A Globally Convergent Interior Point Algorithm for Non-Linear Programming Problems*

Ioannis Akrotirianakis[1] and Berç Rustem[2†]

Abstract

This paper presents a primal-dual interior point algorithm for solving general constrained non-linear programming problems. The initial problem is transformed to an equivalent equality constrained problem, with inequality constraints incorporated into the objective function by means of a logarithmic barrier function. Satisfaction of the equality constraints is enforced through the incorporation of an adaptive quadratic penalty function into the objective. The penalty parameter is determined using a strategy that ensures a descent property for a merit function. It is shown that the adaptive penalty does not grow indefinitely if feasibility accuracy is allowed up to a given finite precision. Different step-sizes are used for the primal and dual variables. The algorithm applies the Newton method to solve the first order optimality conditions of the equivalent equality problem. Global convergence of the algorithm is achieved through the monotonic decrease of a merit function. Finally, computational results are presented which show the efficient performance of the algorithm.

Key Words: Non-linear Programming, primal-dual interior point methods, adaptive penalty parameter, augmented Lagrangian, convergence analysis.

1. Department of Chemical Engineering,
Princeton University,
Princeton, NJ 08540, USA.

2. Department of Computing,
Imperial College,
London SW7 2BZ, U.K.

*The research reported in this paper was done while the first author was at Imperial College. The authors gratefully acknowledge constructive comments from Professor L. C. W. Dixon and an anonymous referee. We are also grateful to Dr Stanislav Zakovic for his very helpful suggestions and comments. Financial support was provided by the EPSRC grant M16016

†Corresponding author. Tel. +44 20 75948345, fax: +44 20 75818024, email: br@doc.ic.ac.uk

1 Introduction

In this paper, we discuss a primal-dual interior point algorithm for solving general nonlinear programming problems. The algorithm is based on two different approaches. The first is the augmented Lagrangian sequential quadratic programming (SQP) framework for general constrained optimization problems, discussed in Rustem [22]. The SQP algorithms possess good theoretical and practical properties and are very efficient for solving general NLP problems [5]. The second approach is the primal-dual interior point method, where a barrier function and a damped Newton framework are used in order to solve NLP problems. This is closely related to the SQP framework, since after the initial incorporation of the inequality constraints into the objective function an equivalent equality constrained problem is obtained. The latter is solved by applying the Newton method to the first order optimality conditions.

The algorithm presented in this paper is motivated by the fact that the solution of the first order optimality conditions of any NLP problem, which is the core of interior point algorithms, is not sufficient to guarantee the convergence to a local minimum, unless the problem is convex. An algorithm that merely solves the first order optimality conditions may converge to a saddle point or even to a local maximum, since these conditions are also satisfied at those points. To avoid such undesirable behavior we use a merit function, whose aim is to guide the iterates of the algorithm toward a local minimum. The merit function incorporates the inequality constraints by means of the logarithmic barrier function and the equality constraints by means of the quadratic penalty function. Furthermore, the subproblem that is used to compute the search direction involves the augmented Lagrangian of the equality constrained barrier problem. The search direction is shown to be descent for the merit function. It is also shown that the penalty parameter in the merit function does not increase indefinitely, if the iterates of the algorithm are not near a feasible point of the barrier problem. This is a particularly important point as it involves the use of the equality constrained problem and the corresponding augmented Lagrangian to establish the finiteness of the penalty parameter. If the iterates of the algorithm are near a feasible point of the barrier problem, then a switch in the merit function is performed. In this case we use the Euclidean norm of the first order optimality conditions as the merit function. The second merit function, on its own, only ensures convergence to a point satisfying the first order perturbed optimality conditions, without distinguishing between a minimum or maximum. However, the second merit function is expected to be activated when the iterates have been placed, by the primary penalty-barrier merit function, within a neighborhood of a local minimum. As will be discussed later, it is this switch of the merit functions that enables the global convergence of the algorithm to a local minimum.

Although our algorithm is related to the approaches proposed by El-Bakry *et al.* [8] and Yamashita [32], it differs in significant aspects, such as the choice of the merit function, the adaptive penalty selection rule and the step-size rules. Other algorithms which use an adaptive penalty have been recently developed by Vanderbei and Shanno [27], Gajulapalli and Lasdon [12] and Gay *et al.* [13]. More recently it has come to our attention that Vanderbei and Shanno [28], report a similar descent property and penalty selection rule. Our algorithm, is based on the adaptive penalty strategy introduced by Rustem in [21], [22], [23] and a switch-activated second merit function of the first order perturbed optimality conditions. One important property that needs to be highlighted is that the penalty parameter need not be increased indefinitely and descent is assured. The first merit function which uses an adaptive penalty adjustment strategy assures descent for infeasible and exactly feasible points.

Recently, general (non-convex) NLP problems have been the subject of intensive research in the optimization community and several primal-dual interior point algorithms have been emerged. The common characteristic of these algorithms is that they need to use a merit function within a line-search or trust-region framework to achieve global convergence. In the rest of this section we mention some of the most recent work. One of the first primal-dual algorithms for constrained optimization was developed by Yamashita [32], where the l_1 penalty function of the barrier problem is used as merit function to globalise the algorithm. The step size for the primal variables is determined by a line search procedure that guarantees sufficient decrease of the merit function, while the step size for the dual variables is determined in such a way that the complementarity condition is always bounded away from zero and bounded from above. El-Bakry *et al.* [8] develop a primal-dual interior point algorithm for general NLP problems. They basically extend the general primal-dual framework for LP problems, proposed by Kojima, Mijuno and Yoshise [16], to NLP problems. The l_2 norm of the residuals is used as merit function in a line search procedure to globalise the algorithm. A difficulty with this algorithm, caused mainly by the choice of the merit function, is that it may converge to a saddle point or a maximum when a minimum is being sought. In fact Simantiraki and Shanno [24] provide appropriate tests that exhibit this behaviour.

A primal-dual interior point algorithm for non-convex NLP problems is developed by Gay *et al.* [13]. They also use two merit functions and a variation of the watchdog strategy [6], in order to force convergence of the iterates to a local minimum. However, no detailed proof of the global convergence of the algorithm is given in [13]. Also, Akrotirianakis and Rustem [1] develop a primal-dual algorithm that uses an approximation of Fletcher's exact and differentiable merit function together with a line-search procedure in order to achieve global convergence. Vanderbei and Shanno [27] extend their quadratic programming solver LOQO [25] to be able to solve general NLP problems. They propose a primal-dual algorithm that uses the quadratic penalty function of the barrier problem as merit function. The major theoretical difficulty of this algorithm is that the penalty parameter, in the merit function, has to tend to infinity to achieve global convergence. As the penalty parameter becomes arbitrarily large, the conditioning of the Hessian matrix becomes arbitrarily bad. In a more recent paper Benson, Shanno and Vanderbei [3] consider three possible remedies to the Wachter and Biegler problem [30] which arises when primal-dual methods fail to converge because the iterates are bounded away from the optimum. The remedies proposed in [3] are: shifting of the slack variables of the problem, a trust-region method and a modified method. Forsgren and Gill [10] extend the classical Fiacco and McCormick [9] penalty-barrier function, by augmenting it with two extra functions that measure proximity to the central path. The merit function is used in a line search procedure to determine the step size and its monotonic decrease ensures the global convergence of the method. Yamashita and Yabe [34] also investigate the use of the classical Fiacco and McCormick penalty-barrier function [9] augmented by an extra logarithmic term involving the complementarity condition and the equality constraints. The Newton's method is applied to solve the shifted perturbed optimality conditions. It is shown that the merit function decreases monotonically at every iteration, ensuring global convergence of the algorithm.

This paper is organized as follows. In section 2 we introduce the basic features of the augmented Lagrangian methods, used in this paper. In section 3 we present the basic algorithmic framework of primal-dual methods for NLP problems. Section 4 describes the primal-dual interior point algorithm. In section 5 we establish the global convergence of the algorithm. In section 6, we report our numerical experience. We also provide an example where we demonstrate how the mechanism that switches between the two merit functions enables the algorithm to converge to a local minimum and avoid saddle points or local maxima. Finally, in section 7 we present our conclusions.

2 Augmented Lagrangian Methods

Penalty methods are mainly used for equality constrained optimization problems. The aim is to eliminate the constraints and augment the cost function with a penalty term that associates a high cost to infeasible points. The severity of the penalty is determined by a parameter, denoted by c. As c takes higher values feasibility is increasingly ensured.

Consider the equality constrained problem

$$\min\ f(x) \text{ subject to } g(x) = 0, \tag{1}$$

where $f : \Re^n \to \Re$, and $g(x) : \Re^n \to \Re^q$ are given smooth functions. The Lagrangian function of this problem is $L(x, y) = f(x) - y^T g(x)$.

Consider (1) augmented with a quadratic penalty term

$$\min\ f(x) + \frac{c}{2} \parallel g(x) \parallel^2 \text{ subject to } g(x) = 0. \tag{2}$$

The Lagrangian of (2) is given by the augmented Lagrangian function

$$L_c(x, y) = f(x) - y^T g(x) + \frac{c}{2} \parallel g(x) \parallel^2 . \tag{3}$$

Problem (2) has the same local minima as problem (1). The gradient and the Hessian of L_c with respect to x are

$$\nabla_x L_c(x, y) = \nabla f(x) + \nabla g(x)^T (cg(x) - y),$$

$$\nabla^2_{xx} L_c(x, y) = \nabla^2 f(x) + \sum_{i=1}^{q} \nabla^2 g_i(x)(cg_i(x) - y_i) + c\nabla g(x)\nabla g(x)^T . \tag{4}$$

In particular, if x_* and y_* satisfy the first order optimality conditions of problem (1), then

$$\nabla_x L_c(x_*, y_*) = \nabla L(x_*, y_*) = 0 \text{ and } \nabla^2_{xx} L_c(x_*, y_*) = \nabla^2_{xx} L(x_*, y_*) + c\nabla g(x_*)\nabla g(x_*)^T .$$

Consequently, x_* is an unconstrained minimum of $L_c(x, y_*)$. For a detailed treatment of penalty and augmented Lagrangian methods we refer to [4] and [5].

3 Basic Iteration in Primal-Dual Methods

Consider the following constrained problem

$$\begin{array}{ll} min & f(x) \\ ST & g(x) = 0, \quad x \geq 0, \end{array} \tag{5}$$

where $x \in \Re^n$, $f : \Re^n \to \Re$ and $g(x) : \Re^n \to \Re^q$. The formulation in (5) is quite general because every equality and inequality constrained optimization problem can be reduced to that form, by adding for example slack variables to the constraints.

In barrier methods, (5) is approximated by augmenting the objective with the logarithmic barrier function $B(x;\mu) : \Re^n \to \Re$, $B(x;\mu) = -\mu \sum_{i=1}^n log(x^i)$. Thus, the initial problem is approximated by

$$\begin{array}{ll} min & f(x) - \mu \sum_{i=1}^n log(x^i) \\ ST & g(x) = 0, \end{array} \tag{6}$$

where $x > 0$ and the barrier parameter μ is a given sufficiently small and strictly positive constant [9], [5]. The optimality conditions of (6) are

$$\begin{array}{rcl} \nabla f(x) - \nabla g(x)^T y - \mu X^{-1} e & = & 0 \\ g(x) & = & 0, \end{array} \tag{7}$$

where X is the diagonal matrix defined as $X = diag(x^1, ..., x^n)$. Also $e \in \Re^n$ is the vector of all ones. Introducing the non-linear transformation $z = \mu X^{-1} e$, (7) becomes

$$\begin{array}{rcl} \nabla f(x) - \nabla g(x)^T y - z & = & 0 \\ g(x) & = & 0 \\ XZe & = & \mu e, \end{array} \tag{8}$$

where $x, z > 0$ and $Z = diag(z^1, ..., z^n)$. The introduction of z is essential to the numerical success of the barrier methods (see for example [8]).

