in Y(x_k)}\{f(x,y)\}$$
$$\hat{y}_{k+1} = \arg\max_{y\in Y}\{f(x_{k+1},y)\}$$

if $f(x_{k+1}, \hat{y}_{k+1}) \le v$ *Stop*: the additional maximizer(s) $\hat{y}_{k+1}$ do not improve the current function value so x_{k+1} is the minimax solution. go to *Step 3*

Step 2(b): If $(\mathcal{M}_{\text{num}} \le 0)$ then descent assured and c_k remains unchanged otherwise no decrease with $\Delta x_k = \Delta x_k(1)$ and $n_{\max} > 1$, a new direction $\Delta x_k(\beta)$ needs to be computed if $(n_{\max} \ne 1)$ and $(\beta = 1)$ then go to *Step 1(b)* otherwise increase penalty parameter c_k :

$$c_{k+1} = max\left\{\frac{\Delta x_k^t \nabla_x\Phi_k - \mu^l\Delta x_k^t X_k^{-1} e + \|\Delta x_k\|_{H_k}^2}{\|g_k\|_2^2}, c_k + \delta\right\}.$$

Step 2(c): Compute w_{k+1} $\alpha_{x_k} = \theta^i\hat{\alpha}_k$, where $i = \min\{0, 1, 2, \dots, \}$ such that:

$$\Psi(x_k + \alpha_{x_k}\Delta x_k; c_{k+1}, \mu) - \Psi(x_k; c_{k+1}, \mu) \le \rho\alpha_{x_k}\nabla_x\Psi(x_k; c_{k+1}, \mu)^t\Delta x_k$$

If $(\|x_k + \alpha_{xk}\Delta x_k\|)/\|x_k\| \le \epsilon_{\text{tol}}$ then go to *Step 2(a)ii*

$$\text{LB}_k^i = \min\left\{\frac{1}{2}m\mu, (x_k^i + \alpha_{x_k}\Delta x_k^i)z_k^i\right\}, \quad \text{UB}_k^i = \max\{2M\mu, (x_k^i + \alpha_{x_k}\Delta x_k^i)z_k^i\}$$
$$\alpha_{z_k}^i = \max\{\alpha > 0:\text{LB}_k^i \le (x_k^i + \alpha_{x_k}\Delta x_k^i)(z_k^i + \alpha\Delta z_k^i) \le \text{UB}_k^i\}.$$
$$\alpha_{z_k} = \min\left\{1, \min_{1\le i\le n}\{\alpha_{z_k}^i\}\right\}.\alpha_{\lambda_k} = \alpha_{z_k}\alpha_k = (\alpha_{x_k}, \alpha_{z_k}, \alpha_{z_k})^t$$
$$w_{k+1} = w_k + \alpha_k\Delta w_k \ \hat{y}_{k+1} = \arg\max_{y\in Y} f(x_{k+1}, y)$$

If $\hat{y}_{k+1} \in Y_k$, $k = k+1$, go to *Step 2(a)(ii)*

Step 2(c): Update the set of potential maximizers $Y_{k+1} = Y_k \cup \hat{y}_{k+1}$, $k = k+1$, go to *Step 1(a).*

When updating the penalty parameter c, the following is computed:

$$\mathcal{M}_{\text{num}} = \Delta x_k^t\nabla\Phi(x_k) - c_k\|g_k\|_2^2 - \mu^l\Delta x_k^t X_k^{-1}e + \|\Delta x_k\|_{H_k}^2.$$

The algorithm moves from one inner iteration to another (with μ fixed) by minimizing the merit function $\Psi(x; c, \mu)$ which is achieved by appropriately selecting the penalty parameter c at each

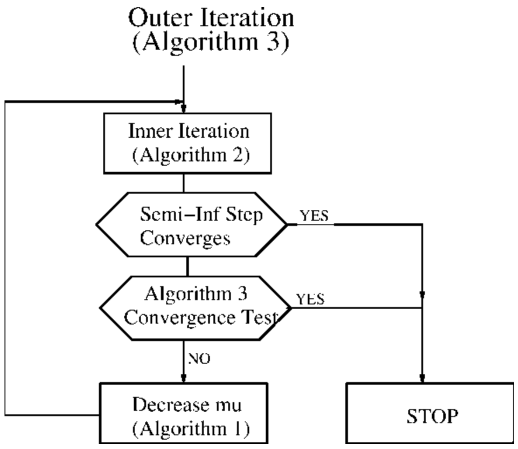

Figure 1. Structure of the overall interior point algorithm.

