manuscript No.
(will be inserted by the editor)

Dynamic Control of Infeasibility for Nonlinear Programming

Abel S. Siqueira · Francisco A. M. Gomes

June 15, 2016

Abstract An effective way of solving general nonlinear programming problems is the adoption of composite-step strategies that combine a step tangent to the constraints and a normal step, alternating between reducing the objective function value and the norm of the infeasibility. However, this kind of method requires the control of the iterates in order to prevent one step from destroying the progress of the other. In the Dynamic Control of Infeasibility algorithm, proposed by Bielschowsky and Gomes for equality constrained problems, the steps are controlled through the use of the so called Trust Cylinders. We present an extension of this algorithm for solving problems with general constraints. We also show numerical experiments that indicate that the new method has a performance that is comparable to well known nonlinear programing codes.

Keywords nonlinear programming · constrained optimization · numerical algorithms · interior point

Mathematics Subject Classification (2000) 65K05 · 90C30

This work was supported by Fundação de Amparo à Pesquisa do Estado de São Paulo (grants 2009/00576-4, 2009/17273-4 and 2013/05475-7) and Conselho Nacional de Desenvolvimento Científico e Tecnológico (grant 501763/2013-9

A. S. Siqueira
Departamento de Matemática, UFPR, Centro Politécnico - Jd. das Américas, Caixa Postal 19081, CEP 81531-980, Curitiba - PR, Brasil
E-mail: abel.s.siqueira@gmail.com

F. A. M. Gomes
Departamento de Matemática Aplicada, IMECC, UNICAMP, Rua Sergio Buarque de Holanda, 651, CEP 13083-859, Campinas, SP, Brazil.
E-mail: chico@ime.unicamp.br

1 Introduction

In this paper, we present a method for solving nonlinear optimization problems with inequality constraints. The method is an extension of the algorithm proposed in Bielschowsky and Gomes [5], originally devised for equality constrained problems. It uses adaptative cylindrical regions around the feasible set, called Trust Cylinders, to control the iterates and step sizes. This strategy ensures that a step will not leave a reliable region around the feasible set, hence the name Dynamic Control of Infeasibility, or DCI for short.

We follow the composite step approach, that consists of dividing the iteration into a normal and a tangential step. These steps are obtained using a quadratic approximation for the Lagrangian function and a linear approximation for the constraints. This strategy is well-known, and can be seen, for instance, in [5, 7, 8, 11, 12, 18, 19, 21, 23, 26, 33]. In addition, we treat the bounds on the variables using interior-point techniques, controlling the steps with fraction-to-the-boundary rules and a logarithm barrier function, as done in [7, 8, 23, 44].

Many strategies exist for globalizing nonlinear programming methods. The most traditional are line search [4, 6, 13, 44], trust region [7, 8, 12, 13, 18, 19, 21, 22, 25, 26, 33, 36], and the filter approach [22, 25, 28, 31, 35, 37, 38, 44]. The trusts cylinders introduce a new way to obtain global convergence for composite steps methods, without relying on merit function or filters. As far as we are concerned, the only similar approach is the trust-funnel method proposed in [11, 30].

This paper is organized as follows. In Section 2 we introduce the DCI method, detailing the algorithm. Sections 3 and 4 present the global and local convergence properties of the method, respectively. Section 5 shows information about our computational implementation and numerical results, followed by some conclusions and open issues in Section 6.

2 The Method

The Dynamic Control of Infeasibility (DCI) method of Bielchowsky and Gomes [5] was originally proposed for equality constrained problems. This work extends the method to handle the inequality constrained problem

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & c_E(x) = 0, \\ & c_I(x) \geq 0, \end{array} \tag{2.1}$$

where $f : \mathbb{R}^n \longrightarrow \mathbb{R}$, $c_E : \mathbb{R}^n \longrightarrow \mathbb{R}^{m_E}$, $c_I : \mathbb{R}^n \longrightarrow \mathbb{R}^{m_I}$ are twice continuously differentiable functions.