Consider now the Lagrangian function of the equality and inequality constrained problem (5)

$$L(x, y, z) = f(x) - y^T g(x) - z^T x, \tag{9}$$

where $y \in \Re^q$ and $z \in \Re^n_+ \equiv \{v \in \Re^n : v \geq 0\}$ are the Lagrange multiplier vectors of the equality constraints, $g(x) = 0$, and non-negativity constraints, $x \geq 0$, respectively. The

KKT conditions of (5) are given by the nonlinear system of equations

$$F(x,y,z) = \begin{pmatrix} \nabla_x L(x,y,z) \\ \\ g(x) \\ \\ XZe \end{pmatrix} = 0, \tag{10}$$

where $x, z \geq 0$, $\nabla_x L(x,y,z) = \nabla f(x) - \nabla g(x)^T y - z$, and $F(x,y,z)$ is a mapping from $\Re^{2n+q}$ to $\Re^{2n+q}$. Note that system (8) differs from the KKT conditions (10) of the initial problem (5), only in the complementarity conditions.

A point $(x(\mu), y(\mu), z(\mu))$ is said to belong to the central path C, if it is the solution of the perturbed KKT conditions (8), for a fixed value of μ. Conditions (8) approximate the KKT conditions (10) increasingly accurately as $\mu \to 0$. Hence, as $\mu \to 0$, the sequence $\{(x(\mu), y(\mu), z(\mu))\}$ converges to the solution of the KKT conditions (10), of the initial constrained problem (5).

Furthermore, primal-dual methods use the Newton or a quasi-Newton method to solve approximately the perturbed KKT conditions (8), for a fixed value of μ. Therefore, the first order change of the above system needs to be found. The k-th Newton iteration for solving (8) can be written as

$$\begin{pmatrix} \nabla^2_{xx} L(x_k, y_k, z_k) & -\nabla g(x_k)^T & -I \\ \\ \nabla g(x_k) & 0 & 0 \\ \\ Z_k & 0 & X_k \end{pmatrix} \begin{pmatrix} \Delta x_k \\ \\ \Delta y_k \\ \\ \Delta z_k \end{pmatrix} = - \begin{pmatrix} \nabla_x L(x_k, y_k, z_k) \\ \\ g(x_k) \\ \\ X_k Z_k e - \mu_k e \end{pmatrix}$$

or in matrix-vector form

$$J(w_k)\Delta w_k = -r(w_k), \tag{11}$$

where $w_k = (x_k, y_k, z_k)$, and $\Delta w_k = (\Delta x_k, \Delta y_k, \Delta z_k)$. The solution of (11) gives a direction vector Δw_k which is used to find the next approximation of the solution of (8). That is, the next iterate is $w_{k+1} = w_k + \mathrm{A}_k \Delta w_k$, where A_k is the diagonal matrix $\mathrm{A}_k = diag(\alpha_{xk} I_n, \alpha_{yk} I_q, \alpha_{zk} I_n)$ and I_n, I_q are the n-th and q-th order identity matrices respectively. The step-lengths α_{xk}, α_{yk}, and α_{zk} belong to the interval $(0, 1]$ and may all be equal to or different from each other.

A unit step along the Newton direction is often not allowed because it violates the non-negativity constraints on x and z in (8). To avoid this violation, the step-sizes α_{xk} and α_{zk} are selected such that the new iterates x_{k+1} and z_{k+1} are strictly positive for all k. When an approximation of the central point, corresponding to the value where μ is fixed, is found, the barrier parameter μ is fixed onto a strictly smaller value and the iterations proceed until μ becomes zero.

4 Description of the Algorithm

The algorithm discussed below solves problem (5) and is based on a sequence of optimization problems characterized by a penalty $c \geq 0$ and a barrier $\mu \geq 0$ parameter. The following assumptions are used throughout the paper.

Assumptions:

A1: The second order derivatives of the objective function f and the constraints g are continuous.

A2: The columns of the matrix $[\nabla g(x), e_i : i \in I_x^0\}$ are linear independent, where $I_x^0 = \{i : \liminf_{k\to\infty} x_k^i = 0,\ i = 1, 2, ..., n\}$ and e_i represents the i-th column of the $n \times n$ identity matrix. Also the sequence $\{x_k\}$ is bounded.

A3: Strict complementarity of the solution $w_* = (x_*, y_*, z_*)$ is satisfied, that is if $z_*^i > 0$ then $x_*^i = 0$, for $i = 1, 2, ..., n$ and vice versa.

A4: The second order sufficiency condition for optimality is satisfied at the solution point, i.e., if for all vectors $0 \neq v \in \Re^n$ such that $\nabla g^i(x_*)^T v = 0$, $i = 1, 2, ..., q$, and $e_i^T v = 0$, for $i \in I_x^0$, then $v^T \nabla_{xx}^2 L(x, y, z) v > 0$.

We note that most nonlinear programming algorithms invoke similar assumptions (see e.g. [8], [13], [32]). Nevertheless, $A1$-$A4$ restrict the applicability of the algorithms to specific problem classes. For example, any algorithm requiring these assumptions would not be applicable to concave minimization problems such as the examples considered in Vanderbei and Shanno [29].

The original equality and inequality constrained optimization problem (5) is approximated by

$$\begin{aligned} &min \quad f(x) + \tfrac{c}{2} \parallel g(x) \parallel_2^2 - \mu \textstyle\sum_{i=1}^n log(x^i) \\ &ST \quad\ g(x) = 0, \end{aligned} \tag{12}$$

for $c, \mu \geq 0$. The objective in (5) is augmented by the penalty and the logarithmic barrier functions. The penalty is used to enforce satisfaction of the equality constraints by adding a high cost to the objective function for infeasible points. The barrier is needed to introduce an interior point method to solve the initial problem (5), since it creates a positive singularity at the boundary of the feasible region. Thus, strict feasibility is enforced, while approaching the optimum solution.

Penalty-barrier methods involve outer and inner iterations [7]. Outer iterations are associated with decreasing the barrier parameter μ, such that μ approaches zero. Inner iterations determine the penalty parameter c and then solve the optimization problem (12) for the corresponding values of μ and c.

The Lagrangian associated with the optimization problem (12) is given by

$$L(x, y; c, \mu) = f(x) + \frac{c}{2} \parallel g(x) \parallel_2^2 - \mu \sum_{i=1}^{n} log(x^i) - g(x)^T y,$$

while the first order optimality conditions are the system of nonlinear equations

$$\begin{pmatrix} \nabla f(x) - \mu X^{-1}e + c\nabla g(x)^T g(x) - \nabla g(x)^T y \\ \\ g(x) \end{pmatrix} = 0,$$

for $x > 0$. By invoking the nonlinear transformation $z = \mu X^{-1}e$ the above conditions become

$$F(x, y, z; c, \mu) = \begin{pmatrix} \nabla f(x) - z + c\nabla g(x)^T g(x) - \nabla g(x)^T y \\ \\ g(x) \\ \\ XZe - \mu e \end{pmatrix} = 0, \tag{13}$$

with $x, z > 0$. For μ fixed, system (13) is solved by using the quasi-Newton method. At the k-th iteration, the Newton system is

$$\begin{pmatrix} H_k & -\nabla g_k^T & -I \\ \\ \nabla g_k & 0 & 0 \\ \\ Z_k & 0 & X_k \end{pmatrix} \begin{pmatrix} \Delta x_k \\ \\ \Delta y_k \\ \\ \Delta z_k \end{pmatrix} = - \begin{pmatrix} \nabla f_k - z_k + c_k \nabla g_k^T g_k - \nabla g_k^T y_k \\ \\ g_k \\ \\ X_k Z_k e - \mu e, \end{pmatrix} \tag{14}$$

where H_k is a positive definite approximation of the Hessian of the augmented Lagrangian defined in (4). In matrix-vector form (14) can be written as

$$J(w_k; c_k)\Delta w_k = -F(w_k; c_k, \mu_k), \tag{15}$$

where $J(w_k; c_k)$ is the Jacobian matrix of the vector function $F(w_k; c_k, \mu_k)$. Equation (15) is different from the corresponding equation (11) due to the introduction of the penalty term in the objective function. The algorithm uses different step-sizes for the primal and dual variables. Hence, the next iterate $w_{k+1} = (x_{k+1}, y_{k+1}, z_{k+1})$ is defined as

$$x_{k+1} = x_k + \alpha_{xk}\Delta x_k, \quad y_{k+1} = y_k + \alpha_{zk}\Delta y_k, \quad z_{k+1} = z_k + \alpha_{zk}\Delta z_k, \tag{16}$$

where α_{xk} and α_{zk} are the step-lengths for the primal variables x and the pair of dual variables y and z, respectively.

To initiate the algorithm, a strictly interior starting point is needed, that is a point $w^0 = (x^0, y^0, z^0)$, with x^0, $z^0 > 0$. By controlling the step lengths α_{xk} and α_{zk}, the algorithm ensures that the generated iterates remain strictly in the interior of the feasible region. Moreover, the algorithm moves from one inner iteration to another inner iteration (i.e., with μ fixed) by seeking to minimize the merit function

$$\Phi(x; c, \mu) = f(x) + \frac{c}{2} \parallel g(x) \parallel_2^2 - \mu \sum_{i=1}^{n} log(x^i), \tag{17}$$

which is basically the objective function of the barrier problem (12). This is achieved by properly selecting the values of the penalty parameter c at each inner iteration. In order to avoid situations where the penalty parameter may grow to large values, we introduce a second merit function defined by the ℓ_2 norm of the KKT residuals of the barrier problem (12). The potential difficulties of the the penalty parameter in the merit function $\Phi(x; c, \mu)$ have been considered by other authors. For example, Bartholomew-Biggs [2] avoids this undesirable situation by constructing search directions based directly on the augmented Lagrangian of the of the barrier problem. As shown later, the monotonic decrease of both merit functions and the rules for determining the primal and dual step-sizes, guarantee that the inner iterates converge to the solution of (12), for a fixed value of μ. Subsequently, by reducing μ, such that $\{\mu\} \to 0$, the optimum of the initial problem (5) is reached.

A detailed description of the algorithm follows. Throughout the algorithm, subscript k indicates variables changed in inner iterations (i.e., while μ is fixed) and superscript l indicates variables changed in outer iterations (i.e., when μ decreases). Superscript i denotes elements of vectors.

Algorithm 1

STEP 0: Initialization:

Choose $\tilde{x}^0, \tilde{z}^0 \in \Re^n$ and $\tilde{y}^0 \in \Re^q$, such that $\tilde{x}^0,\ \tilde{z}^0\ >\ 0$, penalty and barrier parameters $c_0 > 0$, $\mu^0 > 0$ and parameters: $\gamma, \epsilon_0, \eta, \rho \in (0, 1)$, $\delta > 0$.