inner iteration. The monotonic decrease of Equation (5) and the rules for determining the primal and dual step sizes ensure that the inner iterates converge to the solution of Equation (3) for fixed value of μ. By reducing μ and $\{\mu\} \rightarrow 0$, the optimum of the initial problem (1) is obtained.

In the implementation of the algorithm, the following criterion is used for inner iteration: $\|F(x_k, \lambda_k, z_k; \mu)\|_2 \leq \eta\mu$. As long as the criteria is not satisfied, μ stays fixed and new w_k is computed.

The interior point algorithm consists of inner iterations (Algorithm 2) and outer iterations (Algorithm 3). As can be verified, the main work is done in the inner loop, where problem (3) is solved, with μ being fixed.

ALGORITHM 3 *Outer Iteration*

```
Step 0:  Initialization:
    Choose $x^0, \lambda^0, z^0, (\eta, \mu^0, \sigma, \delta, \varepsilon_g, \epsilon_0, \epsilon_{\text{tol}}) > 0$, $M$ and $m$ given by Equation (24) for $(\theta, \gamma) \in (0, 1)$, $\rho \in (0, 1/2)$;
    Compute $y_0 = \arg\max_{y\in Y} \{f(x^0, y)$; set $l = 0, k = 0, Y_0 = \{y_0\}$.

Step 1:  Termination check for overall optimality:
    if $\|F(x^l, \lambda^l, z^l; \mu)\|_2/(1 + \|x_l, \lambda_l, z_l\|_2) \leq \epsilon_0$ stop
    set $(x_k, \lambda_k, z_k) = (x^l, \lambda^l, z^l)$, $\chi_l = \min_i\{x_l^i z_l^i\}$, $\hat{\chi}_l = \max_i\{x_l^i z_l^i\}$

Step 2:  Execute inner iteration – Algorithm 2

Step 3:  Update $\mu$ – Algorithm 1

Step 4:  Set $(x^{l+1}, \lambda^{l+1}, z^{l+1}) = (x_k, \lambda_k, z_k)$; $l = l + 1$,Go to Step 1.
```

The overall structure of the interior point algorithm is given in Figure 1.

6. Computational results

In this section, we report experience with numerical examples. The algorithm defined in the previous sections have been implemented using Fortran 90 on an IBM compatible Personal Computer, using the Intel Pentium IV 3 GHz processor. The following 10 examples are based on

Table 3. Additional constraints for problems in [28].

Problem no.	Constraint(s)
1	$x_1^2 + x_2^2 \le 0.3$
2	$x_1^2 + x_2^2 \le 2.5$
3	$-x_1^2 + x_2^2 \le 1.75$
4	$x_1^2 \le 0.15,\ x_2 \le 0.1$
5	$x_1^2 + x_2^2 \le 0.03,\ x_2^2 + x_3^2 \le 0.06,\ x_3^2 + x_1^2 \le 0.05$
6	$x_1 \ge 0.2,\ x_3 + x_4 \ge 0.8$
7	$x_1 \ge 0.1,\ x_3 + x_4 \le 0.4$
8	$x_2 \ge 0.1,\ x_3 + x_4 \le 1.45$
9	$x_1 + x_2 \ge 0.1,\ x_3 + x_4 \le 1.7$
10	$x_1^2 + x_2^2 \le 1.5,\ x_3 + x_4 \le 0.1$

Table 4. Numerical results (1–5).