Introducing a vector of slack variables, $s \in \mathbb{R}^{m_I}$, we can rewrite (2.1) as

$$\begin{array}{lll} \min & f(x) & \\ \text{s.t.} & c_E(x) = 0, & \\ & c_I(x) - s = 0, & s \geq 0. \end{array}$$

Moreover, defining the vector $z = \begin{bmatrix} x \\ s \end{bmatrix}$ and the function

$$h(z) = \begin{bmatrix} c_E(x) \\ c_I(x) - s \end{bmatrix}, \tag{2.2}$$

we obtain the equivalent problem

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & h(z) = 0, \qquad s \geq 0. \end{array}$$

Following an interior point approach, we define the function

$$\varphi(z, \mu) = f(x) + \mu\beta(z),$$

where $\beta(z) = -\sum_{i=1}^{m_I} \ln s_i$, and reformulate our problem as

$$\min \varphi(z, \mu) \qquad \text{s.t.} \qquad h(z) = 0. \tag{2.3}$$

The Lagrangian function for this new problem is

$$L(z, \lambda, \mu) = \varphi(z, \mu) + \lambda^T h(z).$$

Since (2.3) is an equality constrained problem, we can use the original DCI method to solve it, although some modifications are required in order to ensure convergence.

Let z_c be an approximate solution for (2.3). A straightforward way to obtain a new point $z_c^+ = z_c + d$ is to approximately solve the problem

$$\min_d \; L(z_c + d, \lambda, \mu) \qquad \text{s.t.} \qquad h(z_c + d) = h(z_c).$$

Unfortunately, this problem can be ill-conditioned due to fact that the derivatives of the objective function of (2.3) involve the inverse of the matrix $S = \operatorname{diag}(s_1, \ldots, s_{m_I})$. In fact, it is easy to see that

$$\nabla\varphi(z, \mu) = \begin{bmatrix} \nabla f(x) \\ -\mu S^{-1} e \end{bmatrix}, \qquad \nabla^2\varphi(z, \mu) = \begin{bmatrix} \nabla^2 f(x) & 0 \\ 0 & \mu S^{-2} \end{bmatrix},$$

where the index z of the derivatives was omitted, that is, $\nabla\varphi(z, \mu) = \nabla_z\varphi(z, \mu)$ and $\nabla^2\varphi(z, \mu) = \nabla^2_{zz}\varphi(z, \mu)$. One way to avoid the ill-conditioning is to introduce the scaling matrix

$$\Lambda(z) = \begin{bmatrix} I & 0 \\ 0 & S \end{bmatrix},$$

so the vector d can be rewritten as $d = \Lambda(z_c)\delta$. With this modification, we define the scaled gradient and the scaled Hessians of φ and L, respectively, as

$$g(z, \mu) = \Lambda(z)\nabla\varphi(z, \mu) = \begin{bmatrix} \nabla f(x) \\ -\mu e \end{bmatrix}, \tag{2.4}$$

$$\Gamma(z, \mu) = \Lambda(z)\nabla^2\varphi(z, \mu)\Lambda(z) = \begin{bmatrix} \nabla^2 f(x) & 0 \\ 0 & \mu I \end{bmatrix}, \tag{2.5}$$

$$W(z, \lambda, \mu) = \Lambda(z)\nabla^2_{zz}L(z, \lambda, \mu)\Lambda(z) = \begin{bmatrix} W_x(x, \lambda) & 0 \\ 0 & \mu I \end{bmatrix}, \tag{2.6}$$

where $m = m_E + m_I$, and $W_x(x,\lambda) = \nabla^2 f(x) + \sum_{i=1}^{m} \lambda_i \nabla^2 c_i(x)$. We also define the scaled Jacobian as

$$A(z) = \nabla h(z) \Lambda(z) = \begin{bmatrix} \nabla c_E(x) & 0 \\ \nabla c_I(x) & -I \end{bmatrix} \Lambda(z) = \begin{bmatrix} \nabla c_E(x) & 0 \\ \nabla c_I(x) & -S \end{bmatrix}.$$

The Lagrange multipliers associated to the problem can be approximated by the least squares solution of the dual feasibility scaled equation

$$\Lambda(z) \nabla L(z,\lambda,\mu) = g(z,\mu) + A(z)^T \lambda = 0.$$

These estimates are given by

$$\lambda_{LS}(z,\mu) = \arg\min_{\lambda} \ \frac{1}{2} \left\| A(z)^T \lambda + g(z,\mu) \right\|^2.$$