Set $l = 0$ and $k = 0$. new–merit = **false**

STEP 1: Test for convergence of outer iterations:

If $\| F(\tilde{x}^l, \tilde{y}^l, \tilde{z}^l; c_*, \mu^l) \|_2 \ / (1 + \| (\tilde{x}^l, \tilde{y}^l, \tilde{z}^l) \|_2) \le \epsilon_0$, then Stop.

STEP 2: Start of inner iterations: (μ is fixed to μ^l throughout this step)

Set $(x_k, y_k, z_k) = (\tilde{x}^l, \tilde{y}^l, \tilde{z}^l)$

Step 2.1: Test for convergence of inner iterations:

If $((\| F(x_k, y_k, z_k; c_k, \mu^l) \| \le \eta\mu^l).and.\|g(x)\|^2 \le \epsilon_g)$ then

Set $(\tilde{x}^{l+1}, \tilde{y}^{l+1}, \tilde{z}^{l+1}) = (x_k, y_k, z_k)$ and GoTo step 3

Step 2.2: Solve Newton system (14) to obtain $\Delta w_k = (\Delta x_k, \Delta y_k, \Delta z_k)$

Step 2.3: Penalty parameter selection:

$c_{k+1} = c_k$

If (.not.new–merit) then

$\mathcal{M}_{num} = \Delta x_k^T \nabla f_k - c_k \| g_k \|_2^2 - \mu^l \Delta x_k^T X_k^{-1} e + \| \Delta x_k \|_{H_k}^2$

If $(\mathcal{M}_{num} > 0.\text{and}.\|g_k\|^2 > \epsilon_g)$ then

$c_{k+1} = max\{\frac{\Delta x_k^T \nabla f_k - \mu^l \Delta x_k^T X_k^{-1} e + \|\Delta x_k\|_{H_k}^2}{\|g_k\|_2^2},\ c_k + \delta\}.$

Step 2.4: Step-length selection rules:

Set $\alpha_{xk}^{max} = \min_{1 \le i \le n} \{\frac{-x_k^i}{\Delta x_k^i}\ :\ \Delta x_k^i < 0\}$

If $((\mathcal{M}_{num} > 0.\text{and}.(0 < \|g_k\|^2 \le \epsilon_g)).\text{or}.\text{new–merit})$ then

$\alpha_{zk}^{max} = \min_{1 \le i \le n}\{-\frac{z_k^i}{\Delta z_k^i} | \Delta z_k^i < 0\}$

$\hat{\alpha}_k = \min\{1, \gamma\alpha_{xk}^{max}, \gamma\alpha_{zk}^{max}\}$
Let $\alpha_k = \beta^\theta \hat{\alpha}_k$, where θ is the smallest non-negative integer such that
$\|F(w_k + \alpha_k \Delta w_k)\|^2 - \|F(w_k)\|^2 \leq \rho\alpha_k(\nabla F^t(w_k)F(w_k), \Delta w_k)$
$w_{k+1} = w_k + \alpha_k \Delta w_k$
new–merit = **true**

Else

$\hat{\alpha}_{xk} = \min\{\gamma\alpha_{xk}^{max}, 1\}$.
Let $\alpha_{xk} = \beta^\theta \hat{\alpha}_{xk}$, where θ is the smallest non-negative integer such that
$\Phi(x_{k+1}; c_{k+1}, \mu^l) - \Phi(x_k; c_{k+1}, \mu^l) \leq \rho\, \alpha_{xk} \nabla\Phi(x_k; c_{k+1}, \mu^l)^T \Delta x_k$,
with $x_{k+1} = x_k + \alpha_{xk}\Delta x_k$.
Set $LB_k^i = \min\{\frac{1}{2}m\mu, x_{k+1}^i z_k^i\}$ and $UB_k^i = \max\{2M\mu, x_{k+1}^i z_k^i\}$,
for some $m, M > 0$.
For $i = 1, 2, ..., n$ find: $\alpha_{zk}^i = \max\{\alpha_i : LB_k^i \leq x_{k+1}^i(z_k^i + \alpha_i \Delta z_k^i) \leq UB_k^i\}$
Set $\alpha_{zk} = \min\{1, \min_{1\leq i\leq n}\{\alpha_{zk}^i\}\}$
Set $y_{k+1} = y_k + \alpha_{zk}\Delta y_k$ and $z_{k+1} = z_k + \alpha_{zk}\Delta z_k$.

End if

Step 2.5: Set $k = k + 1$ and GoTo Step 2.1

STEP 3: Reduce barrier parameter as described in section § 4.4.
new–merit = **false**

STEP 4: Set $l = l + 1$ and GoTo Step 1.

4.1 Penalty parameter selection rule

The penalty parameter c plays an important role in the algorithm. At each iteration, its value is determined such that a descent property is ensured for the merit function $\Phi(x; c, \mu)$. For μ fixed, the gradient of Φ at the k-th iteration is

$$\nabla\Phi(x_k; c_k, \mu) = \nabla f_k + c_k \nabla g_k^T g_k - \mu X_k^{-1} e. \tag{18}$$

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

$$\Delta x_k^T \nabla\Phi(x_k; c_k, \mu) \leq 0. \tag{19}$$

By considering the second equation of the Newton system (14), the directional derivative $\Delta x_k^T \nabla\Phi(x_k; c_k, \mu)$ can be written as

$$\Delta x_k^T \nabla\Phi(x_k; c_k, \mu) = \Delta x_k^T \nabla f(x_k) - c_k \parallel g_k \parallel^2 - \mu \Delta x_k^T X_k^{-1} e, \tag{20}$$

where c_k is the value of the penalty parameter at the beginning of the k-th iteration. Since the barrier parameter μ is fixed throughout the inner iterations, we can deduce from (20) that the sign of $\Delta x_k^T \nabla\Phi(x_k; c_k, \mu)$ depends on the value of the penalty parameter. If c_k is not large enough then the descent property (19) may not be satisfied. Thus, a new value

$c_{k+1} > c_k$ must be determined to guarantee the satisfaction of the descent property. The next lemmas show that Algorithm 1 chooses the value of the penalty parameter in such a way that Δx_k is descent direction for the merit function.

Let $\epsilon_g > 0$ denote the finite precision (set to 10^{-8} in the numerical experiments of Section 6). Thus, we have a worst case feasibility precision

$$\|g(x)\|^2 > \epsilon_g. \tag{21}$$

In Lemmas 1 and 4 we show that descent is always guaranteed if (21) holds or $g(x) = 0$ and the penalty parameter $c_k = c_k(\epsilon_g)$ remains finite.

If, at some inner iteration k, we have $0 < \|g(x_k)\|^2 \leq \epsilon_g$ and the descent condition (19) is not satisfied, then a switch to the following merit function

$$\|F(x, y, z; c, \mu)\|^2 \tag{22}$$

is performed for all consecutive inner iterations. Once the convergence of the inner iteration is achieved, the algorithm returns to minimizing merit function (17). This is a variation of the so called "watchdog" technique, which was first suggested by Chamberlain *et al.* in [6]. In the context of interior point methods it was also used by Gay *et al.* in [13]. The convergence criteria for (22) has been well established [8, 35].

Lemma 1 *Let f and g be differentiable functions and let there exist small $\epsilon_g > 0$, such that $\|g_k\|^2 > \epsilon_g$. If Δx_k is calculated by solving the Newton system (14) and c_{k+1} is chosen as in step 2.3 of Algorithm 1 then Δx_k is a descent direction for the merit function Φ at the current point x_k. Furthermore*

$$\Delta x_k^T \nabla \Phi(x_k; c_{k+1}, \mu) \leq - \| \Delta x_k \|_{H_k}^2 \leq 0. \tag{23}$$

Proof In step 2.3, Algorithm 1 initially checks the inequality

$$\Delta x_k^T \nabla f_k - c_k \| g_k \|^2 - \mu \Delta x_k^T X_k^{-1} e + \| \Delta x_k \|_{H_k}^2 \leq 0. \tag{24}$$

If (24) is satisfied then by setting $c_{k+1} = c_k$ and re-arranging (24) we obtain (23). On the other hand if (24) is not satisfied, by setting

$$c_{k+1} = \max\{\frac{\Delta x_k^T \nabla f_k - \mu \Delta x_k^T X_k^{-1} e + \| \Delta x_k \|_{H_k}^2}{\| g_k \|^2}, c_k + \delta\}, \quad \delta > 0,$$

and substituting it into (20) it can be verified that (23) also holds. •

Remark 1 *The role of the parameter δ in the definition of c_{k+1} is to guarantee that the penalty parameter increases by at least a certain amount every time it needs to be updated. We have observed that this technique results in a better performance of Algorithm 1.*

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 Φ when $g_k = 0$, i.e., when feasibility of the equality constraints has been achieved.

Lemma 2 *Let f and g be differentiable functions and let $\Delta w_k = (\Delta x_k, \Delta y_k, \Delta z_k)$ be the Newton direction taken by solving system (14). If for some or all iterations k, $g_k = 0$, then the descent property (23) is satisfied for any choice of the penalty parameter $c_k \in [0, \infty)$.*

Proof If $g_k = 0$ then (20) yields

$$\Delta x_k^T \nabla \Phi(x_k; c_k, \mu) = \Delta x_k^T \nabla f_k - \mu \Delta x_k^T X_k^{-1} e \tag{25}$$

and the second equation of the Newton system (14) becomes

$$\nabla g_k \Delta x_k = 0. \tag{26}$$

Furthermore, solving the third equation of (14) for Δz_k we have

$$\Delta z_k = -X_k^{-1} Z_k \Delta x_k - z_k + \mu X_k^{-1} e. \tag{27}$$

Substituting Δz_k into the first equation of (14) yields

$$\nabla f_k - c_k \nabla g_k^T g_k + \mu X_k^{-1} e = -(H_k + X_k^{-1} Z_k) \Delta x_k + \nabla g_k^T (y_k + \Delta y_k). \tag{28}$$

Pre-multiplying (28) by Δx_k^T yields

$$\Delta x_k^T \nabla f_k - c_k \Delta x_k^T \nabla g_k^T g_k + \mu \Delta x_k^T X_k^{-1} e = -\Delta x_k^T (H_k + X_k^{-1} Z_k) \Delta x_k + \Delta x_k^T \nabla g_k^T (y_k + \Delta y_k).$$

Using (26), the above equation becomes

$$\Delta x_k^T \nabla f_k + \mu \Delta x_k^T X_k^{-1} e = -\Delta x_k^T (H_k + X_k^{-1} Z_k) \Delta x_k, \tag{29}$$

and from (25), equation (29) yields

$$\Delta x_k^T \nabla \Phi(x_k; c_k, \mu) = -\Delta x_k^T (H_k + X_k^{-1} Z_k) \Delta x_k. \tag{30}$$

From the fact that x_k and z_k are strictly positive and Assumption (A4), we have

$$-\Delta x_k^T (H_k + X_k^{-1} Z_k) \Delta x_k < -\Delta x_k^T H_k \Delta x_k. \tag{31}$$

From (30) and (31) it is derived that (23) holds for every $c_k \in [0, \infty)$. •

Lemma 3 *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 2.3. Also, Δx_k is still a descent direction for the merit function Φ at x_k.*

Proof In the previous lemma it was proved that the descent property (23) is satisfied for $g_k = 0$. This basically means that the condition in step 2.3 of Algorithm 1 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 (23) still holds •

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

Proof We note that when $\Delta x_k = 0$, the Newton direction necessarily implies that $g_k = 0$ and therefore this case is explicitly covered by Lemmas 2 and 3. •

Lemma 4 *Let* f *and* g *be continuously differentiable functions and*

$$\Delta x_k^T \nabla f_k - \mu \Delta x_k^T X_k^{-1} e + \| \Delta x_k \|_{H_k}^2 \leq \mathcal{M}^* \leq \infty.$$

Then for μ *fixed:*

(i) there always exists a constant $c_{k+1} \geq 0$*, satisfying step* 2.3 *of Algorithm* 1.

