Switching Stepsize Strategies for SQP

George Tzallas-Regas[*] Berç Rustem[†]

September 13, 2010

Abstract: An SQP algorithm is presented for solving constrained nonlinear programming problems. The algorithm uses three stepsize strategies in order to achieve global and superlinear convergence. Switching rules are implemented that combine the merits and avoid the drawbacks of the three stepsize strategies. A penalty parameter is determined using an adaptive strategy that aims to achieve sufficient decrease of the activated merit function. Global convergence is established and it is also shown that, locally, unity step sizes are accepted, and therefore superlinear convergence is not impeded under standard assumptions. Global convergence and convergence of the stepsizes is displayed on test problems from the Hock and Schittkowski collection.

Keywords: Nonlinear Programming, SQP, Global Convergence, Stepsize Convergence, Merit Functions, Switching Stepsize Strategies

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

[†]Professor, Department of Computing, Imperial College London, London SW7 2AZ, UK

1 Introduction

We consider the nonlinear inequality constrained optimization problem

$$\begin{aligned} &\text{minimize} \quad && f(\mathbf{x}) \\ &\text{subject to} \quad && \mathbf{g}(\mathbf{x}) \leq \mathbf{0} \end{aligned} \tag{ICP}$$

where $f : \mathbb{R}^n \to \mathbb{R}$, $\mathbf{g} : \mathbb{R}^n \to \mathbb{R}^m$. The Lagrangian function for this problem is

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\mu}) = f(\mathbf{x}) + \langle \boldsymbol{\mu}, \mathbf{g}(\mathbf{x}) \rangle,$$

where $\boldsymbol{\mu}$ is the vector of Lagrange multipliers. The presented algorithm can be extended to handle general nonlinear programming problems directly, *i.e.* without converting equality constraints to a pair of inequalities, or adding slack variables. We chose (ICP) in order to emphasize on the stepsize strategy for inequality constraints and to make the presentation easier to follow.

The *Sequential Quadratic Programming* (SQP) framework [2, 3, 4, 5, 6] is one of the most successful methods for solving nonlinearly constrained optimization problems. The convergence properties of SQP are more than satisfactory when we are in a neighborhood of the solution. But the method may fail to converge remote from the solution. In order to induce global convergence a measure of goodness of the iterates must be introduced, which is minimized locally at the solution. Merit functions serve exactly this purpose.

A straightforward merit function is the residual of the KKT conditions. The most convenient form is the squared Euclidean norm of the KKT error. Although this merit function does serve the purpose of guiding the iterates and giving fast local convergence [7], it is not entirely satisfactory, because its solution can be any KKT point. Therefore the iterates may even converge to local maxima or saddle points [8].

Exact penalty functions are penalty functions that are minimized locally by the constrained minimum, and therefore can measure progress more efficiently. A widely used merit function is the l_1 penalty function. It was introduced by Pietrzykowski [9] as a penalty function, and was used as a merit function by Han [10] in a globally convergent SQP algorithm. Powell [11], [12] popularized the l_1 merit function in a successful SQP algorithm and also proposed an adaptive strategy for updating the penalty parameter from iteration to iteration.

The study of local convergence properties of SQP algorithms has shown that the role of the merit function is not restricted in inducing global convergence only. Han [13] and Powell [12] showed that if after a finite number of iterations the stepsize parameter can be chosen to be equal to unity, then the iteration has a superlinear rate of convergence, if further assumptions hold. Therefore, the merit function has to be chosen in such a way as to guide the iterates, and to also admit unity stepsizes, in order to achieve global and superlinear convergence. The work of Maratos [14] showed that the admission of unity stepsizes is not always possible with non-differentiable exact penalty functions. An example was constructed, showing that the stepsize strategy has to truncate the search direction, in order to achieve decrease in the l_1 merit function, although the current iterate was very close to the minimum.

The l_1 penalty function is a non-differentiable penalty function. Han [10] pointed that if the merit function is to be used in a line search procedure differentiability is not important. For differentiable functions though, convergence can be demonstrated with reasonably relaxed conditions. The quadratic or l_2 penalty function is a twice continuously differentiable penalty function, but only for equality constrained problems. Most merit functions for the inequality constrained problems are either non-differentiable or once differentiable. Rustem [15] gave a suitable merit function that was used to formulate two SQP algorithms for general nonlinear programming problems. Both algorithms were shown to be globally convergent. What is more important, the stepsizes were shown to be convergent to unity. The problem with the l_2 merit function though is that the proposed updates for the penalty parameter involve the constraint violations in the denominator of a fraction. Therefore, for feasible or almost feasible points the value of the penalty parameter becomes extremely large, which makes subsequent calculations very difficult.

Having the above in mind, we present an algorithm that tries to exploit the advantages of the aforementioned merit functions and avoid the disadvantages, in an effort to combine global convergence with fast local convergence. To that end we use a technique that is similar to the *watchdog technique* proposed by Chamberlain *et al.* [16]. The l_2 and l_∞ merit functions are used to guide the iterates towards a local minimum, and when necessary we switch to the Euclidean norm of the KKT residual, so as to achieve fast local convergence. The l_∞ merit function was used in a line search procedure by Pshenichnyi and Danilin [17], Mayne and Polak [18]. In [17] the l_∞ merit function was shown to possess good convergence properties. Our algorithm uses a modification of this line search procedure in order to guarantee that unity stepsizes will be admitted when close to the minimum. The rule that switches between merit functions updates on the one hand the penalty parameter so as to guarantee sufficient decrease for the merit function that will be activated. It is also carefully implemented so as to avoid large values of the penalty parameter which will make progress towards the solution harder.

This paper is organized as follows. In Section 2 we present the general framework of the SQP algorithm we are going to develop. The full algorithm is presented in Section 4. In Section 3 we give details of the three stepsize strategies. In Section 5 we give global convergence results, and in Section 6 local convergence properties of the proposed algorithm. Finally, Section 7 contains our numerical experience on small and larger scale problems.

The following notation is used throughout the paper. Superscript T is used for the transpose of a matrix or a vector. Subscript k denotes sequence elements, subscript $*$ the optimum and superscript i vector components. In order to simplify the presentation we suppress iterate subscripts in the proofs of theorems and lemmas. In there the current estimates appear unadorned, and a bar adorns next estimates, therefore $\mathbf{x}_k$ appears as $\mathbf{x}$ and $\mathbf{x}_{k+1}$ appears as $\bar{\mathbf{x}}$, and similarly for all other components. The gradient of a function with respect to $\mathbf{x}$ is denoted by $\boldsymbol{\nabla}$, and $\boldsymbol{\nabla}^2$ denotes the Hessian with respect to $\mathbf{x}$. The symbol $\|\cdot\|$ denotes the Euclidean norm unless explicitly specified. Operator $\langle \cdot, \cdot \rangle$ is taken to mean the dot product between two vectors. The vector operator $[\cdot]^+$ is such that the ith element is

$$([\cdot]^+)^i = \max\left\{0, (\cdot)^i\right\}.$$

In addition we will use

$$G_{(\cdot)} = \int_0^1 (1-t)\langle \mathbf{d}_k, \boldsymbol{\nabla}^2_{\mathbf{x}}(\cdot)(\mathbf{x}_k + t\alpha\mathbf{d}_k)\mathbf{d}_k\rangle dt,$$

where $(\cdot)$ is a function (f or $\mathcal{L}$). $\tilde{\mathbf{x}}_k$ is taken to be $\mathbf{x}_k + t\alpha\mathbf{d}_k$ in order to simplify the presentation. Finally $\Phi(\mathbf{x};\sigma)$ is used to denote the merit function, $\phi(\mathbf{d})$ a measure of sufficient decrease and $P(\mathbf{x})$ a penalty term. The arguments of these functions vary depending on the stepsize strategy chosen. These are made clear when we analyze the particular merit function, but for general discussions we use the arguments presented here.

2 Algorithmic Framework

Sequential Quadratic Programming methods are iterative. A search direction is generated and iterates are updated. In order to determine the search direction for our method, we attempt to solve the following quadratic programming subproblem

$$\begin{aligned} \underset{\mathbf{d}}{\text{minimize}} \quad & \langle \mathbf{d}, \boldsymbol{\nabla} f(\mathbf{x}_k) \rangle + (1/2) \langle \mathbf{d}, \mathbf{B}_k \mathbf{d} \rangle \\ \text{subject to} \quad & \mathbf{g}(\mathbf{x}_k) + \boldsymbol{\nabla} \mathbf{g}(\mathbf{x}_k)^T \mathbf{d} \leq \mathbf{0} \end{aligned} \tag{QPS}$$

where $\mathbf{x}_k$ is the current estimate and $\mathbf{B}_k$ is an approximation to the Hessian of the Lagrangian function. The solution $(\mathbf{d}_k, \boldsymbol{\mu}_{k+1})$ of problem QPS is then used to obtain the next iterate as

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k \tag{1}$$

where α_k is the line search parameter, chosen in such a way that the value of a merit function is decreased at each iteration. The Lagrange multiplier estimate is taken to be the corresponding multiplier $\boldsymbol{\mu}_{k+1}$, although other choices are possible [see 19]. In our algorithm we choose α_k so that it satisfies

$$\Phi(\mathbf{x}_k + \alpha \mathbf{d}_k; \sigma_{k+1}) - \Phi(\mathbf{x}_k; \sigma_{k+1}) \leq -\alpha c \phi(\mathbf{d}_k), \tag{2}$$

where Φ is the merit function that guides our search, $c > 0$ is a predetermined constant and ϕ quantifies the decrease in the merit function. Strategies use different upper bounds for c. In general, the descent function ϕ is usually taken to be the directional derivative of the merit function, but other approaches have also been proposed. These approaches aim to handle non-differentiability, or to economize with the calculations. Our exact choices for Φ, ϕ, c will be clarified in the next sections.