Problem no.		Minimax	SIP	Dim-x	Dim-y
1	CPU	0.094	1.472	2	2
	Iterations	11	350(8)		
2	CPU	0.141	0.329	2	2
	Iterations	16	91(3)		
3	CPU	1.250	3.063	2	2
	Iterations	71	307(12)		
4	CPU	0.109	0.203	2	3
	Iterations	11	46(2)		
5	CPU	0.266	0.469	3	3
	Iterations	18	78(4)		

those in [28]. First five are convex–concave functions (i.e. $f(x, y)$ convex in x and concave in y) and then the further five examples are convex–convex functions, where multiple maximizers are present, and we examine the behaviour of the algorithm in the presence of such points. Problems 1–6 correspond to problems 1–6 in [28], While problems 7–10 correspond to problems 13–16. Problems in [28] are unconstrained in the x variables. We therefore added some constraints to these problems. These are given in Table 3.

The various parameters used in the overall algorithm are as follows. In Algorithm 1, the barrier reduction rule, we set $\eta = 0.1$ and $\sigma = 6$. In Algorithm 2, in Step 1(c), we set $\gamma = 0.995$, in Step 2(a) $\epsilon_g = 10^{-12}$ and in Step 2(b) $\delta = 10$. In Step 2(c), $\theta = 0.5, \rho = 10^{-4}, \epsilon_{\text{tol}} = 10^{-12}, m = 1$, and $M = 10$. The accuracy of the stopping criterion (Algorithm 3) is $\epsilon_0 = 10^{-8}$. The dimensionality of the problems in the x variables is indicated in the Dim-x column of Tables 4 and 5; similarly for the y variables.

The results are obtained using the minimax algorithm presented in this paper and the semi-infinite algorithm (SIP) presented in [36].

6.1 *f(x, y) locally convex in x, concave in y*

The examples in this section illustrate the algorithm when $f(x, y)$ has a unique maximizer for x fixed. In Table 4, results in terms of CPU time and number of iterations are presented. For the SIP case, we present the number of nonlinear programs solved (in brackets) together with the total number of iterations. The minimax algorithm performs better in terms of CPU time and needs less iterations to converge, compared with the SIP algorithm discussed in [36].

Table 5. Numerical results (6–10).

Problem no.		Minimax	SIP	Dim-x	Dim-y
6	CPU	0.031	0.094	4	3
	Iterations	13	41(2)		
7	CPU	0.047	0.125	4	4
	Iterations	16	38(2)		
8	CPU	0.188	0.215	4	4
	Iterations	29	75(3)		
9	CPU	0.094	0.220	4	4
	Iterations	31	84(3)		
10	CPU	0.172	0.204	4	4
	Iterations	53	58(4)		

6.2 $f(x, y)$ *locally convex in x, locally convex in y*

The five examples below involve $f(x, y)$ locally convex in x and y, so the maximizers of $f(x, y)$ lie at the boundary of the feasible region Y. In general, such problems have multiple maximizers. We present five examples and each example is run twice – with the minimax algorithm and the semi-infinite programming algorithm in [36]. The CPU time and number of iterations for each of the problems 6–10 are presented in Table 5.

The results shown in Table 5 are consistent with those from Table 4 and indicate that the minimax algorithm converges faster. It is also worth observing that both algorithms perform better for problems with multiple maximizers. The reason for improved performance is that although at each point x there might be a number of different maximizers, the set of maximizers is discrete as it usually has a finite number of elements. In the case of convex–concave functions, this is not the case as the set:

$$\{y|\Phi(x) = f(x, y), \forall x \in X_f\}$$

has in general an infinite number of elements, as for each x there exists a different y so that $\Phi(x) = f(x, y)$. Therefore, at each iteration global optimization is needed, whereas in the convex–convex case, a switch to the discrete minimax problem is justified (see Step 2(a)(ii), Algorithm 2) once the appropriate number of maximizers, as established in Caratheodory's theorem [25], become available.

7. Conclusions

In this paper, we extend the algorithm for unconstrained continuous minimax problems in Rustem and Howe [28] to include constraints on the minimizing variable x. An interior point approach is used to ensure feasibility of the variables. A semi-infinite programming-based epi-convergent procedure accumulates maximizers. This is realized with a discrete minimax formulation that identifies the potential maximizers and also tests for termination. A quasi-Newton search direction, conditional on approximate maximizers is used and progress is maintained through a merit function.