(ii) assuming that the sequence $\{x_k\}$ *is bounded,* c_k *is increased finitely often, that is, there exists an integer* $k_* \geq 0$ *such that for all* $k \geq k_*$*, we have* $c_* \in [0, \infty)$.

Proof Part (i) is a direct consequence of Lemmas 1 - 3 and Corollary 1, since a finite value c_{k+1} is always generated, in step 2.3. 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 2.3 we can deduce that, if $c_k \to \infty$, then $\|g_k\|^2 \to 0$. Hence, there exists an integer k_1 such that for all $k \geq k_1$ we have: $0 < \|g_k\|^2 \leq \epsilon_g$. As can be seen in step 2.4, however, in the case where $0 < \|g_k\|^2 \leq \epsilon_g$ the algorithm stops increasing the penalty parameter since it switches to the second merit function. Therefore the maximum value that c_k can take is: $c_* = c_{k_1} = \mathcal{M}^*/\epsilon_g$ where $\mathcal{M}_{num}$ and c_* are finite values. Hence we have $c_* < \infty$. 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 $c_k < \infty$. •

4.2 Primal step-size rule

In step 2.4 of the algorithm we adopt Armijo's rule to determine the new iterate x_{k+1}. The maximum allowable step-size is determined by the boundary of the feasible region and is given by

$$\alpha_{xk}^{max} = \min_{1 \leq i \leq n} \{ \frac{x_k^i}{-\Delta x_k^i} \ : \ \Delta x_k^i < 0 \}.$$

This is indeed the maximum allowed step, since α_{xk}^{max} gives an infinitely large value to at least one term of the logarithmic barrier function $\sum_{i=1}^n log(x_{k+1}^i)$. However, if the step-size is in the interval $[0, \alpha_{xk}^{max})$ then the next primal iterate x_{k+1} is strictly feasible and none of the logarithmic terms becomes infinitely large.

We take as initial step $\hat{\alpha}_{xk}$ a number very close to α_{xk}^{max} and we ensure that it is never greater than one, i.e., $\hat{\alpha}_{xk} = min\{\gamma \alpha_{xk}^{max}, 1\}$, with $\gamma \in (0, 1)$. The final step is $\alpha_{xk} = \beta^\theta \hat{\alpha}_{xk}$, where θ is the first non-negative integer for which Armijo's rule is satisfied and the factor β is usually chosen in the interval $[0.1, 0.5]$, depending on the confidence we have on the initial step $\hat{\alpha}_{xk}$. The value of the parameter ρ is usually chosen in the interval $[10^{-5}, 10^{-1}]$.

4.3 Dual step-size rule

In this section we discuss the calculation of the step-size of the dual variables z. The strategy uses the information provided by the new primal iterate x_{k+1}, in order to find the new iterate z_{k+1}. It is a modification of the strategy suggested by Yamashita [32] and Yamashita and Yabe [33].

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

$$\alpha^i_{zk} = \max\{\alpha > 0 :\ LB^i_k \leq (x^i_k + \alpha_{xk}\Delta x^i_k)(z^i_k + \alpha\Delta z^i_k) \leq UB^i_k\}. \tag{32}$$

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

$$LB^i_k = \min\{\frac{1}{2}m\mu, (x^i_k + \alpha_{xk}\Delta x^i_k)z^i_k\} \text{ and } UB^i_k = \max\{2M\mu, (x^i_k + \alpha_{xk}\Delta x^i_k)z^i_k\}, \tag{33}$$

where the parameters m and M are chosen such that

$$0 < m \leq \min\{1, \frac{(1-\gamma)(1-\frac{\gamma}{(M_0)^\mu})\min_i\{x^i_k z^i_k\}}{\mu}\}, \tag{34}$$

and

$$M \geq \max\{1, \frac{\max_i\{x^i_k z^i_k\}}{\mu}\} > 0, \tag{35}$$

with $\gamma \in (0, 1)$ and M_0 a positive large number. These two parameters are always fixed to constants which satisfy (34) and (35), while μ is fixed. The values of m and M change when the barrier parameter μ is decreased.

The common dual step length α_{zk} is the minimum of all individual step lengths α^i_{zk} with the restriction of being always not more than one, namely

$$\alpha_{zk} = min\{1, \min_{1\leq i\leq n}\{\alpha^i_{zk}\}\}.$$

The step-size for the dual variables y can be either $\alpha_{yk} = 1$ or $\alpha_{yk} = \alpha_{zk}$.

The difference between the present step-size rule and the one proposed in [33] lies in the definition of the lower bounds LB^i_k, $i = 1, 2, ..., n$ of the box constraints (32). In particular, the term $1 - \gamma/(M_0)^{\mu_k} \in (0, 1)$ in the definition of the parameter m, given by (34), results in the lower bounds LB^i_k being smaller than the corresponding bounds defined in [33]. Consequently, the step lengths α_{zk} are larger than those in [33]. Also by noting that

$$\lim_{\mu\to 0}(1 - \gamma/(M_0)^{\mu_k}) = 1 - \gamma$$

and

$$z^i_k + \alpha_{zk}\Delta z^i_k \geq (1-\gamma)z^i_k > 0, \text{ for all } i = 1, 2, ...n$$

it can be shown that asymptotically the algorithm accepts the maximum allowable step for the dual variables.

4.4 Barrier parameter selection rule

The barrier parameter μ plays an important role in interior point methods. By gradually reducing it to zero, the algorithms converge to the optimum solution. The efficiency of interior point algorithms heavily depends on the speed by which μ approaches zero. Very fast and premature reduction of μ, however, can cause the failure of these algorithms [31].

The strategy which is used to reduce the barrier parameter in Algorithm 1, derives from two other strategies, presented by Lasdon *et al.* [17] and Gay *et al.* [13]. The basic characteristic of our strategy is that, it determines the new value of μ, by taking into consideration the distance of the current point (x_k, y_k, z_k) from the central path and the optimum solution of the initial problem. The barrier reduction strategy is shown in Figure 1.

$\mu^{l+1} = \min\{0.95\mu^l, 0.01(0.95)^k \parallel F(x_k, y_k, z_k) \parallel_2\}$
If $\parallel F(x_k, y_k, z_k; \mu^l) \parallel_2 \leq 0.1\eta\mu^l$ then
  If $\mu^l < 10^{-4}$ then
    $\mu^{l+1} = \min\{0.85\mu^l, 0.01(0.85)^{k+2\sigma} \parallel F(x_k, y_k, z_k) \parallel_2\}$
  Else
    $\mu^{l+1} = \min\{0.85\mu^l, 0.01(0.85)^{k+\sigma} \parallel F(x_k, y_k, z_k) \parallel_2\}$

Figure 1: Barrier reduction rule.

The vectors $F(x_k, y_k, z_k; \mu^l)$ and $F(x_k, y_k, z_k)$ represent the perturbed and unperturbed KKT conditions of the initial problem (5) and are defined by (8) and (10) respectively. The threshold determining whether the barrier parameter is going to decrease is initially checked in Step 2.1 of Algorithm 1. If the current point is close enough to the central path (i.e., if $\parallel F(x_k, y_k, z_k; \mu^l) \parallel_2 \leq 0.1\eta\mu^l$) and the optimum solution (i.e., if $\mu^l < 10^{-4}$), then the barrier parameter is reduced faster, since it is multiplied by the factor $(0.85)^{2\sigma}$, where $\sigma > 0$. If it is only close to the central path and not close to the optimum solution then the barrier parameter is still reduced but not as fast as before, since it is now multiplied by the larger factor $(0.85)^{\sigma}$. Hence, σ can be thought of as a parameter which accelerates the decrease of μ at appropriate points. In our numerical tests, this barrier reduction rule has performed very effectively.

5 Global Convergence

In this section we show that the algorithm is globally convergent, because it always guarantees progress towards a solution from any starting point. El-Bakry *et al.* [8] and Yamashita [32] have also shown global convergence for their primal-dual algorithms. In [8] the global convergence is achieved by determining a single step-size for all the variables. The Armijo rule is used to guarantee that the Euclidean norm of the KKT conditions, which plays the role of a merit function, is reduced at each iteration.

Algorithm 1 is closely related to that in [32], since both algorithms use the same strategies to determine different step-sizes for the primal and dual variables. They use, however, different merit functions to guarantee global convergence. In [32], the non-differentiable merit function

$$\phi(x;\hat{\rho},\mu) = f(x) + \hat{\rho}\sum_{i=1}^{q}|g^i(x)| - \mu\sum_{i=1}^{n}\log(x^i)$$

is used, while we have chosen the merit function defined by (17), which has the useful property to be continuously differentiable. An advantage of the differentiability of the merit function Φ is that the adaptive strategy, developed by Rustem in [22], is used to determine the penalty parameter c, to ensure a descent property for Φ, leading to the global convergence of the algorithm.

We show that, while the barrier parameter is fixed to a value μ^l, the algorithm produces iterates $w_k(\mu^l) = (x_k(\mu^l), y_k(\mu^l), z_k(\mu^l))$, for $k \geq 0$, which are bounded and converge to a point $w_*(\mu^l) = (x_*(\mu^l), y_*(\mu^l), z_*(\mu^l))$ such that

$$\| F(x_*(\mu^l), y_*(\mu^l), z_*(\mu^l); c_*, \mu^l) \| = 0,$$

where $F(x,y,z;c,\mu)$ is the vector of the perturbed KKT conditions, defined in (13). In other words, we show that the inner iterations (2.1) - (2.5) of Algorithm 1, converge to a solution of the perturbed KKT conditions. For simplicity we suppress the index l, and we use w_k instead of $w_k(\mu^l)$ to denote the iterates produced while $\mu = \mu^l$.

The basic result of Lemmas 1 to 4 is that the direction Δx_k, taken from the solution of the Newton system (14), is a descent direction for the merit function Φ at the current point x_k, that is inequality (23) holds. In the next theorem we show that the sequence $\{\Phi(x_k; c_*, \mu)\}$ is monotonically decreasing if the barrier parameter μ is fixed. We also show that the step α_{xk}, chosen in step 2.4 is always positive.