The Karush-Kuhn-Tucker conditions for (QPS) are as follows

$$
\begin{aligned}
\boldsymbol{\nabla} f(\mathbf{x}_k) + \mathbf{B}_k \mathbf{d}_k + \boldsymbol{\nabla}\mathbf{g}(\mathbf{x}_k)\boldsymbol{\mu}_{k+1} &= \mathbf{0} && \text{(3a)}\\
\boldsymbol{\nabla}\mathbf{g}(\mathbf{x}_k)^T \mathbf{d}_k + \mathbf{g}(\mathbf{x}_k) &\leq \mathbf{0} && \text{(3b)}\\
\mu^i_{k+1}\big(\langle \mathbf{d}_k, \boldsymbol{\nabla} g^i(\mathbf{x}_k)\rangle + g^i(\mathbf{x}_k)\big) &= 0, \quad i \in \mathcal{M} && \text{(3c)}\\
\boldsymbol{\mu}_{k+1} \in \mathbb{R}^m, \boldsymbol{\mu}_{k+1} &\geq \mathbf{0}
\end{aligned}
$$

where, for convenience, we define $\mathcal{M} \triangleq \{1, \ldots, m\}$.

The following assumptions will be used in the paper

(A1) Functions $f, \mathbf{g}$ are twice continuously differentiable.

(A2) The constraint normals are Lipschitz continuous

$$\left\|\boldsymbol{\nabla} g^i(\mathbf{x}) - \boldsymbol{\nabla} g^i(\mathbf{y})\right\| \leq \gamma \left\|\mathbf{x} - \mathbf{y}\right\|, \qquad i \in \mathcal{M}, \gamma > 0.$$

(A3) The feasible region of the (QPS) is non-empty.

(A4) The constraint normals of the binding constraints at the solution of the (QPS) are linearly independent.

(A5) Strict complementarity holds at the solution of problem ICP.

In relation to Assumption A4, we note that the active set in a QP is constructed to contain constraint gradients that are linearly independent, and further changes to the working set (addition or deletion of constraint indices), maintains this linear independence property (see for example Nocedal and Wright [20], section 16.4). A similar approach is taken by Gill *et al.* [21], Fukushima [22], Panier and Tits [23], Powell and Yuan [24], Yamashita [25].

3 Stepsize Strategies

Stepsize strategies are very important to the success of nonlinear programming algorithms. The success of algorithms lies mainly in the choice of the stepsize parameter α_k. The stepsize parameter is chosen so that an Armijo-type condition like (2) is satisfied. There are two key elements in this condition. The first one is the merit function Φ, which we aim to minimize locally; the second one is the right hand-side of the condition, for which we use the symbol ϕ.

Computationally Φ and ϕ have to be easy to compute. Theoretically they have to be combined in such a way that guarantees monotonic decrease of the merit function. This implies convergence from arbitrary points. For local convergence though, a stepsize of unity has to be admitted, so as to give rise to fast local convergence.

Most stepsize strategies employ penalty functions as merit functions, which involve a penalty parameter. In carefully designed stepsize strategies, the penalty parameter is modified adaptively in order to guarantee decrease of the merit function. The update of the penalty parameter is very important computationally because large values make further arithmetical operations very unstable. The motivation for the penalty parameter has changed as time goes by. In Sequential Unconstrained Minimization methods it had to guarantee equivalence between the constrained and the unconstrained problem. In Sequential methods that solve linearly constrained subproblems (as in our case), it has to guarantee that the search direction is descent for the merit function. In recent work by Byrd *et al.* [26] the penalty parameter is updated so as to guarantee prescribed level of linear feasibility. And then in the work by Fletcher and Leyffer [27], a method is proposed that dispenses with the use of penalty parameters.

Our approach generally falls under the second category and is based on the following remarks:

- The squared norm of the KKT conditions is not forceful in dispatching the sequence $\{\mathbf{x}_k\}$ to its optimal solution when used in a close neighborhood of the latter. However it may, on occasion, allow the algorithm to converge to a saddle point or a maximum. It can be shown to converge to unity stepsizes and consequently to allow superlinear convergence.
- The l_2 penalty function converges to a minimum and stepsizes converge to unity, but the penalty parameter may go to infinity.
- The l_∞ penalty function also converges to a minimum and is less problematic unless the multipliers tend to infinity, but it cannot be sown to attain unity stepsizes.

Our algorithm uses three stepsize strategies. Each stepsize strategy uses one of the above merit function. The stepsize should satisfy an Armijo condition that has the form

$$\Phi_t(\mathbf{x}_k + \alpha \mathbf{d}_k; \sigma_{k+1}) - \Phi_t(\mathbf{x}_k; \sigma_{k+1}) \leq -\alpha c_t \phi_t(\mathbf{d}_k),$$

where $t \in \{1, 2, 3\}$ is used to denote the merit function used.

3.1 The l_∞ Merit Function

This stepsize strategy corresponds to $t = 1$ in the algorithm. The Armijo condition at iteration k is

$$\Phi_1(\mathbf{x}_{k+1}; \sigma_{k+1}) - \Phi_1(\mathbf{x}_k; \sigma_{k+1}) > -\alpha c_1 \phi_1(\mathbf{x}_k, \boldsymbol{\mu}_{k+1}; \sigma_{k+1}), \tag{4}$$

where $c_1 \in (0,1)$. For this stepsize strategy the merit function is

$$\Phi_1(\mathbf{x}; \sigma) = f(\mathbf{x}) + \sigma P_1(\mathbf{x})$$

where

$$P_1(\mathbf{x}) = \max_{i \in \mathcal{M} \cup \{0\}} \left\{ g^i(\mathbf{x}) \right\}$$

and $g^0(\mathbf{x}) = 0$. In the right hand side of the Armijo condition we use

$$\phi_1(\mathbf{x}, \boldsymbol{\mu}; \sigma) = \sigma P_1(\mathbf{x}) + \langle \boldsymbol{\mu}, \mathbf{g}(\mathbf{x}) \rangle.$$

A stepsize strategy using this merit function was introduced in the *Linearization method* of Pshenichnyi and Danilin [17]. In the original method though, Pshenichnyi and Danilin used

$$\phi_{PD}(\mathbf{d}) = \|\mathbf{d}\|^2$$

as the right hand side of the Armijo condition. Their proofs show that if the penalty parameter is sufficiently large, the search direction is a descent direction for the merit function and thus global convergence is induced. The result using ϕ_{PD} though, cannot be used to show that the stepsize parameter converges to zero, therefore fast local convergence may not be guaranteed. We modified the stepsize strategy using ϕ_1 in an effort to show that fast local convergence will not be impeded by the effort to achieve sufficient decrease in the merit function. In our stepsize strategy, the condition for descent in Φ_1 remains the same (see Lemma 5.1) but in order to prove that the stepsizes converge to unity (Theorem 6.1), we need to prove an intermediate result (Lemma 5.2). This intermediate result relies on two conditions, namely

$$
\begin{aligned}
w_k = \langle \mathbf{d}, \boldsymbol{\nabla} f(\mathbf{x}_k) \rangle + \langle \mathbf{d}, \mathbf{B}_k \mathbf{d} \rangle &> 0 \qquad (5) \\
\|\mathbf{d}_k\| &> \epsilon_d, \qquad (6)
\end{aligned}
$$

where ϵ_d is some predetermined tolerance on the magnitude of the search direction. In the absence of these conditions, convergence of the stepsizes generated by (4) is not guaranteed and we avoid using this stepsize strategy in such cases.

3.2 The l_2 Merit Function

This stepsize strategy corresponds to $t = 2$ in the algorithm. The Armijo condition at iteration k is

$$
\Phi_2(\mathbf{x}_{k+1}, \boldsymbol{\pi}_{k+1}; \sigma_{k+1}) - \Phi_2(\mathbf{x}_k, \boldsymbol{\pi}_{k'}; \sigma_{k+1}) > -\alpha c_2 \phi_2(\mathbf{x}_k, \mathbf{d}_k, \boldsymbol{\pi}_{k'}; \sigma_{k+1}), \quad (7)
$$

where $c_2 \in (0, (1/2))$. In this stepsize strategy the merit function is

$$
\Phi_2(\mathbf{x}, \boldsymbol{\pi}; \sigma) = f(\mathbf{x}) + (1/\,(2\sigma))\, P_2(\mathbf{x}, \boldsymbol{\pi}),
$$

where

$$P_2(\mathbf{x}, \boldsymbol{\pi}) = \left\| [\sigma \mathbf{g}(\mathbf{x}) + \boldsymbol{\pi}]^+ \right\|^2 .$$

In the right hand side of the Armijo condition we use the directional derivative of this merit function, that is

$$\phi_2(\mathbf{x}, \mathbf{d}, \boldsymbol{\pi}; \sigma) = -\langle \mathbf{d}, \boldsymbol{\nabla}\Phi_2(\mathbf{x}, \boldsymbol{\pi}; \sigma)\rangle.$$

Vector $\boldsymbol{\pi} \in \mathbb{R}^m_+$ is an auxiliary vector. At the intermediate step k', $\pi^i_{k'}$ is computed using

$$\pi^i_{k'} = \begin{cases} -\sigma_{k+1} g^i(\mathbf{x}_k), & \text{if } g^i(\mathbf{x}_k) \leq 0 \\ \pi^i_k, & \text{if } g^i(\mathbf{x}_k) > 0 \end{cases}. \tag{8a}$$

Vector π^i_{k+1} is updated using

$$\pi^i_{k+1} := \begin{cases} -\sigma_{k+1} g^i(\mathbf{x}_{k+1}), & \text{if } g^i(\mathbf{x}_{k+1}) \leq 0 \\ \pi^i_{k'}, & \text{if } g^i(\mathbf{x}_{k+1}) > 0 \end{cases}. \tag{8b}$$

This stepsize strategy, along with the updates for $\boldsymbol{\pi}$, was introduced by Rustem [15]. It is shown by the author that if the penalty parameter is large enough, then we can decrease this merit function along the Newton direction. The author also displays convergence of the stepsizes to unity.

