COMISEF Computational Optimization Methods in Statistics, Econometrics and Finance

- Marie Curie Research and Training Network funded by the EU Commission through MRTN-CT-2006-034270 -

COMISEF WORKING PAPERS SERIES

WPS-019 06/11/2009

An Interior-Point algorithm for Nonlinear Minimax Problems

E. Obasanjo
G. Tzallas-Regas
B. Rustem

www.comisef.eu

An Interior-Point algorithm for Nonlinear Minimax Problems

E. Obasanjo[*], G. Tzallas-Regas[†], B. Rustem[‡]

[*] Equity Derivatives Group – Systematic Trading, Barclays Capital, London, UK (Enitan.Obasanjo@barclayscapital.com)

[†] Research Associate, Department of Computing, Imperial College London, London SW7 2AZ, UK (gt299@doc.ic.ac.uk)

[‡] Professor, Department of Computing, Imperial College London, London SW7 2AZ, UK (br@doc.ic.ac.uk)

Abstract. We present a primal-dual interior-point method for constrained nonlinear, discrete minimax problems where the objective functions and constraints are not necessarily convex. The algorithm uses two merit functions to ensure progress toward the points satisfying the first-order optimality conditions of the original problem. Convergence properties are described and numerical results provided.

Key Words. Discrete min-max, Constrained nonlinear programming, Primal-dual interior-point methods, Stepsize strategies.

1 Introduction

A number of interior-point algorithms for linear and nonlinear programming have been developed in the years following the introduction of Karmarkar's method in Ref. 1. El-Bakry *et al.* provided the formulation and theory of the Newton interior-point method for nonlinear programming in Ref. 2, extending the general primal-dual framework proposed for LP problems by Kojima *et al.* (Ref. 3). Their algorithm used the l_2 norm of the residuals as merit function in the line-search procedure. This choice of merit function, however may lead the algorithm to converge to a saddle point or maximum when a minimum is being sought. Facilitated by advances in computing, the development of interior-point methods for nonlinear problems has been of great interest over the last two decades. In light of the success of some interior-point methods to this class of problems (see e.g. Ref. 4), the research presented in this paper aimed to develop an efficient and robust algorithm for general nonlinear minimax problems (Refs. 5, 6) using a primal-dual interior-point method. The resulting algorithm is presented in this paper.

The algorithm is based on the primal-dual interior-point method described in Ref. 7 and the minimax approach of Rustem (Ref. 8). It differs from Ref. 8 mainly in the choice of merit function, stepsize rule and computation of search direction. The objective and constraint functions of the optimization problem are assumed to be smooth and twice differentiable. Newton's method is used to solve the primal-dual system of the problem, generating descent directions for the merit function. The use of slack variables in the inequality constraints relaxes the requirement of feasibility at each iteration. An adaptive penalty parameter is employed to ensure progress towards optimality and feasibility. Two merit functions are used to ensure convergence towards a solution of the optimization problem.

The algorithm code is linked with the mathematical programming language `AMPL`, which provides first and second order derivative information, but the approach is also applicable to quasi-Newton methods. Global and local convergence results are established for the algorithm. The performance of the method is illustrated through some numerical examples, and is compared with the results obtained using `MINOS`, `LOQO` and `IPOPT` on the `NEOS` server 4.0 on problems drawn from the `CUTE` test set.

2 Description of Algorithm

In this section we describe the algorithm we are going to present. First we discuss a reformulation of the minimax problem and then propose how to solve the reformulation with a primal-dual interior-point framework.

2.1 Minimax Problem

We consider the following equality constrained minimax problem:

$$\min_{x \in R^n} \max_{j=1,\ldots,m} f^j(x) \text{ subject to } g(x) = 0, x \geq 0, \tag{1}$$

where $f^j : R^n \to R, j = 1, \ldots, m$ and $g : R^n \to R^q$ are smooth, twice differentiable, nonlinear functions.

It has been shown in Ref. 8, from the original proof Medanić and Andjelić (Refs. 9, 10), that problem (1) is equivalent to the following min-max formulation with pooled objectives

$$\min_{x \in R^n} \max_{\alpha \in \Xi^m_+} \{\langle \alpha, f(x) \rangle \,|g(x) = 0, x \geq 0\}, \tag{2}$$

where

$$\Xi^m_+ = \{\alpha \in R^m | \alpha \geq 0, \langle \mathbf{e}, \alpha \rangle = 1\}, \tag{3}$$

$\mathbf{e} \in R^m$ is a vector of all ones, and $f : R^n \to R^m$ with

$$f(x) = [f_1(x), \ldots, f_m(x)]^T.$$

This follows from the fact that the maximum of m numbers is equal to the maximum of their convex combination.

By introducing an additional variable, v, the discrete minimax problem can be represented as the nonlinear program (NLP)

$$\min_{x \in R^n, v \in R^1} \{v | f(x) \leq v\mathbf{e}, g(x) = 0, x \geq 0\}. \tag{4}$$

To enable comparison with other algorithms for nonlinear programming, the minimax problems discussed in this paper are implemented using the above NLP. However, both the equivalent formulations (2) and (4) are used in this paper to represent problem (1) whenever convenient.

2.2 Primal-Dual Method

The use of slack variables, as was done in Ref. 11, yields the following reformulation of the minimax problem (4):

$$\min_{x \in R^n, v \in R^1, s \in R^m} \{v | f(x) - v\mathbf{e} + s = 0, g(x) = 0, x \geq 0, s \geq 0\}. \tag{5}$$

The equality constraints can be combined to form a new constraint vector $G : R^{n+m+1} \rightarrow R^{m+q}$ defined as

$$G(v, x, s) = \left(\begin{array}{c} f(x) + s - v\mathbf{e} \\ g(x) \end{array} \right). \tag{6}$$

For notational compactness, the decision variables x and slack variables s can be combined into a single variable:

$$\bar{x} = \left(\begin{array}{c} x \\ s \end{array} \right),$$

where $\bar{x} \in R^N$ and $N = n + m$. Thus, we can re-write problem (5) as:

$$\min_{\bar{x} \in R^N, v \in R^1} \{v | G(v, \bar{x}) = 0, \bar{x} \geq 0\}. \tag{7}$$

The inequality constraints in (7) above can then be incorporated into a barrier term

$$B(\bar{x}) = \sum_{i=1}^{N} \log \bar{x}^i,$$

leading to the problem:

$$\min_{\bar{x} \in R^N, v \in R^1} \{v - \mu B(\bar{x}) \,| G(v, \bar{x}) = 0\} \tag{8}$$

for $\bar{x} > 0$ with barrier parameter $\mu \geq$0.

The Lagrangian function associated with the constrained optimization problem in (8) is given by

$$L(v, \bar{x}, y; \mu) = v + \langle y, G(v, \bar{x}) \rangle - \mu B(\bar{x}), \tag{9}$$

where $y \in R^p$, $p = q + m$, are the Lagrange multipliers of the equality constraints.

The Karush-Kuhn-Tucker (KKT) optimality conditions for this problem are

$$\begin{array}{rclcl} \nabla_v L(v,\bar{x},y;\mu) & = & 1+\langle y, \nabla_v G(v,\bar{x})\rangle & = & 0, \\ \nabla_{\bar{x}} L(v,\bar{x},y;\mu) & = & \nabla_{\bar{x}} G(v,\bar{x})^T y - \mu \bar{X}^{-1}\mathbf{e} & = & 0, \\ \nabla_y L(v,\bar{x},y;\mu) & = & G(v,\bar{x}) & = & 0, \end{array} \tag{10}$$

where $\bar{X} = \operatorname{diag}(\bar{x})$, is the diagonal matrix with elements $\bar{x}^i$, and $\nabla_a G(v,\bar{x})$ is the Jacobian matrix of $G(v,\bar{x})$ with respect to variable a.

By introducing the nonlinear transformation $z = \mu \bar{X}^{-1}\mathbf{e}$ we modify Eqs. (10) to produce the standard primal-dual system:

$$E(v,\bar{x},z,y;\mu) = \begin{pmatrix} 1+\langle y, \nabla_v G(v,\bar{x})\rangle \\ \nabla_{\bar{x}} G(v,\bar{x})^T y - z \\ \bar{X}Z\mathbf{e} - \mu\mathbf{e} \\ G(v,\bar{x}) \end{pmatrix} = 0, \tag{11}$$

where $Z = \operatorname{diag}(z)$, with $(\bar{x},z) > 0$.

2.3 Newton System and Search Direction

Primal-dual methods use the Newton or quasi-Newton method to solve approximately the perturbed KKT conditions for a fixed value of μ. The Newton system for (11) above is:

$$\begin{bmatrix} 0 & 0 & 0 & -\mathbf{e}^T \\ 0 & \nabla^2_{\bar{x}\bar{x}} L(v,\bar{x},y;\mu) & -I & \nabla_{\bar{x}} G(v,\bar{x})^T \\ 0 & Z & \bar{X} & 0 \\ -\mathbf{e} & \nabla_{\bar{x}} G(v,\bar{x}) & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta v \\ \Delta \bar{x} \\ \Delta z \\ \Delta y \end{bmatrix} = -E(v,\bar{x},z,y;\mu), \tag{12}$$

since

$$\nabla^2_{vv} L(v,\bar{x},y;\mu) = 0, \tag{13}$$

and

$$\nabla_{v\bar{x}} L(v,\bar{x},y;\mu) = \nabla_{\bar{x}v} L(v,\bar{x},y;\mu) = 0. \tag{14}$$

Let $w^p = (v,\bar{x})$ represent the primal variables and $w = (v,\bar{x},z,y)$ all the variables of problem (7). System (12) is then solved to give the search directions $\Delta w = (\Delta v, \Delta\bar{x}, \Delta z, \Delta y)$. The algorithm proceeds iteratively from an initial point $(v_0, \bar{x}_0, z_0, y_0)$ through a sequence $w_{k+1} = w_k + \tau_k \Delta w_k$, where

$$\tau_k = \operatorname{diag}(\tau_{pk} I_v, \tau_{pk} I_{\bar{x}}, \tau_{dk} I_z, \tau_{dk} I_y),$$

with τ_{pk} being the primal stepsize, τ_{dk} the dual stepsize and I_a being a unity matrix of dimension related to the length of a. System (12) can be expressed as:

$$\nabla E(w;\mu)\Delta w = -E(w;\mu). \tag{15}$$

The Newton system (12) can be re-written as:

$$\begin{aligned}
-\langle \mathbf{e}, \Delta y\rangle &= -\left[1 - \langle y, \mathbf{e}\rangle\right] && \text{(16a)}\\
Q\Delta\bar{x} + \nabla_{\bar{x}} G(v,\bar{x})^T \Delta y - \Delta z &= -[\nabla_{\bar{x}} G(v,\bar{x})^T y - z] && \text{(16b)}\\
Z\Delta\bar{x} + \bar{X}\Delta z &= -\left(\bar{X}Z\mathbf{e} - \mu\mathbf{e}\right) && \text{(16c)}\\
-\Delta v\mathbf{e} + \nabla_{\bar{x}} G(v,\bar{x})\Delta\bar{x} &= -G(v,\bar{x}) && \text{(16d)}
\end{aligned}$$

where Q is the Hessian of the Lagrangian function, $L(v,\bar{x},y;\mu)$, or an approximation of the latter.

Remark 2.1 Due to Eqs. (13), (14) there holds

$$\langle \Delta w^p, \nabla^2_{w^p} L(v,\bar{x},y;\mu)\Delta w^p\rangle = \langle \Delta\bar{x}, \nabla^2_{\bar{x}} L(v,\bar{x},y;\mu)\Delta\bar{x}\rangle = \langle \Delta\bar{x}, Q\Delta\bar{x}\rangle.$$

Remark 2.2 From the definition of w^p, Eq. (16d) can be expressed as:

$$\nabla_{w^p} G(w^p)\Delta w^p = -G(w^p),$$

where

$$\nabla_{w^p} G(w^p) = \left[\begin{array}{cc} \nabla_v G(v,\bar{x}) & \nabla_{\bar{x}} G(v,\bar{x}) \end{array}\right],$$

and $\Delta w^p = (\Delta v, \Delta\bar{x})$.

