J Optim Theory Appl (2008) 136: 87–103
DOI 10.1007/s10957-007-9290-1

Convergence of an Interior Point Algorithm for Continuous Minimax

B. Rustem · S. Žaković · P. Parpas

Published online: 15 November 2007

Abstract We propose an algorithm for the constrained continuous minimax problem. The algorithm uses a quasi-Newton search direction, based on subgradient 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 toward epiconvergence. 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 toward the solution is maintained using merit functions.

Keywords Worst case analysis · Continuous minimax algorithms · Interior point methods · Semi–infinite programming

1 Formulation of the Problem

In a companion paper [1] we propose an algorithm for the continuous minimax problem, we presented our motivation and discussed numerical results. In this paper, we focus on the theoretical properties of the algorithm and establish its convergence. The notation we use is identical with [1]. For ease of exposition we repeat some of the basic definitions and concepts. Before we consider the convergence proof of the algorithm we also give an outline of its basic steps. Readers interested in the motivation and computational implementation and numerical performance of the algorithm are referred to in [1].

Communicated by L.C.W. Dixon.

B. Rustem · S. Žaković · P. Parpas (✉)
Department of Computing, Imperial College, London, UK
e-mail: pp500@doc.ic.ac.uk

The problem we are concerned with is given by

$$\min_x \max_{y\in Y}\{f(x,y) \mid g(x)=0,\ x\geq 0\}. \tag{1}$$

The exact properties of the functions and sets involved are detailed below. For now it is sufficient to say that f, and g are assumed to be differentiable, and Y compact. The proposed algorithm combines semi-infinite programming with quasi-Newton search directions. The latter direction is generated using subgradient information. We use the interior point algorithmic framework and a merit function to enforce positivity of the iterates and encourage feasibility.

The applications of the continuous minimax problem span many areas where optimization methods can be fruitfully applied. Details are given in [1]. Likewise, there are many different solution methods for the problem. There exist algorithms that are very efficient when the cardinality of Y is finite [2–5]. Efficient algorithms are also available when f is convex in its first argument, and concave in its second (see e.g. [6, 7]). The contribution of this paper is to propose an algorithm that is efficient in the general case i.e. when Y is infinite, and when the convexity of f in x is not assumed. Applications and solution algorithms for the continuous minimax problem are reviewed in [1], see also [8–13].

The max function and the set of maximizers will be denoted by

$$\Phi(x)=\max_{y\in Y} f(x,y), \qquad \hat{Y}(x)=\{y\in Y \mid f(x,y)=\Phi(x)\}.$$

The transformed minimax problem, for $x>0$, is given by

$$\min_x \max_{y\in Y}\left\{f(x,y)-\mu\sum_{i=1}^{n}\log(x^i) \mid g(x)=0\right\}. \tag{2}$$

Consider the following augmented objective function:

$$P(x,y;c,\mu)=f(x,y)+\frac{c}{2}\|g(x)\|_2^2-\mu\sum_{i=1}^{n}\log(x^i); \tag{3}$$

the maximum of the preceding equation will be denoted by $\Psi(x;c,\mu)$,

$$\Psi(x;c,\mu)=\max_{y\in Y} P(x,y;c,\mu). \tag{4}$$

The basic idea behind the algorithm is to use the augmented objective function (3) and generate a descent direction where the current incumbent can be improved in terms of feasibility and reduction of the objective function. Despite the differentiability of (3), its maximum may fail to be differentiable. For this reason a straightforward application of Newton's or other descent methods is impossible. In order to use a descent-type method, we will need to work with the subdifferentials of both the max and merit functions,

$$\partial\Phi(x)=\mathrm{conv}\{\nabla_x f(x,y) \mid y\in\hat{Y}(x)\}=\sum_{y\in\hat{Y}(x),\beta\in\mathcal{B}}\beta^y\nabla_x f(x,y),$$

$$\partial\Psi(x;c,\mu)=\mathrm{conv}\{\nabla_x P(x,y;c,\mu)\mid y\in\hat{Y}(x)\}=\sum_{y\in\hat{Y}(x),\beta\in\mathcal{B}}\beta^y\nabla_x P(x,y;c,\mu),$$

where $\mathcal{B}=\{\beta\mid\sum_y\beta^y=1,\ \beta^y\geq 0\}$. Using Caratheodory's theorem [14], a vector $\nabla\Psi(x;c,\mu)\in\partial\Psi(x;c,\mu)$ can be characterized with at most $(n+1)$ vectors $\nabla_x P(x,y;c,\mu)\in\partial\Psi(x;c,\mu)$ so that

$$\nabla\Psi(x;c,\mu)=\sum_{y\in\hat{Y}(x),\beta\in\mathcal{B}}\beta^y\nabla_x f(x,y)+c\nabla g(x)^t g(x)-\mu X^{-1}e, \tag{5}$$

where X^{-1} is the diagonal matrix, defined as $X^{-1}=\mathrm{diag}(\frac{1}{x^1},\frac{1}{x^2},\dots,\frac{1}{x^n})$. Similarly, we have $\nabla\Phi(x)=\sum_{y\in\hat{Y}(x),\beta\in\mathcal{B}}\beta^y\nabla_x f(x,y)$. Working directly with the whole set of maximizers at a given point is extremely difficult. For this reason we define the sets $Y_i\subset Y$ so that, $Y_0=\{y_0\}$, $y_i=\arg\max_{y\in Y} f(x_i,y)$, $Y_i=Y_{i-1}\cup y_i$, $i=1,2,\dots$. A finite set of maximizers at the current point x_k is denoted by

$$Y(x_k)=\{y\in Y_k\,|\,f(x_k,y)=\Phi(x_k)\}.$$

Note that $Y(x_k)\subset Y_k$. The proposed algorithm works with $Y(x_k)$. This makes the method more efficient numerically as the whole set of maximizers can not, in general, be computed. The first-order conditions of problem (3) are given by

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

where $\nabla_x\Phi(x)$ is a sub-gradient of $\Phi(x)$ and its evaluation is considered in the next section. Invoking the nonlinear transformation $z=\mu X^{-1}e$ yields

$$F^T=\big[\nabla\Phi(x)-z-\nabla g^t(x)\lambda,\ g(x),\ XZe-\mu e\big]=0, \tag{6}$$

where $F=F(x,\lambda,z;\mu)$ and Z is the diagonal matrix $Z=\mathrm{diag}(z_1,z_2,\dots,z_k)$. These equations represent the perturbed optimality conditions for problem (1). The solution of the Newton system associated with (6) is given by

$$\begin{aligned}\Delta x_k&=\Omega_k^{-1}\nabla g_k^t\Delta\lambda_k-\Omega_k^{-1}h_k,\\ \Delta\lambda_k&=-[\nabla g_k\Omega_k^{-1}\nabla g_k^t]^{-1}(g_k-\nabla g_k\Omega_k^{-1}h_k),\\ \Delta z_k&=-z_k+\mu X_k^{-1}e-X_k^{-1}Z_k\Delta x_k,\end{aligned} \tag{7}$$

where $\Omega_k=H_k+X_k^{-1}Z_k$, and $h_k=\nabla\Phi(x_k)-\mu X_k^{-1}e-\nabla g_k^t\lambda_k$. Introducing 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 (7) can be written as

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

H_k is a positive definite approximation of the Hessian of the Lagrangian associated with (2).[1] H_k is approximated using the updating formula suggested by Powell [15]. Starting from an initial point w_0, the algorithm generates a sequence $\{w_k\}$: $w_{k+1} = w_k + \alpha_k \Delta w_k$. In order to maintain feasibility of w_{k+1}, the algorithm needs to ensure $x_{k+1}, z_{k+1} > 0$. The algorithm generates a descent direction based on a sub–gradient of $\Phi(x)$ and an approximate Hessian. 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 epigraphs to determine potential multiple maximizers. The overall iterative process is in two stages: first, (2) is solved for μ fixed. This is the *inner iteration*. Once (2) is solved, μ is reduced, convergence criteria are checked (*outer iteration*) and, if necessary, another *inner iteration* is performed. We now turn to the delicate issue of computing a descent direction for the max-function.