The directional derivative of Φ_2 is

$$\langle \mathbf{d}, \boldsymbol{\nabla}\Phi_2(\mathbf{x}, \boldsymbol{\pi}; \sigma)\rangle = \langle \mathbf{d}, \boldsymbol{\nabla} f(\mathbf{x})\rangle + \langle \mathbf{d}, \boldsymbol{\nabla}\mathbf{g}(\mathbf{x})[\sigma \mathbf{g}(\mathbf{x}) + \boldsymbol{\pi}]^+\rangle.$$

At iteration k, if we consider (3b), and [15, Lemma 2.2] we obtain the following bound[1]

$$-\phi_2(\mathbf{x}, \mathbf{d}, \boldsymbol{\pi}_{k'}; \sigma) \leq \langle \mathbf{d}, \boldsymbol{\nabla} f(\mathbf{x}) \rangle - \bar{\sigma} \left\| [\mathbf{g}(\mathbf{x})]^+ \right\|^2 - \langle [\mathbf{g}(\mathbf{x})]^+, \boldsymbol{\pi} \rangle. \tag{9}$$

From this inequality it is seen that in order to achieve decrease in Φ_2, there must hold

$$\sigma_{k+1} \geq \left(\langle \mathbf{d}_k, \boldsymbol{\nabla} f(\mathbf{x}_k) \rangle - \langle [\mathbf{g}(\mathbf{x}_k)]^+, \boldsymbol{\pi}_k \rangle \right) / \left\| [\mathbf{g}(\mathbf{x}_k)]^+ \right\|^2$$

If the iterates are close to becoming feasible though, then σ_{k+1} becomes extremely large and subsequent numerical calculations are practically difficult.

Therefore the computational success of this stepsize strategy relies on

$$\left\| [\mathbf{g}(\mathbf{x}_k)]^+ \right\|^2 > \epsilon_g \tag{10}$$

where $\epsilon_g > 0$ is some feasibility tolerance. If this condition is not satisfied we avoid using this stepsize strategy.

We intend to use this merit function when conditions (5), (6) are not simultaneously satisfied. We observe though that if condition (5) is not satisfied at the current point, then the search direction is a direction of descent for Φ_2 and we do not have to update the penalty parameter. Therefore in such cases we can use this stepsize strategy and we do not have to consider condition (10). This fact is exploited in the design of the switches in our algorithm.

[1] Iterate subscripts are suppressed as in proofs for this equation only; only k' is shown

3.3 The Squared KKT Residual

This stepsize corresponds to $t = 3$ in the algorithm. The Armijo condition at iteration k is

$$\Phi_3(\mathbf{z}_{k+1}) - \Phi_3(\mathbf{z}_k) > -\alpha c_3 \phi_3(\mathbf{z}_k), \tag{11}$$

where $c_3 \in (0, (1/2))$. In this stepsize strategy

$$\mathbf{z}_k = \begin{pmatrix} \mathbf{x}_k \\ \boldsymbol{\mu}_k \end{pmatrix}, \quad \delta\mathbf{z}_k = \begin{pmatrix} \mathbf{d}_k \\ \boldsymbol{\mu}_{k+1} - \boldsymbol{\mu}_k \end{pmatrix}, \mathbf{z}_{k+1} = \mathbf{z}_k + \alpha\delta\mathbf{z}_k.$$

The merit function used is

$$\Phi_3(\mathbf{z}) = (1/2) \left\|\mathbf{E}(\mathbf{z})\right\|^2$$

where

$$\mathbf{E}(z) = \begin{bmatrix} \boldsymbol{\nabla}_x \mathcal{L}(\mathbf{x}, \boldsymbol{\mu}) \\ \mu^1 g^1(\mathbf{x}) \\ \vdots \\ \mu^m g^m(\mathbf{x}) \end{bmatrix}.$$

In the right hand side of the Armijo condition we define

$$\phi_3(\mathbf{z}) = -\langle \delta\mathbf{z}, \hat{\mathbf{J}}(\mathbf{z})\mathbf{E}(\mathbf{z}) \rangle,$$

where

$$\hat{\mathbf{J}}(\mathbf{z}_k) = \begin{bmatrix} \mathbf{B}_k & \boldsymbol{\nabla}\mathbf{g}(\mathbf{x}_k) \\ \mathbf{M}_k \boldsymbol{\nabla} g^T(\mathbf{x}_k) & \mathbf{G}(\mathbf{x}_k). \end{bmatrix}$$

This stepsize strategy is activated if conditions (5), (6) and (10) are not satisfied. Such a stepsize strategy was studied by Rustem [28]. It has been shown by the author that the search direction is a direction of descent, and that the stepsize strategy attains unity stepsizes.

4 Algorithm Formulation

The full algorithm is presented in Algorithm 1. We remind that in the algorithm $t \in \{1, 2, 3\}$ controls which merit function will be activated in the Armijo search (steps 28–36). The Armijo condition, depending on the value of t, will take the form (4), (7) or (11) respectively.

5 Global Convergence

In this section we establish the global convergence of the proposed algorithm. The main convergence result is Theorem 5.2. In order to instantiate convergence of our algorithm we also invoke the following assumption, a standard assumption in SQP algorithms that use line searches:

(B1) There exist constants $\beta_2 \geq \beta_1 > 0$ such that

$$\beta_1 \|\mathbf{d}\|^2 \leq \langle \mathbf{d}, \mathbf{B}_k \mathbf{d} \rangle \leq \beta_2 \|\mathbf{d}\|^2, \qquad \forall \mathbf{d} \in \mathbb{R}^n, k = 0, 1, \ldots$$

Algorithm 1 A globally and superlinearly convergent algorithm

Choose $\mathbf{x}_0 \in \mathbb{R}^n$, $\mathbf{B}_0 \in \mathbb{R}^{n \times n}$, $\sigma_0 \in (0, +\infty)$.
Choose $\epsilon_g, \epsilon_d, \theta \in (0,1)$, $\delta \in (0, +\infty)$, $c_1 \in (0, 1/2), c_2, c_3 \in (0,1)$
Set $k := 0$, $\boldsymbol{\pi}_k = [-\sigma_k \mathbf{g}(\mathbf{x}_k)]^+$
repeat
  Obtain a Kuhn-Tucker point $(\mathbf{d}_k, \boldsymbol{\mu}_{k+1})$ of (QPS)
  Set $w_k := \langle \mathbf{d}_k, \mathbf{B}_k \mathbf{d}_k \rangle + \langle \mathbf{d}_k, \boldsymbol{\nabla} f(\mathbf{x}_k) \rangle$
  if $(w_k > 0)$ **then**
    if $(\|\mathbf{d}_k\| > \epsilon_d)$ **then**
      $t := 1$
      $\sigma_{k+1} := \max\left\{1.1, \sqrt{m}/\sqrt{\epsilon_g}\right\} \cdot \sum_{i \in \mathcal{M}} \mu^i_{k+1}$
    else
      if $(\left\|[\mathbf{g}(\mathbf{x}_k)]^+\right\|^2 \leq \epsilon_g)$ **then**
        $t := 3$
        $\sigma_{k+1} := \sigma_k$
      else
        $t := 2$
        if $(w_k \leq \sigma_k \left\|[\mathbf{g}(\mathbf{x}_k)]^+\right\|^2 + \langle [\mathbf{g}(\mathbf{x}_k)]^+, \boldsymbol{\pi}_k \rangle)$ **then**
          $\sigma_{k+1} := \sigma_k$
        else
          $\sigma_{k+1} := \max\left\{\dfrac{w_k - \langle [\mathbf{g}(\mathbf{x}_k)]^+, \boldsymbol{\pi}_k \rangle}{\|[\mathbf{g}(\mathbf{x}_k)]^+\|^2}, \sigma_k + \delta\right\}$
        end if
      end if
    end if
  else
    $t := 2$
    $\sigma_{k+1} := \sigma_k$
  end if
  $j_k := 0$
  $\mathbf{x}_{k+1} := \mathbf{x}_k + \mathbf{d}_k$
  Update (if necessary) $\boldsymbol{\pi}_{k'}$, $\boldsymbol{\pi}_{k+1}$
  while $\Big(\Phi_t(\mathbf{x}_{k+1}; \sigma_{k+1}) - \Phi_t(\mathbf{x}_k; \sigma_{k+1}) > -\alpha_{j_k} c_t \phi_t(\mathbf{d}_k)\Big)$ **do**
    $j_k := j_k + 1$
    $\alpha_{j_k} := (\theta)^{j_k}$
    $\mathbf{x}_{k+1} := \mathbf{x}_k + \alpha_{j_k} \mathbf{d}_k$
    Update (if necessary) $\boldsymbol{\pi}_{k+1}, \boldsymbol{\mu}_{k+1}$
  end while
  Update $\mathbf{B}_k$ to $\mathbf{B}_{k+1}$
  Set $k := k + 1$
until Convergence

5.1 The l_∞ Merit Function

In this section we assume that conditions (5) and (6) are satisfied at the solution of the (QPS).

In the next lemma we show that the Armijo condition (4) for $t = 1$ requires that Φ_1 decreases from iteration to iteration.

Lemma 5.1 (The sign of ϕ_1) *If the penalty parameter is sufficiently large, that is*

$$\sigma_{k+1} \geq \sum_{i \in \mathcal{M}} \mu_{k+1}^i \tag{12}$$

then

$$\phi_1(\mathbf{x}_k, \boldsymbol{\mu}_{k+1}; \sigma_{k+1}) \geq 0.$$

Proof : If $(\mathbf{d}, \bar{\boldsymbol{\mu}})$ is a Kuhn-Tucker pair for (QPS), then from Eqs. (3c), (12) we have

$$\begin{aligned} \langle \mathbf{d}, \boldsymbol{\nabla}\mathbf{g}(\mathbf{x})\bar{\boldsymbol{\mu}} \rangle &= -\langle \mathbf{g}(\mathbf{x}), \bar{\boldsymbol{\mu}} \rangle \\ &\geq -\sum \bar{\mu}^i P_1(\mathbf{x}) \\ &\geq -\bar{\sigma} P_1(\mathbf{x}) \end{aligned} \tag{13}$$

or

$$\bar{\sigma} P_1(\mathbf{x}) + \langle \mathbf{d}, \boldsymbol{\nabla}\mathbf{g}(\mathbf{x})\bar{\boldsymbol{\mu}} \rangle \geq 0,$$

which is the required result. $\square$

Note that in our algorithm (Step 10) the choice of σ_{k+1} guarantees satisfaction of (12) with a strict inequality. In the following proofs we assume that the penalty parameter is sufficiently large. Condition (12) is the condition that has been used in [17, 29] to establish convergence properties for the linearization method, and is used in our algorithm to update the penalty parameter (Step 10), in a suitable form.