The design of the algorithm used to solve problem (8) is based on the following assumptions:

(A1) Smoothness: $G(v,\bar{x})$ is a smooth twice continuously differentiable function of v and $\bar{x}$.

(A2) Regularity : The columns of the matrix $[\nabla G(v,\bar{x})]$ corresponding to the binding constraints are linearly independent.

(A3) Complementarity: At a solution point, $w_* = (v_*, \bar{x}_*, z_*, y_*)$, strict complementarity is satisfied, *i.e.* if $\bar{x}^i_* > 0$ then $z^i_* = 0$ and if $z^i_* > 0$, $\bar{x}^i_* = 0$ for $i = 1, 2, \ldots, N$.

(A4) Second order sufficiency: The second order sufficiency condition is satisfied at a solution point.

(A5) Feasibility: The feasible region is bounded[1].

(A6) There exists a constant $m_1 > 0$ such that

$$\|Q_k\| \leq m_1, \qquad \text{for all } k.$$

(A7) There exists a constant $m_2 > 0$ such that

$$m_2 \|p\|^2 \leq \langle p, Q_k p \rangle$$

for all k and all $0 \neq p \in R^N$ such that $\nabla_{\bar{x}} G(w_k^p) p = 0$.

2.4 Merit Functions

In this section we describe the merit functions used for determining the stepsize of the algorithm.

Let $\epsilon_G > 0$ be a finite precision to which the equality constraints are satisfied, *i.e.* we have a worst-case feasibility precision

$$\|G(w_k^p)\|_2^2 > \epsilon_G. \tag{17}$$

While (17) holds, the inner iterations k use the following merit function

$$\Phi_1(w^p; c, \mu) = v + (c/2) \|G(w^p)\|_2^2 - \mu B(\bar{x}). \tag{18}$$

For any inner iteration, k, where $0 < \|G_k\|_2^2 \leq \epsilon_G$, the algorithm switches to the second merit function,

$$\Phi_2(w) = (1/2) \|E(w; 0)\|_2^2 , \tag{19}$$

for all consecutive inner iterations, where $E(w; 0)$ is as defined in (11) with $\mu = 0$. Once the convergence of the inner iterations is achieved, the algorithm switches back to the original merit function.

[1] This may be enforced, if required, in the usual way of placing upper and lower bounds on $\bar{x}$.

2.5 Penalty Parameter Selection Rule

The penalty parameter selection strategy is an important part of the algorithm. At each iteration for which (17) holds, a value of c is determined such that the descent property

$$\Delta v_k - \mu \Delta \bar{x}_k^T \bar{X}_k^{-1} \mathbf{e} - c_k \left\| G(v_k, \bar{x}_k) \right\|_2^2 + d_k(\Delta \bar{x}_k) \leq 0, \tag{20}$$

associated with Φ_1 holds. The monotonic descent of function Φ_1 is discussed in Theorem 3.1. Inequality (20) plays an important role to the proof of Theorem 3.1. Term $d_k(\Delta \bar{x}_k)$ is a function of the search direction and should have the property that

$$d_k(\Delta \bar{x}_k) \geq 0 \tag{21}$$

in order to prove Theorem 3.1 and generally establish global convergence of our algorithm. The exact choice of $d_k(\Delta \bar{x}_k)$ depends on the framework used. For example, in a quasi-Newton framework exploiting hereditary positive-definiteness of Hessian updates the natural choice would be

$$d_k(\Delta \bar{x}_k) = \langle \Delta \bar{x}_k, Q_k \Delta \bar{x}_k \rangle. \tag{22}$$

In such frameworks Q_k is chosen positive semi-definite, therefore (21) is satisfied. In a Newton framework such a choice for $d_k(\Delta \bar{x}_k)$ is not entirely sufficient. In our algorithm we choose

$$d_k(\Delta \bar{x}_k) = \left\{ \begin{array}{ll} \langle \Delta \bar{x}_k, Q_k \Delta \bar{x}_k \rangle, & \text{if } G_k = 0 \\ \sigma_k \left\| \Delta \bar{x}_k \right\|^2, & \text{otherwise} \end{array} \right. , \tag{23}$$

where $\sigma_k \in (0, 1/2]$ can be constant throughout the algorithm, or can be updated from iteration to iteration[1].

Remark 2.3 In order for $d_k(\Delta \bar{x}_k)$, as chosen by (23), to satisfy property (21), then when $G_k = 0$, there must hold that $\langle \Delta \bar{x}_k, Q_k \Delta \bar{x}_k \rangle \geq 0$. If we take into account (16d), then $\langle \Delta \bar{x}_k, Q_k \Delta \bar{x}_k \rangle \geq 0$ if Assumption A7 is satisfied.

Since μ is fixed throughout the inner iterations, it can be deduced that the sign of the term on the left hand side of (20) is dependent on the value of c. For insufficiently large values of c, the descent property (20) may not be satisfied. When this is the case, the penalty parameter is updated to a new

[1] We do not present an iteration scheme for σ_k, and consider it a constant positive value.

value $c_{k+1} > c_k$ which guarantees the satisfaction of (20). Lemma 3.2 shows that the algorithm chooses the value of the penalty parameter to satisfy (20). In Lemmas 3.2 and 3.4 we show that descent is always guaranteed if (17) holds or if $G(v_k, \bar{x}_k) = 0$, and that the penalty parameter $c_k = c_k(\epsilon_G)$ remains finite, due to the use of two merit functions. This scheme is a variation of the *watchdog* technique, first suggested by Chamberlain *et al.* (Ref. 12), and used in the context of interior-point methods by Gay *et al.* (Ref. 13). The convergence property of an algorithm based on the merit function (19) is discussed in Ref. 2. The algorithm in Ref. 2 has two salient features that need to be noted. The first is the choice of penalty parameter selection rule for (18) which extends the sequential quadratic programming approach proposed in (Refs. 14, 15) for nonlinear programming and in Rustem and Nguyen (Ref. 16) for minimax. The second, and perhaps more important, aspect is the objective and method of application of the second merit function, defined in (19). We illustrate, through an example in Section 4, that the l_2 norm of the KKT conditions on its own is unsuitable for use as a merit function for non-convex optimization, but this is overcome with a combination of merit functions as defined in (18) and (19).

2.6 Line Search Strategies

The algorithm utilizes an Armijo line search strategy to determine stepsizes in the search direction. The search strategy should decrease the value of the merit function chosen by the algorithm.

When Φ_1 is in use, then for , $\hat{\tau} > 0, \beta \in (0, 1)$ the stepsize strategy in the algorithm determines the stepsize τ_{pk} as the largest value of $\tau = \hat{\tau}\beta^j, j = 0, 1, 2, \ldots$ such that w^p_{k+1} given by $\bar{x}_{k+1} = \bar{x}_k + \tau_{pk}\Delta\bar{x}_k, v_{k+1} = \max_j f^j(\bar{x}_{k+1})$ satisfies the inequality

$$\Phi_1(w^p_{k+1}; c_{k+1}, \mu) - \Phi_1(w^p_k; c_k, \mu) \leq \rho\tau_{pk}\phi_1(w^p_k, \Delta w^p_k; c_{k+1}, \mu), \qquad (24)$$

where $\rho \in (0, 1/2)$ is a given scalar and

$$\phi_1(w^p, \Delta w^p; c, \mu) = \langle \Delta w^p, \nabla_{w^p}\Phi_1(w^p; c, \mu)\rangle \qquad (25)$$

is the directional derivative of Φ_1 at w^p_k along the primal search direction Δw^p_k. At iteration k, using the definition of Φ_1 from (18) and Remark 2.2 we can write ϕ_1 as

$$\phi_1(w^p_k, \Delta w^p_k; c_k, \mu) = \Delta v_k - \mu\Delta\bar{x}^T_k\bar{X}^{-1}_k\mathbf{e} - c_{k+1}\|G_k\|^2. \qquad (26)$$

When Φ_2 is in use, then for $\hat{\tau} > 0, \beta \in (0,1)$ the stepsize strategy in the algorithm determines the stepsize τ_k as the largest value of $\tau = \hat{\tau}\beta^j, j = 0, 1, \ldots$, such that w_{k+1} given by

$$w_{k+1} = w_k + \tau_k \Delta w_k$$

satisfies the inequality

$$\Phi_2(w_k + \tau_k \Delta w_k) - \Phi_2(w_k) \leq \rho \tau_k \phi_2(w_k, \Delta w_k). \qquad (27)$$

where, again, ϕ_2 is the directional derivative of Φ_2 at w_k along the search direction Δw_k. At iteration k we can write ϕ_2, using (15), as

$$\phi_2(w_k, \Delta w_k) = \langle \Delta w_k, \nabla_w \Phi_2(w_k) \rangle = \langle \nabla E^T(w_k; 0) E(w_k; 0), \Delta w_k \rangle. \qquad (28)$$

Algorithm 1 : Primal-Dual Interior-Point algorithm

```
Step 0: Initialization:
        Given $\tilde{\bar{x}}^0 \in R^N, \tilde{\bar{x}}^0 > 0$.
        Choose $\tilde{y}^0 \in R^p, \tilde{z}^0 \in R^N, \tilde{z}^0 > 0$.
        Choose penalty and barrier parameters $c_0 > 0$, $\mu^0 > 0$.
        Choose $\beta, \epsilon_0, \epsilon_G, \eta, \theta \in (0,1), \rho \in (0, 1/2)$, $\delta, M_0, \omega > 0, \sigma_k = 1$.
        Set $\tilde{v}^0 = \max_j f^j(\tilde{\bar{x}}^0)$.
        Set $l = 0$, $k = 0$ and new–merit = false.

Step 1: Convergence of outer iterations:
        If $\left\| E(\tilde{v}^l, \tilde{\bar{x}}^l, \tilde{z}^l, \tilde{y}^l; \mu^l) \right\|_2 / (1 + \left\| (\tilde{v}^l, \tilde{\bar{x}}^l, \tilde{z}^l, \tilde{y}^l) \right\|_2) \leq \epsilon_0$ then STOP.

Step 2: Inner iterations: ($\mu$ is fixed to $\mu^l$ throughout this step)
        Set $(v_k, \bar{x}_k, z_k, y_k) = (\tilde{v}^l, \tilde{\bar{x}}^l, \tilde{z}^l, \tilde{y}^l)$

  Step 2.1: Convergence of inner iterations:
            If $\left\| E(v_k, \bar{x}_k, z_k, y_k; \mu^l \right\|_2 \leq \eta \mu^l$ and $\|G_k\|_2^2 \leq \epsilon_G$ then set
            $(\tilde{v}^{l+1}, \tilde{\bar{x}}^{l+1}, \tilde{z}^{l+1}, \tilde{y}^{l+1}) = (v_k, \bar{x}_k, z_k, y_k)$ and go to Step 3.

  Step 2.2: Search direction generation:
            Solve Newton system (12) to obtain $\Delta w_k = (\Delta v_k, \Delta \bar{x}_k, \Delta z_k, \Delta y_k)$.

  Step 2.3: Penalty parameter selection:
            Compute $d_k(\Delta \bar{x}_k)$ using (23).
```