Note that, if $A(z)$ has full rank, then

$$\lambda_{LS}(z,\mu) = -[A(z)A(z)^T]^{-1} A(z) g(z,\mu).$$

Rewriting $\lambda_{LS}(z,\mu) = \left[\lambda_E(z,\mu)^T \ \lambda_I(z,\mu)^T\right]^T$, where $\lambda_E(z,\mu) \in \mathbb{R}^{m_E}$ and $\lambda_I(z,\mu) \in \mathbb{R}^{m_I}$, we define the scaled projection of the gradient at z onto the null space of $A(z)$ as

$$g_p(z,\mu) = \Lambda(z) \nabla_z L(z, \lambda_{LS}(z,\mu), \mu) = g(z,\mu) + A(z)^T \lambda_{LS}(z,\mu)$$

The iteration of our method is decomposed into a normal and a tangential step. The purpose of the normal step is to bring the iterates close to the feasible set, while the tangential step drives the iterates towards optimality, keeping feasibility under control.

The DCI method is based on what we call *trust cylinders*. These cylinders are regions around the feasible set defined by

$$\mathcal{C}(\rho) = \{ z \in \mathbb{R}^n : \|h(z)\| \leq \rho \},$$

where ρ is called the radius of the trust cylinder. From an iterate z^{k-1}, the normal step gives z_c^k inside a cylinder with radius ρ^k, and the tangent step obtains z^k inside the cylinder of radius $2\rho^k$. Figure 1 exemplifies the process.

On the algorithm, we use an approximation λ^k to $\lambda_{LS}(z_c^k, \mu^k)$ and define the projected gradient

$$\zeta^k = g(z_c^k, \mu^k) + A(z_c^k)^T \lambda^k = \begin{bmatrix} \nabla f(x_c^k) + \nabla c(x_c^k)^T \lambda^k \\ -\mu^k e - S_c^k \lambda_I^k \end{bmatrix},$$

which is an approximation to $g_p(z_c^k, \mu^k)$. We use the norm of the projected gradient as an optimality measure, and keep the cylinder radii proportional to this measure, which means that $\rho = \mathcal{O}(\|\zeta\|)$.

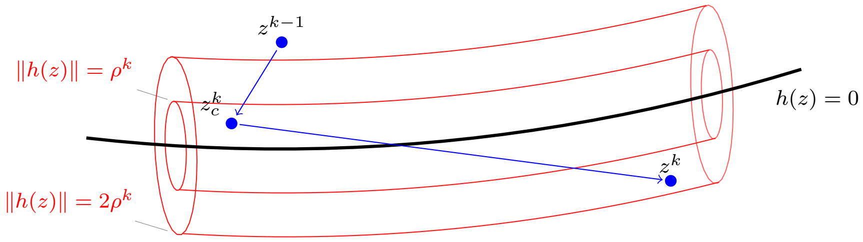


Fig. 1: The feasible set, the cylinders, and the steps of the algorithm. The "normal" iterate z_c^k satisfies $\|h(z_c^k)\| \le \rho^k$ and the "tangential" iterate z^k satisfies $\|h(z^k)\| \le 2\rho^k$.

The iterates are kept strictly feasible with respect to the bounds on the variables. To ensure this, we use a fraction-to-the-boundary rule, i.e., given a point z and a direction d, the step length α must satisfy

$$s + \alpha d_s \ge \varepsilon_\mu s, \tag{2.7}$$

where d_s is the component of d corresponding to s and $\varepsilon_\mu > 0$ is a small constant. An outline of the method is given below.

Algorithm 1 DCI Method

1: Parameters: $\varepsilon_g > 0$, $\varepsilon_h > 0$, $\varepsilon_a > 0$, $\varepsilon_\mu \in (0,1)$ and $\nu \in [10^{-4}, 1]$.
2: Initial Values: z^0, $\rho^0 > 0$, μ^0, $k = 1$, $\rho^0_{\max}$.
3: **Normal Step.** Obtain z_c^k, λ^k, ρ^k, μ^k and $\zeta^k = g(z_c^k, \mu^k) + A(z_c^k)^T\lambda^k$ such that

$$\left\|h(z_c^k)\right\| \le \left\|h(z^{k-1})\right\|, \quad \left\|h(z_c^k)\right\| \le \rho^k \le \nu \frac{\|\zeta^k\|\,\rho^{k-1}_{\max}}{\|g(z_c^k,\mu^k)\| + 1}, \quad \text{and} \quad s_c^k \ge \varepsilon_\mu s^{k-1}$$

4: **if** $\left\|h(z_c^k)\right\| < \varepsilon_h$, $\|\zeta^k\| < \varepsilon_g$ and $\left|(s_c^k)^T\lambda_I^k\right| < \varepsilon_a$, **STOP** with $z^* = z_c^k$.
5: **Update** $\rho^k_{\max}$.
6: **Tangential Step.** Obtain z_k that sufficiently reduces the Lagrangian function and satisfies

$$\left\|h(z^k)\right\| \le 2\rho^k \qquad \text{and} \qquad s^k \ge \varepsilon_\mu s_c^k.$$

7: Increment k and return to Step 3.