The following three Lemmas give necessary bounds on various quantities of the algorithm. The first and third one are bounds on the search direction, whereas the second one is a bound on the penalty term. They are all necessary to establish global convergence and convergence of the stepsizes to unity of the iteration presented in Algorithm 1.

Lemma 5.2 *There exists $\beta_3 > 0$ such that:*

$$\|\mathbf{d}_k\| \leq \beta_3 P_1(\mathbf{x}_k) \tag{14}$$

Proof : From (5) it is evident that $\mathbf{d} \neq \mathbf{0}$. Eq. (3a), (3c) yield

$$\begin{aligned}
\|\mathbf{d}\| \left((\langle \mathbf{d}, \mathbf{B}\mathbf{d} \rangle + \langle \mathbf{d}, \boldsymbol{\nabla} f(\mathbf{x}) \rangle) \,/\, \|\mathbf{d}\| \right) &= \langle \mathbf{d}, \boldsymbol{\nabla} f(\mathbf{x}) \rangle + \langle \mathbf{d}, \mathbf{B}\mathbf{d} \rangle \\
&= \langle \bar{\boldsymbol{\mu}}, \mathbf{g}(\mathbf{x}) \rangle \\
&\leq \sum \bar{\mu}^i P_1(\mathbf{x}) \\
&\leq \bar{\sigma} P_1(\mathbf{x}), \qquad (15)
\end{aligned}$$

which for

$$\beta_3 = \bar{\sigma}\left(\|\mathbf{d}\| \,/\, (\langle \mathbf{d}, \mathbf{B}\mathbf{d}\rangle + \langle \mathbf{d}, \boldsymbol{\nabla} f(\mathbf{x})\rangle)\right)$$

gives the desired result. $\square$

Lemma 5.3 *There exists $\beta_4 > 0$, such that:*

$$P_1(\mathbf{x}_k) \leq \beta_4 \phi_1(\mathbf{x}_k, \boldsymbol{\mu}_{k+1}; \sigma_{k+1}).$$

Proof : From the definition of ϕ_1 we have

$$\begin{aligned} \phi_1(\mathbf{x}, \bar{\boldsymbol{\mu}}; \bar{\sigma}) &= \bar{\sigma} P_1(\mathbf{x}) - \langle \bar{\boldsymbol{\mu}}, \mathbf{g}(\mathbf{x})\rangle \\ &\geq \bar{\sigma} P_1(\mathbf{x}) - \sum_{i \in \mathcal{M}} \bar{\mu}^i P_1(\mathbf{x}) \\ &= \left(\bar{\sigma} - \sum_{i \in \mathcal{M}} \bar{\mu}^i\right) P_1(\mathbf{x}). \end{aligned}$$

If the penalty parameter is sufficiently large, that is

$$\bar{\sigma} > \sum_{i \in \mathcal{M}} \bar{\mu}^i,$$

then define

$$\beta_4 = \left(1 \Big/ \left(\bar{\sigma} - \sum_{i \in \mathcal{M}} \bar{\mu}^i\right)\right)$$

and the result of the lemma holds. $\square$

The next corollary is a direct consequence of Lemmas 5.2 and 5.3. It will be used to show that this stepsize strategy, based on satisfaction of (5) achieves global convergence and convergence of the stepsizes to unity.

Corollary 5.1 *There exists* $\beta_5 > 0$ *such that:*

$$\|\mathbf{d}_k\| \leq \beta_5 \phi_1(\mathbf{x}_k, \boldsymbol{\mu}_{k+1}; \sigma_{k+1}).$$

The next theorem shows that the stepsize strategy that employs Φ_1 is well defined.

Theorem 5.1 *Let assumptions A1–A2, B1 be satisfied. Also let conditions* (5), (6) *be satisfied at the solution of* (QPS)*. If the penalty parameter is chosen sufficiently large, then there exists* $\alpha_k \in (0, 1]$ *such that the Armijo condition* (4) *is satisfied for* $c_1 \in (0, 1]$.

Proof : We begin by finding suitable bounds for the components of the merit function, and use them along with the aforementioned Lemmas in order to show that the merit function decreases at each iteration.

A first order Taylor series expansion on $g^i(\bar{\mathbf{x}})$ around the point $\bar{\mathbf{x}} = \mathbf{x} + \alpha\mathbf{d}$ using Assumption A2 and (3b) yields

$$\begin{aligned}
g^i(\bar{\mathbf{x}}) &= g^i(\mathbf{x}) + \alpha\langle \mathbf{d}, \boldsymbol{\nabla} g^i(\mathbf{x} + t^i\alpha\mathbf{d})\rangle, \qquad 0 \leq t^i \leq 1 \\
&= g^i(\mathbf{x}) + \alpha\langle \mathbf{d}, \boldsymbol{\nabla} g^i(\mathbf{x})\rangle + \alpha\langle \mathbf{d}, (\boldsymbol{\nabla} g^i(\mathbf{x} + t^i\alpha\mathbf{d}) - \boldsymbol{\nabla} g^i(\mathbf{x}))\rangle \\
&\leq g^i(\mathbf{x}) + \alpha\langle \mathbf{d}, \boldsymbol{\nabla} g^i(\mathbf{x})\rangle + \alpha \|\mathbf{d}\| \left\|\boldsymbol{\nabla} g^i(\mathbf{x} + t^i\alpha\mathbf{d}) - \boldsymbol{\nabla} g^i(\mathbf{x})\right\| \\
&\leq g^i(\mathbf{x}) + \alpha\langle \mathbf{d}, \boldsymbol{\nabla} g^i(\mathbf{x})\rangle + \alpha^2\gamma \|\mathbf{d}\|^2 \\
&\leq g^i(\mathbf{x}) - \alpha g^i(\mathbf{x}) + \alpha^2\gamma \|\mathbf{d}\|^2 .
\end{aligned}$$

By maximizing both sides, we obtain

$$P_1(\bar{\mathbf{x}}) \leq (1-\alpha)P_1(\mathbf{x}) + \alpha^2 \gamma \|\mathbf{d}\|^2 . \tag{16}$$

Similarly, a second order Taylor series expansion on $f(\bar{\mathbf{x}})$ around the same point

$$\begin{aligned} f(\bar{\mathbf{x}}) &= f(\mathbf{x}) + \alpha \langle \mathbf{d}, \boldsymbol{\nabla} f(\mathbf{x}) \rangle + \alpha^2 G_f \\ &\leq f(\mathbf{x}) + \alpha \langle \mathbf{d}, \boldsymbol{\nabla} f(\mathbf{x}) \rangle + (1/2)\, \alpha^2 \langle \mathbf{d}, \mathbf{B}\mathbf{d} \rangle + \alpha^2 \|\mathbf{d}\|^2 \eta \end{aligned}$$

where

$$\eta = \int_0^1 (1-t) \left\| \boldsymbol{\nabla}^2 f(\mathbf{x} + t\alpha \mathbf{d}) - \mathbf{B} \right\| dt.$$

If we now use (3a), the second and third term of the last inequality give

$$\alpha \langle \mathbf{d}, \boldsymbol{\nabla} f(\mathbf{x}) \rangle + (1/2)\, \alpha^2 \langle \mathbf{d}, \mathbf{B}\mathbf{d} \rangle \;=\; -\alpha \langle \mathbf{d}, \boldsymbol{\nabla} \mathbf{g}(\mathbf{x}) \bar{\boldsymbol{\mu}} \rangle + \big((1/2)\, \alpha^2 - \alpha \big) \langle \mathbf{d}, \mathbf{B}\mathbf{d} \rangle.$$

The last inequality holds from Assumption B1 for any $\alpha \in [0,1]$. Thus we obtain

$$f(\bar{\mathbf{x}}) \leq f(\mathbf{x}) - \alpha \langle \mathbf{d}, \boldsymbol{\nabla} \mathbf{g}(\mathbf{x}) \bar{\boldsymbol{\mu}} \rangle + \alpha^2 \|\mathbf{d}\|^2 \eta. \tag{17}$$

Combining inequalities (16), (17)

$$
\begin{aligned}
\Phi_1(\bar{\mathbf{x}};\bar{\sigma}) &= f(\bar{\mathbf{x}}) + \bar{\sigma} P_1(\bar{\mathbf{x}}) \\
&\leq f(\mathbf{x}) - \alpha \langle \mathbf{d}, \boldsymbol{\nabla}\mathbf{g}(\mathbf{x})\bar{\boldsymbol{\mu}} \rangle + \alpha^2 \left\|\mathbf{d}\right\|^2 \eta + \\
&\quad + \bar{\sigma}(1-\alpha) P_1(\mathbf{x}) + \bar{\sigma}\alpha^2 \gamma \left\|\mathbf{d}\right\|^2 \\
&= \Phi_1(\mathbf{x};\bar{\sigma}) - \alpha \left(\bar{\sigma} P_1(\mathbf{x}) + \langle \mathbf{d}, \boldsymbol{\nabla}\mathbf{g}(\mathbf{x})\bar{\boldsymbol{\mu}} \rangle\right) + \\
&\quad + \alpha^2 \left\|\mathbf{d}\right\|^2 (\eta + \bar{\sigma}\gamma) \\
&= \Phi_1(\mathbf{x};\bar{\sigma}) - \alpha \phi_1(\mathbf{x}, \bar{\boldsymbol{\mu}}; \bar{\sigma}) + \alpha^2 \left\|\mathbf{d}\right\|^2 (\eta + \bar{\sigma}\gamma)\,,
\end{aligned}
$$

where in the last relation we have used the definition of ϕ_1. If we employ Lemma 5.2 and Corollary 5.1 we obtain the following bound

$$
\begin{aligned}
\Delta\Phi_1 &\leq -\alpha \phi_1(\mathbf{x}, \bar{\boldsymbol{\mu}}; \bar{\sigma}) + \alpha^2 \left\|\mathbf{d}\right\|^2 (\eta + \bar{\sigma}\gamma) \\
&\leq -\alpha \phi_1(\mathbf{x}, \bar{\boldsymbol{\mu}}; \bar{\sigma}) + \beta_3\beta_5\alpha^2 \phi_1(\mathbf{x}, \bar{\boldsymbol{\mu}}; \bar{\sigma}) P_1(\mathbf{x}) (\eta + \bar{\sigma}\gamma) \\
&= -\alpha \phi_1(\mathbf{x}, \bar{\boldsymbol{\mu}}; \bar{\sigma}) \left(1 - \alpha\beta_3\beta_5 P_1(\mathbf{x}) (\eta + \bar{\sigma}\gamma)\right) \qquad (18)
\end{aligned}
$$

where for convenience we have set $\Delta\Phi_1 = \Phi_1(\bar{\mathbf{x}};\bar{\sigma}) - \Phi_1(\mathbf{x};\bar{\sigma})$.