Theorem 1 *Assume that the following conditions hold*

i. the objective function f and the constraints g are twice continuously differentiable,

ii. the approximate Hessian matrix H_k, is such that, for every iteration k and for every vector $v \in \Re^n$ there exist constants $M' > m' > 0$, such that

$$m' \| v \|_2^2 \leq v^T H_k v \leq M' \| v \|_2^2,$$

iii. for each iteration k, there exists a triple $(\Delta x_k, \Delta y_k, \Delta z_k)$, as a solution to the Newton system (14),

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 the penalty parameter restriction in step 2.3

$$\Delta x_k^T \nabla f_k - c_k(\epsilon_g) \| g_k \|_2^2 - \mu \Delta x_k^T X_k^{-1} e + \| \Delta x_k \|_{H_k}^2 \leq 0$$

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

Then the step-size computed in step 2.4 is such that $\alpha_{xk} \in (0,1]$ *and hence the sequence* $\{\Phi(x_k; c_*, \mu)\}$ *is monotonically decreasing, for* $k \geq k_*$ *and* μ *fixed.*

Proof Consider the case $\|g_k\|^2 \notin (0, \epsilon_g)$ and first order approximation with remainder of the function $\Phi(x; c_*, \mu)$ around the point $x_{k+1} = x_k + \alpha_{xk}\Delta x_k$

$$\Phi(x_{k+1}; c_*, \mu) - \Phi(x_k; c_*, \mu) =$$

$$\alpha_{xk}\Delta x_k^T \nabla\Phi(x_k; c_*, \mu) + \alpha_{xk}^2 \int_0^1 (1-t)\Delta x_k^T \nabla_x^2 \Phi(x_k + t\alpha_{xk}\Delta x_k; c_*, \mu)\Delta x_k \, dt.$$

The above equation, after adding and subtracting the Hessian matrix H_k in the remainder, yields

$$\Phi(x_{k+1}; c_*, \mu) - \Phi(x_k; c_*, \mu) \ \leq \ \alpha_{xk}\Delta x_k^T \nabla\Phi(x_k; c_*, \mu) \ + \ \tfrac{1}{2}\alpha_{xk}^2 \Delta x_k^T H_k \Delta x_k$$

$$+\alpha_{xk}^2 \int_0^1 (1-t) \mid \Delta x_k^T \ (\ \nabla_x^2 \Phi(x_k + t\alpha_{xk}\Delta x_k; c_*, \mu) - H_k \ ) \ \Delta x_k \mid dt. \tag{36}$$

Furthermore, inequality (36) can take the following equivalent form

$$\Phi(x_{k+1}; c_*, \mu) \ - \ \Phi(x_k; c_*, \mu) \ \leq$$

$$\alpha_{xk}\Delta x_k^T \nabla\Phi(x_k; c_*, \mu) \ + \ \frac{\alpha_{xk}^2}{2}\Delta x_k^T H_k \Delta x_k + \alpha_{xk}^2 \psi_k \parallel \Delta x_k \parallel_2^2, \tag{37}$$

where

$$\psi_k = \int_0^1 (1-t) \ \parallel \nabla_x^2 \Phi(x_k + t\alpha_{xk}\Delta x_k; c_*, \mu) - H_k \ \parallel_2 \ dt.$$

Furthermore, from assumption (ii) we have

$$\parallel \Delta x_k \parallel_2^2 \leq \frac{1}{m'}\Delta x_k^T H_k \Delta x_k, \tag{38}$$

and from Lemmas 1 and 2

$$\Delta x_k^T H_k \Delta x_k \leq -\Delta x_k^T \nabla\Phi(x_k; c_*, \mu). \tag{39}$$

Substituting (38) and (39) into (37) yields

$$\Phi(x_{k+1}; c_*, \mu) - \Phi(x_k; c_*, \mu) \ \leq \alpha_{xk}\Delta x_k^T \nabla\Phi(x_k; c_*, \mu)(1 - \alpha_{xk}(\frac{1}{2} + \frac{\psi_k}{m'})). \tag{40}$$

The scalar ρ in Armijo's rule in step 2.4 determines a step-length α_{xk} such that

$$\rho \leq 1 - \alpha_{xk}(\frac{1}{2} + \frac{\psi_k}{m'}) \leq \frac{1}{2}.$$

Since from Lemmas 1 to 4 we always have $\Delta x_k^T \nabla\Phi(x_k; c_*, \mu) \leq 0$, there must exist $\alpha_{xk} \in (0,1]$, to ensure (40) and Armijo's rule in step 2.4. Assume that α^0 is the largest step in the interval $(0,1]$ satisfying both (40) and Armijo's rule. Consequently for every $\alpha \leq \alpha^0$, inequality (40) and Armijo's rule are also satisfied. Hence the strategy in step 2.4 always selects a step-length $\alpha_{xk} \in [\beta\alpha^0, \alpha^0]$, where $0 < \beta \leq 1$. From the above analysis, it follows that the sequence $\{\Phi(x_k; c_*, \mu)\}$ is monotonically decreasing. •

Remark 2 *The results of the above theorem can be proved to hold before the penalty parameter* c_k *achieves a constant value* c_*. *This can be done by considering the difference* $\Phi(x_{k+1}; c_{k+1}, \mu) - \Phi(x_k; c_{k+1}, \mu)$ *and the Taylor expansion of the function* $\Phi(x; c_{k+1}, \mu)$ *instead of* $\Phi(x; c_*, \mu)$. *In the above theorem we chose to prove the case where* $c_k = c_*$ *has been achieved, in order to show that asymptotically,* Φ *is monotonically decreasing and the strategy in step* 2.4 *selects a step-length* $\alpha_{xk} \in (0, 1]$.

An immediate consequence of the above theorem is that the sequence $\{x_k\}$ is bounded away from zero. This is established in the following corollary.

Corollary 2 *The sequence* $\{x_k\}$ *of primal variables generated by Algorithm 1, with* μ *fixed, is bounded away from zero.*

Proof Assume to the contrary that the sequence $\{x_k\} \to 0$. Then $\{-\sum_{i=1}^{n} \log(x_k^i)\} \to \infty$. 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 $\{\Phi(x_k; c_*, \mu)\} \to \infty$ which contradicts the monotonic decrease of Φ. •

The following lemma, proved by Yamashita in [32], shows that the dual step-size rule, used by the Algorithm 1, generates iterates z_k which are also bounded above and away from zero.

Lemma 5 *While* μ *is fixed, the lower bounds* LB_k^i *and the upper bounds* UB_k^i, $i = 1, 2, ..., n$, *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 [32]. •

Having established that the sequences of iterates $\{x_k\}$ and $\{z_k\}$ are bounded above and away from zero, we show that the iterates $\{y_k\}$, $k \geq 0$ are also bounded. In particular Lemma 7 shows that if at each iteration of the algorithm we take a unit step along the direction Δy_k, then the resulting sequence $\{y_k + \Delta y_k\}$ is bounded. In addition to this, Lemma 7 also shows that the Newton direction $\Delta w_k = (\Delta x_k, \Delta y_k, \Delta z_k)$ is bounded, while μ is fixed. We first establish the following technical result.

Lemma 6 *Let* w_k *be a sequence of vectors generated by Algorithm* 1 *for* μ *fixed. Then the matrix sequence* $\{\Theta_k^{-1}\}$ *is bounded, where*

$$\Theta_k = \begin{pmatrix} 0 & \nabla g_k \\ -\nabla g_k^T & H_k + X_k^{-1} Z_k \end{pmatrix}.$$

Proof The inverse of the partitioned matrix Θ_k is

$$\Theta_k^{-1} = \begin{pmatrix} [\nabla g_k \Omega_k \nabla g_k^T]^{-1} & -[\nabla g_k \Omega_k \nabla g_k^T]^{-1} \nabla g_k \Omega_k \\ \Omega_k \nabla g_k^T [\nabla g_k \Omega_k \nabla g_k^T]^{-1} & \Omega_k - \Omega_k \nabla g_k^T [\nabla g_k \Omega_k \nabla g_k^T]^{-1} \nabla g_k \Omega_k \end{pmatrix},$$

where $\Omega_k = (H_k + X_k^{-1} Z_k)^{-1}$. According to Assumption (A4), Corollary 2 and Lemma 5, the matrices Ω_k and $[\nabla g_k \Omega_k \nabla g_k^T]^{-1}$ exist and are bounded. Hence the matrix Θ_k^{-1} is bounded, since all matrices involved in it are bounded. •

Lemma 7 *Let w_k is a sequence of vectors generated by Algorithm* 1 *for μ fixed. Then the sequence of vectors $\{(\Delta x_k, y_k + \Delta y_k, \Delta z_k)\}$ is bounded.*

Proof Solving the third equation of the Newton system (14) for Δz_k yields

$$\Delta z_k = -z_k + \mu X_k^{-1} e - X_k^{-1} Z_k \Delta x_k. \tag{41}$$

Substituting (41) into the first equation of (14) and re-arranging the first two equations, yields the following reduced system

$$\begin{pmatrix} 0 & \nabla g_k \\ -\nabla g_k^T & H_k + X_k^{-1} Z_k \end{pmatrix} \begin{pmatrix} y_k' \\ \Delta x_k \end{pmatrix} = - \begin{pmatrix} g_k \\ \nabla f_k + c_k \nabla g_k^T g_k - \mu X_k^{-1} e \end{pmatrix} \tag{42}$$

where $y_k' = y_k + \Delta y_k$. From the previous lemma we have that the inverse of the matrix in the left side of (42) exists and is bounded. Hence the sequences $\{\Delta x_k\}$ and $\{y_k'\}$ are also bounded. Considering now (41), we can easily deduce that the sequence $\{\Delta z_k\}$ is bounded, since it is a sum of bounded sequences. •

Lemmas 8 and 9 provide the necessary results needed by Theorem 2, which shows that the sequence of $\{w_k\}$ converges to a point $w_* = (x_*, y_*, z_*)$, satisfying the KKT conditions of problem (12).

Lemma 8 *Let the assumptions of Theorem 1 be satisfied and the barrier parameter μ is fixed. Also suppose, for some iteration $k_0 \geq 0$, the level set*

$$S_1 = \{x \in \Re_+^n : \Phi(x; c_*, \mu) \leq \Phi(x_{k_0}; c_*, \mu)\} \tag{43}$$

is compact. Then for all $k \geq k_0$ we have

$$\lim_{k \to \infty} \Delta x_k^T \nabla \Phi(x_k; c_*, \mu) = 0. \tag{44}$$

Proof The scalar $\rho \in (0, 1/2)$ in the step-size strategy at step 2.4, determines α_{xk} such that

$$\rho \leq 1 - \alpha_{xk}(\frac{1}{2} + \frac{\psi_k}{m'}) \leq \frac{1}{2},$$

and by solving for α_{xk} we obtain

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

Hence the largest value that the step-length α_{xk} can take and still satisfy Armijo's rule in step 2.4 is

$$\alpha_{xk}^0 = \min\{1, \frac{1-\rho}{1/2 + \psi_k/m'}\}.$$

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 merit function Φ is twice continuously differentiable and the level set S_1 is bounded, there exists a scalar $\bar{M} < \infty$ such that

$$\psi_k = \int_0^1 (1-t) \parallel \nabla_x^2 \Phi(x_k + t\alpha_{xk}\Delta x_k; c_*, \mu) - H_k \parallel_2 dt \leq \bar{M} < \infty.$$

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 1 and 2 we have

$$\Phi(x_{k+1}; c_*, \mu) - \Phi(x_k; c_*, \mu) \leq \rho\alpha_{xk}\nabla\Phi(x_k; c_*, \mu)^T \Delta x_k < 0. \tag{45}$$

From our assumption that the level set S_1 is bounded, it can be deduced that

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

Consequently, from (45)

$$\lim_{k\to\infty} (\rho\alpha_{xk}\nabla\Phi(x_k; c_*, \mu)^T \Delta x_k) = 0.$$

Finally, since ρ, $\alpha_{xk} > 0$ it can be deduced that (44) holds. •

Lemma 9 *Let the assumptions of the previous lemma hold. Then*

$$\lim_{k\to\infty} \parallel \Delta x_k \parallel_{H_k}^2 = 0. \tag{46}$$

Proof From (23) we have

$$-\nabla\Phi(x_k; c_*, \mu)^T \Delta x_k \geq \parallel \Delta x_k \parallel_{H_k}^2 . \tag{47}$$

Hence from (47) and (44) we have that (46) holds. •

Theorem 2 *Let the assumptions of the previous lemma hold and let $\epsilon_g = 10^{-8}$. Then the algorithm converges, in the limit, to a point satisfying*

$$F(x,y,z;c,\mu) = \begin{pmatrix} \nabla f(x) - z + c\nabla g(x)^T g(x) - \nabla g(x)^T y \\ g(x) \\ XZe - \mu e \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \qquad (48)$$

for μ fixed.