To compute the normal step, we apply a method to solve the problem

$$\min_{s \ge 0} \quad \frac{1}{2}\|h(z)\|^2. \tag{2.8}$$

The conditions on what the method must satisfy are generally satisfied by the usual descent methods, and they'll be described on the convergence hypothesis. In our practical algorithm, each application of the method starts with the previous z_c and a new iterate z_c^+ is obtained solving the subproblem

$$\min \frac{1}{2}\|h(z_c + d)\|^2 \quad \text{s.t} \quad \|d\|_\infty \le \Delta_N, \; s_c + d_s \ge \varepsilon_\mu s^{k-1} \tag{2.9}$$

where $\Delta_N > 0$ is the trust region radius for the normal step.

Step 3 of the algorithm also comprises the update of the trust region cylinder ρ, the penalty parameter μ and the projected gradient ζ. After solving the normal subproblem to find z_c^+, we update the multipliers, the projected gradient and the radius ρ, and verify if z_c^+ remains inside the new cylinder. If it is not the case, the whole process is repeated. Algorithm 2 outlines the normal step.

Algorithm 2 Normal Step

1: Parameters: $\alpha_\rho > 0$, $\alpha_h > 0$ and $\varepsilon_h > 0$.
2: Initial Value: $z_c = z^{k-1}$.
3: Compute λ, $\zeta = g(z_c, \mu) + A(z_c)^T\lambda$ and update ρ and μ.
4: **while** $\|h(z_c)\| > \max\{\rho, \varepsilon_h\}$ **do**
5: Find z_c such that $\|h(z_c)\| \leq \rho$ and $s_c \geq \varepsilon_\mu s^{k-1}$ from (2.9).
6: If that is not possible, stop with *possible infeasibility* flag.
7: Compute λ, $\zeta = g(z_c, \mu) + A(z_c)^T\lambda$, ρ and μ.
8: **end while**
9: Define $\lambda^k = \lambda$, $\zeta^k = \zeta$ and $\rho^k = \rho$.
10: Define $\mu^k = \min\left\{\mu^{k-1}, \alpha_\rho\rho_k, \alpha_\rho\rho_k^2, \dfrac{(s_c^k)^T\max\{0, -\lambda_I^k\}}{m_I}, \alpha_h h(z^k)\right\}$.

The main difficulty in the normal step is obtaining a new z_c in Step 5 of Algorithm 2, solving approximately (2.9). To solve this subproblem, we use a modification of the Dogleg method proposed in [24]. In general terms, we take steps that are combinations of the Gauss-Newton step

$$d_{GN} = -J^T(JJ^T)^{-1}h(z_c)$$

and the Cauchy step $d_C = -\alpha_C J^T h(z_c)$, with $J = \nabla h(z_c)$. Although it may seem that the loop on Step 4 is very expensive, since it may require several z_c updates, we will show on Subsection 5.1 that, in practice, less than one loop is usually required per iteration, making the normal step no more expensive than an inexact restoration method.

In this algorithm, we compute λ so that it remains asymptotically close to $\lambda_{LS}(z_c, \mu)$, ensuring that the sign of the multipliers associated with the inequalities are correct when μ tends to 0. This is done using the formula

$$\lambda_i = \begin{cases} \lambda_{LS_i}(z_c, \mu), & \text{if } i \in E, \\ \min\{\lambda_{LS_i}(z_c, \mu), \alpha\mu^r\}, & \text{if } i \in I, \end{cases}$$

where $\alpha > 0$ and $r > 0$ are chosen appropriately. After that, we compute ζ and update ρ according to Algorithm 3. The update of ρ is partly empirical, including some safeguards for very small values, but it also include some theorical properties required for convergence, which are discussed in the next sections.