Set $c_{k+1} = c_k$.
If new–merit is false then set

$$\mathcal{M}_{num} = \Delta v_k - \mu \Delta \bar{x}_k^T \bar{X}_k^{-1} \mathbf{e} - c_k \|G_k\|^2 + d_k(\Delta \bar{x}_k)$$

else go to Step 2.4
If new–merit is false and $\mathcal{M}_{num} > 0$ and $\|G_k\|_2^2 > \epsilon_G$ then set

$$c_{k+1} = \max\left\{ \left(\Delta v_k - \mu \Delta \bar{x}_k^T \bar{X}_k^{-1} \mathbf{e} + d_k(\Delta \bar{x}_k) \right) / \|G_k\|^2, c_k + \delta \right\}$$

Step 2.4: Step-length selection:
Set $\tau_{pk}^{max} = \min\limits_{1 \le i \le N} \left\{ -\bar{x}_k^i / \Delta \bar{x}_k^i \,\middle|\, \Delta \bar{x}_k^i < 0 \right\}$.
If ($\mathcal{M}_{num} > 0$ and $0 < \|G_k\|_2^2 \le \epsilon_G$) or new–merit is true then
    Set $\tau_{dk}^{max} = \min\limits_{1 \le i \le N} \left\{ -z_k^i / \Delta z_k^i \,\middle|\, \Delta z_k^i < 0 \right\}$.
    Set $\hat{\tau}_k = \min\{1, \theta \tau_{pk}^{\max}, \theta \tau_{dk}^{\max}\}$.
    Let $\tau_k = \beta^j \hat{\tau}_k$, where j is the smallest non-negative integer:

$$\Phi_2(w_k + \tau_k \Delta w_k) - \Phi_2(w_k) \le \rho \tau_k \phi_2(w_k, \Delta w_k).$$

    Set new–merit = true.

else

    Set $\hat{\tau}_{pk} = \min\{1, \theta \tau_{pk}^{\max}\}$.
    Let $\tau_{pk} = \beta^j \hat{\tau}_{pk}$, where j is the smallest non-negative integer:

$$\Phi_1(w_{k+1}^p; c_{k+1}, \mu) - \Phi_1(w_k^p; c_{k+1}, \mu) \le \rho\, \tau_{pk} \phi_1(w_k^p, \Delta w_k^p; c_{k+1}, \mu)$$

    with $\bar{x}_{k+1} = \bar{x}_k + \tau_{pk} \Delta \bar{x}_k$ and $v_{k+1} = \max_j f^j(x_{k+1})$.
    Choose $\hat{m}, \hat{M} > 0$.
    Set

$$\begin{aligned} LB_k^i &= \min\left\{ (1/2) \hat{m} \mu, \bar{x}_{k+1}^i z_k^i \right\} \\ UB_k^i &= \min\left\{ 2 \hat{M} \mu, \bar{x}_{k+1}^i z_k^i \right\} \end{aligned}$$

    Set for $i = 1, \dots N$

$$\tau_{dk}^i = \max\left\{ \tau_i > 0 : LB_k^i \le \bar{x}_{k+1}^i \left(z_k^i + \tau_i \Delta z_k^i \right) \le UB_k^i \right\}.$$

Set $\tau_{dk} = \min\left\{1, \min_{1\leq i\leq N}\left\{\tau^i_{dk}\right\}\right\}$.

Set

$$y_{k+1} = y_k + \tau_{dk}\Delta y_k, \qquad z_{k+1} = z_k + \tau_{dk}\Delta z_k.$$

Step 2.5: Start a new inner iteration:
Set $k = k + 1$ and go to Step 2.1.

Step 3: Barrier parameter adjustment:
Reduce the barrier parameter μ as described in Section 2.8.

Step 4: Start a new outer iteration:
Set new–merit = false
Set $l = l + 1$ and go to Step 1.

2.7 Stepsize Rules

In this section we discuss the step size rules associated with Φ_1. A treatment of the step size rules associated with Φ_2 can be found in Ref. 2.

Primal Stepsize Rule. For the primal variables, (x, s), represented by $\bar{x}$, we determine the maximum allowable stepsizes by the boundary of the feasible region:

$$\tau^{max}_{pk} = \min_{1\leq i\leq N}\left\{-\bar{x}^i_k/\Delta\bar{x}^i_k \,\middle|\, \Delta\bar{x}^i_k < 0\right\}.$$

This is indeed the maximum step allowed because τ^{max}_{pk} gives an infinitely large value to at least one term of the logarithmic barrier function $B(\bar{x}_{k+1})$. However, if the stepsize is in the interval $[0, \tau^{max}_{pk})$ then the next primal iterate $\bar{x}_{k+1}$ is strictly feasible and none of the logarithmic terms become infinitely large. As an initial step, $\hat{\tau}_{pk}$, we choose a number close to τ^{max}_{pk} but never greater than one, *i.e.* $\hat{\tau}_{pk} = \min[\theta\tau^{max}_{pk}, 1]$. with $\theta \in (0, 1)$. The Armijo rule is then applied to give $\tau_{pk} = \beta^j\hat{\tau}_{pk}$, with β chosen to be in the interval $[0.1, 0.5]$, dependent on the confidence on the initial step $\hat{\tau}_{pk}$.

Dual Stepsize Rule. The dual stepsize strategy used by the algorithm follows that described in Ref. 7. It is a modification of the strategy suggested by Yamashita (Ref. 17) and Yamashita and Yabe (Ref. 18).

For a fixed value of the barrier parameter μ, we determine a stepsize τ^i_{dk} along the direction Δz^i_k for each dual variable $z^i_k, i = 1, 2, \ldots, N$, such that the following box constraints are satisfied:

$$\tau^i_{dk} = \max\left\{\tau_i > 0 : LB^i_k \leq \bar{x}^i_{k+1}\left(z^i_k + \tau_i \Delta z^i_k\right) \leq UB^i_k\right\} \tag{29}$$

with LB^i_k and UB^i_k defined, as in Step 2.4 of the algorithm, as

$$\begin{aligned} LB^i_k &= \min\{(1/2)\hat{m}\mu, \bar{x}^i_{k+1} z^i_k\} \\ UB^i_k &= \min\{2\hat{M}\mu, \left(\bar{x}^i_{k+1}\right)\left(z^i_k\right)\}. \end{aligned}$$

The parameters $\hat{m}$ and $\hat{M}$, as proposed in Ref. 19, are chosen to satisfy

$$0 < \hat{m} \leq \min\left\{1, \left((1-\eta)(1-\eta/M_0^{\mu^l}) \min_i\{(\tilde{\bar{x}}^l)^i(\tilde{z}^l)^i\}\right)/\mu^l\right\} \tag{30}$$

and

$$\hat{M} \geq \max\left\{1, \max_i\{(\tilde{\bar{x}}^l)^i(\tilde{z}^l)^i\}/\mu^l\right\} > 0, \tag{31}$$

for a fixed value of μ^l, with $\eta \in (0, 1)$ and M_0 a large positive number.

The common dual step τ_{dk}, is the minimum of all individual step-lengths τ^i_{dk} with the restriction of always being no greater than one. Numerically:

$$\tau_{dk} = \min\left\{1, \min_{1\leq i\leq N}\left\{\tau^i_{dk}\right\}\right\}.$$

The step-lengths for the dual variables y can be set to either τ_{dk} or 1. It is clear from the construction of the lower and upper bounds that

$$LB^i_k / \left(\bar{x}^i_k + \tau_{pk}\Delta\bar{x}^i_k\right) \leq z^i_k \leq UB^i_k / \left(\bar{x}^i_k + \tau_{pk}\Delta\bar{x}^i_k\right)$$

for all $i = 1, \ldots, N$.

The dual stepsize rule defines a neighborhood of the central path beyond which the iterates cannot stray. It does so by setting a lower bound which prevents the products $\bar{x}^i_k z^i_k$ from becoming very much smaller than the current value of the barrier parameter, preventing them reaching the boundary of the feasible region prematurely. The dual stepsize rule also prevents the variables $\bar{x}^i_k$ and z^i_k from becoming unbounded by setting an upper bound on them that is proportional to the barrier parameter.

2.8 Selection of Barrier Parameter

The selection of the barrier parameter follows the strategy outlined in Ref. 7. It is restated here for convenience

Set $\mu^{l+1} = \min\{0.95\mu^l, 0.01(0.95)^k \|E(w_k; 0)\|_2\}$

if $\left\|E(w_k; \mu^l)\right\|_2 \leq 0.1\eta\mu^l$ and $\mu^l < 10^{-4}$ the set

$$\mu^{l+1} = \min\{0.95\mu^l, 0.01(0.95)^{k+2\omega} \|E(w_k; 0)\|_2\}.$$

if $\left\|E(w_k; \mu^l)\right\|_2 \leq 0.1\eta\mu^l$ and $\mu^l \geq 10^{-4}$ the set

$$\mu^{l+1} = \min\{0.95\mu^l, 0.01(0.95)^{k+\omega} \|E(w_k; 0)\|_2\}.$$

In the above, $\omega > 0$ is a parameter which accelerates the decrease of μ at appropriate points. The algorithm checks if the current point is close enough to the central path (*i.e.* if $\left\|E(w_k; \mu^l)\right\|_2 \leq 0.1\eta\mu^l$) and to the optimum solution (i.e if $\mu^l < 10^{-4}$), in which case the barrier parameter is multiplied by $(0.95)^{2\omega}$, reducing it at a faster rate. If the current point is close to the central path but not an optimum solution, the barrier parameter is multiplied by the larger factor $(0.95)^{\omega}$.

3 Convergence Results

In this section the algorithm is shown to be globally convergent, in the sense that it guarantees progress towards a solution of the approximate KKT conditions (11) from any starting point and for a fixed value of the barrier parameter $\mu^l = \mu$.

Lemma 3.1 *Let f and g be differentiable functions and let there exist a small $\epsilon_G > 0$ such that $||G_k||_2^2 > \epsilon_G$. If $\Delta\bar{x}_k$ is the search direction for variable $\bar{x}$ calculated by the solution of the Newton system* (12)*, c_{k+1} is chosen as in Step 2.4 of the algorithm above and $d_k(\Delta\bar{x})$ satisfies* (21)*, then $\Delta\bar{x}_k$ is a descent direction for Φ_1. Furthermore*

$$\phi_1(w_k^p, \Delta w_k^p; c_{k+1}, \mu) \leq -d_k(\Delta\bar{x}_k) \leq 0.$$

Proof. Step 2.3 of the algorithm checks if

$$\Delta v_k - \mu \Delta \bar{x}_k^T \bar{X}_k^{-1} \mathbf{e} - c_k \|G_k\|^2 + d_k(\Delta \bar{x}_k) \leq 0. \tag{32}$$

If this is satisfied then by setting $c_{k+1} = c_k$ and rearranging (32) we obtain

$$\phi_1(w_k^p, \Delta w_k^p; c_{k+1}, \mu) \leq -d_k(\Delta \bar{x}_k) \leq 0. \tag{33}$$

On the other hand, if (32) is not satisfied, then by setting

$$c_{k+1} \geq \left(\Delta v_k - \mu \Delta \bar{x}_k^T \bar{X}_k^{-1} \mathbf{e} + d_k(\Delta \bar{x}_k)\right) / \|G_k\|^2$$

and substituting into (26), it can be verified from (21) that

$$\phi_1(w_k^p, \Delta w_k^p; c_{k+1}, \mu) \leq -d_k(\Delta \bar{x}_k) \leq 0.$$

This proves the lemma. ∎

The above lemma establishes the descent property of the search direction for $\|G_k\|_2^2 > \epsilon_G$. We now proceed to demonstrate the descent property for the merit function when $\|G_k\|_2^2 = 0$. We stress that assumption A7 is needed in order to prove descent for Φ_1 in this lemma only.

Lemma 3.2 *Let f and g be differentiable functions and also let A7 hold. If $(\Delta v_k, \Delta \bar{x}_k, \Delta y_k, \Delta z_k)$ is the solution to the Newton system (12) and $G_k = 0$ for some k, then the descent property*

$$\phi_1(w_k^p, \Delta w_k^p; c_k, \mu) \leq -d_k(\Delta \bar{x}_k) \leq 0$$

is satisfied for any choice of the penalty parameter, $c_k \in [0, \infty)$. Therefore the search direction $\Delta \bar{x}_k$ obtained from the solution of the Newton system of equations is still a descent direction after feasibility of the equality constraints has been achieved.