The quadratic approximation to $P(x, y; c, \mu)$ in (3) at x_k is given by

$$P_k(\Delta x_k, y; c, \mu) = P(x_k, y; c, \mu) + \Delta x^t \nabla_x P(x_k, y; c, \mu) + \frac{1}{2}\|\Delta x_k\|^2_{H_k}. \tag{8}$$

In the presence of multiple maximizers, the computation of the subgradients of $\Phi(x)$ is a nontrivial matter. We consider two approaches for determining the β's in (5). In the first, β_k is computed as follows:

$$\beta_k = \arg\max_{\beta \in \mathcal{B}} \left\{ \Delta x_k(\beta)^t \left(\sum_{y \in Y(x_k)} \beta^y \nabla_x f(x_k, y) + c_k \nabla g_k^t g_k - \mu X_k^{-1} e \right) + \frac{1}{2}\|\Delta x_k(\beta)\|^2_{H_k} \right\}. \tag{9}$$

For later reference, we define

$$F_k^{\beta_k} = \left[\sum_{y \in Y(x_k)} \beta_k^y \nabla_x f(x_k, y) - z_k - \nabla_x g_k^t \lambda_k,\ g_k,\ X_k Z_k e - \mu e \right]^T. \tag{10}$$

The second approach is given by: $\beta_k^y = 1$ for some $y = y_{k+1} \in Y(x_k)$ or $\beta_k^y = 0, \forall y \neq y_{k+1}, y \in Y(x_k)$. 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

$$y_{k+1} = \arg\max_{y \in Y(x_k)} \left\{ \Delta x_k(1)^t (\nabla f(x_k, y) + c \nabla g_k^t g_k - \mu X_k^{-1} e) + \frac{1}{2}\|\Delta x_k(1)\|^2_{H_k} \right\}. \tag{11}$$

We therefore, consider two possible directions $\Delta w(1)$ and $\Delta w(\beta_k)$, depending on the subgradient $\nabla \Phi(x)$ used. The direction $\Delta w(1)$ is easier to compute, as it does not entail the solution of the quadratic programming problem (9). We define two characterizations of $\nabla P(x_k, ., c_k, \mu)$. One corresponds to the maximizer y_{k+1} responsible

[1] This is a strong assumption that can be relaxed, for example, by considering the Hessian of the augmented Lagrangian. We omit further discussion on this for brevity.

for the worst-case descent direction and the other corresponds to multiple maximizers, $y \in Y(x_k)$. These are given by

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

These two characterizations of $\nabla P(x_k, ., c_k, \mu)$ lead to two possible sub-gradient choices for $\nabla \Phi(x_k)$ and $\nabla \Psi(x_k; c_k, \mu)$.The new maximizer y_{k+1} is chosen as a maximizer of the following augmented quadratic approximation to $\Psi(x; c, \mu)$:

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

But if there exists a $\mathcal{C} \geq 0$ such that $\Phi(x_k) - f(x_k, y_{k+1}) = 0$ ensures (11). The two penalty parameters play an important role in the algorithm. By $\{(x_*(\mu), \lambda_*(\mu), z_*(\mu))\}$ we will denote the solution of the system in (6). The trajectory containing the solution to the perturbed system is called the central path. As μ approaches zero, the path converges to the solution of the original system. A numerically efficient procedure for the update of the barrier parameter μ is suggested in [1]. For the theoretical properties of the algorithm, it is sufficient to have any sequence μ going to zero.

A much more subtle role in the algorithm is played by the penalty parameter c. This will be briefly discussed next. For a more detailed discussion we refer the interested reader to [1].

The subgradient of $\Psi(x_k; c, \mu)$ at the kth iteration is

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

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

$$(\nabla \Phi(x_k) + c_k \nabla g_k^t g_k - \mu X_k^{-1} e, \Delta x_k) + \frac{1}{2} \|\Delta x_k\|_{H_k}^2 \leq 0. \tag{12}$$

Using the fact that the Newton system must satisfy $\nabla g_k^T \Delta x_k = -g_k$, the directional derivative $\Delta x_k^t \nabla \Psi(x_k; c_k, \mu)$ can be written as

$$\Delta x_k^t \nabla \Psi(x_k; c_k, \mu) = \Delta x_k^t \nabla \Phi(x_k) - c_k \|g_k\|^2 - \mu \Delta x_k^t X_k^{-1} e, \tag{13}$$

where c_k is the value of the penalty parameter at the beginning of the k-th iteration. Since μ is fixed we can deduce that if c_k is large enough then the descent in (13) can be achieved. When a single maximizer is present then descent can be assured by increasing c. In the case of multiple maximizers the direction of the single maximizer can be used as long as it ensures descent without an increase in c. When a single maximizer cannot ensure descent, problem (9) is solved and a new sub-gradient, that 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

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

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

When Δx_k is not a descent direction for the merit function (4), and $0 < \|g_k\|_2^2 < \epsilon_g$, then x_{k+1} and v are given by the solution of the following discrete minimax problem: $\min_{x \in X_f} \max_{y \in Y(x_k)}\{f(x, y)\}$. The value v is used in Step 2(a) of the algorithm. At x_{k+1}, the new maximizer is computed as: $\hat{y}_{k+1} = \arg\max_{y \in Y} f(x_{k+1}, y)$. The algorithm terminates if $f(x_{k+1}, \hat{y}_{k+1}) \le v$. Otherwise the new maximizer is added to the set of maximizers $Y_{k+1} = Y_k \cup \hat{y}_{k+1}$, and a new iteration is performed. 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 1 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 $\Delta w_k(\beta_k)$. We now have all the definitions and basic concepts required to state the algorithm.

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 $\|F(x_k, y_k, \lambda_k, z_k; \mu)\|_2 \le \eta\mu$, inner iteration converged. Go to outer iteration.