Proof Consider the case $\|g_k\|^2 \notin (0, \epsilon_g)$ and let $x_*(\mu)$, $z_*(\mu) \in \Re^n$ and $y_*(\mu) \in \Re^q$ be such that $\lim_{k\to\infty} x_k = x_*(\mu)$, $\lim_{k\to\infty} z_k = z_*(\mu)$, and $\lim_{k\to\infty} y_k = y_*(\mu)$, $\forall k \geq k_*$, $k \in K \subseteq \{1,2,...\}$. The existence of such points is ensured since by Assumption A2 and Lemmas 5 and 7, the sequence $\{(x_k(\mu), y_k(\mu), z_k(\mu))\}$ is bounded for μ fixed, and by Theorem 1 the algorithm always decreases the merit function Φ sufficiently at each iteration, thereby ensuring $x_k \in S_1$, with S_1 compact.

We first prove that for k sufficiently large, the dual step, α_{zk}, becomes the unit one, by showing that

$$\lim_{k\to\infty} \| z_k + \Delta z_k - \mu X_{k+1}^{-1} e \| = 0 \qquad (49)$$

By adding $-\mu X_{k+1}^{-1} e$ to both sides of (41) we have

$$\| z_k + \Delta z_k - \mu X_{k+1}^{-1} e \| \leq \| -X_k^{-1} Z_k \| \; \| \Delta x_k \| + \mu \| X_k^{-1} - X_{k+1}^{-1} \| \; \| e \| \qquad (50)$$

Moreover

$$\begin{aligned} \| X_k^{-1} - X_{k+1}^{-1} \|^2 &\leq n \max_{1\leq i\leq n} \{ (\frac{1}{x_k^i} - \frac{1}{x_{k+1}^i})^2 \} \\ &= n \max_{1\leq i\leq n} \{ \frac{(\alpha_{xk})^2 (\Delta x_k^i)^2}{(x_k^i)^2 \; (x_{k+1}^i)^2} \} \end{aligned}$$

Since we always have $\alpha_{xk} \in (0,1]$, $(\Delta x_k^i)^2 \leq \| \Delta x_k \|^2$ and the sequence $\{x_k\}$ bounded away from zero, from the above inequality and (46) we can derive

$$\lim_{k\to\infty} \| X_k^{-1} - X_{k+1}^{-1} \|^2 \leq n \lim_{k\to\infty} \max_{1\leq i\leq n} \{ \frac{\| \Delta x_k \|^2}{(x_k^i)^2 \; (x_{k+1}^i)^2} \} = 0 \qquad (51)$$

Hence letting $k \to \infty$ in (50), and using (46) and (51) we can deduce that (49) holds. Consequently, $z_{k+1} = z_k + \Delta z_k$, for k sufficiently large.

Furthermore, using (41) and for k sufficiently large, the complementarity condition becomes

$$X_{k+1} z_{k+1} = X_{k+1}(z_k + \Delta z_k) = X_{k+1} X_k^{-1} (-Z_k \Delta x_k + \mu e) \qquad (52)$$

From (46) and the fact that the elements of the diagonal matrix $X_{k+1} X_k^{-1}$ can be written as

$$\frac{x_{k+1}^i}{x_k^i} = 1 + \alpha_{xk} \frac{\Delta x_k^i}{x_k^i}, \qquad \text{for all } i = 1, 2, ..., n,$$

we can derive that

$$\lim_{k\to\infty} X_{k+1}X_k^{-1} = I_n \tag{53}$$

where I_n is the $n \times n$ identity matrix. Letting $k \to \infty$ in (52), and using (46) and (53) yields

$$\lim_{k\to\infty} X_{k+1}z_{k+1} = X_*(\mu)z_*(\mu) = \mu e \tag{54}$$

Also for $k \to \infty$, the second equation of the Newton system (14) and (46) yield

$$\lim_{k\to\infty} (\nabla g_k \Delta x_k) = g(x_*(\mu)) = 0. \tag{55}$$

The first equation of the Newton system (14) can be written as

$$\nabla f_k - \nabla g_k^T y_{k+1} + c_* \nabla g_k^T g_k - \mu X_k^{-1} e = -(H_k + X_k^{-1}Z_k)\Delta x_k$$

where $y_{k+1} = y_k + \Delta y_k$. Letting $k \to \infty$, and using (46) the above equation yields

$$\lim_{k\to\infty} \| \nabla f_k - \nabla g_k^T y_{k+1} + c_* \nabla g_k^T g_k - \mu X_k^{-1} e \| = 0 \tag{56}$$

From the assumptions that the functions f and g have continuous gradients and ∇g_k^T has full column rank and using (51), equation (56) yields

$$\lim_{k\to\infty} \| \nabla f_{k+1} - \nabla g_{k+1}^T y_{k+1} + c_* \nabla g_{k+1}^T g_{k+1} - \mu X_{k+1}^{-1} e \| = 0$$

or equivalently

$$\nabla f(x_*(\mu)) - \nabla g(x_*(\mu))^T y_*(\mu) + c_* \nabla g(x_*(\mu))^T g(x_*(\mu)) - \mu X_*(\mu)^{-1} e = 0 \tag{57}$$

From (57), (55) and (54) we can conclude that the vector $(x_*(\mu), y_*(\mu), z_*(\mu))$ is a solution of the perturbed KKT conditions (13).

The convergence result for $\|g_k\|^2 \in (0, \epsilon_g)$ is a consequence of El-Bakry *et al.* [8] and Zakovic *et al.* [35]. •

An immediate consequence of Theorem 2 is that, for any convergent subsequence, produced by the algorithm, for $\mu = \mu^l$, there is an iteration $\tilde{k}$, such that

$$\| F(x_{\tilde{k}}, y_{\tilde{k}}, z_{\tilde{k}}; c_*, \mu) \| \leq \eta\mu, \tag{58}$$

for all $k \geq \tilde{k}$, where $\eta \geq 0$ and $F(x, y, z; c, \mu)$ is given by (13). At this point, we record the value of the current iterate

$$(\tilde{x}^l, \tilde{y}^l, \tilde{z}^l) = (x_{\tilde{k}}, y_{\tilde{k}}, z_{\tilde{k}}),$$

and set μ to a smaller value $\mu^{l+1} < \mu^l$. Therefore a sequence of approximate central points $\{(\tilde{x}^l, \tilde{y}^l, \tilde{z}^l)\}$ is generated.

In the remaining part of this section, we show that the sequence of approximate central points converges to a KKT point $(\tilde{x}^*, \tilde{y}^*, \tilde{z}^*)$ of the initial constrained optimization problem (5).

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

$$S_2(\epsilon) = \{(\tilde{x}^l, \tilde{y}^l, \tilde{z}^l) : \epsilon \leq \| F(\tilde{x}^l, \tilde{y}^l, \tilde{z}^l; c_*, \mu^l) \| \leq \| F(\tilde{x}^0, \tilde{y}^0, \tilde{z}^0; c_*, \mu^0) \|, \ \forall \mu^l < \mu^0\}.$$

If $\epsilon > 0$ then the step-size rules, described in section 4 guarantee that $\tilde{x}^l, \tilde{z}^l \in S_2(\epsilon)$ are bounded away from zero, for $l \geq 0$. Consequently $(\tilde{x}^l)^T \tilde{z}^l$ is also bounded away from zero in $S_2(\epsilon)$. The following lemma shows that the sequence $\{\tilde{y}^l\}$ is bounded if the sequence $\{\tilde{z}^l\}$ is also bounded.

Lemma 10 *Assuming that the columns of $\nabla g(\tilde{x}^l)$ are linearly independent 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*

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

Proof By defining $r^l = \nabla f(\tilde{x}^l) - \tilde{z}^l + c\nabla g(\tilde{x}^l)^T g(\tilde{x}^l) - \nabla g(\tilde{x}^l)^T \tilde{y}^l$ and solving for $\nabla g(\tilde{x}^l)^T \tilde{y}^l$ we obtain

$$\nabla g(\tilde{x}^l)^T \tilde{y}^l = \nabla f(\tilde{x}^l) - \tilde{z}^l + c\nabla g(\tilde{x}^l)^T g(\tilde{x}^l) - r^l.$$

From our assumptions the above equation can be written as

$$\begin{aligned} \tilde{y}^l &= [\nabla g(\tilde{x}^l)\nabla g(\tilde{x}^l)^T]^{-1}\nabla g(\tilde{x}^l)\,(\nabla f(\tilde{x}^l) + c\nabla g(\tilde{x}^l)^T g(\tilde{x}^l) - r^l) \\ &\quad -[\nabla g(\tilde{x}^l)\nabla g(\tilde{x}^l)^T]^{-1}\nabla g(\tilde{x}^l)\,\tilde{z}^l. \end{aligned}$$

Taking norms in both sides of the above equation yield

$$\begin{aligned} \| \tilde{y}^l \| &\leq \| [\nabla g(\tilde{x}^l)\nabla g(\tilde{x}^l)^T]^{-1}\nabla g(\tilde{x}^l) \| \, \| \nabla f(\tilde{x}^l) + c\nabla g(\tilde{x}^l)^T g(\tilde{x}^l) - r^l \| \\ &\quad + \| [\nabla g(\tilde{x}^l)\nabla g(\tilde{x}^l)^T]^{-1}\nabla g(\tilde{x}^l) \| \, \| \tilde{z}^l \| \\ &\leq M_1(1 + \| \tilde{z}^l \|). \end{aligned}$$

where the constant M_1 is defined as

$$\begin{aligned} M_1 &\geq \max\{ \| [\nabla g(\tilde{x}^l)\nabla g(\tilde{x}^l)^T]^{-1}\nabla g(\tilde{x}^l) \| \, \| \nabla f(\tilde{x}^l) + c\nabla g(\tilde{x}^l)^T g(\tilde{x}^l) - r^l \|, \\ &\qquad \| [\nabla g(\tilde{x}^l)\nabla g(\tilde{x}^l)^T]^{-1}\nabla g(\tilde{x}^l) \| \}. \end{aligned}$$

and is finite according to our assumptions. •

Lemma 11 *If $(\tilde{x}^l, \tilde{y}^l, \tilde{z}^l) \in S_2(\epsilon)$ for all $l \geq 0$, then the sequence $\{(\tilde{x}^l, \tilde{y}^l, \tilde{z}^l)\}$ is bounded above.*

Proof From Lemma 10, it suffices to prove that the sequences $\{\tilde{x}^l\}$ and $\{\tilde{z}^l\}$ are bounded from above. By assumption (A2), the sequence $\{\tilde{x}^l\}$ is bounded. Assume that there exists a non-empty set I_z^∞, which contains the indices i of those elements, $(\tilde{z}^l)^i$, of the vector $\tilde{z}^l$, for which $\lim_{l\to\infty}(\tilde{z}^l)^i = \infty$. From the boundedness of the sequences $\{(\tilde{x}^l)^i(\tilde{z}^l)^i\}$, $i = 1, 2, ..., n$, we obtain $\liminf_{l\to\infty}(\tilde{x}^l)^i = 0$, for those indices $i \in I_z^\infty$. Furthermore from the definition of the set I_x^0, in Assumption (A4), it is evident that $I_z^\infty \subseteq I_x^0$.