Proof. If $G_k = 0$, (26) can be written as:

$$\phi_1(w_k^p, \Delta w_k^p; c_{k+1}, \mu) = \Delta v_k - \mu \Delta \bar{x}_k^T \bar{X}_k^{-1} \mathbf{e}. \tag{34}$$

We can also write (16d) as:

$$-\Delta v_k \mathbf{e} + \nabla_{\bar{x}} G_k \Delta \bar{x}_k = 0.$$

Multiplying the last one by $(y_k + \Delta y_k)^T$ and using (16a) we get

$$\Delta v_k = \langle \Delta \bar{x}_k, \nabla_{\bar{x}} G_k^T (y_k + \Delta y_k) \rangle. \tag{35}$$

Rearranging (16b) we get

$$\nabla_{\bar{x}} G_k^T (y_k + \Delta y_k) = -Q_k \Delta \bar{x}_k + z_k + \Delta z_k$$

and if we multiply it on the left by $\Delta \bar{x}_k^T$ we obtain

$$\langle \Delta \bar{x}_k, \nabla_{\bar{x}} G_k^T (y_k + \Delta y_k) \rangle = -\langle \Delta \bar{x}_k, Q_k \Delta \bar{x}_k \rangle + \langle \Delta \bar{x}_k, z_k + \Delta z_k \rangle.$$

and therefore write (35) as

$$\Delta v_k = -\langle \Delta \bar{x}_k, Q_k \Delta \bar{x}_k \rangle + \langle \Delta \bar{x}_k, z_k + \Delta z_k \rangle. \tag{36}$$

Next, solving equation (16c) for $\Delta z_k + z_k$ gives

$$\Delta z_k + z_k = -\bar{X}_k^{-1} Z_k \Delta \bar{x}_k + \mu \bar{X}_k^{-1} \mathbf{e}$$

and multiplying it by $\Delta \bar{x}_k^T$ yields

$$\langle \Delta \bar{x}_k, \Delta z_k + z_k \rangle = -\Delta \bar{x}_k^T \bar{X}_k^{-1} Z_k \Delta \bar{x}_k + \mu \Delta \bar{x}_k^T \bar{X}_k^{-1} \mathbf{e}. \tag{37}$$

Then we substitute $\langle \Delta \bar{x}_k, \Delta z_k + z_k \rangle$ from (37) in (36) to obtain

$$\Delta v_k = -\langle \Delta \bar{x}_k, Q_k \Delta \bar{x}_k \rangle - \Delta \bar{x}_k^T \bar{X}_k^{-1} Z_k \Delta \bar{x}_k + \mu \Delta \bar{x}_k^T \bar{X}_k^{-1} \mathbf{e} \tag{38}$$

Finally, using (38) in (34) we can write

$$\begin{aligned}
\phi_1(w_k^p, \Delta w_k^p; c_{k+1}, \mu) &= -\langle \Delta \bar{x}_k, Q_k \Delta \bar{x}_k \rangle - \Delta \bar{x}_k^T \bar{X}_k^{-1} Z_k \Delta \bar{x}_k \\
&\quad + \mu \Delta \bar{x}_k^T \bar{X}_k^{-1} \mathbf{e} - \mu \Delta \bar{x}_k^T \bar{X}_k^{-1} \mathbf{e} \\
&= -\langle \Delta \bar{x}_k, Q_k \Delta \bar{x}_k \rangle - \Delta \bar{x}_k^T \bar{X}_k^{-1} Z_k \Delta \bar{x}_k \\
&\leq -\langle \Delta \bar{x}_k, Q_k \Delta \bar{x}_k \rangle \\
&= -d_k(\Delta \bar{x}_k) \leq 0.
\end{aligned}$$

The last inequality holds since x, z are kept strictly feasible, and the last equality from the definition of $d_k(\Delta \bar{x}_k)$ in (23). ∎

Lemma 3.3 *Let f and g be differentiable. If Δw_k is the direction calculated by solving the Newton system (12) with c_{k+1} chosen as in Algorithm 1, then Δw_k is a descent direction for Φ_2 when $0 < \|G_k\|_2^2 \leq \epsilon_G$.*

Proof. When $0 < \|G_k\|_2^2 \leq \epsilon_G$, Step 2.4 of the algorithm uses $\Phi_2(w_k)$. Using (15) we can write

$$\langle \Delta w_k, \nabla \Phi_2(w_k)\rangle = \langle \Delta w_k, (\nabla E_k^T E_k)\rangle = -\|E_k\|^2 \leq 0$$

from which the descent property is established. ▮

Corollary 3.1 *If $\|\Delta w_k\|$=0, the algorithm chooses $c_{k+1} = c_k$.*

Proof. From (16d) and the definition of G_k when $\Delta w_k = 0$, $G_k = 0$, therefore this situation is covered by Lemma 3.2 (See Ref. 7). ▮

Lemma 3.4 *Let f and g satisfy Assumption A1. Then, for a fixed value μ^l of μ:*

(i) There exists a constant $c_{k+1} \geq 0$, satisfying Step 2.3 of the algorithm.

(ii) Assuming that the sequence $\bar{x}_k$ is bounded, c_k is increased finitely often. That is, there exists an integer $k_* \geq 0$, and $c_* \geq 0$, such that for all $k \geq k_*$, $c_k = c_* < \infty$.

Proof. Part (i) follows directly from the preceding Lemmas, since a finite value c_{k+1} is always generated in Step 2.3. Part (ii) can be shown by contradiction. Assume that $c_k \to \infty$ as $k \to \infty$. From the definition of c_{k+1} in Step 2.3, we can deduce that if $c_k \to \infty$ then $\|G_k\|_2^2 \to 0$. Hence, there exists an integer k_1 such that for all $k \geq k_1$ we have $\|G_k\|_2^2 \leq \epsilon_G$. In Step 2.4 of the algorithm, however, when $\|G_k\|_2^2 \leq \epsilon_G$, the algorithm stops increasing the penalty parameter and switches to the second merit function. Therefore, the maximum value that c_k can attain is $c_* = c_{k_1} = M_{num}/\epsilon_G$ where M_{num} and c_* are finite values. Hence, $c_* < \infty$, contrary to our assumption that $c_k \to \infty$ as $k \to \infty$. Hence, the penalty parameter does not increase indefinitely and there exists an integer $k_* \geq 0$ such that for all $k \geq k_*$ we have $c_k < \infty$. ▮

3.1 Convergence of the Inner Iterations

Consider the objective function

$$\max_{\alpha \in \Xi_m^+} L(x, \alpha) = \max_{\alpha \in \Xi_m^+} \langle \alpha, f\rangle = \max_j (f^j),$$

and its linear approximation with respect to x at x_k

$$\begin{aligned} \max_{\alpha\in\Xi_m^+} L_k(x,\alpha) &= \max_{\alpha\in\Xi_m^+}\langle \alpha, f(x_k)+\nabla f^T(x_k)(x-x_k)\rangle \\ &= \max_j (f_k^j + \langle \nabla f_k^j, \Delta x_k\rangle). \end{aligned} \tag{39}$$

This objective function is used in the following theorem.

Theorem 3.1 *Monotonic Descent of* Φ_1

- *Let Assumptions A1, A6, A7 hold,*
- *For each* k*, let there exist a bounded point* $\Delta w_k = (\Delta v_k, \Delta\bar{x}_k, \Delta z_k, \Delta y_k)$ *as a solution to the Newton system* (12),
- *Let there exist an iteration* k_**, small* $\epsilon_G > 0$*,* $\|G_k\|_2^2 \notin (0,\epsilon_G)$ *and a scalar* $c_* \geq 0$*,* $c_* = c_*(\epsilon_G)$*, such that the condition*

  $$\Delta v_k - \mu\Delta\bar{x}_k^T\bar{X}_k^{-1}\mathbf{e} - c_k(\epsilon_G)\|G_k\|^2 + d_k(\Delta\bar{x}_k) \leq 0,$$

  is satisfied for all $k \geq k_*$ *and* $c_{k+1}(\epsilon_G) = c_k(\epsilon_G) = c_*(\epsilon_G)$.

Then the stepsize computed in Step 2.4 is such that $\tau_{pk} \in (0,1]$ *and hence the sequence* $\{\Phi_1(w_k^p; c_k,\mu)\}$ *is monotonically decreasing for* $k \geq k_*$ *and* μ *fixed.*

Proof. From Step 2.4 of the algorithm we wish to find a bound for

$$\Delta\Phi_1 = \Phi_1(w_{k+1}^p; c_{k+1},\mu) - \Phi_1(w_k^p; c_k,\mu) \tag{40}$$

for $\bar{x}_{k+1} = \bar{x}_k + \tau_{pk}\Delta\bar{x}_k$, $v_{k+1} = \max_j \{f^j(x_{k+1})\}^2$.

We have that

$$\begin{aligned} u_{k+1} &= \max_{\alpha\in\Xi_m^+}\left\{\langle \alpha, f(x_k) + \tau_{pk}\Delta x_k^T\nabla f(x_k)\rangle\right\} \\ &= \max_{\alpha\in\Xi_m^+}\{L_k(x_k+\tau_{pk}\Delta x_k,\alpha)\} \end{aligned}$$

where L_k is defined in (39). From the convexity of $\max_{\alpha\in\Xi_m^+}\{L_k(x,\alpha)\}$

$$\begin{aligned} u_{k+1} &\leq \max_{\alpha\in\Xi_m^+}\{L_k(x_k,\alpha)\} + \tau_{pk}\left(\max_{\alpha\in\Xi_m^+}\{L_k(x_k+\tau_{pk}\Delta x_k,\alpha)\} - \max_{\alpha\in\Xi_m^+}\{L_k(x_k,\alpha)\}\right) \\ &\leq u_k + \tau_{pk}\left((u_k+\Delta u_k) - v_k\right) \\ &= u_k + \tau_{pk}\Delta v_k. \end{aligned} \tag{41}$$

[2] The dependency of $\Delta\Phi_1$ on $(w_{k+1}^p, w_k^p, c_{k+1}, c_k, \mu)$ has been dropped so as to avoid long equations.

Using the definition of Φ_1 from (18) we can write (40) as

$$\begin{aligned}
\Delta\Phi_1 &= u_{k+1} - u_k + (c_*/2)\left(\|G_{k+1}\|^2 - \|G_k\|^2\right) + \mu\left(B(\bar{x}_{k+1}) - B(\bar{x}_k)\right) \\
&\leq \tau_{pk}\Delta v_k - \tau_{pk}c_*\|G_k\|^2 - \tau_{pk}\mu\langle\Delta\bar{x}_k, \bar{X}_k^{-1}\mathbf{e}\rangle + (\tau_{pk}^2/2)F_{\Phi_1} \\
&= \tau_{pk}\left(\Delta v_k - c_*\|G_k\|^2 - \mu\langle\Delta\bar{x}_k, \bar{X}_k^{-1}\mathbf{e}\rangle\right) + (\tau_{pk}^2/2)F_{\Phi_1} \\
&= \tau_{pk}\phi_1(\Delta w_k^p, w_k^p; c_*, \mu) + (\tau_{pk}^2/2)F_{\Phi_1}.
\end{aligned}$$

The first term on the right of the inequality comes from (41); the second and third terms come from a Taylor series expansion of $G(x)$ and $B(\bar{x})$, respectively; we use the fact that $c_k = c_{k+1} = c_*$ for $k \geq k_*$ according to Lemma 3.4; second order terms of the Taylor expansion are in F_{Φ_1}, which is defined as

$$F_{\Phi_1} = \int_0^1 (1-t)\langle\Delta\bar{x}_k, \nabla^2_{\bar{x}_k^2}\Phi_1(v_k, \bar{x}_k + t\tau_{pk}\Delta\bar{x}_k; c_*, \mu)\Delta\bar{x}_k\rangle dt.$$

Finally, the last equality comes from from (26).