In the tangential step, the dual infeasibility is reduced through the minimization of a quadratic approximation of the Lagrangian, subject to a scaled

Algorithm 3 ρ Update

1: Given z_c, μ, $\rho_{\max}$ and ρ
2: Compute $n_p = \|\zeta\| / (\|g(z_c, \mu)\| + 1)$.
3: **if** $\rho > 2\rho_{\max} n_p$ **then**
4: Choose $\rho^+ \leq \rho_{\max} n_p$
5: **else**
6: Choose $\rho^+ = \max\{\rho, \min\{\rho_{\max} n_p, \max\{10^{-4}\rho_{\max} n_p, 0.75\rho_{\max}\}\}\}$.
7: **end if**

trust region. The step δ_t is an approximate solution to the problem

$$\begin{array}{ll} \min\limits_{\delta} & q_k(\delta) = \frac{1}{2}\delta^T B^k \delta + \delta^T \zeta^k \\ \text{s.t.} & A(z_c^k)\delta = 0, \quad \left\|A(z_c^k)\delta\right\|_\infty \leq \Delta_T, \quad S_c^k \delta_s \geq (\varepsilon_\mu - 1)s_c^k, \end{array} \tag{2.10}$$

where $B^k = \text{diag}(B_x^k, B_s^k)$ is a block diagonal approximation to $W(z_c^k, \lambda^k, \mu^k)$, and $\Delta_T > 0$ is the trust region radius for the step. This subproblem is solved with a modification of the Steihaug method [42] to handle the bounds on a similar way of what it does to the trust region.

The step δ_t is required to be at least as good as a Cauchy step, and to remain inside the cylinder $\mathcal{C}(2\rho^k)$. If the tangent step significantly undermines the reduction of the infeasibility obtained in the normal step, we also make a second order correction, as suggested in [10]. This is verified by the conditions

$$\|h(z_c + \delta_t)\| > \min\{2\rho, 2\|h(z_c)\| + 0.5\rho\}$$

or

$$\|h(z_c)\| \leq 10^{-5} \quad \text{and} \quad \|h(z_c + \delta_t)\| > \max\{10^{-5}, 2\|h(z_c)\|\}.$$

Following an idea given in [34], the correction direction d^+ is given by

$$\begin{aligned} d^+ &= \arg\min_{\delta} \frac{1}{2}\left\|A(z_c^k)\delta + h(z_c^k + \delta_t) - h(z_c^k)\right\| \\ &= -A(z_c^k)^T[A(z_c^k)A(z_c^k)^T]^{-1}[h(z_c^k + \delta_t) - h(z_c^k)] \end{aligned}$$

The correction is defined as $\delta_{soc} = \alpha d^+$, where $\alpha \in (0, 1)$ is chosen so that $\delta_t + \delta_{soc}$ satisfy the same fraction-to-the-boundary condition as δ in (2.10). Algorithm 4 shows an outline of the tangential step.

To close this section, let us show how to update ρ_{max} in step 5 of Algorithm 1. This parameter is used to control the size of the cylinder radius at each iteration, and depends not only on the norm of the projected gradient, but also on the variation of the Lagrangian, that is given by

$$\Delta L_c^k = L(z_c^k, \lambda^k, \mu^k) - L(z_c^{k-1}, \lambda^{k-1}, \mu^{k-1}) = \Delta L_T^{k-1} + \Delta L_N^k,$$

where the contributions of the normal and tangential steps to the Lagrangian variation are, respectively,

$$\begin{aligned} \Delta L_N^k &= L(z_c^k, \lambda^k, \mu^k) - L(z^{k-1}, \lambda^{k-1}, \mu^{k-1}), \\ \Delta L_T^k &= L(z^k, \lambda^k, \mu^k) - L(z_c^k, \lambda^k, \mu^k). \end{aligned}$$

The update of $\rho_{\max}$ is described in Algorithm 5.

Algorithm 4 Tangential Step

Parameters: $\eta_1 \in (0, \frac{1}{2}]$, $\eta_2 > \eta_1$, $\alpha_R \in (0, \frac{3}{4}]$, $\alpha_I > 1$.
Initial Values: Δ_T, $r = 0$ and $z^+ = z_c^k$.
while $\left\|h(z^+)\right\| > 2\rho^k$ or $r < \eta_1$ **do**
  Compute the Cauchy Step $\delta_{CP} = -\alpha_{CP}\zeta^k$, where α_{CP} is the solution of

$$\min_{\alpha > 0} \; q_k(-\alpha\zeta^k) \qquad \text{s.t.} \qquad \left\|\alpha \Lambda(z_c^k)\zeta^k\right\|_\infty \leq \Delta_T, \quad -\alpha S_c^k(\zeta^k)_s \geq (\varepsilon_\mu - 1)s_c^k$$