Compute the descent direction if $\beta = 1$ then

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

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

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 $\Delta x_k = \Delta x_k(\beta_k)$ which are actually computed in Step 1(c)

Step 1(c): *Interior point step*

$$\Delta w_k(\beta_k) = -(\nabla F_k^{\beta_k})^{-1} F_k^{\beta_k}; \qquad \alpha_{xk}^{\max} = \min_{1 \le i \le n} \left\{ \frac{-x_k^i}{\Delta x_k^i} : \Delta x_k^i < 0 \right\},$$
$$\hat{\alpha}_{x_k} = \min\{\gamma \alpha_{x_k}^{\max}, 1\}. \tag{14}$$

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 $((M_{\text{num}} \ge 0)$ 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*,

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

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

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 ($M_{\mathrm{num}} \le 0$), 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 the penalty parameter* c_k,

$$c_{k+1} = \max\{(\Delta x_k^t \nabla_x \Phi_k - \mu^l \Delta x_k^t X_k^{-1} e + \|\Delta x_k\|_{H_k}^2)/(\|g_k\|_2^2),\ c_k + \delta\}. \tag{15}$$

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

$$\Psi(x_k + \alpha_{x_k}\Delta x_k; c_{k+1},\mu) - \Psi(x_k; c_{k+1},\mu) \le \rho\alpha_{x_k}\nabla_x\Psi(x_k; c_{k+1},\mu)^t \Delta x_k. \tag{16}$$

If $\frac{\|x_k+\alpha_{xk}\Delta x_k\|}{\|x_k\|} \le \epsilon_{tol}$, then go to Step 2(a)(ii) $LB_k^i = \min\{\frac{1}{2}m\mu, (x_k^i + \alpha_{xk}\Delta x_k^i)z_k^i\}$, $UB_k^i = \max\{2M\mu, (x_k^i + \alpha_{xk}\Delta x_k^i)z_k^i\}$ $\alpha_{zk}^i = \max\{\alpha > 0 : LB_k^i \le (x_k^i + \alpha_{xk}\Delta x_k^i)(z_k^i + \alpha\Delta z_k^i) \le UB_k^i\}$. $\alpha_{zk} = \min\{1, \min_{1\le i\le n}\{\alpha_{zk}^i\}\}$. $\alpha_{\lambda_k} = \alpha_{zk}$. $\alpha_k = (\alpha_{x_k}, \alpha_{z_k}, \alpha_{z_k})^t$. $w_{k+1} = w_k + \alpha_k \Delta w_k$. $\hat{y}_{k+1} = \arg\max_{y\in Y} f(x_{k+1},y)$. If $\hat{y}_{k+1} \in Y_k$, $k = k+1$, go to Step 2(a)(ii).

Step 3: *Update the set of potential maximizers* $Y_{k+1} = Y_k \cup \hat{y}_{k+1}$, $k = k+1$, go to Step 1(a).

The outer iteration involves the reduction of μ to zero, and termination checks for the over problem.

2 Convergence Results

The semi-infinite programming steps in Step 2(a)(ii) of the algorithm provide a safety-net for the method which tries to progress using descent steps if a sufficient number of (multiple) maximizers are identified. The interior point approach looks for new maximizers at each x_{k+1} assuming the merit function can be reduced. Semi-infinite programming, solves the discrete minimax problem for the given set of discrete maximizers. Convergence of the semi-infinite programming algorithm is based on the discretization of (1), given by: $\min_{x\in X}\max_{y\in Y_k}\{f(x,y)\}$, where Y_k is some finite subset of Y. Let $\bar{Y}$ denote all finite subsets of Y. At some k, $Y_k \subset Y$, $x_k \in X_f$, the semi-infinite programming algorithm steps compute x_{k+1} and attempt to find a $\widehat{y} \in Y$ such that: $f(x_{k+1},\widehat{y}) > f(x_{k+1},y)$, $\forall y \in Y_k$. The semi-infinite algorithm can be expressed as a point to set mapping [16]. The search for a $\widehat{y}$ can be viewed as: $\beta : X_f \times \bar{Y} \to \bar{Y}$, where β is given by

$$\beta(x_k, Y_k) = \{Y_k \cup \{\widehat{y}\} \mid f(x_k, \widehat{y}) > f(x_k, y), \forall y \in Y_k\}.$$

As before, we let $Y_{k+1} = Y_k \cup \hat{y}_{k+1}$. Let $\Phi_k(x) = \max_{y \in Y_k} f(x, y)$ and let $\varepsilon_k^\alpha > 0$ indicate the neighborhood: $\{x \in X_f \mid \Phi(x_k) \le \Phi(x), x \in \|x_k - x\| \le \varepsilon_k^\alpha\}$. The computation of x_{k+1}, realized by minimizing $\Phi_k(x)$, can be defined as

$$\alpha(\beta(x_k, Y_k)) : Y_{k+1} \to X_f \times \mathcal{R}^+,$$

$$\alpha(Y_{k+1}) = \left\{x \in X_f \mid \text{is an } \varepsilon_k^\alpha \text{ local solution to } \min_{x \in X_f} \max_{y \in Y_{k+1}} f(x, y)\right\}.$$

Lemma 2.1 *$\alpha \circ \beta$ is closed.*

Proof The proof can be found in [17]. □

Lemma 2.2 *Let x_*, Y_* be accumulation points of the sequences $\{x_k\}$ and $\{Y_k\}$ generated by the algorithm. Then, x_* is a solution to (1).*

Proof The proof can be found in [17]. □

In the rest of this section we focus on the basis that sufficient number of maximizers are available and the algorithm is able to generate a descent direction. If descent cannot be assured and new maximizers are needed, the maximizers found so far are retained and utilized by the semi-infinite programming algorithm. We show that, while the barrier parameter is fixed to a value μ^l, the algorithm produces iterates $w_k(\mu^l) = (x_k(\mu^l), \lambda_k(\mu^l), z_k(\mu^l))$, for $k \ge 0$, which are bounded and converge to a point $w_*(\mu^l) = (x_*(\mu^l), \lambda_*(\mu^l), z_*(\mu^l))$ such that: $\|F(x_*(\mu^l), \lambda_*(\mu^l), z_*(\mu^l); \mu^l)\| = 0$. In other words, we show that the inner iteration, converges to a solution of the perturbed optimality conditions (6). For simplicity we suppress the index l, and we use w_k instead of $w_k(\mu^l)$ to denote the iterates produced while $\mu = \mu^l$.

Lemma 2.3 *Let f and g be differentiable functions and suppose that there exists a small $\epsilon_g > 0$, such that $\|g_k\|^2 > \epsilon_g$. If Δx_k is calculated by (7) and c_{k+1} is chosen as in (15), then Δx_k is a descent direction for the merit function Ψ at the current point x_k. Furthermore,*

$$\Delta x_k^t \nabla \Psi(x_k; c_{k+1}, \mu) + \frac{1}{2}\|\Delta x_k\|_{H_k}^2 \le -\frac{1}{2}\|\Delta x_k\|_{H_k}^2 \le 0. \tag{17}$$

Proof Assume that Δw_k is given by $\Delta w(1)$ and that we are using y_{k+1} to determine the search direction: $y_{k+1} = \arg\max_{y \in Y(x_k)}\{\Delta x_k^t \nabla_x P(x_k, y; c_k, \mu) + \frac{1}{2}\|\Delta x_k\|_{H_k}^2\}$.