If we add and subtract $(\tau_{pk}^2/2)\langle\Delta\bar{x}_k, Q_k\Delta\bar{x}_k\rangle$ from the right hand side, we can write

$$\begin{aligned}
\Delta\Phi_1 &\leq \tau_{pk}\phi_1(\Delta w_k^p, w_k^p; c_*, \mu) + (\tau_{pk}^2/2)F_{\Phi_1} \\
&\quad + (\tau_{pk}^2/2)\left(\langle\Delta\bar{x}_k, Q_k\Delta\bar{x}_k\rangle - \langle\Delta\bar{x}_k, Q_k\Delta\bar{x}_k\rangle\right) \\
&= \tau_{pk}\phi_1(\Delta w_k^p, w_k^p; c_*, \mu) + (\tau_{pk}^2/2)\langle\Delta\bar{x}_k, Q_k\Delta\bar{x}_k\rangle \\
&\quad + (\tau_{pk}^2/2)\left(F_{\Phi_1} - \langle\Delta\bar{x}_k, Q_k\Delta\bar{x}_k\rangle\right) \\
&\leq \tau_{pk}\phi_1(\Delta w_k^p, w_k^p; c_*, \mu) + (\tau_{pk}^2/2)\langle\Delta\bar{x}_k, Q_k\Delta\bar{x}_k\rangle \\
&\quad + (\tau_{pk}^2/2)\xi_k\|\Delta\bar{x}_k\|^2 \qquad (42)
\end{aligned}$$

where

$$\xi_k = \int_0^1 (1-t)\left\|\nabla^2_{\bar{x}_k^2}\Phi_1(v_k, \bar{x}_k + t\tau_{pk}\Delta\bar{x}_k; c_*, \mu) - Q_k\right\| dt.$$

Using the Cauchy-Schwarz inequality and Assumption A6 we can write

$$\langle\Delta\bar{x}_k, Q_k\Delta\bar{x}_k\rangle \leq \|Q_k\|\,\|\Delta\bar{x}_k\|^2 \leq m_1\|\Delta\bar{x}_k\|^2$$

therefore inequality (42) can be written as

$$\begin{aligned}
\Delta\Phi_1 &\leq \tau_{pk}\phi_1(\Delta w_k^p, w_k^p; c_*, \mu) + (\tau_{pk}^2/2)m_1\|\Delta\bar{x}_k\|^2 + (\tau_{pk}^2/2)\xi_k\|\Delta\bar{x}_k\|^2 \\
&= \tau_{pk}\phi_1(\Delta w_k^p, w_k^p; c_*, \mu) + (\tau_{pk}^2/2)(m_1 + \xi_k)\|\Delta\bar{x}_k\|^2.
\end{aligned}$$

In Lemmas 3.1, 3.2 we have shown that for $\|G_k\|^2 \notin (0, \epsilon_G)$

$$\phi_1(\Delta w_k^p, w_k^p; c_*, \mu) \leq -d_k(\Delta \bar{x}_k).$$

If we choose $M_1 = \min\{\sigma_k, m_2\}$, then from (23)

$$d_k(\Delta \bar{x}_k) \geq M_1 \|\Delta \bar{x}_k\|^2,$$

therefore combining the last two

$$\|\Delta \bar{x}_k\|^2 \leq -(1/M_1)\phi_1(\Delta w_k^p, w_k^p; c_*, \mu). \tag{43}$$

In this case we can write

$$\begin{aligned} \Delta\Phi_1 &\leq \tau_{pk}\phi_1(\Delta w_k^p, w_k^p; c_*, \mu) - (\tau_{pk}^2/2M_1)(m_1 + \xi_k)\phi_1(\Delta w_k^p, w_k^p; c_*, \mu) \\ &= \tau_{pk}\phi_1(\Delta w_k^p, w_k^p; c_*, \mu)\left(1 - \tau_{pk}((m_1 + \xi_k)/2M_1)\right). \end{aligned} \tag{44}$$

The stepsize strategy in Step 2.4 of the algorithm determines τ_{pk} such that

$$\rho \leq 1 - \tau_{pk}((m_1 + \xi_k)/2M_1) \leq 1/2.$$

Since by Lemmas 3.2–3.4 the descent property $\phi_1(w_k^p, \Delta w_k^p; c_{k+1}, \mu) \leq 0$ holds, there exists a τ_{pk} to ensure that (44) and hence Eq. (24) holds. Therefore $\{\Phi_1(w_k^p; c_k, \mu)\}$ is monotonically decreasing. ∎

Corollary 3.2 *The sequence $\{\bar{x}_k\}$ generated by the algorithm is bounded away from zero.*

Proof. Assume, to the contrary, that the sequence $\{\|\bar{x}_k\|\} \rightarrow 0$. Then

$$\{B(\bar{x}_k)\} \rightarrow \infty.$$

From Assumption A5 (the feasible region is bounded), the sequences $f(x_k)$, $\{\|G(v_k, \bar{x}_k)\|\}$ and $\{\bar{x}_k\}$ must be bounded. Hence the sequence

$$\{\Phi_1(w_k^p; c_*, \mu)\} \rightarrow \infty.$$

This contradicts the monotonic decrease of this sequence established in Theorem 3.1. ∎

Lemma 3.5 *For fixed μ, the lower bounds LB_k^i and the upper bounds UB_k^i, $i = 1, 2 \ldots n$ of the box constraints in the dual stepsize rule are bounded away from zero and bounded from above respectively, if the corresponding $\bar{x}_k^i$, of the iterates $\bar{x}_k$ are also bounded above and away from zero.*

Proof. Established in Ref. 18 for the dual stepsize rule. ∎

This also implies that $\{z_k\}$ is a sequence bounded above and away from zero. We have thus established that both $\{\bar{x}_k\}$ and $\{z_k\}$ are bounded sequences. Next we attempt to show that $\{y_k\}$ is also bounded. We establish an intermediate result in the next lemma and then proceed to prove that a unit step along Δy_k results in sequence $\{y_k + \Delta y_k\}$ being bounded.

In the next paragraph we derive a reduced Newton system for our algorithm. Iterate subscripts and function arguments are omitted.

From the Newton system (16), solving (16c) for Δz_k we obtain

$$\Delta z = -\bar{X}^{-1} Z \Delta \bar{x} - z + \mu \bar{X}^{-1} \mathbf{e}. \tag{45a}$$

Substituting the latter in (16b), we can write (16) as

$$\begin{array}{rcl} -\langle \mathbf{e}, \Delta y \rangle &=& -\left[1 - \langle y, \mathbf{e} \rangle\right] \\ Q\Delta\bar{x} + \nabla_{\bar{x}} G(v,\bar{x})^T \Delta y + \bar{X}^{-1} Z \Delta \bar{x} + z + \mu \bar{X}^{-1}\mathbf{e} &=& -\nabla_{\bar{x}} G(v,\bar{x})^T y + z \\ -\Delta v \mathbf{e} + \nabla_{\bar{x}} G(v,\bar{x}) \Delta \bar{x} &=& -G(v,\bar{x}) \end{array}$$

which after cancellations and some re-arrangement gives

$$\begin{array}{lcrcrcl} & & -\langle \mathbf{e}, \Delta y + y \rangle & & & = & -1 \\ \Delta v \mathbf{e} & & & - & \nabla_{\bar{x}} G(v,\bar{x}) \Delta \bar{x} & = & G(v,\bar{x}) \\ & & \nabla_{\bar{x}} G(v,\bar{x})^T (\Delta y + y) & + & (Q + \bar{X}^{-1} Z) \Delta \bar{x} & = & -\mu \bar{X}^{-1} \mathbf{e} \end{array}$$

which in matrix form is written as

$$\begin{pmatrix} 0 & -\mathbf{e}^T & 0 \\ \mathbf{e} & 0 & -\nabla_{\bar{x}} G(v,\bar{x}) \\ 0 & \nabla_{\bar{x}} G(v,\bar{x})^T & Q + \bar{X}^{-1} Z \end{pmatrix} \begin{pmatrix} \Delta v \\ y + \Delta y \\ \Delta \bar{x} \end{pmatrix} = - \begin{pmatrix} 1 \\ -G(v,\bar{x}) \\ \mu \bar{X}^{-1} \mathbf{e} \end{pmatrix}. \tag{45b}$$

Lemma 3.6 *Let w_k be a sequence of vectors generated by our algorithm. Then the matrix sequence $\{R_k^{-1}\}$ is bounded, where*

$$R_k = \begin{pmatrix} 0 & -\mathbf{e}^T & 0 \\ \mathbf{e} & 0 & -\nabla_{\bar{x}} G(v,\bar{x}) \\ 0 & \nabla_{\bar{x}} G(v,\bar{x})^T & Q + \bar{X}^{-1} Z \end{pmatrix}.$$

Proof. For simplicity, we shall suppress arguments and subscripts. The reduced matrix can be written as

$$R = \begin{pmatrix} 0 & A \\ -A^T & B \end{pmatrix} = \begin{pmatrix} 0 & -\mathbf{e}^T & 0 \\ \mathbf{e} & 0 & -\nabla_{\bar{x}} G \\ 0 & \nabla_{\bar{x}} G^T & Q + \bar{X}^{-1} Z \end{pmatrix}$$

where

$$A = \begin{pmatrix} -\mathbf{e} & 0 \end{pmatrix}, \qquad B = \begin{pmatrix} 0 & -\nabla_{\bar{x}} G \\ \nabla_{\bar{x}} G^T & Q + \bar{X}^{-1} Z \end{pmatrix}.$$

By assumptions A1, A2, Corollary 3.2 and Lemma 3.5 matrix B^{-1} exists and is bounded, with

$$B^{-1} = \begin{pmatrix} [\nabla_{\bar{x}} G \Omega \nabla_{\bar{x}} G^T]^{-1} & -[\nabla_{\bar{x}} G \Omega \nabla_{\bar{x}} G^T]^{-1} \nabla_{\bar{x}} G \Omega \\ \Omega \nabla_{\bar{x}} G^T [\nabla_{\bar{x}} G \Omega \nabla_{\bar{x}} G^T]^{-1} & \Omega - \Omega \nabla_{\bar{x}} G^T [\nabla_{\bar{x}} G \Omega \nabla_{\bar{x}} G^T]^{-1} \nabla_{\bar{x}} G \Omega \end{pmatrix}$$

where $\Omega = (Q + \bar{X}^{-1} Z)^{-1}$. In addition R_k is invertible with

$$R^{-1} = \begin{pmatrix} [AB^{-1}A^T]^{-1} & -[AB^{-1}A^T]^{-1} AB^{-1} \\ B^{-1}A^T [AB^{-1}A^T]^{-1} & B^{-1} - B^{-1}A^T [AB^{-1}A^T]^{-1} AB^{-1} \end{pmatrix}$$

and is bounded, since all the matrices involved are bounded. ∎

Lemma 3.7 *Let $\{w_k\}$ be a sequence of vectors generated by the algorithm. Then the sequence of vectors, $\{\Delta v_k, \Delta \bar{x}_k, y_k + \Delta y_k, \Delta z_k\}$ is bounded.*

Proof. We have shown in Lemma 3.6 that the matrix on the left of the reduced system (45b) exists and is bounded. Therefore the sequences $\{\Delta v_k\}$, $\{\Delta \bar{x}_k\}$ and $\{y_k + \Delta y_k\}$ are also bounded. From (45a) and since all elements on the left are bounded, we deduce that $\{\Delta z_k\}$ is also bounded. ∎

Lemma 3.8 *Let the assumptions of Theorem 3.1 be satisfied and let, for some k_0 and $k \geq k_0$, the set*

$$U = \{w^p \in R^{N+1} | \Phi_1(w^p; c_*, \mu) \leq \Phi_1(w^p_{k_0}; c_*, \mu)\} \tag{46}$$

be compact. Then for all $k \geq k_0$

$$\lim_{k \to \infty} \phi_1(w^p_k, \Delta w^p_k; c_*, \mu) = 0. \tag{47}$$

Proof. The scalar $\rho \in (0, 1/2)$ determined by the stepsize strategy of Step 2.4 is such that

$$\rho \leq 1 - \tau_{pk}((m_1 + \xi_k) / 2M_1) \leq 1/2.$$

By solving for τ_{pk} we can write,

$$\tau_{pk} \leq 2M_1((1 - \rho) / (m_1 + \xi_k)).$$

From the last one, the largest value τ_{pk} can take whilst ensuring satisfaction of the Armijo rule in Step 2.4 is

$$\tilde{\tau}_{pk} = \min\{1, 2M_1((1 - \rho) / (m_1 + \xi_k))\}.$$

According to Step 2.4, τ_{pk} is chosen by reducing by a factor of β the maximum allowed step length $\hat{\tau}_{pk}$ until inequality (24) is satisfied. Therefore $\tau_{pk} \in [\beta\tilde{\tau}_{pk}, \tilde{\tau}_{pk}]$ and such a choice also satisfies (24).