From inequality (18), and since $0 < c_1 < 1$, there is $\alpha \in (0,1]$ such that

$$
c_1 \leq 1 - \alpha\beta_3\beta_5 P_1(\mathbf{x}) (\eta + \bar{\sigma}\gamma) \leq 1. \qquad (19)
$$

By Lemma 5.1, we have that $\phi_1(\mathbf{x}, \bar{\boldsymbol{\mu}}; \bar{\sigma}) \geq 0$, therefore the Armijo condition (4) must be satisfied for this α. Assume that $\tilde{\alpha}$ is the largest step in the interval $(0,1]$ satisfying inequality (4). Then for every $\alpha \leq \tilde{\alpha}$ Armijo's condition is satisfied, and the selected $\alpha \in [\theta\tilde{\alpha}, \tilde{\alpha}]$, where $\theta \in (0, 1]$. $\square$

Lemma 5.4 *Let the assumptions of Theorem 5.1 be satisfied, and let for $k_0 \geq 0$ and $k \geq k_0$, the set*

$$\mathcal{F}_1 = \left\{ \mathbf{x} \in \mathbb{R}^n \,\middle|\, \Phi_1(\mathbf{x}; \sigma) \leq \Phi_1(\mathbf{x}_{k_0}; \sigma) \right\}$$

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

$$\lim_{k \to \infty} \phi_1(\mathbf{x}_k, \boldsymbol{\mu}_{k+1}; \sigma_{k+1}) = 0.$$

Proof : The scalar $c_1 \in (0, 1)$ in the Armijo stepsize strategy corresponding to $t = 1$ (Step 31), determines a stepsize α such that

$$c_1 \leq 1 - \alpha\beta_3\beta_5 P_1(\mathbf{x}) \left(\eta + \bar{\sigma}\gamma\right) \leq 1.$$

Solving for α we obtain

$$\alpha \leq \left(1 - c_1\right) / \beta_3\beta_5 P_1(\mathbf{x}) \left(\eta + \gamma\bar{\sigma}\right). \tag{20}$$

Therefore, in order to satisfy the Armijo condition, the largest value the stepsize parameter α can take is

$$\tilde{\alpha} = \min \left\{1, \left(1 - c_1\right) / \beta_3\beta_5 P_1(\mathbf{x}) \left(\eta + \gamma\bar{\sigma}\right)\right)\right\}.$$

As the objective function is twice continuously differentiable and the level set $\mathcal{F}_1$ is bounded, it follows that there exists a scalar $\tilde{M}_1$ such that

$$\eta \leq \tilde{M}_1 < \infty.$$

If parameter $\bar{\sigma}$ is not unbounded we have that

$$\alpha \geq \tilde{\alpha} > 0.$$

Hence, the stepsize α is always bounded away from zero. From the Armijo rule and Lemma 5.1 we now obtain that

$$\Phi_1(\bar{\mathbf{x}}; \bar{\sigma}) - \Phi_1(\mathbf{x}; \bar{\sigma}) \leq -c_1 \phi_1(\mathbf{x}, \bar{\boldsymbol{\mu}}; \bar{\sigma}) \leq 0. \tag{21}$$

By the boundedness assumption on $\mathcal{F}_1$, we deduce that

$$\{|\Phi_1(\bar{\mathbf{x}}; \bar{\sigma}) - \Phi_1(\mathbf{x}; \bar{\sigma})|\} \to 0$$

and since $c_1, \alpha > 0$ the lemma follows from (21). □

5.2 The l_2 Merit Function

The l_2 merit function is activated if either i) condition (5) is not satisfied, or ii) conditions (5), (10) are satisfied and condition (6) is violated.

The convergence properties of stepsize strategies that employ the l_2 merit function have been studied in an algorithm by Rustem [15], and we shall not repeat them here. In this section, the main result (Corollary 5.2) is a bound between the directional derivative of the l_2 merit function and the direction of search. Lemmas 5.5–5.8 support this result and show that the direction of search generated by the (QPS) is a descent direction for Φ_2. This is exploited to instantiate global convergence of the presented algorithm, with a suitable modification, presented later.

Lemma 5.5 *If* $(\mathbf{d}_k, \boldsymbol{\mu}_{k+1})$ *is calculated by the* (QPS) *and* σ_{k+1} *is chosen as in Step 14 then* Φ_2 *is decreased at the current point.*

Proof : Using equation (3a), inequality (9) and Assumption B1 we have that

$$\begin{aligned}
-\phi_2(\mathbf{x}, \mathbf{d}, \boldsymbol{\pi}; \bar{\sigma}) &\leq \langle \mathbf{d}, \boldsymbol{\nabla} f(\mathbf{x}) \rangle \\
&= -\langle \mathbf{d}, \mathbf{B}\mathbf{d} \rangle + \langle \bar{\boldsymbol{\mu}}, \mathbf{g}(\mathbf{x}) \rangle \\
&\leq -\langle \mathbf{d}, \mathbf{B}\mathbf{d} \rangle \\
&\leq -\beta_1 \|\mathbf{d}\|^2 \leq 0
\end{aligned}$$

which yields the required result. □

Lemma 5.6 *If* $(\mathbf{d}_k, \boldsymbol{\mu}_{k+1})$ *is calculated by the* (QPS) *and* σ_{k+1} *is chosen as in Step 18, then* Φ_2 *is decreased at the current point.*

Proof : Using the fact that in this step

$$\langle \mathbf{d}, \boldsymbol{\nabla} f(\mathbf{x}) \rangle + \langle \mathbf{d}, \mathbf{B}\mathbf{d} \rangle \leq \sigma \left\| [\mathbf{g}(\mathbf{x})]^+ \right\|^2 + \langle [\mathbf{g}(\mathbf{x})]^+, \boldsymbol{\pi} \rangle,$$

and (9) we obtain

$$\begin{aligned}
-\phi_2(\mathbf{x}, \mathbf{d}, \boldsymbol{\pi}; \bar{\sigma}) &\leq \langle \mathbf{d}, \boldsymbol{\nabla} f(\mathbf{x}) \rangle - \langle [\mathbf{g}(\mathbf{x})]^+, \boldsymbol{\pi} \rangle - \bar{\sigma} \left\| [\mathbf{g}(\mathbf{x})]^+ \right\|^2 \\
&\leq -\langle \mathbf{d}, \mathbf{B}\mathbf{d} \rangle \\
&\leq -\beta_1 \|\mathbf{d}\|^2 \leq 0.
\end{aligned}$$

which proves the lemma. □

Lemma 5.7 *If* $(\mathbf{d}_k, \boldsymbol{\mu}_{k+1})$ *is calculated by the* (QPS) *and* σ_{k+1} *is chosen as in Step 20, then* Φ_2 *is decreased at the current point.*

Proof : Using inequality (9) and the update of the penalty parameter in this step, the following bound is obtained

$$
\begin{aligned}
-\phi_2(\mathbf{x}, \mathbf{d}, \boldsymbol{\pi}; \bar{\sigma}) &\leq \langle \mathbf{d}, \boldsymbol{\nabla} f(\mathbf{x})\rangle - \langle [\mathbf{g}(\mathbf{x})]^+, \boldsymbol{\pi}\rangle - \bar{\sigma} \left\| [\mathbf{g}(\mathbf{x})]^+ \right\|^2 \\
&\leq -\langle \mathbf{d}, \mathbf{B}\mathbf{d}\rangle \\
&\leq -\beta_1 \left\| \mathbf{d} \right\|^2 \leq 0.
\end{aligned}
$$

which gives the required result. □

Lemma 5.8 *If* $(\mathbf{d}_k, \boldsymbol{\mu}_{k+1})$ *is calculated by the* (QPS) *and* σ_{k+1} *is chosen as in Step 10, then* Φ_2 *is decreased at the current point.*

Proof : If $\mathbf{x}$ is feasible, then any descent property associated with Φ_1 will also hold for Φ_2. Therefore we assume that $\mathbf{x}$ is not feasible and in fact that $\|[\mathbf{g}(\mathbf{x})]^+\| > \sqrt{\epsilon_g}$. Note that in this case $P_1(\mathbf{x}) = \|[\mathbf{g}(\mathbf{x})]^+\|_\infty$. From the update of the penalty parameter in this step we obtain

$$
\begin{aligned}
\bar{\sigma} &\geq \left(\sqrt{m}/\sqrt{\epsilon_g}\right) \sum_{i \in \mathcal{M}} \bar{\mu}^i \\
&= \left(\sum_{i \in \mathcal{M}} \bar{\mu}^i \left\| [\mathbf{g}(\mathbf{x})]^+ \right\|^2 / \left\| [\mathbf{g}(\mathbf{x})]^+ \right\|^2 \right) \left(\sqrt{m}/\sqrt{\epsilon_g}\right) \\
&\geq \frac{\sum_{i \in \mathcal{M}} \bar{\mu}^i \left\| [\mathbf{g}(\mathbf{x})]^+ \right\|_1}{\left\| [\mathbf{g}(\mathbf{x})]^+ \right\|^2} \frac{\left\| [\mathbf{g}(\mathbf{x})]^+ \right\|}{\sqrt{m}} \frac{\sqrt{m}}{\sqrt{\epsilon_g}} && (22) \\
&= \left(\sum_{i \in \mathcal{M}} \bar{\mu}^i \sum_{i \in \mathcal{M}} [g^i(\mathbf{x})]^+ / \left\| [\mathbf{g}(\mathbf{x})]^+ \right\|^2 \right) \left(\left\| [\mathbf{g}(\mathbf{x})]^+ \right\| / \sqrt{\epsilon_g} \right) && (23)
\end{aligned}
$$

where inequality (22) follows from standard norm equivalence inequalities and equality (23) follows from the definition of the l_1 vector norm.