Then, it follows that: $\Delta x_k(1)^t \nabla \Psi(x_k; c_{k+1}, \mu) = \Delta x_k(1)^t \nabla_x f(x_k, y_{k+1}) - c_k \|g_k\|_2^2 - \mu \Delta x_k(1)^t X_k^{-1} e$. In Step 2, the algorithm initially checks the inequality

$$\Delta x_k(1)^t \nabla_x f(x_k, y_{k+1}) - c_k \|g_k\|^2 - \mu \Delta x_k(1)^t X_k^{-1} e + \|\Delta x_k(1)\|_{H_k}^2 \le 0. \quad (18)$$

If (18) is satisfied, then by setting $c_{k+1} = c_k$ and rearranging (18), we obtain (17). On the other hand, if (18) is not satisfied, by setting

$$\begin{aligned} c_{k+1} = \max\{&(\Delta x_k(1)^t \nabla_x f(x_k, y_{k+1}) - \mu \Delta x_k(1)^t X_k^{-1} e \\ &+ \|\Delta x_k(1)\|_{H_k}^2)/(\|g_k\|^2), c_k + \delta\}, \end{aligned}$$

for $\delta > 0$, and substituting into (13), it can be verified that (17) also holds. If $n_{\max} > 1$, when Δw_k is given by $\Delta w(\beta)$, we have

$$\begin{aligned} &\Delta x_k(\beta_k)^t \nabla \Psi(x_k; c, \mu) + \|\Delta x_k(\beta_k)\|_{H_k}^2 \\ &\quad = \max_{\beta^y} \Delta x_k(\beta_k)^t \left(\sum_{\beta^y} \beta^y \nabla_x P(x_k, y)\right) + \frac{1}{2}\|\Delta x_k(\beta_k)\|_{H_k}^2 + \frac{1}{2}\|\Delta x_k(\beta_k)\|_{H_k}^2 \\ &\quad \le \max_{\beta^y} \left\{ \Delta x_k(\beta^y)^t \left(\sum_{\beta^y} \beta^y \nabla_x P(x_k, y)\right) + \frac{1}{2}\|\Delta x_k(\beta^y)\|_{H_k}^2 \right\} + \frac{1}{2}\|\Delta x_k(\beta_k)\|_{H_k}^2 \\ &\quad = \left\{ \Delta x_k(\beta_k)^t \left(\sum_{\beta_k} \beta_k \nabla_x P(x_k, y)\right) + \frac{1}{2}\|\Delta x_k(\beta_k)\|_{H_k}^2 \right\} + \frac{1}{2}\|\Delta x_k(\beta_k)\|_{H_k}^2 \\ &\quad = \Delta x_k(\beta_k)^t \left(\sum_{y \in Y(x_k)} \beta_k^y \nabla_x f(x_k, y)\right) - c_k \|g_k\|_2^2 \\ &\qquad - \mu \Delta x_k(\beta_k)^t X_k^{-1} e + \|\Delta x_k(\beta_k)\|_{H_k}^2 \\ &\quad = \Delta x_k(\beta_k)^t \nabla \Phi(x_k) - c_k \|g_k\|_2^2 - \mu \Delta x_k(\beta_k)^t X_k^{-1} e + \|\Delta x_k(\beta_k)\|_{H_k}^2, \end{aligned}$$

where β_k is obtained solving (9). In Step 2, the algorithm checks the inequality

$$\Delta x_k(\beta_k)^t \nabla_x \Phi(x_k) - c_k \|g_k\|^2 - \mu \Delta x_k(\beta_k)^t X_k^{-1} e + \|\Delta x_k(\beta_k)\|_{H_k}^2 \le 0. \quad (19)$$

If (19) is satisfied then by setting $c_{k+1} = c_k$ and rearranging (19), we obtain (17). On the other hand, if (19) is not satisfied, by setting

$$c_{k+1} = \max\left\{ \frac{\Delta x_k(\beta_k)^t \nabla_x \Phi(x_k) - \mu \Delta x_k(\beta_k)^t X_k^{-1} e + \|\Delta x_k(\beta_k)\|_{H_k}^2}{\|g_k\|^2}, c_k + \delta \right\},$$

for $\delta > 0$, and substituting into (13), it can be verified that (17) also holds. □

In the previous lemma it is assumed that $\|g_k\|^2 > \epsilon_g$. The next lemma demonstrates that Δx_k remains a descent direction for the merit function Ψ when $g_k = 0$, i.e., when feasibility of the equality constraints has been achieved.

Lemma 2.4 *Let f and g be differentiable functions and let $\Delta w_k = (\Delta x_k, \Delta\lambda_k, \Delta z_k)$ be the Newton direction taken by solving system* (7). *If for some or all iterations k, $g_k = 0$, then the descent property* (17) *is satisfied for any choice of the penalty parameter $c_k \in [0, \infty)$.*

Proof If $g_k = 0$, then (8) yields

$$\Delta\lambda_k = M_k^{-1}\nabla g_k \Omega_k^{-1} h_k; \qquad \Delta x_k = -\mathcal{P}_k \Omega_k^{-1}\nabla_x \Psi_k. \tag{20}$$

From the fact that x_k and z_k are strictly positive and assuming that the second order sufficiency condition for optimality is satisfied at the solution point, we have $\Delta x_k^t H_k \Delta x_k \leq \Delta x_k^t (H_k + X_k^{-1} Z_k)\Delta x_k$. Now, for $g_k = 0$, we have

$$\begin{aligned}\Delta x_k^t \nabla\Psi(x_k; c, \mu) + \|\Delta x_k\|_{H_k}^2 &\leq \Delta x_k^t \nabla\Psi(x_k; c, \mu) + \|\Delta x_k\|_{\Omega_k}^2 \\ &= -\nabla\Psi^t \mathcal{P}_k \Omega_k^{-1}\nabla\Psi + \nabla\Psi^t \mathcal{P}_k \Omega_k^{-1}\nabla\Psi = 0.\end{aligned}$$

Therefore, $\Delta x_k^t \nabla\Psi(x_k; c, \mu) + \frac{1}{2}\|\Delta x_k\|_{H_k}^2 \leq -\frac{1}{2}\|\Delta x_k\|_{H_k}^2$, which establishes the descent property (17). □

Lemma 2.5 *Let the assumptions of the previous lemma hold and let $g_k = 0$, for some k. Then, the algorithm chooses $c_{k+1} = c_k$ in Step* 2. *Also, Δx_k is still a descent direction for the max of merit function Ψ at x_k.*

Proof In the previous lemma it was proved that the descent property (17) is satisfied for $g_k = 0$. This basically means that the condition in Step 2 of the algorithm is always satisfied. Consequently, the algorithm does not need to increase the value of the penalty parameter and simply sets $c_{k+1} = c_k$. For this choice of the penalty parameter it can be verified that the descent property (17) still holds. □

Lemma 2.6 *Let f and g be continuously differentiable functions and $\Delta x_k^t \nabla\Phi(x_k) - \mu\Delta x_k^t X_k^{-1} e + \|\Delta x_k\|_{H_k}^2 \leq \mathcal{M}^* < \infty$. Then, for μ fixed*:

(i) *there exists a constant $c_{k+1} \geq 0$, satisfying Step* 2 *the algorithm*;
(ii) *assuming that the sequence $\{x_k\}$ is bounded away from zero, c_k is increased finitely often, then there exist an integer $k^* \geq 0$ and $c^* \in [0, \infty)$ such that $c_k = c^*$ for all $k \geq k^*$.*

Remark 2.1 In the preceding lemma we assume that the sequence x_k is bounded. Similarly, El-Bakry et al. in [18] define the following set:

$$\Theta(\epsilon) \equiv \{w_k \mid \epsilon \leq \|F(x_k, \lambda_k, z_k; 0)\|_2^2 \leq \|F(x_0, \lambda_0, z_0; 0)\|_2^2\}.$$

The sequence $\{\mu X_k^{-1} e\}$ converges to z_k at the end of the inner iteration. It is then shown, in Lemma 6.1 of [18] that if $w_k \subset \Theta(\epsilon)$, then the iteration sequence w_k is bounded above and in addition $\{(z_k, x_k)\}$ is componentwise bounded away from zero. This is used in the following proof.

Proof of Lemma 2.6 Part (i) is a direct consequence of Lemmas 2.3–2.5, since a finite value c_{k+1} is always generated, in step 2. Part (ii) will be shown by contradiction. Assume that $c_k \to \infty$ as $k \to \infty$. From the way c_{k+1} is defined in Step 2 we can deduce that, if $c_k \to \infty$, then $\|g_k\|^2 \to 0$. Thus, there exists an integer k_1 such that for all $k \geq k_1$ we have: $0 < \|g_k\|^2 \leq \epsilon_g$. However, in the case $0 < \|g_k\|^2 \leq \epsilon_g$, the algorithm stops increasing the penalty parameter since it switches to a procedure to generate more maximizers. Therefore the maximum value that c_k can take is: $c_* = c_{k_*} = \mathcal{M}^*/\epsilon_g < \infty$ where $\mathcal{M}^*$ and ϵ_g are finite values. This contradicts our assumption that $c_k \to \infty$ as $k \to \infty$. Hence the penalty parameter does not increase indefinitely, that is, there exists an integer $k_* \geq 0$ such that for all $k \geq k_*$, we have $c_k < \infty$. □

The basic result of Lemmas 2.3 to 2.6 is that the direction Δx_k, generated by (7), is a descent direction for the merit function Ψ at the current point x_k, that is inequality (17) holds. In the next theorem we show that the sequence $\{\Psi(x_k; c_*, \mu)\}$ generated by the interior point section of the algorithm is monotonically decreasing for barrier parameter μ fixed. It is assumed that at most $(n+1)$ maximizers are identified through the epi-convergent procedure in Step 2. We also show that the step α_{xk}, chosen by the Armijo step size strategy in Step 2 is always positive.

Lemma 2.7 *Consider the quadratic approximation* (8) *to* $P(x, y; c, \mu)$ *in* (3) *at* x_k. *We note that, for* $\alpha_k \in [0, 1]$,

$$\max_{y \in Y} P(x_k + \alpha_k \Delta x_k, y; c, \mu) = \max_{y \in Y(x_k + \alpha_k \Delta x_k)} P(x_k + \alpha_k \Delta x_k, y; c, \mu),$$

$$\max_{y \in Y(x_k)} P_k(0) = \max_{y \in Y} P(x_k, y; c, \mu) = \max_{y \in Y(x_k)} P(x_k, y; c, \mu),$$

and due to continuity, for $\alpha_k > 0$ *sufficiently small,*

$$\max_{y \in Y(x_k + \alpha_k \Delta x_k)} P_k(\alpha_k \Delta x_k, y; c_*, \mu) \leq \max_{y \in Y(x_k)} P_k(\alpha_k \Delta x_k, y; c_*, \mu). \tag{21}$$

Proof The above equalities follow by definition of Y, $Y(x_k)$ and P_k. Inequality (21) follows trivially if $Y(x_k) \supseteq Y(x_k + \alpha_k \Delta x_k)$, $\forall \alpha_k > 0$. Otherwise, consider the semi-infinite setting of: $\min_{x \in X_f, \tau} \{\tau \mid f(x, y) \leq \tau \ \forall y \in Y\}$. Where we have $f(x_k, y) = \tau$ if$y \in Y(x_k)$, or $f(x_k, y) < \tau$ otherwise. Let $\tau(\alpha)$ be the solution of the problem: $\min_\alpha \{\tau(\alpha) \mid f(x_k + \alpha \Delta x_k, y) - \tau(\alpha) \leq 0, \ \forall y \in Y\}$. At x_k, $y \in Y(x_k)$, moving from x_k along Δx_k, yields $\tau(\alpha)$ such that, by continuity of $f(x, y)$: $f(x_k + \alpha \Delta x_k, y) = \tau(\alpha)$ if $y \in Y(x_k)$, and $f(x_k + \alpha \Delta x_k, y) < \tau(\alpha)$ otherwise. Hence, there exists $\alpha_k > 0$ such that $Y(x_k) \supseteq Y(x_k + \alpha_k \Delta x_k)$. □

We observe that $\alpha_k > 0$ sufficiently large may lead to the case $Y(x_k + \alpha_k \Delta x_k) \nsubseteq Y(x_k)$. Nevertheless, for $\alpha_k > 0$ sufficiently small, the required result is assured. Practical applications also support this. It is also possible to compute $\max_{y \in Y} P_k(\alpha_k \Delta x_k, y; c_*, \mu)$, to define $\Psi(x_{k+1}; c_*, \mu)$. However, this involves additional computation.

Theorem 2.1 *Assume that*:

(i) *f and g are twice continuously differentiable*;
(ii) *the approximate Hessian matrix H_k is positive definite and bounded*;
(ii) *for each iteration k, there exists a solution to the Newton system* (7);
(iv) *there exists an iteration k_*, small $\epsilon_g > 0$, $\|g_k\|^2 \notin (0, \epsilon_g)$ and a scalar $c_* \geq 0$ ($c_* = c*(\epsilon_g)$) such that*

$$\Delta x_k^t \nabla \Psi(x_k; c, \mu) + \frac{1}{2}\|\Delta x_k\|_{H_k}^2 \leq -\frac{1}{2}\|\Delta x_k\|_{H_k}^2$$

is satisfied for all $k \geq k_*$ with $c_k(\epsilon_g) = c_*(\epsilon_g)$.