As f and g are twice continuously differentiable and U is a compact set, there is a scalar $\bar{\xi} < \infty$ such that $\xi_k \leq \bar{\xi}$. Thus, since $M_1, m_1 > 0$, we have

$$\tau_{pk} \geq \bar{\tau}_{pk} > 0,$$

for all k, for the stepsize strategy defined by inequality (24) and for some

$$\bar{\tau}_{pk} = \min\{1, 2M_1((1 - \rho) / \left(m_1 + \bar{\xi}\right))\}.$$

It is then obvious that the stepsize τ_{pk} is always bounded away from zero.

In addition, from Armijo's rule and Lemmas 3.1, 3.2 we have

$$\Phi_1(w_{k+1}^p; c_*, \mu) - \Phi_1(w_k^p; c_*, \mu) \leq \rho\tau_{pk}\phi_1(w_k^p, \Delta w_k^p; c_*, \mu) < 0. \tag{48}$$

Having assumed that the level set U is bounded, we can deduce that

$$\lim_{k\to\infty} \left|\Phi_1(w_{k+1}^p; c_*, \mu) - \Phi_1(w_k^p; c_*, \mu)\right| = 0,$$

which when combined with (48) and the fact that $\rho, \tau_{pk} > 0$ proves the lemma. ∎

Lemma 3.9 *Let the assumptions of Lemma 3.8 be satisfied. Then*

$$\lim_{k\to\infty} \|\Delta\bar{x}_k\| = 0 \tag{49a}$$

and

$$\lim_{k\to\infty} \|\Delta v_k\| = 0. \tag{49b}$$

Equivalently

$$\lim_{k\to\infty} \|\Delta w_k^p\| = 0. \tag{49c}$$

Proof. We have shown in (43) that for $\|G_k\|^2 \notin (0, \epsilon_G)$

$$\|\Delta \bar{x}_k\|^2 \leq -(1/M_1)\phi_1(\Delta w_k^p, w_k^p; c_*, \mu),$$

for $M_1 = \min\{\sigma_k, m_2\}$. Since $M_1 > 0$, (49a) holds from the previous lemma. Bearing in mind Remark 2.1 we derive the validity of (49c) which in turn yields the validity of (49b). $\blacksquare$

Theorem 3.2 *Let the assumptions of the previous lemma hold. Then for μ fixed, the algorithm asymptotically generates a point that satisfies the perturbed KKT conditions* (11)*.*

Proof. Let $(v_*(\mu), \bar{x}_*(\mu), z_*(\mu), y_*(\mu))$ be a point such that $\{v_k\} \to v_*(\mu)$, $\{\bar{x}_k\} \to x_*(\mu)$, $\{z_k\} \to z_*(\mu)$, $k_* \leq k \in K$. The existence of such a point is ensured since—by Assumption A2, Corollary 3.2 and Lemmas 3.5 and 3.7—the sequence of points $(v_*(\mu), \bar{x}_*(\mu), z_*(\mu), y_*(\mu))$ is bounded for μ fixed and by Theorem 3.1, the algorithm sufficiently decreases Φ_1, ensuring thus that $(v_k, \bar{x}_k) \in U$, with U compact.

We begin our proof by showing that for k sufficiently large, τ_{dk} becomes unity by establishing that

$$\lim_{k\to\infty} \left\|z_k + \Delta z_k - \mu \bar{X}_{k+1}^{-1}\mathbf{e}\right\| = 0 \tag{50}$$

Adding $-\mu X_{k+1}^{-1}\mathbf{e}$ to both sides of equation (45a) used in the reduced system and rearranging gives

$$\left\|z_k + \Delta z_k - \mu \bar{X}_{k+1}^{-1}\mathbf{e}\right\| \leq \left\|-\bar{X}_k^{-1} Z_k\right\| \left\|\Delta \bar{X}_k\right\| + \mu \left\|\bar{X}_k^{-1} - \bar{X}_{k+1}^{-1}\right\| \|\mathbf{e}\| . \tag{51}$$

Furthermore,

$$\begin{aligned} \left\|\bar{X}_k^{-1} - \bar{X}_{k+1}^{-1}\right\|_2^2 &\leq N \max_{1\leq i\leq N}\{\left((1/\bar{x}_k^i) - (1/\bar{x}_{k+1}^i)\right)^2\} \\ &= N \max_{1\leq i\leq N}\{(\tau_{pk})^2(\Delta x_k^i)^2/(\bar{x}_k^i)^2(\bar{x}_{k+1}^i)^2\}. \end{aligned}$$

Since $\tau_{pk} \in (0,1], (\Delta \bar{x}_k^i)^2 \leq \|\Delta \bar{x}_k\|_2^2$ and the sequence $\{\bar{x}_k\}$ is bounded away from zero, from the above inequality and (49a), we can derive

$$\lim_{k\to\infty} \left\|\bar{X}_k^{-1} - \bar{X}_{k+1}^{-1}\right\|_2^2 \leq N \lim_{k\to\infty} \max_{1\leq i\leq N}\{\|\Delta \bar{x}_k\|_2^2 / (\bar{x}_k^i)^2(\bar{x}_{k+1}^i)^2\} = 0. \tag{52}$$

Consequently, letting $k \to \infty$ in (51) and using (49a) and (52), we can deduce that

$$\lim_{k\to\infty} \left\| z_k + \Delta z_k - \mu \bar{X}_{k+1}^{-1}\mathbf{e} \right\| = 0.$$

As a result, for sufficiently large k,

$$z_{k+1} = z_k + \Delta z_k. \tag{53}$$

Moreover, using $\Delta z_k = -z_k + \mu \bar{X}_k^{-1}\mathbf{e} - \bar{X}_k^{-1} Z_k \Delta \bar{x}_k$, and for k sufficiently large, the complementarity condition becomes

$$\bar{X}_{k+1} z_{k+1} = \bar{X}_{k+1}(z_k + \Delta z_k) = \bar{X}_{k+1}\bar{X}_k^{-1}(\mu\mathbf{e} - Z_k \Delta \bar{x}_k) \tag{54}$$

From (49a) and the fact that the elements of the diagonal matrix $\bar{X}_{k+1}\bar{X}_k^{-1}$ can be written as $(\bar{x}_{k+1}^i)/(\bar{x}_k^i) = 1 + \tau_{pk}\Delta\bar{x}_k^i/(\bar{x}_k^i)$, for all $i = 1, 2, \ldots, m$, it is clear that

$$\lim_{k\to\infty} \bar{X}_{k+1}\bar{X}_k^{-1} = I_N, \tag{55}$$

where I_N is the $N \times N$ identity matrix. Therefore, letting $k \to \infty$ in (54) and using (49a) and (55) yields

$$\lim_{k\to\infty} \bar{X}_{k+1} z_{k+1} = \bar{X}_*(\mu) z_*(\mu) = \mu\mathbf{e}. \tag{56}$$

Also, for $k \to \infty$, Eqs (16d) and (49) yield:

$$\lim_{k\to\infty} \nabla G_k \Delta \bar{x}_k = G(\bar{x}_*(\mu)) = 0. \tag{57}$$

For $z_{k+1} = z_k + \Delta z_k$, $y_{k+1} = y_k + \Delta y_k$, Eq (16b) can be written as

$$\nabla G_k^T y_{k+1} - z_{k+1} = -Q_k \Delta \bar{x}_k. \tag{58}$$

Letting $k \to \infty$ in the above and using (49a), we get

$$\lim_{k\to\infty} \left\| \nabla G_k^T y_{k+1} - z_{k+1} \right\| = 0 \tag{59}$$

or equivalently

$$\nabla G(\bar{x}_*(\mu))^T y_*(\mu) - z_*(\mu) = 0. \tag{60}$$

Also, from (53) and considering equation (16a),

$$1 = \langle y_{k+1}, \mathbf{e} \rangle \tag{61}$$

and therefore letting $k \to \infty$, we get

$$1 = \langle y_*(\mu), \mathbf{e} \rangle. \tag{62}$$

From (62), (60), (58) and (57), we conclude that $(v_*(\mu), \bar{x}_*(\mu), z_*(\mu), y_*(\mu))$ is a solution of the perturbed KKT conditions given in (12). ∎

A consequence of Theorem 3.2 is that, for any convergent subsequence produced by the algorithm, for $\mu = \mu^l$, there is an iteration $k \geq \tilde{k}$, such that

$$\|E(\tilde{v}_k, \bar{x}_{\tilde{k}}, z_{\tilde{k}}, y_{\tilde{k}}; \mu)\|_2 \leq \eta \mu^l \tag{63}$$

for all $k \geq \tilde{k}$ where $\eta \geq 0$. Record the value of the current iterate as

$$(\tilde{v}^l, \tilde{\bar{x}}^l, \tilde{z}^l, \tilde{y}^l) = (\tilde{v}_k, \bar{x}_{\tilde{k}}, z_{\tilde{k}}, y_{\tilde{k}}),$$

and set $\mu^{l+1} < \mu^l$. In this way we generate a sequence of approximate central points $\{\tilde{v}^l, \tilde{\bar{x}}^l, \tilde{z}^l, \tilde{y}^l\}$.

3.2 Convergence of the Outer Iterations

In this section it will be shown that the sequence of approximate central points converge to a KKT point $w_* = (v_*, \bar{x}_*, z_*, y_*)$ of the original constrained minimax problem (7).

For a given and sufficiently small $\varepsilon \geq 0$, considering the set of all approximate central points, generated by Algorithm 1,

$$S(\varepsilon) = \{\tilde{w}^l : \varepsilon \leq \left\|E(\tilde{w}^l; \mu^l)\right\|_2 \leq \left\|E(\tilde{w}^0; \mu^0)\right\|_2, \forall \mu^l < \mu^0\}. \tag{64}$$

If $\varepsilon > 0$, the stepsize rules described in previous sections guarantee that $\tilde{x}^l$ and $\tilde{z}^l \in S(\varepsilon)$ are bounded away from zero for $l \geq 0$. As a result $(\tilde{\bar{x}}^l)^T(\tilde{z}^l)$ are also bounded away from zero in $S(\varepsilon)$. The following lemma shows that the sequence $\{\tilde{y}^l\}$ is bounded if the sequence $\{\tilde{z}^l\}$ is also bounded.