Bearing in mind that the summands of the numerator of the first fraction are both positive we obtain

$$\begin{aligned}\bar{\sigma} &\geq \left(\sum_{i\in\mathcal{M}} \bar{\mu}^i [g^i(\mathbf{x})]^+ / \left\|[\mathbf{g}(\mathbf{x})]^+\right\|^2\right) \left(\left\|[\mathbf{g}(\mathbf{x})]^+\right\| / \sqrt{\epsilon_g}\right) \\ &\geq \left(\sum_{i\in\mathcal{M}} \bar{\mu}^i g^i(\mathbf{x}) / \left\|[\mathbf{g}(\mathbf{x})]^+\right\|^2\right) \left(\left\|[\mathbf{g}(\mathbf{x})]^+\right\| / \sqrt{\epsilon_g}\right)\end{aligned}$$

where the last inequality follows from standard max-function properties. Finally using (3a), and the fact that $\|[\mathbf{g}(\mathbf{x})]^+\|^2 > \epsilon_g$ we obtain

$$\begin{aligned}\bar{\sigma} &\geq \left((\langle \mathbf{d}, \boldsymbol{\nabla} f(\mathbf{x})\rangle + \langle \mathbf{d}, \mathbf{B}\mathbf{d}\rangle) / \left\|[\mathbf{g}(\mathbf{x})]^+\right\|^2\right) \left(\left\|[\mathbf{g}(\mathbf{x})]^+\right\| / \sqrt{\epsilon_g}\right) \\ &> \left((\langle \mathbf{d}, \boldsymbol{\nabla} f(\mathbf{x})\rangle + \langle \mathbf{d}, \mathbf{B}\mathbf{d}\rangle) / \left\|[\mathbf{g}(\mathbf{x})]^+\right\|^2\right) \\ &\geq \left((\langle \mathbf{d}, \boldsymbol{\nabla} f(\mathbf{x})\rangle + \langle \mathbf{d}, \mathbf{B}\mathbf{d}\rangle - \langle [\mathbf{g}(\mathbf{x})]^+, \boldsymbol{\pi}\rangle) / \left\|[\mathbf{g}(\mathbf{x})]^+\right\|^2\right).\end{aligned}$$

Therefore $\bar{\sigma}$ is large enough to guarantee descent of Φ_2 along $\mathbf{d}$, and in fact

$$-\phi_2(\mathbf{x}, \mathbf{d}, \boldsymbol{\pi}; \bar{\sigma}) \leq -\beta_1 \|\mathbf{d}\|^2 \leq 0.$$

which proves the lemma. □

Lemma 5.9 *If* $(\mathbf{d}_k, \boldsymbol{\mu}_{k+1})$ *is calculated by the* (QPS) *and* σ_{k+1} *chosen as in Step 26, then* Φ_2 *is decreased at the current point.*

Proof : Since

$$\langle \mathbf{d}, \boldsymbol{\nabla} f(\mathbf{x})\rangle + \langle \mathbf{d}, \mathbf{B}\mathbf{d}\rangle \leq 0,$$

then using (9), we obtain

$$\begin{aligned}
-\phi_2(\mathbf{x}, \mathbf{d}, \boldsymbol{\pi}; \bar{\sigma}) &\leq \langle \mathbf{d}, \boldsymbol{\nabla} f(\mathbf{x})\rangle - \bar{\sigma} \left\|[\mathbf{g}(\mathbf{x})]^+\right\|^2 - \langle [\mathbf{g}(\mathbf{x})]^+, \boldsymbol{\pi}\rangle \\
&\leq -\langle \mathbf{d}, \mathbf{B}\mathbf{d}\rangle - (\bar{\sigma} \left\|[\mathbf{g}(\mathbf{x})]^+\right\|^2 + \langle [\mathbf{g}(\mathbf{x})]^+, \boldsymbol{\pi}\rangle) \\
&\leq -\langle \mathbf{d}, \mathbf{B}\mathbf{d}\rangle \\
&\leq -\beta_1 \|\mathbf{d}\|^2 \leq 0
\end{aligned}$$

which proves the lemma. □

The next corollary summarizes the results of lemmas 5.5–5.9. This is an important result that states that convergence of the algorithm can be monitored using the l_2 merit function.

Corollary 5.2 *Let assumptions A1, B1 hold. If* $(\mathbf{d}_k, \boldsymbol{\mu}_{k+1})$ *is calculated by the* (QPS)*, then*

$$-\phi_2(\mathbf{x}_k, \mathbf{d}_k, \boldsymbol{\pi}_{k'}; \sigma_{k+1}) \leq -\beta_1 \|\mathbf{d}_k\|^2 \leq 0. \tag{24}$$

5.3 Overall Algrorithm

Lemma 5.10 *Let Assumptions A1,A4 hold, and let* $(\mathbf{d}_k, \boldsymbol{\mu}_{k+1})$ *be generated by* (QPS)*. The update of the penalty parameter is such that* σ_k *does not become unbounded.*

Proof : We shall prove the lemma by contradiction. Let us assume that $\sigma \to \infty$ as $k \to \infty$. The penalty parameter changes value in steps 10, 20 of Algorithm 1. Assumption A4 guarantees that the Lagrange multiplier vector $\bar{\boldsymbol{\mu}}$ will be bounded, therefore the update in Step 10 precludes such a behaviour.

From the way the penalty parameter is chosen in Step 20, $\sigma \to \infty$ only if $\|[\mathbf{g}(\mathbf{x})]^+\| \to 0$. Hence, there exists an integer k_1 such that, for all $k \geq k_1$ we have that

$$0 < \left\|[\mathbf{g}(\mathbf{x})]^+\right\| \leq \sqrt{\epsilon_g}.$$

But if this is the case, then the penalty parameter is not updated, as seen in Step 14. Therefore the maximum value that σ can take is

$$\sigma_* = \mathcal{D}/\epsilon_g$$

where, both ϵ_g, $\mathcal{D}$ are finite values. Thence $\sigma_* < \infty$, which contradicts our assumption that $\sigma \to \infty$ as $k \to \infty$. $\square$

For the proof of the main result (Theorem 5.2) we need to show that $\|\mathbf{d}_k\| \to 0$. In order to prove the latter we must guarantee that a sequence, related to the search direction, is decreasing monotonically.

If after an iteration the algorithm only activates the stepsize strategy that corresponds to $t = 1$, then from Theorem 5.1 we see that $\{\Phi_1(\mathbf{x}_k; \sigma_{k+1})\}$ is indeed monotonically decreasing, and hence from Corollary 5.1 and Lemma 5.4 we can deduce that $\|\mathbf{d}_k\| \to 0$. A similar conclusion holds if the algorithm only activates the stepsize strategy that corresponds to $t = 2$ or $t = 3$ (the related proofs are in [15, 28]).

For many problems though, this behavior may not be observed. In order to establish convergence we modify our algorithm and require that Φ_2 decreases at every iteration, irrespective of which stepsize strategy is activated. We change our stepsize acceptance condition in Step 31 so that in addition to the Armijo condition

$$\Phi_t(\mathbf{x}_k + \alpha \mathbf{d}_k; \sigma_{k+1}) - \Phi_t(\mathbf{x}_k; \sigma_{k+1}) \leq -\alpha c_t \phi_t(\mathbf{d}_k), \tag{25a}$$

which takes the form of (4), (7) or (11) depending on t, we also satisfy

$$\Phi_2(\mathbf{x}_k + \alpha \mathbf{d}_k; \sigma_{k+1}) < \Phi_2(\mathbf{x}_k; \sigma_{k+1}). \tag{25b}$$

Such a requirement can be met since, as shown in Corollary 5.2, the search direction is a direction of descent for Φ_2. Note that in (25b) we require that Φ_2 merely decreases and that it does not decrease sufficiently, in order to intervene as little as possible with the stepsizes generated by the Armijo condition (4), (7) or (11) (depending on whether $t = 1, 2$ or 3 is chosen by the algorithm).

Lemma 5.11 *Let the assumptions of Corollary 5.2 hold. Assume that the stepsize satisfies* (25b) *in addition to the Armijo condition of the activated stepsize strategy (*(4), (7) *or* (11)*). In addition let for $k_0 \geq 0$ and $k \geq k_0$, the set*

$$\mathcal{F}_2 = \left\{ \mathbf{x} \in \mathbb{R}^n \,\middle|\, \Phi_2(\mathbf{x}; \sigma_*) \leq \Phi_2(\mathbf{x}_{k_0}; \sigma_*) \right\}$$

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

$$\lim_{k \to \infty} \|\mathbf{d}_k\| = 0. \tag{26}$$

Proof : Having established monotonic decrease of Φ_2 and Corollary 5.2 the lemma is easy to establish [see 15]. □

Theorem 5.2 (Algorithm convergence) *Let Assumptions A1–A4, B1 be satisfied. Then*

1. *The algorithm either terminates at a Kuhn-Tucker point of problem ICP, or generates an infinite sequence* $\{\mathbf{x}_k\}$*, for which there exists a subsequence* $\{\mathbf{x}_k\}$*,* $k_* \le k \in \mathcal{K} \subseteq \{1, 2, \ldots\}$ *every accumulation point* $\mathbf{x}_*$ *of which is a Kuhn-Tucker point of problem ICP.*

2. *If furthermore, strict complementarity holds at the solution of problem QPS for large* $k, k_* \le k \in \mathcal{K}$*,* $\boldsymbol{\mu}_k$ *predicts the active inequality constraints at* $\mathbf{x}_*$*.*

Proof : By Lemma 5.11 there exists a subsequence $\{\mathbf{x}_k\}$, $k_* \le k \in \mathcal{K}$, such that $\|\mathbf{d}_k\| \to 0$. Let there exist $(\mathbf{x}_*, \boldsymbol{\mu}_*)$ such that $\{\mathbf{x}_k\} \to \mathbf{x}_*$, $\{\boldsymbol{\mu}_k\} \to \boldsymbol{\mu}_*$, $k_* \le k \in \mathcal{K}$. The existence of such points is ensured since the algorithm decreases Φ_2 monotonically, therefore $\mathbf{x}_k$ belongs to $\mathcal{F}_2$ and $\mathcal{F}_2$ is compact.