Then, the stepsize computed in (14–16) *is such that $\alpha_k \in (0, 1]$ and the sequence $\{\Psi(x_k; c_*, \mu)\}$ is monotonically decreasing for $k \geq k_*$ and μ fixed.*

Proof Consider the case when the direction Δw_k is given by $\Delta w_k(1)$ in Table 1. Note that

$$\begin{aligned}\Psi(x_{k+1}; c_*, \mu) &= \max_{y\in Y}\{P(x_{k+1}, y; c_*, \mu)\}\\ &= \max_{\beta\in\mathcal{B}}\left\{\sum_{y\in Y(x_{k+1})} \beta^y P(x_{k+1}, y; c_*, \mu)\right\}. \end{aligned} \tag{22}$$

The second order expansion of $P(x, y; c_*, \mu)$ with respect to x yields

$$\begin{aligned}&P(x_k + \alpha_k \Delta x_k, y; c_*, \mu)\\ &\quad = P(x_k, y; c_*, \mu) + \alpha_k \nabla_x P(x_k, y; c_*, \mu)^t \Delta x_k\\ &\qquad + \alpha_k^2 \int_0^1 (1-t)\Delta x_k^t(\nabla_x^2 P(x_k + t\alpha_k \Delta x_k, y; c_*, \mu) + H_k - H_k)\Delta x_k dt)\}. \end{aligned} \tag{23}$$

Using $P_k(\Delta x_k, y; c_*, \mu)$ given by (8), we evaluate the maximum on both sides of (23) with respect to y $\in Y$. Noting $x_{k+1} = x_k + \alpha_k \Delta x_k$, we have

$$\Psi(x_{k+1}; c_*, \mu) \leq \max_{y\in Y}\{P_k(\alpha_k \Delta x_k, y; c_*, \mu)\} + \alpha_k^2 \phi_k \frac{1}{2}\|\Delta x_k\|_2^2,$$

$$\phi_k = \max_{y\in Y}\left\{\int_0^1 (1-t)\Delta x_k^t(\nabla_x^2 P(x_k + t\alpha_k \Delta x_k, y; c_*, \mu) - H_k)\Delta x_k dt\right\}. \tag{24}$$

Using (21) in Lemma 2.7, we have

$$\begin{aligned}\max_{y\in Y}\{P_k(\alpha_k \Delta x_k, y; c_*, \mu)\} &= \max_{y\in Y(x_k+\alpha_k\Delta x_k)}\{P_k(\alpha_k \Delta x_k, y; c_*, \mu)\}\\ &\leq \max_{y\in Y(x_k)}\{P_k(\alpha_k \Delta x_k, y; c_*, \mu)\}. \end{aligned} \tag{25}$$

As the first term on the right in (25) is a convex function, we have

$$\begin{aligned}
&\max_{y\in Y(x_k)}\{P_k(\alpha_k\Delta x_k, y; c_*, \mu)\}\\
&\quad\le \alpha_k \max_{y\in Y(x_k)}\{P_k(\Delta x_k, y; c_*, \mu)\} + (1-\alpha_k)\max_{y\in Y(x_k)}\{P_k(0, y; c_*, \mu)\}\\
&\quad= \alpha_k \max_{y\in Y(x_k)}\{P_k(\Delta x_k, y; c_*, \mu)\} + (1-\alpha_k)\max_{y\in Y(x_k)}\{P(x_k, y; c_*, \mu)\}\\
&\quad= \max_{y\in Y(x_k)}\{(P(x_k, y; c_*, \mu)\}\\
&\qquad + \alpha_k\Big(\max_{y\in Y(x_k)}\{P_k(\Delta x_k, y; c_*, \mu)\} - \max_{y\in Y(x_k)}\{P(x_k, y; c_*, \mu)\}\Big)\\
&\quad\le \max_{y\in Y(x_k)}\{(P(x_k, y; c_*, \mu)\} + \alpha_k\Bigg(\max_{y\in Y(x_k)}\{P(x_k, y; c_*, \mu)\}\\
&\qquad + \max_{y\in Y(x_k)}\left\{\Delta x^t\nabla_x P(x_k, y; c_*, \mu) + \frac{1}{2}\|\Delta x_k\|^2_{H_k}\right\}\\
&\qquad - \max_{y\in Y(x_k)}\{P(x_k, y; c_*, \mu)\}\Bigg)\\
&\quad= \max_{y\in Y(x_k)}\{(P(x_k, y; c_*, \mu)\} + \alpha_k \max_{y\in Y(x_k)}\left\{\Delta x^t\nabla_x P(x_k, y; c_*, \mu) + \frac{1}{2}\|\Delta x_k\|^2_{H_k}\right\},
\end{aligned}$$

where the last inequality is due to taking the maximum of each subcomponent on the right. Thus, using (24), we have

$$\begin{aligned}
\Psi(x_{k+1}; c_*, \mu) \le \Psi(x_k; c_*, \mu) + \alpha_k \max_{y\in Y(x_k)}\left\{\Delta x^t\nabla_x P(x_k, y; c_*, \mu) + \frac{1}{2}\|\Delta x_k\|^2_{H_k}\right\}\\
+ \alpha_k^2\phi_k\|\Delta x_k\|_2^2. \qquad (26)
\end{aligned}$$

Consider the choice of $\Delta x_k(1), y_{k+1}$. By Lemmas 2.3 and 2.4, $\Delta x_k(1), y_{k+1}$ in (11) are chosen to ensure the inequality

$$\max_{y\in Y(x_k)}\left\{\Delta x^t\nabla_x P(x_k, y; c_*, \mu) + \frac{1}{2}\|\Delta x_k\|^2_{H_k}\right\} \le -\frac{1}{2}\|\Delta x_k\|^2_{H_k} \le -\frac{m'}{2}\|\Delta x_k\|^2 \qquad (27)$$

as established in Lemmas 2.3 and 2.4. From assumption (ii), we have: $\|\Delta x_k\|_2^2 \le \frac{1}{m'}\Delta x_k^t H_k \Delta x_k$, and from Lemmas 2.3 and 2.4, $\Delta x_k^t H_k \Delta x_k \le -\Delta x_k^t \nabla\Psi(x_k; c_*, \mu)$. Thus, (24) may be expressed as

$$\begin{aligned}
\Psi(x_{k+1}; c_*, \mu) \le \Psi(x_k; c_*, \mu) + \alpha_k \Delta x_k^t \nabla\Psi(x_k; c_*, \mu)\\
+ \frac{1}{2}\alpha_k^2\|\Delta x_k\|^2_{H_k} + \alpha_k^2\phi_k\|\Delta x_k\|_2^2.
\end{aligned}$$

Computing β, by Lemmas 2.3 and 2.4, $\Delta x_k(\beta_k)$ and β_k in (9) are also chosen to ensure (27). For β_k^y chosen as $\beta_k = \arg\max_{\beta\in\mathcal{B}}\{\Delta x_k^t \sum_{y\in Y(x_k)} \beta^y \nabla_x P(x_k, y; c, \mu) +$

$\frac{1}{2}\|\Delta x_k\|^2_{H_k}\}$, we have the corresponding directional derivative

$$\Delta x_k^t \nabla_x \Psi(x_k; c_*, \mu) = \Delta x_k^t \sum_{y\in Y(x_k)} \beta_k^y \nabla_x P(x_k, y; c_*, \mu).$$

From (24), we have

$$\Psi(x_{k+1}; c_*, \mu) \le \Psi(x_k; c_*, \mu) + \alpha_k \Delta x_k^t \sum_{y\in Y(x_k)} \beta_k^y \nabla_x P(x_k, y; c_*, \mu) + \frac{1}{2}\alpha_k^2 \|\Delta x_k\|^2_{H_k} + \alpha_k^2 \phi_k \|\Delta x_k\|_2^2. \tag{28}$$