Lemma 3.10 *If Assumption A2 holds, and the iterates $\tilde{x}^l$ are in a compact set for $l \geq 0$, then there exists a constant $M_1 > 0$ such that*

$$\left\|\tilde{y}^l\right\| \leq M_1(1 + \left\|\tilde{z}^l\right\|).$$

Proof. We define $\tilde{r}^l = \nabla G(\tilde{x}^l)^T \tilde{y}^l - \tilde{z}^l$, and solve for $\nabla G(\tilde{x}^l)\tilde{y}^l$ to obtain

$$\nabla G(\tilde{x}^l)^T \tilde{y}^l = \tilde{z}^l + \tilde{r}^l.$$

From the assumptions made, the above equation can be written as:

$$\tilde{y}^l = [\nabla G(\tilde{x}^l)\nabla G(\tilde{x}^l)^T]^{-1}\nabla G(\tilde{x}^l)\left(\tilde{z}^l + \tilde{r}^l\right). \tag{65}$$

Taking norms of both sides of the above equation yields:

$$\begin{aligned} \left\|\tilde{y}^l\right\| &\leq \left\|[\nabla G(\tilde{x}^l)\nabla G(\tilde{x}^l)^T]^{-1}\nabla G(\tilde{x}^l)\right\|\left(\left\|(\tilde{z}^l)\right\| + \left\|\tilde{r}^l\right\|\right) \\ &\leq M_1(1 + \left\|\tilde{z}^l\right\|) \end{aligned} \tag{66}$$

where the constant M_1 is defined as

$$M_1 \geq [\nabla G(\tilde{x}^l)\nabla G(\tilde{x}^l)^T]^{-1}\nabla G(\tilde{x}^l)\max\{1, \left\|\tilde{r}^l\right\|\}.$$

and is finite, according to our assumptions. ■

Lemma 3.11 *If the sequence $\{\tilde{v}^l, \tilde{\tilde{x}}^l, \tilde{z}^l, \tilde{y}^l\}$ stays in $S(\varepsilon)$ for all $l \geq 0$, then it is bounded above.*

Proof. From Lemma 3.10 it suffices to show that the sequences $\{\tilde{v}^l\}$, $\{\tilde{\tilde{x}}^l\}$, and $\{\tilde{z}^l\}$ are bounded from above. Sequence $\{\tilde{\tilde{x}}^l\}$ is bounded from above; since f is continuous, by A5 we have that $\{\tilde{v}^l\}$ is also bounded. Therefore we only need to show that $\{\tilde{z}^l\}$ is bounded from above. This proof can be found in Ref. 7. ■

Theorem 3.3 *Let $\{\mu^l\}$ be a positive monotonically decreasing sequence of barrier parameters with $\{\mu^l\} \to 0$, and let $\{\tilde{v}^l, \tilde{\tilde{x}}^l, \tilde{z}^l, \tilde{y}^l\}$ be a sequence of approximate central points satisfying* (63) *for $\mu = \mu^l, l \geq 0$. Then, the sequence $\{\tilde{v}^l, \tilde{\tilde{x}}^l, \tilde{z}^l, \tilde{y}^l\}$ is bounded and all of its limit points $(\tilde{v}^*, \tilde{\tilde{x}}^*, \tilde{z}^*, \tilde{y}^*)$ satisfy the KKT conditions of problem* (8)*.*

Proof. From Lemma 3.11 the sequence $\{\tilde{v}^l, \tilde{\tilde{x}}^l, \tilde{z}^l, \tilde{y}^l\}$ is bounded and remains in the compact set $S(\varepsilon)$. Thus it has a limit point in $S(\varepsilon)$, denoted by $(\tilde{v}^*, \tilde{\tilde{x}}^*, \tilde{z}^*, \tilde{y}^*)$. From (63) and the fact that $\{\mu^l\} \to 0$, we easily obtain that

$$\lim_{l\to\infty} \left\|E(\tilde{v}^l, \tilde{\tilde{x}}^l, \tilde{z}^l, \tilde{y}^l; \mu)\right\|_2 = 0$$

and $(\tilde{v}^*, \tilde{\tilde{x}}^*, \tilde{z}^*, \tilde{y}^*)$ is a KKT point of the constrained problem (8). ■

4 Numerical Results

In this section we present numerical experience related to the concepts presented. We first start by giving two examples that illustrate the reason why Φ_2 on its own is not suitable for use as a merit function for non-convex optimization. The examples also highlight the usefulness of the proposed method to overcome the problem. Subsequently we present some numerical results obtained using an implementation of our method on the `CUTE` test set. Finally we describe the progress of our algorithm on two of the test problems, to note that the switch between the two merit functions indeed occurs.

Example 4.1 Consider the following problem

$$\begin{array}{rlcl} \min & f & & \\ \text{s.t} & -x^3 & \leq & f, \\ & x^3+1 & \leq & f, \\ & x+1 & \leq & f. \end{array}$$

As seen in Figure 1 it has a saddle point at $x = -0.19$ and a minimum at $x = -0.7937$. Suppose that we start at $x_0 = (1.2)$.

1. Using solely Φ_2, after 7 iterations we converge to $f^* = 0.99$, which is the above the saddle point.

2. Using Φ_1, and Φ_2 as in Algorithm 1, after 9 iterations we converge to $f^* = 0.5$ which is the above sought minimum.

Example 4.2 Consider the following example

$$\begin{array}{rlcl} \min & L & & \\ \text{s.t.} & xy - 1 + x^2 + (y-1)^2 & \leq & L, \\ & xy + 1 - x^2 - (y-1)^2 & \leq & L. \end{array}$$

As seen in Figure 2 it has a maximum at $(x, y) = (0.66667, 1.33333)$ and a global minimum at $(x, y) = (-0.86603, 1.5)$. For this problem suppose that we start from point $(x_0, y_0) = (0.6, 1.3)$.

1. Using solely Φ_2, after 6 iterations we converge to $L^* = 1.33333$ which is the above local maximum.

2. Switching from Φ_1 to Φ_2 as in our algorithm, we converge after 13 iterations to $L^* = -1.29904$ which is the global minimum of the problem.

The algorithm presented in the preceding section has been implemented in standard C, and interfaced with AMPL. AMPL (Ref. 20) is a powerful mathematical modeling language that allows the optimization problems to be written in a simple algebraic notation, and also provides first and second order derivatives. The code was tested on a 2.4GHz workstation 2Gb of memory. The test set consists of thirty-three minimax problems drawn from the CUTE set, available in AMPL format (Ref. 21).

The algorithm parameters were set as follows. In Step 1, the accuracy of the stopping criteria is $\epsilon_0 = 10^{-6}$. In Step 2.3, $\epsilon_G = 10^{-4}$ and $\delta = 10$. In Step 2.4, we set $\theta = 0.9995, \beta = 0.5, \rho = 10^{-4}, m_1 = 1$ and $M_1 = 10$. In the barrier selection rule we set $\omega = 6$.

The numerical results are summarized in Table 1, where we use the following abbreviations:

- Problem : The name of the CUTE set problem as described in Ref. 21.
- Itns : The total number of iterations required to find the optimal solution of the problem.
- c_0 : Initial value of the penalty parameter.
- c_* : Final value of the penalty parameter.
- k_* : The iteration after which the penalty parameter was unchanged.
- c_* : the final value of the penalty parameter.

The majority starting points are the same as those in Ref. 21. For problems goffin, haifas and mifflin2 (0.0) was used instead. Additionally, for madsen we used (0.3, 1.0), for oet2 we used (1.0), for polak1 (1.0, 0.05) and for polak6 (-1.0, 0.0, -1.0, 0.0) were used as the starting point. Finally, for makela4 the first 10 variables were initialized to (1.0) and variables 11 to 20 were initialized to a value of (-1.0).

The performance of the minimax algorithm in terms of the number of iterations is provided, along with the results obtained using LOQO, MINOS and IPOPT on the NEOS server 4.0. LOQO is an infeasible primal dual interior-point algorithm which uses a hybrid filter and merit function approach. MINOS uses a linear simplex method for linear problems, a reduced gradient method with quasi-Newton approximations for nonlinear objective functions, and sparse projected Lagrangian method for nonlinear constraints. IPOPT is an interior-point line search filter method that aims to find a local solution to general nonlinear programming problems. Our implementation does not permit timing experiments, or experiments on large scale problems since we use dense linear algebra operations. We hope that the addition of sparse large-scale linear algebra routines will allow a more elaborate comparison. In addition, in Table 1 we compare the performance of the algorithm when the single augmented Lagrangian merit function is used (column labelled Φ_1) with the performance when both merit functions are used (column labelled Φ_1, Φ_2).

Table 1 shows, for the problems solved, that in many cases our algorithm compares favorably against MINOS, LOQO and IPOPT. For example, in problem haifam, where MINOS and LOQO failed, it took our algorithm 23 iterations and IPOPT was more than three times slower (74). Similarly, our algorithm solved problem polak3 in 20 iterations, a bit faster than LOQO (22), but MINOS failed and IPOPT was six times slower than our code (132). A similar result is obtained for problem polak6, where MINOS and IPOPT converged after 232 and 283 iterations, respectively, LOQO in 60 iterations, and our code in only 19 iterations (22 when switching merit functions). On the other hand, for problem oet2, MINOS failed, LOQO and our code were very slow (162 and 160 iterations, respectively) whereas IPOPT took only 19 iterations.

To demonstrate the switch between the two merit functions, the progress of the algorithm for problems chaconn1 and mifflin1 is shown in Table 2.

5 Conclusions

We have presented a primal dual-interior-point algorithm for constrained nonlinear, discrete minimax problems. In the inner iterations of the algorithm we employ a line search framework that uses two merit functions. One of the merit functions is the l_2 merit function. This merit function is initially used in the inner iteration to guide iterates towards feasibility. Once feasibility has been attained, a switch is made to the second merit function. The second merit function is the squared norm of the perturbed KKT residual. The latter is used for the rest of the inner iterations of the algorithm, i.e. until the perturbed KKT conditions have been satisfied. For the outer iterations of the algorithm an adaptive barrier parameter is used. Global convergence results have been presented. Numerical experimentation on small to medium scale problems shows that practical value of the proposed approach.

Acknowledgment. The authors would like to thank the two anonymous for their useful comments.

References

[1] Karmarkar, N., *A New Polynomial-Time Algorithm for Linear Programming*, Combinatorica, Vol.4, No.4, pp.373–395, 1984.

[2] El-Bakry, A. S., Tapia, R. A., Tsuchiya, T., and Zhang, Y., *On the formulation and theory of the Newton interior-point method for nonlinear programming*, Journal of Optimization Theory and Applications, Vol.89, No.3, pp.507–541, 1996.

[3] Kojima, M., Mizuno, S., and Yoshise, A., *A primal-dual interior point method for linear programming*, Progress in Mathematical Programming, Edited by N. Megiddo, New York, Springer-Verlag, pp. 29–47, 1989.

[4] Benson, H. Y., Shanno, D. F., and Vanderbei, R. J., *Interior-point methods for nonconvex nonlinear programming: jamming and numerical testing*, Mathematical Programming, Vol.99, No.1, Ser. A, pp.35–48, 2004.

[5] Du, D.-Z. and Pardalos, P. M., *Minimax and Applications*, Nonconvex Optimization and Its Applications, Springer, Dordrecht, Holland, 1995.

[6] Rustem, B. and Howe, M., *Algorithms for worst-case design and applications to risk management*, Princeton University Press, Princeton, NJ, 2002.

[7] Akrotirianakis, I. and Rustem, B., *A Globally Convergent Interior Point Algorithm For Non-Linear Problems*, Journal of Optimization Theory and Applications, Vol.125, No.3, pp.497–521, 2005.

[8] Rustem, B., *A constrained min-max algorithm for rival models of the same economic system*, Mathematical Programming, Vol.53, No.3, Ser. A, pp.279–295, 1992.