From (58) and the fact that $\{\mu^l\} \to 0$ we have that the sequence

$$\{\| \nabla f(\tilde{x}^l) - \tilde{z}^l + c_* \nabla g(\tilde{x}^l)^T g(\tilde{x}^l) - \nabla g(\tilde{x}^l)^T \tilde{y}^l \|\}$$

is bounded. Using this and the fact that $\{\| \nabla f(\tilde{x}^l) \|\}$ and $\{\| c\nabla g(\tilde{x}^l)^T g(\tilde{x}^l) \|\}$ are bounded, we conclude that $\{\| -\tilde{z}^l - \nabla g(\tilde{x}^l)^T \tilde{y}^l \|\}$ is also bounded. Hence, we have

$$\frac{\| \tilde{z}^l + \nabla g(\tilde{x}^l)^T \tilde{y}^l \|}{\| (\tilde{y}^l, \tilde{z}^l) \|} \to 0 \tag{59}$$

By setting $\tilde{u}^l = \frac{(\tilde{y}^l, \tilde{z}^l)}{\|(\tilde{y}^l, \tilde{z}^l)\|}$, we have $\{\tilde{u}^l\}$ bounded and $\{\tilde{u}^l\} \to \tilde{u}^*$. It is clear that $\| \tilde{u}^* \| = 1$ and the components of $\tilde{u}^*$, corresponding to those indices $i \notin I_z^\infty$, i.e., $\{(\tilde{z}^l)^i\} < \infty$, are zero. If $\hat{u}^*$ is the vector consisting of the components of $\tilde{u}^*$ which correspond to the indices $i \in I_z^\infty$, then $\| \hat{u}^* \| = \| \tilde{u}^* \| = 1$. Furthermore, from (59) we have

$$\frac{\nabla g(\tilde{x}^l)^T \tilde{y}^l + \tilde{z}^l}{\| (\tilde{y}^l, \tilde{z}^l) \|} = \frac{[\nabla g(\tilde{x}^l)^T,\ I_n]\, (\tilde{y}^l, \tilde{z}^l)}{\| (\tilde{y}^l, \tilde{z}^l) \|} = [\nabla g(\tilde{x}^l)^T,\ e_i : i \in I_x^0]\, \hat{u}^* \to 0.$$

However, this result contradicts Assumption (A2). Hence, the set I_z^∞ is empty, or for all indices $i = 1, 2, ..., n$, the sequences $\{(\tilde{z}^l)^i\}$ are bounded. Consequently, $\{\tilde{z}^l\}$ is also bounded. •

The following theorem shows that the sequence $\{(\tilde{x}^l, \tilde{y}^l, \tilde{z}^l)\}$ converges to $(\tilde{x}^*, \tilde{y}^*, \tilde{z}^*)$ which is a KKT point of the initial constrained optimization problem (5).

Theorem 3 *Let $\{\mu^l\}$ be a positive monotonically decreasing sequence of barrier parameters with $\{\mu^l\} \to 0$, and let $\{(\tilde{x}^l, \tilde{y}^l, \tilde{z}^l)\}$ be a sequence of approximate central points satisfying (58) for $\mu = \mu^l$, $l \geq 0$. Then the sequence $\{(\tilde{x}^l, \tilde{y}^l, \tilde{z}^l)\}$ is bounded and its limit point $(\tilde{x}^*, \tilde{y}^*, \tilde{z}^*)$ satisfies the following conditions*

$$F(x, y, z; c, \mu) = \begin{pmatrix} \nabla f(x) - z + c\nabla g(x)^T g(x) - \nabla g(x)^T y \\ g(x) \\ XZe \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}. \tag{60}$$

Proof From Lemma 10 the sequence $\{(\tilde{x}^l, \tilde{y}^l, \tilde{z}^l)\}$ is bounded and remains in the compact set $S_2(\epsilon)$. Thus it has a limit point in $S_2(\epsilon)$, denoted as $(\tilde{x}^*, \tilde{y}^*, \tilde{z}^*)$. From (58) and the fact that $\mu^l \to 0$ we easily obtain that $\lim_{l\to\infty} \| F(\tilde{x}^l, \tilde{y}^l, \tilde{z}^l) \| = 0$. Therefore,

$$\begin{aligned} \nabla f(\tilde{x}^*) - \tilde{z}^* + c\nabla g(x)^T g(x) - \nabla g(\tilde{x}^*)^T \tilde{y}^* &= 0 \\ g(\tilde{x}^*) &= 0 \\ \tilde{X}^* \tilde{Z}^* e &= 0. \end{aligned}$$

Clearly from the above equations we may derive that $(\tilde{x}^*, \tilde{y}^*, \tilde{z}^*)$ is a KKT point of the initial constrained optimization problem (5). •

6 Numerical Results

The test-problems solved by the algorithm fall into two categories. The first category consists of small size problems drawn mainly from the Hock and Schittkowski collection [15]. The second category consists of real-world problems of larger size. These problems were taken from Vanderbei's web site [26]. The implementation of the algorithm has been done using standard C and is interfaced with the powerful mathematical programming language AMPL [11].

The various parameters used in the algorithm are as follows. In Step 1, the accuracy of the stopping criterion is $\epsilon_0 = 10^{-8}$. In Step 2.3, $\delta = 10$ and $\epsilon_g = 10^{-8}$. In Step 2.4, we set $\gamma = 0.995$, $\beta = 0.5$, $\rho = 10^{-4}$, $m = 1$, and $M = 10$. In the barrier reduction rule, described in section § 4.4, we set $\sigma = 6$. Furthermore, our implementation provides two options to the user, regarding the calculation of the Hessian matrix H_k. In the first option, the Hessian can be approximated by a positive definite matrix using Powell's modification to BFGS updating formula [19]. In the second option, the exact Hessian, provided through the interface with AMPL, is used. In this case, in order to ensure that the Newton direction Δx_k in (23) is descent, the exact Hessian has to be sufficiently positive definite at every iteration. Hence, H_k is replaced by a positive definite matrix $\hat{H}_k$, which is generated by the modified Cholesky factorization as described by Gill *et al* [14], and has the form $\hat{H}_k = H_k + E_k$, where E_k is a nonnegative diagonal matrix that is zero if H_k is positive definite. This technique has worked very well in practice enabling the algorithm to solve large problems. It should be stated, however, that the algorithm, using either the exact or the BFGS approximation of the Hessian, has demonstrated fast local convergence.

Tables 1 summarize the numerical results for the test-problems taken from the Hock and Schittkowski collection. The following abbreviations have been used:

Prob: The problem number given in the Hock and Schittkowski collection [15].

Iter: The total number of inner iterations required to find the optimum solution of the original problem (5). It is the final value of the inner iteration counter k in Algorithm 1.

c_*: The final value of the penalty parameter.

All the numerical results have been obtained by using the exact Hessian, provided by AMPL, except those marked with $*$ (i.e., tests 74, 75, 93 97), which were solved using the BFGS updating formula [19]. The algorithm solved all the problems to the given accuracy. For all of the problems the initial value c_0 of the penalty parameter c is set to zero. Its final value c_* is usually kept at low levels. For some tests, however, its final value needed to become large in order to achieve convergence, which was achieved in all tests. We have also observed that usually the penalty parameter becomes large for the problems, whose starting point, provided by Hock and Schittkowski [15], is close to the boundary of the feasible region. In such cases, Vanderbei and Shanno [27] suggest that the starting point should be set to a 90%-10% mixture of the two bounds of the box constraints, with the higher value placed on the nearer bound. If the algorithm starts from such a point, the final value of the penalty parameter can be kept low, with similar convergence.

For problems 102 and 103 we needed to follow the advice of Gay *et al* [13] and increase the initial value of the barrier parameter, μ_0, to 10^4 in order to allow the algorithm to converge in less than one hundred iterations. However, this technique did not help in problem 101, which was also solved but in more than one hundred iterations. For all the remaining problems the initial value of the barrier parameter was set to $\mu_0 = 1$. It should also be mentioned that among the problems of the Hock and Schittkowski collection, solved by our algorithm, there exist some particularly difficult problems. For example, problems 64 and 72 are highly nonconvex, problem 108 has a singular reduced Hessian, and problems 88 to 92 have very large condition numbers of the coefficient matrix in the Newton system.

Moreover, Table 2 summaries the number of variables and constraints of the larger problems solved by the algorithm. All of these problems are limited up to 300 variables and constraints, since we use the student version of AMPL. In order to overcome this limitation and be able to test the algorithm on a diverse collection of problems, we have reduced, in some problems, the number of variables or constraints to 300, by modifying some parameters in their definition. Hence, problems "markowitz100", "1_*lp*_150", "minsurf288", "oet1_148", "oet3_148" and "svanberg299" are smaller versions of the original tests "markowitz", "1_*lp*", "minsurf", "oet1", "oet3" and "svanberg". Table 3 summaries the numerical results for the large problems. As can be seen the algorithm can be very efficient and robust in solving difficult and larger problems.

We present an example to show the importance of the mechanism that switches between the two merit functions. Consider the following box-constrained optimization problem:

$$\begin{aligned} f(x) &= (x_1-1)(x_1-2)(x_1-3)+(x_1-2)(x_1-3)(x_2-1)- \\ &\quad (x_1-3)(x_2-1)(x_2-2)-(x_2-1)(x_2-2)(x_2-3), \\ s.t. &\quad -5 \le x_i \le 5, \quad i=1,2. \end{aligned} \tag{61}$$

Problem (61) has three local minima:

$$x^1_{min} = (-5, -0.69783), x^2_{min} = (3.39512, 5), x^3_{min} = (2.500, 1.500)$$

and two saddle points:

$$x^1_{sad} = (1.29289, 1.29289), x^2_{sad} = (2.70711, 2.70711)$$

We solve (61) with the original Algorithm 1 and with a variant that uses only the norm of the KKT conditions as the merit function in all iterations. Table 4 shows solutions to which both versions of the algorithm have converged, starting from different points. It is clear that using the KKT residual norm as the only merit function there is no guarantee that the algorithm will converge to a local minimum. On the other hand, the switching of merit functions provides a robust and efficient way to guarantee the convergence of the algorithm to a local minimum.

As a final note we would like to mention that the performance of the algorithm is very good and comparable with other primal-dual interior point algorithms. For example, our algorithm seems to have superior performance, in terms of the number of iterations, than the interior point algorithms described by Vanderbei and Shanno [27], and Gay *et al.* [13].

7 Summary and Conclusions

A primal-dual interior point algorithm for general nonlinear programming problems has been presented. Its distinct feature is the incorporation of two differentiable merit functions, which guide the algorithm to a local minimum. A switching mechanism is employed in order to decide which merit function is most appropriate to be used. This mechanism guarantees progress towards to a local minimum and avoid convergence to saddle points and local maxima. By ensuring the monotonic decrease of each merit function, it has been shown that the algorithm is globally convergent. The numerical results demonstrate that the algorithm can be used to successfully solve various nonlinear programming problems in an efficient and robust way.