To illustrate the performance of the algorithm, a number of numerical examples are considered and compared with a purely semi-infinite programming approach. The results reported underline a superior performance of the algorithm in this paper. The main reason for this is that the discrete minimax procedure is able to add new maximizers, if required, at every iteration of the interior point algorithm whereas semi-infinite programming adds the maximizers as new constraints at the end of a full minimization. Costly line searches are avoided by terminating the global maximization procedure after the first maximizer satisfying the Armijo step size rule is identified.

Acknowledgements

Financial support of the EPSRC grants GR/R51377/01 and EP/C513584/1 are gratefully acknowledged. The authors are also grateful to Professor Elijah Polak for valuable suggestions and advice on this paper. Any remaining errors are the responsibility of the authors.

Note

1. The algorithm utilizes a positive definite approximation to this Hessian.

References

[1] I. Akrotirianakis and B. Rustem, *A globally convergent interior point algorithm for general non-linear programming problems*, J. Optim. Theory Appl. 125 (2005), pp. 497–521.

[2] L. Armijo, *Minimization of functions having Lipschitz continuous first-partial derivatives*, Pac. J. Math. 16 (1966), pp. 1–3.

[3] R.G. Becker et al., *The simultaneous use of rival models in policy optimization*, Econ. J. 96 (1986), pp. 425–448.

[4] J.W. Blankenship and J.E. Falk, *Infinitely constrained optimization problems*, J. Optim. Theory Appl. 19 (1976), pp. 261–281.

[5] M. Breton and S. El Hachem, *Algorithms for the solution of stochastic dynamic minimax problems*, Comput. Optim. Appl. 4 (1995), pp. 317–345.

[6] R.M. Chamberlain et al., *The watchdog technique for forcing convergence in algorithms for constrained optimization*, Math. Program. Study 16 (1982), pp. 1–17.

[7] V.F. Demyanov and V.N. Malozemov, *Introduction to Minimax* (Louvish, D., transl.), John Wiley, New York, 1974.

[8] V.F. Demyanov and A.B. Pevnyi, *Numerical methods for finding saddle points*, USSR Comp. Math. Math. Phys. 12 (1972), pp. 1099–1127.

[9] A.S. El–Bakry et al., *On the formulation and theory of the Newton interior point method for nonlinear programming*, J. Optim. Theory Appl. 89 (1996), pp. 507–541.

[10] D.M. Gay, M.L. Overton, and M.H. Wright, *A primal-dual interior method for nonconvex nonlinear programming*, Technical Report 97–4–08, Bell Laboratories, Murray Hill, USA, 1997.

[11] S.A. Gustafson, *A three-phase algorithm for semi-infinite programming*, in *Semi-infinite Programming and Applications*, *Lecture Notes in Economics and Mathematical Systems*, A.V. Fiacco and O. Kortanek, eds., Vol. 215, Springer, Berlin, 1981, pp. 138–157.

[12] W.W. Hager and D.L. Presler, *Dual techniques for minimax*, SIAM J. Control Optim. 25 (1987), pp. 660–685.

[13] R. Hettich and K.O. Kortanek, *Semi-infinite programming: theory, methods, and applications*, SIAM Rev. 35 (1993), pp. 380–429.

[14] K.C. Kiwiel, *A direct method of linearization for continuous minimax problems*, J. Optim. Theory Appl. 55 (1987), pp. 271–287.

[15] L.S. Lasdon, J. Plummer, and Y. Gang, *Primal-dual and primal interior point algorithms for general nonlinear programs*, ORSA J. Comput. 7 (1995), pp. 321–332.

[16] C.T. Lawrence and A.L. Tits, *Feasible sequential quadratic programming for finely discretized problems from SIP*, in *Semi-infinite Programming*, R. Reemtsen and J.J. Ruckmann, eds, Kluwer Academic Publishers, The Netherlands, 1998, pp. 159–193.

[17] N. Maratos, *Exact penalty function algorithms for finite dimensional and optimization problems*, Ph.D. Thesis, Imperial College London, UK, 1978.

[18] E. Obasanjo and B. Rustem, *An interior point algorithm for nonlinear minimax problems*, DoC Research Report, Imperial College, 2005.