[9] Medanić, J. and Andjelić, M., *On a class of differential games without saddle-point solutions*, Journal of Optimization Theory and Applications, Vol.8, pp.413–430, 1971.

[10] Medanić, J. and Andjelić, M., *Minimax solution of the multiple-target problem*, IEEE Transactions on Automatic Control, Vol.17, No.5, pp.597–604, 1972.

[11] Vanderbei, R. J. and Shanno, D. F., *An interior point algorithm for nonconvex nonlinear programming*, Computational Optimization and Applications, Vol.13, pp.231–252, 1999.

[12] Chamberlain, R. M., Powell, M. J. D., Lemarechal, C., and Pedersen, H. C., *The Wathdog technique for Forcing Convergence in Algorithms for Constrained Optimization*, Mathematical Programming Studies, Vol.16, pp.1–17, 1982.

[13] Gay, D. M., Overton, M. L., and Wright, M. H., *A primal-dual interior method for nonconvex nonlinear programming*, Advances in nonlinear programming (Beijing, 1996), Edited by Y. Yuan, volume 14 of *Appl. Optim.* Kluwer Academic Publishers, Dordrecht, pp. 31–56, 1998.

[14] Rustem, B., *Convergent Stepsizes for Constrained Optimization Algorithms*, Journal of Optimization Theory and Applications, Vol.49, pp.135–160, 1986.

[15] Rustem, B., *Equality and Inequality Constrained Optimization Algorithms with Convergent Stepsizes*, Journal of Optimization Theory and Applications, Vol.76, No.3, pp.429–453, 1993.

[16] Rustem, B. and Nguyen, Q., *An algorithm for the inequality-constrained discrete min-max problem*, SIAM Journal on Optimization, Vol.8, No.1, pp.265–283 (electronic), 1998.

[17] Yamashita, H., *A globally convergent primal–dual interior point method for constrained optimization*, Optimization Methods and Software, Vol.10, pp.443–469, 1998.

[18] Yamashita, H. and Yabe, H., *Superlinear and quadratic convergence of some primal–dual interior point methods for constrained optimization*, Mathematical Programming, Vol.75, pp.377–397, 1996.

[19] Tzallas-Regas, G., *Switching stepsize strategies for Nonlinear Programming*, PhD thesis, University of London, Imperial College London, London, 2007.

[20] Fourer, R., Gay, D. M., and Kernighan, B. W., *AMPL – A Modeling Language for Mathematical Programming*, The Scientific Press, 1993.

[21] Benson, H., The Cute AMPL models, `http://www.orfe.princeton.edu/~rvdb/ampl/nlmodels/cute`, 1996.

List of Figures

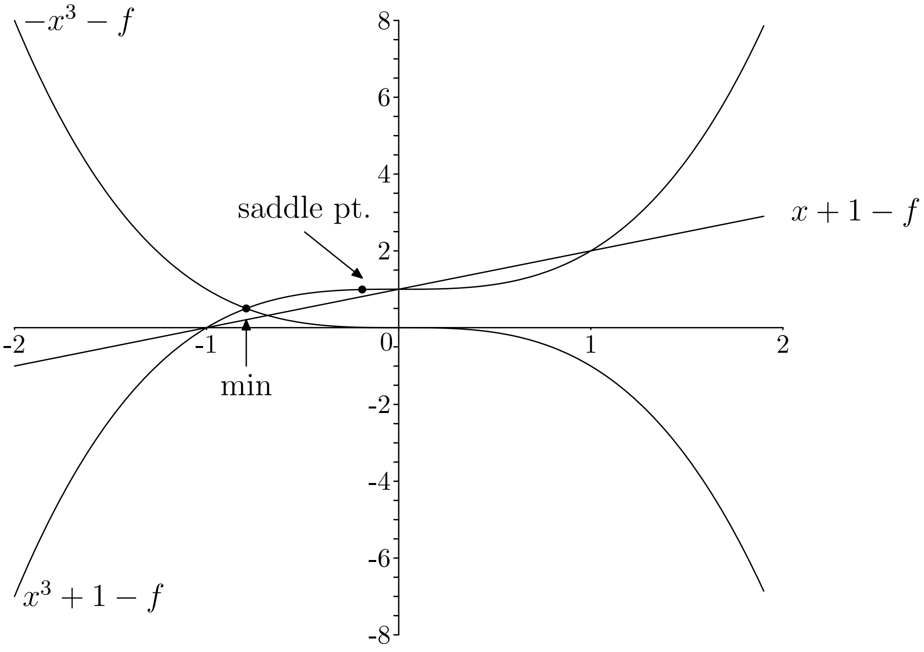

Figure 1: Graph for problem of Example 4.1

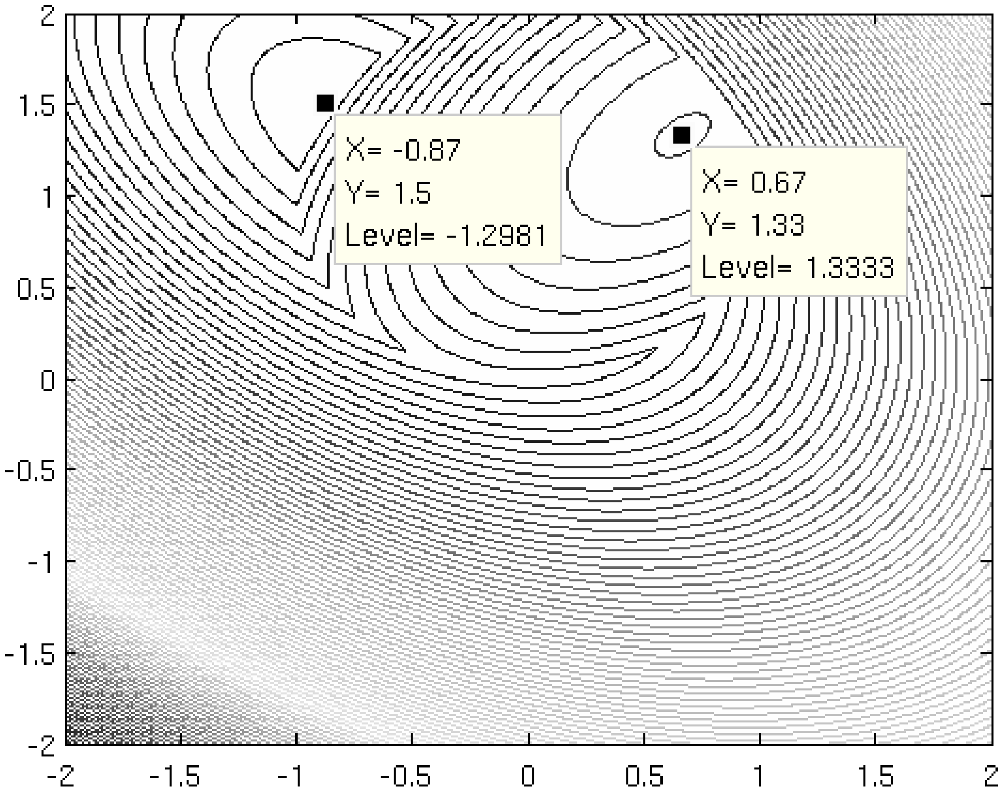

Figure 2: Graph for problem of Example 4.2

List of Tables

	MINOS	LOQO	IPOPT	Minimax algorithm						
					Φ_1			Φ_1, Φ_2		
	Itns	Itns	Itns	c_0	Itns	c_*	k_*	Itns	c_*	k_*
cb2	11	8	7	0.1	8	6.39e11	8	8	1.71e8	6
cb3	2	9	6	0.1	9	4.53e11	9	9	6.94e8	7
chaconn1	24	9	7	0.1	9	3.08e12	9	9	3.7e8	7
chaconn2	2	9	6	0.1	9	4.53e11	9	9	6.94e8	7
congigmz	16	33	30	0.1	27	1.30e12	25	24	3.55e8	22
coshfun	450	22	23	0.0	48	1.81e11	27	48	2.693e6	21
demymalo	30	16	16	0.1	15	1.14e12	15	15	9.91e7	13
gigomez1	40	16	15	0.1	15	2.03e10	15	15	1.75e8	14
goffin	1	11	6	0.0	3	4.88e8	3	3	4.88e8	3
haifam	F^a	max^b	74	0.1	23	3.94e7	23	23	3.94e7	23
haifas	59	12	8	0.1	10	1.03e12	10	10	4.55e8	8
kiwcresc	17	14	20	10	11	9.45e10	11	11	8.51e8	10
madsen	27	11	9	0	23	7.42e8	19	19	1.91e6	18
madsschj	1083	22	22	0.0	29	3.81e5	29	29	3.81e5	29
makela1	26	14	18	1.0	13	1.69e12	13	13	1.19e8	11
makela2	21	12	8	0.1	11	2.63e9	11	11	7.5e7	10
makela3	22	11	6	0.1	10	1.65e2	8	10	1.65e2	8
makela4	1	11	6	0.1	4	0.1	1	4	0.1	1
mifflin1	24	9	7	0.1	9	2.81e12	9	9	2.52e8	7
mifflin2	11	10	8	1	10	4.34e10	10	12	6.2e7	8
minmaxrb	24	10	10	1	15	1e12	15	17	2.94e8	15
oet1	159	14	39	0.0	17	730.57	1	17	730.57	1
oet2	F^a	162	19	0.1	159	2.12e10	155	160	2.12e10	155
polak1	F^a	11	6	0.1	7	2.04e11	7	7	5.3e7	5
polak2	136	17	19	0.1	9	2.44e8	9	9	2.44e8	9
polak3	F^a	22	132	1	20	2.55e12	20	20	1	1
polak4	7	12	14	1	9	1.44e8	8	9	1.44e8	8
polak5	16	38	31	0.1	14	6.04e8	9	14	6.04e8	9
polak6	232	60	283	1	19	3.09e8	19	22	3.09e8	19
pt	1	14	20	0.0	31	1.06e12	31	32	3.77e8	29
rosenmmx	87	18	73	0.1	16	3.27e8	16	16	3.27e8	16
spiral	62	24	17	1	23	1.41e12	22	23	8.32e8	17
womflet	35	88	17	0.1	15	9e11	15	15	8.88e8	13

[a] failure: superbasics limit too small.

[b] unable to solve within maximum number of iterations (default=500)

Table 1: Numerical results comparison

Problem	Itns	Merit Function	c	f	x_k
chaconn1					
	1	Φ_1	10.1	3.94732	(1.00258, 0.28172)
	2	Φ_1	10.1	2.02875	(1.17320, 0.84020)
	3	Φ_1	10.1	2.00637	(1.12765, 0.88404)
	4	Φ_1	10.1	1.95879	(1.13755, 0.89775)
	5	Φ_1	2.6e4	1.95294	(1.13914, 0.89915)
	6	Φ_1	3.3e6	1.95230	(1.13907, 0.89950)
	7	Φ_1	3.7e8	1.95223	(1.13904, 0.89955)
	8	Φ_2	3.7e8	1.95222	(1.13904, 0.89956)
	9	Φ_2	3.7e8	1.95222	(1.13904, 0.89956)
mifflin1					
	1	Φ_1	45.08	-0.04027	(0.99856, 0.98040)
	2	Φ_1	45.08	-0.654271	(1.13630, 0.43687)
	3	Φ_1	45.08	-0.87024	(1.11624, 0.00233)
	4	Φ_1	45.08	-0.982121	(1.0176, 0.00003)
	5	Φ_1	1.9e4	-0.99805	(1.00195, 0)
	6	Φ_1	2.1e6	-0.99980	(1.00020, 0)
	7	Φ_1	2.5e8	-0.99998	(1.00002, 0)
	8	Φ_2	2.5e8	-0.99999	(1, 0)
	9	Φ_2	2.5e8	-0.99999	(1, 0)

Table 2: Progress of the algorithm for `chaconn1`, `mifflin`.