From assumption (ii) and Lemmas 2.3 and 2.4, we also have

$$\|\Delta x_k\|_2^2 \le \frac{1}{m'} \Delta x_k^t H_k \Delta x_k; \qquad \Delta x_k^t H_k \Delta x_k \le -\Delta x_k^t \nabla \Psi(x_k; c_*, \mu).$$

Thus, using Lemmas 2.3 to 2.6, (26) and (28) can be written as

$$\Psi(x_{k+1}; c_*, \mu) - \Psi(x_k; c_*, \mu) \le \alpha_k \Delta x_k^t \nabla \Psi(x_k; c_*, \mu)\left(1 - \alpha_k \frac{m' + 2\phi_k}{2m'}\right). \tag{29}$$

The scalar ρ in Armijo's rule (16) determines a steplength α_{x_k} such that

$$\rho \le 1 - \alpha_{x_k} \frac{m' + 2\phi_k}{2m'} \le \frac{1}{2}.$$

From Lemmas 2.3 to 2.6 we always have that $\Delta x_k^t \nabla \Psi(x_k; c_*, \mu) \le -\|\Delta x_k\|^2_{H_k} \le 0$, there must exist $\alpha_{x_k} \in (0, 1]$ to ensure (29) and Armijo's rule. Let α^0 be the largest such number. Consequently, for every $\alpha \le \alpha^0$ Armijo's rule and (29) are also satisfied. Therefore, step-length $\alpha_{x_k} \in [\beta\alpha^0, \alpha^0]$ is selected, where $0 < \beta < 1$. From all the said it follows that the sequence $\Psi(x_k; c_*, \mu)$ is monotonically decreasing. □

The direct consequence of the above theorem is that the sequence $\{x_k\}$ is bounded away from zero, which is established in the following corollary.

Corollary 2.1 *The sequence $\{x_k\}$ of primal variables generated by the algorithm, with μ fixed, is bounded away from zero.*

Proof Assume to the contrary that the sequence $\{x_k\} \to 0$. Then $\{-\sum_{i=1}^n \log(x^i)\} \to \infty$. From the assumption that the feasible region is bounded, we conclude that the sequences $\{f(x_k)\}$ and $\{\|g(x_k)\|\}$ are also bounded. Hence, $\{\Psi(x_k; c_*, \mu)\} \to \infty$ which contradicts the monotonic decrease of Ψ. □

The following lemma, proved by Yamashita in [19], shows that the dual stepsize rule, used by the algorithm, generates iterates z_k which are also bounded above and away from zero.

Lemma 2.8 *While μ is fixed, the lower bounds LB_k^i and the upper bounds UB_k^i, $i = 1, 2, \dots, n$, of the box constraints in the dual step size rule, are bounded away from zero and bounded from above respectively, if the corresponding components x_k^i, of the iterates x_k are also bounded above and away from zero.*

Proof The proof can be found in [19]. □

Having established that the sequences of iterates $\{x_k\}$ and $\{z_k\}$ are bounded above and away from zero, it can further be shown that the iterates $\{y_k\}$, $k \geq 0$ are also bounded.

Lemma 2.9 *Let w_k is a sequence of vectors generated by the algorithm for μ fixed. Then, the sequence of vectors $\{(\Delta x_k, y_k + \Delta y_k, \Delta z_k)\}$ is bounded.*

Proof The proof can be found in [20]. □

Lemma 2.10 below establishes the results required in Theorem 2, to demonstrate the convergence of the sequence $\{w_k\}$ to $w_* = (x_*, y_*, z_*)$, satisfying the first order necessary conditions of optimality for (2).

Lemma 2.10 *Let the assumptions of the previous theorem hold and the barrier parameter μ is fixed. Assume also that, for some iteration k_0, the level set $S = \{x \in \mathcal{R}_+^n : \Psi(x; c_*, \mu) \leq \Psi(x_{k_0}; c_*, \mu)\}$ is compact. Then, for all $k \geq k_0$, we have*

$$\lim_{k\to\infty} \max_{\beta\in\mathcal{B}} \Delta x_k^t \sum_{y\in Y(x_k)} (\beta\nabla_x f(x_k, y) + c\nabla g_k^t g_k - \mu X_k^{-1} e) = 0. \tag{30}$$

Proof The scalar $\rho \in (0, 1/2)$ in the stepsize strategy at step 2, determines α_{xk} such that $\rho \leq 1 - \alpha_{xk}(0.5 + \phi_k/m') \leq \frac{1}{2}$, and by solving for α_{xk} we obtain

$$\frac{1/2}{1/2 + \phi_k/m'} \leq \alpha_{xk} \leq \frac{1-\rho}{1/2 + \phi_k/m'}.$$

Hence, the largest value that the step-length α_{xk} can take and still satisfy Armijo's rule in step 2 is $\alpha_{xk}^0 = \min\{1, \frac{1-\rho}{1/2+\phi_k/m'}\}$. Recall that the step-length α_{xk} is chosen by reducing the maximum allowable step-length $\hat{\alpha}_{xk}$ until Armijo's rule is satisfied. Therefore $\alpha_{xk} \in [\beta\alpha_{xk}^0, \alpha_{xk}^0]$ and thereby also satisfies Armijo's rule. As the augmented objective function $P(x, y; c, \mu)$ is twice continuously differentiable and the level set S_1 is bounded, there exists a scalar $\bar{M} < \infty$ such that

$$\phi_k = \int_0^1 (1-t) \left\| \sum_{y\in Y(x_k)} \beta^y \nabla_x^2 P(x_k + t\alpha_{xk}\Delta x_k, y; c_*, \mu) - H_k \right\|_2 dt \leq \bar{M} < \infty.$$

Thus, we always have $\alpha_{xk} \geq \bar{\alpha}_{xk} > 0$, where $\bar{\alpha}_{xk} = \min\{1, \frac{1-\rho}{1/2+\bar{M}/m'}\}$. Hence the step size α_{xk} is always bounded away from zero. Furthermore, from Armijo's rule and Lemmas 2.3 and 2.6 we have

$$\begin{aligned}\Psi(x_{k+1}; c_*, \mu) - \Psi(x_k; c_*, \mu) &\le \rho\alpha_{xk}\nabla\Psi(x_k; c_*, \mu)\\ &= \max_{\beta\in\mathcal{B}} \Delta x_k^t \sum_{y\in Y(x_k)} \beta\nabla_x P(x_k, y; c_*, \mu) < 0.\end{aligned} \tag{31}$$

From our assumption that the level set S is bounded, it can be deduced that

$$\lim_{k\to\infty} |\Psi(x_{k+1}; c_*, \mu) - \Psi(x_k; c_*, \mu)| = 0.$$

Consequently, from (31)

$$\lim_{k\to\infty}\left(\rho\alpha_{xk} \max_{\beta\in\mathcal{B}} \Delta x_k^t \sum_{y\in Y(x_k)} \beta\nabla_x P(x_k, y; c_*, \mu)\right) = 0.$$