Prob.	Iter.	c_*	Prob.	Iter.	c_*	Prob.	Iter.	c_*	Prob.	Iter.	c_*
1	27	0	29	47	5.8×10^{11}	56	11	9.4×10^{8}	86	16	7.6×10^{12}
2	15	0	30	10	1.6×10^{11}	57	22	18.9	87	13	1.7×10^{12}
3	10	0	31	9	1.7×10^{11}	59	12	1.7×10^{11}	88	16	2.1×10^{9}
4	10	0	32	45	0	60	11	0	89	16	1.5×10^{11}
5	10	0	33	10	0	61	20	9.5×10^{5}	90	20	1.1×10^{11}
6	6	0	34	19	3.4×10^{11}	62	12	0	91	16	1.0×10^{11}
7	9	2.3×10^{6}	35	12	0	63	20	2.3×10^{6}	92	18	7.1×10^{11}
8	5	0	36	20	0	64	23	4.1×10^{11}	93*	15	2.3×10^{12}
9	5	0	37	57	175.4	65	12	1.2×10^{11}	95	31	1.0×10^{12}
10	14	1.2×10^{12}	38	25	0	66	20	9.8×10^{12}	96	31	4.3×10^{12}
11	11	1.9×10^{12}	39	8	5.3×10^{6}	67	45	3.8×10^{11}	97*	22	1.9×10^{8}
12	15	3.8×10^{11}	40	25	5.8×10^{5}	68	11	2.3×10^{7}	98*	19	3.1×10^{5}
14	11	2.3×10^{11}	41	36	1.2×10^{11}	69	43	4.2×10^{5}	99	35	1.2×10^{10}
15	22	3.6×10^{6}	42	13	2.1×10^{7}	70	14	3.0×10^{12}	100	14	6.1×10^{12}
16	13	6.6×10^{12}	43	13	0	71	25	2.8×10^{11}	102	69	7.8×10^{12}
17	55	2.3×10^{12}	44	23	0	72	18	6.1×10^{10}	103	66	1.3×10^{12}
18	12	4.4×10^{12}	45	22	0	73	12	4.1×10^{11}	104	59	6.6×10^{12}
19	21	8.1×10^{12}	46	26	33.69	74	8*	6.0×10^{9}	105	12	0
20	16	6.7×10^{11}	47	20	71.8	75	9*	104	107	15	7.1×10^{12}
21	10	0	48	5	0	76	12	0	108	21	6.8×10^{12}
22	12	5.3×10^{12}	49	19	0	77	12	3243	110	11	0
23	20	3.6×10^{12}	50	9	0	78	50	1.3×10^{4}	111	13	13957
24	19	9.7×10^{10}	51	5	0	79	7	10	112	26	0
25	31	0	52	4	0	80	14	2.2×10^{6}	113	15	4.8×10^{12}
26	21	10	53	9	0	81	56	465	114	48	2.7×10^{12}
27	9	7065	54	13	0	83	16	2.8×10^{10}	117	18	1.1×10^{12}
28	4	0	55	11	4.1×10^{12}	84	35	3.6×10^{5}	119	19	1157.6

Table 1: Numerical Results for the Hock-Schitkowski problems

Prob.	Vars	Constr	Prob.	Vars	Constr
Antenna	49	166	dea2	6	100
Catenary	198	100	dea_lp	10	126
markowitz100	300	201	dea_lp2	10	126
1_lp_150	8	152	dea_frac_lin	8	100
nnls	300	0	fir_linear	11	154
nnls2	300	201	fir_convex	11	154
oet1_148	3	298	fir_socp	12	155
oet3_148	4	298	fir_exp	12	155
minsurf	289	0	svanberg299	299	298
obstclal	64	0	vanderm1	100	99
dea	6	100	vanderm3	100	99

Table 2: Characteristics of large problems

Prob.	Iter.	c_*	Prob.	Iter.	c_*
Antenna	79	1.3×10^{10}	dea2	12	0
Catenary	23	1.5×10^{8}	dea_lp	11	1481
markowitz100	31	25.1	dea_lp2	11	1481
1_lp_150	18	9.1×10^{11}	dea_frac_lin	14	0
nnls	32	0	fir_linear	20	5.0×10^{12}
nnls2	17	0	fir_convex	21	2.4×10^{11}
oet1_148	16	0	fir_socp	20	2.3×10^{11}
oet3_148	18	0	fir_exp	20	4.1×10^{10}
minsurf	9	0	svanberg299	26	1.7×10^{10}
obstclal	24	0	vanderm1	74	3.9×10^{11}
dea	25	0	vanderm3	78	1.1×10^{10}

Table 3: Numerical Results for large problems

starting point	Original Algorithm 1	Algorithm 1 with only the KKT residual norm as merit function
(-5,-5)	x^1_{min}	x^1_{sad}
(-5, 5)	x^1_{min}	x^1_{sad}
(5,-5)	x^1_{min}	x^3_{min}
(5, 5)	x^2_{min}	x^2_{sad}
(1, 1)	x^3_{min}	x^1_{sad}
(3, 3)	x^2_{min}	x^2_{sad}
(-5, 0)	x^1_{min}	x^1_{sad}
(5, 0)	x^1_{min}	x^3_{min}
(0,-5)	x^1_{min}	x^1_{sad}
(0, 5)	x^3_{min}	x^2_{sad}

Table 4: Convergence of Algorithm 1 and its variant that uses only the KKT residual norm as merit function starting points

References

[1] I. Akrotirianakis and B. Rustem. A primal-dual interior point algorithm with an exact and differentiable merit function for nonlinear programming. *Optimization Methods and Software*, 14:1–36, 2000.

[2] M. C. Bartholomew-Biggs. IP from an SQP point of view. *Optimization Methods and Software*, 16:69–84:, 2001.

[3] H. Y. Benson, D. F. Shanno, and R. J. Vanderbei. Interior point methods for nonconvex nonlinear programming: Jamming and comparative numerical results. Technical Report ORFE-00-02, Department of Operations Research and Financial Engineering, Princeton University, Princeton, NJ, 2000.

[4] D. Bertsekas. *Constrained Optimization and Lagrange Multiplier Methods.* Academic Press, 1982.

[5] D. Bertsekas. *Nonlinear Programming.* Athena Scientific Publishing, 1995.

[6] R. M. Chamberlain, C. Lemarechal, H. C. Pedersen, and M. J. D. Powell. The watchdog technique for forcing convergence in algorithms for constrained optimization. *Mathematical Programming Study*, 16:1–17, 1982.

[7] D. denHertog. *Interior point approach to Linear, Quadratic and Convex Programming.* Kluwer Academic Publishers, 1994.

[8] A. S. El-Bakry, R. A. Tapia, T. Tsuchiya, and Y. Zhang. On the formulation and theory of the newton interior point method for nonlinear programming. *JOTA*, 89(3):507–541, 1996.

[9] A. Fiacco and G. McCormick. *Nonlinear Programming: Sequential Unconstrained minimization Techniques.* John Wiley and Sons, New York, 1968.

[10] A. Forsgren and P. E. Gill. Primal-dual interior point methods for nonconvex nonlinear programming. *SIAM Journal on Optimization*, 8:1132–1152, 1998.

[11] R. Fourer, G. M. Gay, and B. W. Kernigham. *AMPL: A Modeling Language for Mathematical Programming.* ITP, 1993.

[12] R. S. Gajulapalli and L. S. Lasdon. Computational experience with a safeguarded barrier algorithm for sparse nonlinear programming. Technical report, University of Texas at Austin, Austin, TX, 1997.

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

[14] P. E. Gill, W. Murray, and M. H. Wright. *Practical Optimization.* Academic Press, 1981.

[15] W. Hock and K. Schittkowski. *Test Examples for Nonlinear Programming Codes.* Springer Verlag, New York, NY, 1981.

[16] M. Kojima, S. Mizuno, and A. Yoshise. Primal-dual interior point algorithm for linear programming. In N. Megiddo, editor, *Progress in Mathematical Programming: Interior Point and Related Methods*, pages 29–47. Springer-Verlag, New York, 1989.

[17] L. S. Lasdon, J. Plummer, and Y. Gang. Primal-dual and primal interior point algorithms for general nonlinear programs. *ORSA J. on Computing*, 7(3):321–332, 1995.

[18] N. Maratos. Exact Penalty Function Algorithms for Finite Dimensional and Optimization Problems. *Ph. D. Thesis, Imperial College London, U. K.*, 1978.

[19] M. J. D. Powell. A fast algorithm for nonlinearly constrained optimization calculations. In Watson G. A., editor, *Numerical Analysis Proceedings, Biennial Conference, Dundee*, Lecture Notes in Mathematics (630), pages 144–157, Berlin, 1978. Springer Verlag.

[20] B. Rustem. A class of superlinear convergent projection algorithms with relaxed stepsizes. *Applied Mathematics and Optimization*, 12:29–43, 1984.

[21] B. Rustem. Convergent stepsizes for constrained optimization algorithms. *JOTA*, 49:135–160, 1986.

[22] B. Rustem. Equality and inequality constrained optimization algorithms with convergent stepsizes. *JOTA*, 76(3):429–453, 1993.

[23] B. Rustem. *Algorithms for Nonlinear Programming and Multiple Objective Decisions.* J. Wiley, 1998.

[24] E. Simantiraki and D. F. Shanno. Interior point methods for linear and nonlinear programming. In Duff I. S. and Watson G. A., editors, *The State of the Art in Numerical Analysis*, pages 339–352. Oxford University Press, New York, 1997.

[25] R. J. Vanderbei. Loqo: An interior point code for quadratic programming. Technical Report 94-15, SOR, Princeton University, Princeton, NJ, 1994.

[26] R. J. Vanderbei. Large-scale nonlinear *ampl* models. Princeton University, Princeton, NJ, http://www.princeton.edu/ rvdb/ampl/nlmodels, 1997.

[27] R. J. Vanderbei and D. F. Shanno. An interior point algorithm for nonconvex nonlinear programming. Technical Report SOR-97-21, CEOR, Princeton University, Princeton, NJ, 1997.

[28] R. J. Vanderbei and D. F. Shanno. An interior point algorithm for nonconvex nonlinear programming. Technical Report SOR-97-21 (Revised May 1998), CEOR, Princeton University, Princeton, NJ, 1997.

[29] R. J. Vanderbei and D. F. Shanno. An interior point method for nonconvex nonlinear programming: Orderings and higher-order methods. Technical Report SOR-99-5, Program in Statistics and Operations Research, Princeton University, 1999.

[30] A. Wachter and L. Biegler. Failure of global convergence of a class of interior point methods for nonlinear programming. *Mathematical Programming*, 88(3):565–574, 2000.

[31] S. J. Wright. *Primal-Dual Interior-Point Methods.* SIAM, 1997.

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

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

[34] H. Yamashita and H. Yabe. An interior point method with a primal-dual l_2 barrier penalty function for nonlinear optimization. Technical report, Department of Applied Mathematics, Science University of Tokyo, 1-3 Kagurazaka, Shinjuku-ku, Tokyo, Japan, March 1999.

[35] S. Zakovic, B. Rustem and C. Pantelides. An interior point algorithm for computing saddle points of constrained continuous minimax. *Annals of Operations Research*, 99:59–78, 2000.