  Starting at δ_{CP}, compute δ_t such that

$$q_k(\delta_t) \leq q_k(\delta_{CP}), \qquad \left\|\Lambda(z_c^k)\delta_t\right\|_\infty \leq \Delta_T \quad S_c^k\delta_s^k \geq (\varepsilon_\mu - 1)s_c^k$$

  If necessary, compute a second order correction δ_{soc}.
  $d^+ = \Lambda(z_c^k)(\delta_t + \delta_{soc})$.
  $z^+ = z_c^k + d^+$.
  $\Delta L_T^k = L(z^+, \lambda^k, \mu^k) - L(z_c^k, \lambda^k, \mu^k)$.
  $r = \Delta L_T^k / q_k(\delta_t)$.
  if $\left\|h(z^+)\right\| > 2\rho^k$ or $r < \eta_1$ **then**
    $\Delta_T = \alpha_R \Delta_T$,
  else if $r > \eta_2$ **then**
    $\Delta_T = \alpha_I \Delta_T$.
  end if
end while
Define $z^k = z^+$.

Algorithm 5 $\rho_{\max}$ Update

Initial Value: L_{ref} defined in the previous iteration. For $k = 0$, we set $L_{ref} = \infty$.
$\Delta L_N^k = L(z_c^k, \lambda^k, \mu^k) - L(z^{k-1}, \lambda^{k-1}, \mu^{k-1})$
if $\Delta L_N^k \geq \frac{1}{2}[L_{ref} - L(z^{k-1}, \lambda^{k-1}, \mu^{k-1})]$ **then**
  $\rho_{\max}^k = \rho_{\max}^{k-1}/2$
else
  $\rho_{\max}^k = \rho_{\max}^{k-1}$
end if
if $\Delta L_N^k > -\frac{1}{2}\Delta L_T^{k-1}$ **then**
  $L_{ref} = L(z_c^k, \lambda^k, \mu^k)$
end if

3 Global Convergence

The convergence theory of the method is based on relatively weak assumptions. The four hypotheses that are needed to show that Algorithm 1 is globally convergent are described below.

Hypothesis H1 *The functions f, c_E and c_I are C^2.*

Hypothesis H2 *The sequences $\{z_c^k\}$ and $\{z^k\}$, the approximations B^k and the multipliers $\{\lambda^k\}$ remains uniformly bounded.*

Hypothesis H3 *The restoration is always possible and is reasonably short, i.e., given ρ and y, it is always possible to find y_c such that $\|h(y_c)\| \leq \rho$ and*

$$\|y_c - y\| = \mathcal{O}(\|h(y)\|). \tag{3.1}$$

Hypothesis H4 $\|\delta_{soc}^k\| = \mathcal{O}(\|\delta_t^k\|^2)$.

Hyphotesis H1 is natural, since the method uses exact or approximate second derivatives. H2 is required to avoid the possibility of taking unbounded descent directions. H3 is important because the restoration can fail, although we show in Section 3.1 that, when it happens, at least a stationary point of the infeasibility is found. Finally, H4 is a traditional hypothesis for second order correction steps.

If H1 holds, then the remaining hypotheses will also hold if, for example, the level set $\{z \mid \|h(z)\| \le \|h(z_0)\|\}$ is compact, the feasible set is not empty and $\nabla h(z)$ has full rank on this set.

We present now some properties of the algorithm, starting with those related to the trust cylinder. From the definition of the cylinder radius on step 3 of Algorithm 1, the conditions established on steps 3 and 6 of the same algorithm, and the way we update $\rho_{\max}$ on Algorithm 5, we have

$$\begin{aligned} \rho^k &\le \rho_{\max}^{k-1}\|\zeta^k\| \le 2\rho_{\max}^k\|\zeta^k\|, && (3.2)\\ \rho_{\max}^k &\le \rho_{\max}^{k-1} \le 10^4\rho^k\frac{\|g(z_c^k,\mu^k)\|+1}{\|\zeta^k\|}, && (3.3)\\ \|h(z_c^k)\| &\le \|h(z^{k-1})\| \le 2\rho^{k-1}. && (3.4)\end{aligned}$$