Finally, since $\rho, \alpha_{xk} > 0$, it can be deduced that (30) holds. □

A consequence of the above is that, under the same assumptions, the following holds:

$$\lim_{k\to\infty} \|\Delta x_k\|_{H_k}^2 = 0.$$

Theorem 2.2 *Let the assumptions of the previous theorem hold. Then, the algorithm terminates at a point satisfying the first-order necessary conditions of problem* (2) *and at that point the perturbed conditions* (6) *are satisfied for μ fixed.*

Proof Let $\lim_{k\to\infty}(x_k, \lambda_k, z_k) = (x_*(\mu), \lambda_*(\mu), z_*(\mu)), \forall k \ge k_*, k \in K \subseteq \{1, 2, \ldots\}$. The existence of such points is ensured since by Lemmas 2.8 and 2.9, the sequence $\{(x_k(\mu), y_k(\mu), z_k(\mu))\}$ is bounded for μ fixed, and by Theorem 2.1 the algorithm always decreases the merit function Ψ sufficiently at each iteration, thereby ensuring $x_k \in S$, with S compact. First we show that for k sufficiently large the dual step α_{z_k} becomes unity. To this end, we need to show the following:

$$\lim_{k\to\infty} \|z_k + \Delta z_k - \mu X_{k+1}^{-1} e\| = 0.$$

From (7), we have

$$\|z_k + \Delta z_k - \mu X_{k+1}^{-1} e\| \le \|X_k^{-1} Z_k\| \|\Delta x_k\| + \mu \|X_k^{-1} - X_{k+1}^{-1}\| \|e\|. \tag{32}$$

Furthermore, we have $\|X_k^{-1} - X_{k+1}^{-1}\|^2 \le n \max_{1\le i\le n}(\alpha_{x_k}^2 (\Delta x_k^i)^2)/((x_k^i)^2 (x_{k+1}^i)^2)$. Since $\alpha_{x_k} \in (0, 1]$ and $(\Delta x_k^i)^2 \le \|\Delta x_k\|^2$, using the previous theorem yields

$$\lim_{k\to\infty} \|X_k^{-1} - X_{k+1}^{-1}\|^2 \le n \lim_{k\to\infty} \max_{1\le i\le n} \frac{\|\Delta x_k\|^2}{(x_k^i)^2 (x_{k+1}^i)^2}.$$

Therefore, (32) holds and $z_{k+1} = z_k + \Delta z_k$ for k sufficiently large. The complementarity condition becomes $X_{k+1} z_{k+1} = X_{k+1} X_k^{-1}(-Z_k \Delta x_k + \mu e)$. Using the fact $\lim_{k\to\infty}\Delta x_k = 0$, we can derive that: $\lim_{k\to\infty} X_{k+1}X_k^{-1} = I$. Letting $k\to\infty$ yields $\lim_{k\to\infty} X_{k+1}z_{k+1} = X_*(\mu)z_*(\mu) = \mu e$. From (7), it follows that $g(x_*(\mu)) = \lim_{k\to\infty} g(x_k) = \lim_{k\to\infty}\nabla g_k\Delta x_k = 0$. Finally, the first equation of the Newton system (7) gives $\nabla\Phi_k - \nabla g_k^t\lambda_{k+1} - \mu X_k^{-1}e = -(H_k + X_k^{-1}Z_k)\Delta x_k$. Letting $k\to\infty$, the above equation becomes $\lim_{k\to\infty}\|\nabla\Phi_k - \nabla g_k^t\lambda_{k+1} - \mu X_k^{-1}e\| = 0$. From the assumptions that the functions f and g have continuous gradients and ∇g_k^t has full column rank, the above equation becomes $\lim_{k\to\infty}\|\nabla\Phi_{k+1} - \nabla g_{k+1}^t\lambda_{k+1} - \mu X_{k+1}^{-1}e\| = 0$, or $\nabla\Phi(x_*(\mu)) - \nabla g(x_*(\mu))^t\lambda_*(\mu) - \mu X_*(\mu)^{-1}e = 0$. □

Acknowledgements Financial support of EPSRC Grants GR/R51377/01, EP/C513581/1 and GR T02560/01 is gratefully acknowledged. The authors are also grateful to Professor Elijah Polak for valuable suggestions and advice on this paper.

References

1. Rustem, B., Zakovic, S., Parpas, P.: An interior point algorithm for continuous minimax: implementation and computation. Comput. Optim. Appl. (2008, to appear)
2. Womersley, R.S., Fletcher, R.: An algorithm for composite nonsmooth optimization problems. J. Optim. Theory Appl. **48**, 493–523 (1986)
3. Polak, E., Mayne, D.Q., Higgins, E.J.: A superlinearly convergent minimax algorithm for minimax problems. UCB/ERL M86/103, Department of Electrical Engineering, University California, Berkeley, CA (1988)
4. Rustem, B., Nguyen, Q.: An algorithm for inequality constrained discrete minimax. SIAM J. Optim. **8**, 256–283 (1998)
5. Obasanjo, E., Rustem, B.: An interior point algorithm for nonlinear minimax problems (to appear)
6. Demyanov, V.F., Pvnyi, A.B.: Numerical methods for finding saddle points. USSR Comput. Math. Math. Phys. **12**, 1099–1127 (1972)
7. Rustem, B., Howe, M.A.: Algorithms for Worst-case Design with Applications to Risk Management. Princeton University Press, Princeton (2001)
8. Panin, V.M.: Linearization method for continuous min–max problems. Kibernetika **2**, 75–78 (1981)
9. Polak, E.: Optimization Algorithms and Consistent Approximations. Springer, Berlin (1997)
10. Polak, E., Royset, J.O.: Algorithms for finite and semi-infinite min–max–min problems using adaptive smoothing techniques. J. Optim. Theory Appl. **119**, 421–457 (2003)
11. Polak, E., Royset, J.O., Womersley, R.S.: Algorithms with adaptive smoothing for finite minimax problems. J. Optim. Theory Appl. **119**, 459–484 (2003)
12. Sasai, H.: An interior penalty method for minimax problems with constraints. SIAM J. Control Optim. **12**, 643–649 (1974)
13. Stein, O., Still, G.: Solving semi-infinite optimization problems with interior point techniques. SIAM J. Control Optim. **42**, 769–788 (2003)
14. Rockafellar, R.T.: Convex Analysis. Princeton University Press, Princeton (1970)
15. Powell, M.J.D.: A fast algorithm for nonlinearly constrained optimization calculations. In: Lecture Notes in Mathematics, vol. 630, pp. 144–157. Springer, Berlin (1978)
16. Hogan, W.W.: Point-to-set maps in mathematical programming. SIAM Rev. **15**, 591–603 (1973)
17. Parpas, P., Rustem, B.: An algorithm for the global optimization of a class of continuous minimax problems. J. Optim. Theory Appl. (2008, to appear)
18. El-Bakry, A.S., Tapia, R.A., Tsuchiya, T., Zhang, Y.: On the formulation and theory of the Newton interior point method for nonlinear programming. J. Optim. Theory Appl. **89**, 507–541 (1996)
19. Yamashita, H.: 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)
20. Akrotirianakis, I., Rustem, B.: A globally convergent interior point algorithm for general non-linear programming problems. J. Optim. Theory Appl. **125**, 497–521 (2005)