We need to show that $(\mathbf{x}_*, \boldsymbol{\mu}_*)$ satisfy the first order optimality conditions of problem ICP. In order to achieve that, we consider the KKT conditions (3a) of problem QPS, and let $k \to \infty, k_* \le k \in \mathcal{K}$, using Lemma 5.11. We have

$$\begin{aligned}
\boldsymbol{\nabla}\mathcal{L}(\mathbf{x}_*, \boldsymbol{\mu}_*) &= \mathbf{0} && \text{(27a)} \\
\mu_*^i g^i(\mathbf{x}_*) &= 0, \quad i \in \mathcal{M} && \text{(27b)} \\
\mathbf{g}(\mathbf{x}_*) &\le \mathbf{0} && \text{(27c)} \\
\boldsymbol{\mu}_* \in \mathbb{R}^m, \boldsymbol{\mu}_* &\ge \mathbf{0} && \text{(27d)}
\end{aligned}$$

which is the required result.

For the second part, we note that from (27b), (27d), for sufficiently large k and with strict complementarity holding at the solution of problem QPS that for inactive constraints

$$\langle \mathbf{d}_k, \boldsymbol{\nabla} g^i(\mathbf{x}_k)\rangle + g^i(\mathbf{x}_k) < 0, \quad \mu^i_{k+1} = 0$$

as we approach $\mathbf{x}_*$. Therefore

$$g^i(\mathbf{x}_*) < 0, \quad \mu^i_* = 0$$

that is to say, inactive inequality constraints are not predicted to be active at the solution. □

6 Local Convergence

In this section we present local convergence properties of the proposed algorithm. In order to ensure superlinear convergence, it is necessary to show that $\mathbf{x}_k + \mathbf{d}_k$ is actually accepted by the stepsize strategy. It is actually not easy to establish convergence to unity stepsize for the l_∞ merit function (and generally non-differentiable merit functions).

We have isolated the use of the l_∞ merit function in order to demonstrate the overall convergence of the stepsize to unity. The l_2 merit function has been shown to lead to unit steps in [15], whereas a similar result has been established for the Euclidean norm of the KKT residual in [28]. In the following result we show the attainment of unity stepsizes by the l_∞ merit function.

Theorem 6.1 (Step length convergence) *Let the assumptions of Lemma 5.4 be satisfied. Also, suppose that the sequence* $\{\mathbf{x}_k\}$ *converges to a KKT point. Then for the stepsize strategy employing the* l_∞ *merit function, we have that for sufficiently large* $k \geq k_0$

$$\{\alpha_k\} \rightarrow 1.$$

Proof : Following the argument of Lemma 5.4 we have that the largest value the stepsize parameter can take is

$$\tilde{\alpha}_k = \min\left\{1, (1 - c_1) / \beta_3 \beta_5 P_1(\mathbf{x}_k) \left(\eta_k + \gamma \sigma_{k+1}\right)\right\}.$$

The terms multiplied by $P_1(\mathbf{x}_k)$ are bounded. Also for sufficiently large k we have that $P_1(\mathbf{x}_k) \rightarrow 0$ and therefore for sufficiently large k we obtain $\tilde{\alpha}_k = 1$.

Since α_k is obtained by reducing the maximum allowable step length ($\tilde{\alpha}_k$) until Armijo's condition is satisfied, it follows that Armijo's condition will be satisfied immediately without any reductions, hence proving the desired result. □

7 Numerical Results

For the numerical testing of the algorithm we solved two categories of problems. The first category of problems consists of small problems from the collection of Hock and Schittkowski [1]. The second category consists of large scale problems. The algorithm was implemented using `ANSI C`. Most of the problems of the first category were coded using standard `C` and interfaced with the solver. The algorithm was also interfaced with the mathematical programming language

AMPL, and the large scale problems were solved using AMPL.

We experimented with two different ways for updating the Hessian matrix. The first option involves the `damped BFGS` formula suggested by Powell [11]. In the second option, the exact Hessian is calculated, with second derivatives being drawn either programmatically or from AMPL.

The stopping criteria of the algorithm are as follows. Either the KKT conditions are satisfied at the current point or no further progress can be made The tolerance for the residual of the KKT conditions is 10^{-8}. The lack of progress is signified by

$$\max_{i=1,\ldots,n} \left\{ \left| d_k^i \right| / \max \left\{ x_k^i, 1/typx^i \right\} \right\} < 10^{-8},$$

where in our implementation **typx** is a vector of all ones. We also check if the initial point is a KKT point. In that case the tolerance is decreased by 1000. In the merit function switch $\epsilon_g = \epsilon_d = 10^{-6}$, and $\delta = 1$. The initial value of the penalty parameter is $\sigma_0 = 1.0$. In the line search procedure $\theta = 0.5$ and $c = 10^{-4}$.

Tables 1, 2 summarize the numerical results for the test-problems found in the Hock and Schittkowski collection. Table 3 summarizes the results for the large-scale problems. In the tables the following abbreviations are used :

- No is the name of the problem.
- k is the number of iterations to find the optimum solution.
- σ_* is the final value of the penalty parameter.
- k_* is the iteration after which the search direction decreases the merit function without truncations.

- Φ_1 shows the number of times the l_∞ merit function was used in the problem.

- Φ_2 shows the number of times the l_2 merit function was used in the problem.

- Φ_3 shows the number of times the Euclidean norm of the KKT conditions was used in the problem.

- n is the number of variables (for the larger-scale problems).

- m is the total number of constraints, not including simple bounds (for the larger-scale problems).

All the numerical results were obtained by using an exact initial Hessian, and the `BFGS` updating formula. For problems marked with u, the initial Hessian matrix is a diagonal positive definite matrix and the updates are performed using the `BFGS` formula. For problems 88 to 92, the condition number of the Hessian was too large, and we therefore avoided using an exact initial Hessian. For problem 1 the algorithm converged with the standard choice in 44 iterations, but the alternative choice gave much better results. In problem 2, the standard choice converged to the same point as other codes, but not the one reported in [1], and therefore the alternative initial Hessian was used. In problem 31, the expected results were obtained, but the modification yielded convergence with a step less. But for problem 38 convergence was much slower with the standard choice (89 iterations). In problem 73, the algorithm converged within two digits for the iterates (9 iterations), but the convergence using the alternative was much better and faster. In problem 81 we exceeded the maximum number of iterations when using the standard choice, and therefore reverted to the alternative one.

For Problems marked with e exact Hessian approximations were used. In problems 7, 12 there was no problem converging with the standard choice[2], but the reported alternative gave marginally better results. The same is true for problems 47, 49, 64, 72, 74, 77, 80, 102, 103, 112[3]. In problems 27, 56 the maximum number of iterations was exceeded, and the objective function converged to 4 digits, but exact Hessian approximations gave superior results. For problem 56 we note that there was only one truncation at the beginning of the calculation, but then unity stepsizes were employed throughout. The search directions were quite small though, and thus the convergence was not as fast as expected.

Inconsistent constraint linearizations (Assumption A3) were handled using the procedure mentioned by Powell [11]. We also introduced a trap in order to cater for cases in which Assumption A4 cannot be met. Namely, if the value of the penalty parameter as obtained in Step 10 was above ϵ_μ, then the penalty parameter is updated as in steps (16)–(21). We set $\epsilon_\mu = 10^6$ in the algorithm. This modification also contributed in attaining small values for the penalty parameter, as displayed in the accompanying results. We also note that the high values of the penalty parameters in problems 63, 72 are mainly due to high values of the Lagrange multipliers.

The algorithm was compared to algorithms `SNOPT`, `LOQO` and `KNITRO` and it turned out to achieve less iterations in many of the small-scale problems. The behavior of the algorithm was very much similar when tested with the problems of Table 3, and the performance there was much better than the aforementioned algorithms, concluding thus that the algorithm is quite efficient in solving some large problems with respect to the number of iterations.

[2] in 12, 10 iterations respectively

[3] in 31, 28, 23, 36, 10, 19, 12, 28, 23, 59 iterations respectively

No	k	σ_*	k_*	Φ_1	Φ_2	Φ_3	No	k	σ_*	k_*	Φ_1	Φ_2	Φ_3
1^{u}	20	1	10	1	18	0	41	10	13.093	0	1	9	0
2^{u}	8	1	3	1	6	0	42	10	5.53978	0	7	0	3
3	3	1	0	0	2	0	43	12	4.11779	3	4	4	3
4	2	1	0	0	1	0	44	7	1	0	0	6	0
5	8	1	0	0	7	0	45	5	1.66667	0	1	3	0
6	6	1	4	5	1	0	46	51	1.06768	51	33	15	3
7^{e}	8	1.79348	0	7	0	0	47^{e}	20	4.90631	0	5	15	0
8	5	4.55556	1	4	0	0	48	1	1	0	0	1	0
9	7	1.04189	7	2	5	0	49^{e}	20	1.00153	0	1	17	2
10	11	1.53305	0	9	0	2	50	12	2.15665	0	1	10	1
11	7	4.05403	0	6	0	0	51	1	1	0	1	0	0
12^{e}	8	1.5	5	1	6	0	52	1	14.9312	0	1	0	0
13	46	1	0	0	45	0	53	1	11.2326	0	1	0	0
14	5	4.44108	0	5	0	0	54	1	1.85714	0	1	0	0
15	4	8,289.54	0	2	1	0	55	2	5.66667	0	1	0	0
16	6	1,995	0	2	3	0	56^{e}	102	3.5839	2	19	40	42
17	20	6.69685	13	5	14	0	57	5	1.00068	0	3	1	0
18	7	1.37538	1	3	3	0	59	14	4.70045	4	1	12	0
19	6	2,582.38	0	5	0	0	60	10	1	5	1	8	0
20	7	6,993	0	1	5	0	61	9	14.3959	0	6	1	1
21	2	1.04	0	1	0	0	62	8	6,388.18	1	2	5	0
22	5	2.33333	0	4	0	0	63	10	728,135	1	6	1	2
23	7	5.41843	0	3	3	0	64^{e}	19	83,027	15	18	0	1
24	5	1.63492	0	1	3	0	65	11	1.08215	6	2	7	1
25	29	1	14	0	28	0	66	6	1.86678	1	4	1	1
26^{e}	23	1	0	1	22	0	70	94	1	92	0	93	0
27^{e}	14	4.97038	8	12	0	2	71	10	3.15084	5	8	0	1
28	1	1	0	0	1	0	72^{e}	25	101,755	1	15	0	9
29	12	1.7098	5	3	6	2	73^{u}	9	68.2886	6	5	2	1
30	16	1	0	0	16	0	74^{e}	5	2,916.74	0	5	0	0
31^{u}	11	9.35691	0	6	2	2	75	6	2,916.74	0	4	1	0
32	1	1	0	0	1	0	76	1	1	0	0	1	0
33	8	1	0	0	7	0	77^{e}	11	1.1177	6	4	5	2
34	7	1.52114	1	3	2	1	78	8	5.56929	0	4	0	3
35	4	1	0	0	4	0	79	7	1	0	0	6	0
36	5	217.68	0	1	3	0	80^{e}	7	1.08334	0	7	0	0
37	8	1	0	0	6	1	81^{u}	14	2.0455	10	12	0	2
38^{u}	46	1	19	9	36	0	83	4	1,710.4	0	2	1	0
39	10	3.97537	0	8	0	2	84	18	1	11	0	17	0
40	12	8.76795	7	9	0	2	86	6	1	0	0	5	0