[19] V.M. Panin, *Linearization method for continuous min–max problems*, Kibernetika, 2 (1981), pp. 75–78.

[20] E. Polak, *Optimization – Algorithms and Consistent Approximations*, Springer, New York, 1997.

[21] E. Polak and J.O. Royset, *Algorithms for finite and semi-infinite min–max–min problems using adaptive smoothing techniques*, J. Optim. Theory Appl. 119 (2003), pp. 421–457.

[22] E. Polak, D.Q. Mayne, and E.J. Higgins, *A superlinearly convergent min–max algorithm for min–max problems*, Memorandum No: UCB/ERL M86/103, Department of Electrical Engineering, University Of California, Berkeley, CA, 1988.

[23] E. Polak, J.O. Royset, and R.S. Womersley, *Algorithms with adaptive smoothing for finite minimax problems*, J. Optim. Theory Appl. 119 (2003), pp. 459–484.

[24] M.J.D. Powell, *A fast algorithm for nonlinearly constrained optimization calculations*, in *Numerical Analysis Proceedings, Biennial Conference, Dundee, Lecture Notes in Mathematics*, G.A. Watson, ed., Vol. 630, Springer Verlag, Berlin, 1978, pp. 144–157.

[25] R.T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.

[26] R.T. Rockafellar and R.J.–B. Wets, *Variational Analysis*, Springer, Berlin, 2004.

[27] J.B. Rosen, *Existence and uniqueness of equilibrium points for concave n-person games*, Econometrica 33 (1965), pp. 520–534.

[28] B. Rustem and M.A. Howe, *Algorithms for Worst-case Design with Applications to Risk Management*, Princeton University Press, Princeton, 2001.

[29] B. Rustem and Q. Nguyen, *An algorithm for inequality constrained discrete min–max problems*, SIAM J. Optim. 8 (1998), pp. 256–283.

[30] B. Rustem, S. Zakovic, and P. Parpas, *Convergence of an interior point algorithm for continuous minimax*, J. Optim. Theory Appl. 136 (2007), pp. 87–103.

[31] H. Sasai, *An interior penalty method for minimax problems with constraints*, SIAM J. Control 12 (1974), pp. 643–649.

[32] O. Stein and G. Still, *Solving semi-infinite optimization problems with interior point techniques*, SIAM J. Control Optim. 42 (2003), pp. 769–788.

[33] R.S. Womersley and R. Fletcher, *An algorithm for composite nonsmooth optimization problems*, J. Optim. Theory Appl. 48 (1986), pp. 493–523.

[34] H. Yamashita, *A globally convergent primal-dual interior point method for constrained optimization*, Technical report, Mathematical Systems Institute Inc., 2-5-3 Shinjuku, Shinjuku-ku, Tokyo, Japan, May 1995.

[35] H. Yamashita and H. Yabe, *Superlinear and quadratic convergence of some primal-dual interior point methods for constrained optimization*, Math. Program. 75 (1996), pp. 377–397.

[36] S. Zakovic and B. Rustem, *Semi-infinite programming and applications to minimax problems*, Ann. Oper. Res. 124 (2003), pp. 81–110.

[37] S. Zakovic, B. Rustem, and V. Wieland, *A continuous minimax problem and its application to inflation targeting*, in *Decision and Control in Management Science*, G. Zaccour, ed., Kluwer Academic Publishers, Dordrecht, 2002.

[38] Y. Zhang, R.A. Tapia, and J.E. Dennis, Jr., *On the superlinear and quadratic convergence of primal-dual interior point linear programming algorithms*, SIAM J. Optim. 2 (1992), pp. 304–324.

[39] Y. Zhang, R.A. Tapia, and F. Potra, *On the superlinear convergence of interior point algorithms for a general class of problems*, Technical Report TR90–9, Department of Mathematical Sciences, Rice University, USA, 1990.

[40] J.L. Zhou and L.A. Tits, *An SQP algorithm for finely discretized continuous minimax problems and other minimax problems with many objective functions*, SIAM J. Numer. Anal. 6 (1996), pp. 461–487.