Table 1: Hock & Schittkowski results

No	k	σ_*	k_*	Φ_1	Φ_2	Φ_3	No	k	σ_*	k_*	Φ_1	Φ_2	Φ_3
87	9	62.079	0	6	1	1	102^{e}	17	15,347.4	6	15	0	1
88^{u}	19	1048.58	14	16	0	2	103^{e}	17	16,060.9	8	15	0	1
89^{u}	30	12,243.9	14	22	0	7	104	16	11.622	3	11	1	3
90^{u}	36	11,447.6	30	12	3	20	105	54	51.2349	24	8	45	0
91^{u}	33	1,048.12	28	23	4	5	106	18	12,285.7	13	9	7	1
92^{u}	25	1,041.72	22	18	1	6	108	18	3.7406	15	14	3	1
93	21	136.555	14	17	1	2	110	5	1	1	1	3	0
95	2	181.261	0	1	0	0	112^{e}	12	38.976	0	2	10	0
96	2	181.261	0	1	0	0	113	9	5.18661	0	4	1	3
97	7	471.531	4	3	3	0	116	23	4,222.57	18	18	3	2
98	7	471.531	4	3	3	0	117	20	1	0	0	19	0
100^{e}	8	2.50842	3	3	4	0	118	1	1	0	0	1	0
101	23	8,543.64	10	14	4	4	119^{e}	8	601,497	0	1	4	3

Table 2: Hock & Schittkowski results

No	n	m	k	σ_*	k_*	Φ_1	Φ_2	Φ_3
fir_linear	11	304	1	8.41662	0	1	0	0
fir_convex	11	304	4	9.0178	0	2	1	0
fir_socp	12	305	4	9.57933	0	3	0	0
fir_exp	12	305	6	10.8008	0	4	0	1
hadamard	56	256	1	2.41667	0	1	0	0
markowitz	8	1	1	1	0	0	1	0
minsurf	36	0	5	1	2	2	2	0
model	60	32	1	1	0	0	1	0
nnls2	150	393	1	162.718	0	1	0	0
obstclal	64	0	2	1	0	0	1	0
oet1	3	1,002	2	2	0	1	0	0
oet3	4	1,002	1	2	0	1	0	0
steenbra	432	108	6	1E+00	0	0	4	2
optctrl6	118	80	11	95,739.3	8	3	7	0
qpcstair	385	356	1	1	0	0	1	0
qpnstair	385	356	77	1	0	0	39	37
reading3	202	102	1	1	0	0	0	0
trimloss	142	72	4	2.75333	3	4	0	0
dittert	327	264	126	46.7225	0	3	113	9
core2	157	122	10	75.0005	4	10	0	0
gouldqp2	699	349	1	1	0	0	1	0
grouping	100	125	1	1	0	0	0	0

Table 3: Large scale problems

8 Conclusions

An SQP algorithm for general nonlinear programming problems has been presented. The algorithm uses three stepsize strategies. Each strategy has its own advantages and disadvantages. Switch rules are implemented that combine all three strategies so as to complement their merits and avoid their drawbacks. The penalty parameter is updated in order to guarantee sufficient decrease of the merit function. Convergence from arbitrary starting point is achieved. It is also shown that when close to the optimum, unity step sizes will be accepted, giving thus rise to fast local convergence. The numerical results of the algorithm demonstrate that the algorithm can be used to solve a variety of nonlinear programming problems effectively.

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

References

[1] Hock, W. and Schittkowski, K., *Test examples for nonlinear programming codes*, Lecture Notes in Economics and Mathematical Systems #187. Springer-Verlag, Berlin, Heidelberg and New York, 1981.

[2] Fletcher, R., *Practical Methods of Optimization,* 2nd ed., John Wiley, New York, 1987.

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

[4] Pardalos, P. M. and Resende, M. G. C., *Handbook of applied optimization*, Oxford University Press, Oxford, 2002.

[5] Floudas, C. A. and Pardalos, P. M., *Encyclopedia of optimization. Vol. I–VI*, Kluwer Academic Publishers, Dordrecht, 2001.

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

[7] 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*, J. Optim. Theory Appl., Vol. 89, No. 3, pp. 507–541, 1996.

[8] Shanno, D. F. and Simantiraki, E. M., *Interior point methods for linear and nonlinear programming*, The state of the art in numerical analysis (York, 1996), volume 63 of *Inst. Math. Appl. Conf. Ser. New Ser.* Oxford Univ. Press, New York, pp. 339–362, 1997.

[9] Pietrzykowski, T., *An Exact Potential Method for Constrained Maxima*, SIAM J. Numer. Anal., Vol. 6, No. 2, pp. 299–304, 1969.

[10] HAN, S. P., *A globally convergent method for nonlinear programming*, J. Optim. Theory Appl., Vol. 22, pp. 297–309, 1977.

[11] POWELL, M. J. D., *A Fast Algorithm for Nonlinearly Constrained Optimization Calculations*, Lecture notes in Mathematics, Numerical Analysis, Dundee 1977, Edited by G.A. Watson, Dundee, Scotland, Springer-Verlag, 1977.

[12] POWELL, M. J. D., *Algorithms for Nonlinear Constraints That Use Lagrangian Functions*, Math. Program., Vol. 14, 1978.

[13] HAN, S. P., *Superlinearly convergent variable metric algorithms for general nonlinear programming problems*, Math. Program., Vol. 11, pp. 263–282, 1976.

[14] MARATOS, N., *Exact penalty function algorithms for finite-dimensional and control optimization problems*, PhD thesis, University of London, 1978.

[15] RUSTEM, B., *Equality and Inequality Constrained Optimization Algorithms with Convergent Stepsizes*, J. Optim. Theory Appl., Vol. 76, No. 3, pp. 429–453, 1993.

[16] CHAMBERLAIN, R. M., POWELL, M. J. D., LEMARECHAL, C., AND PEDERSEN, H. C., *The Wathdog technique for Forcing Convergence in Algorithms for Constrained Optimization*, Math. Program. Stud., Vol. 16, pp. 1–17, 1982.

[17] PSHENICHNYI, B. N. AND DANILIN, Y. M., *Numerical Methods in Extremal problems*, Hayka, 1975. English translation, 1978.

[18] MAYNE, D. Q. AND POLAK, E., *A Superlinearly Convergent Algorithm for Constrained Optimization Problems*, Math. Program. Stud., Vol. 16, pp. 45–61, 1982.

[19] Tapia, R. A., *Quasi-Newton methods for equality constrained optimization: equivalence of existing methods and a new implementation*, Nonlinear Programming 3, Edited by O. L. Mangasarian and R. R. Meyer and S. M. Robinson. Academic Press, New York, NY, pp. 125–164, 1978.

[20] Nocedal, J. and Wright, S. J., *Numerical Optimization*, Springer-Verlag, New York, 1999.

[21] Gill, P. E., Murray, W., Saunders, M. A., and Wright, M. H. Some theoretical properties of an augmented lagrangian merit function. Technical Report SOL 86-6R, SOL, Stanford University, 1986.

[22] Fukushima, M., *A Successive Quadratic Programming Algorithm with Global and Superlinear Convergence Properties*, Math. Program., Vol. 35, pp. 253–264, 1986.

[23] Panier, E. R. and Tits, A. L., *Avoiding the Maratos Effect by Means of a Nonmonotone Line Search I. General Constrained Problems*, SIAM J. Numer. Anal., Vol. 28, No. 4, pp. 1183–1195, 1991.

[24] Powell, M. J. D. and Yuan, Y., *A recursive quadratic programming algorithm that uses differentiable penalty functions*, Math. Program., Vol. 7, pp. 265–278, 1986.

[25] Yamashita, H., *A globally convergent constrained quasi-Newton method with an augmented Lagrangian type penalty function*, Math. Program., Vol. 23, No. 1, pp. 75–86, 1982.

[26] Byrd, R. H., Nocedal, J., and Waltz, R. A. Steering exact penalty methods. Technical Report Report OTC 05, Optimization Technology Center, Northwestern University, Evanston Il 60208, 2005.

[27] Fletcher, R. and Leyffer, S., *Nonlinear programming without a penalty function*, Math. Program., Vol. 91, No. 2, Ser. A, pp. 239–269, 2002.

[28] Rustem, B., *Convergent Stepsizes for Constrained Optimization Algorithms*, J. Optim. Theory Appl., Vol. 49, pp. 135–160, 1986.

[29] Bertsekas, D. P., *Constrained optimization and Lagrange multipliers*, Academic Press, New York, NY, 1982.