Besides, we require the iterates to follow the fraction-to-the-boundary rule given in (2.7), which means that

$$s_c^k \ge \varepsilon_\mu s^{k-1}, \qquad s^k \ge \varepsilon_\mu s_c^k, \qquad s^+ \ge \varepsilon_\mu s_c^k. \tag{3.5}$$

Moreover, from the definition of μ^k in step 10 of Algorithm 2, we also have

$$\mu^k \le \alpha_\rho \min\{\rho^k, (\rho^k)^2\}. \tag{3.6}$$

From this point forward, we assume that the sequences $\{z_c^k\}$ and $\{z^k\}$ generated by the algorithm satisfy H1-H4. To ease the reading, we also define the matrices $\Lambda_c^k = \Lambda(z_c^k)$, $\Lambda^k = \Lambda(z^k)$ and $\Lambda^+ = \Lambda(z^+)$, and denote the full normal step by $\delta_N^k = z_c^k - z^{k-1}$. From the sequence of normal steps in loop 4 in Algorithm 2, and Hypothesis H3, we have

$$\|z_c^{k+1} - z^k\| = \mathcal{O}(\|h(z^k)\|). \tag{3.7}$$

Besides, Hypotheses H1-H4 allow us to choose a constant $\delta_{\max} > 0$, such that, for every iteration k,

$$\|\delta_t^k\| + \|\delta_{soc}^k\| + \|\delta_N^k\| \le \delta_{\max}. \tag{3.8}$$

The hypotheses and (3.7) also allow us to define $\xi_0 > 0$, such that, for all k, if $\left\|z - z_c^k\right\| \leq \delta_{\max}$ and $\mu \leq \mu_0$, then

$$\|A_j(z)\| \leq \xi_0, \qquad j = 1, \ldots, m, \tag{3.9}$$
$$\left\|\nabla^2 h_j(z)\right\| \leq \xi_0, \qquad j = 1, \ldots, m, \tag{3.10}$$
$$\|\nabla f(x)\| \leq \xi_0, \tag{3.11}$$
$$\left\|\nabla^2 f(x)\right\| \leq \xi_0, \tag{3.12}$$
$$\|g(z, \mu)\| \leq \xi_0, \tag{3.13}$$
$$\|\Gamma(z, \mu)\| \leq \xi_0, \tag{3.14}$$
$$\left\|B^k\right\| \leq \xi_0, \tag{3.15}$$
$$\left\|\lambda^k\right\| \leq \xi_0, \tag{3.16}$$
$$\left\|\delta_{soc}^k\right\| \leq \xi_0 \left\|\delta_t^k\right\|^2, \tag{3.17}$$
$$\left\|\Lambda_c^k\right\| \leq \xi_0, \tag{3.18}$$
$$\left\|z_c^{k+1} - z^k\right\| \leq \xi_0 \left\|h(z^k)\right\|, \tag{3.19}$$
$$s_c^k \leq \xi_0 s^{k-1}, \tag{3.20}$$
$$s^k \leq \xi_0 s_c^k. \tag{3.21}$$

Our main global convergence result is given by Theorem 1, which is based on six lemmas. The first one shows that the normal step is well-defined.

Lemma 1 *Under Hypothesis H1-H3, the loop on step 4 of Algorithm 2 terminates finitely.*

Proof Denote the point and the radius at the beginning of the loop as z_c and ρ, and at the end as z_c^+ and ρ^+. Notice that $\|h(z_c^+)\| \leq \rho < \|h(z_c)\|$. Now, if the condition on step 3 of Algorithm 3 is not satisfied, then, from step 6 of the same algorithm, we have $\rho^+ \geq \rho \geq \|h(z_c^+)\|$, and z_c^+ is accepted. Otherwise, if step 3 is never satisfied, then we have $\rho^+ < \rho/2$. Hence ρ converges to zero, and eventually we'll have $\rho < \varepsilon_h$, thus $\|h(z_c^+)\| < \rho < \varepsilon_h$. □

The next lemma provides a bound on the increase of the infeasibility caused by the tangential step in relation to the intermediate point z_c^k, generated by the normal step.

Lemma 2 *The trial iterate z^+ generated on step 8 of Algorithm 4 satisfies $\left\|h(z^+) - h(z_c^k)\right\| \leq \overline{\xi}_0 \left\|\delta_t\right\|^2$, where $\overline{\xi}_0$ is a positive constant.*

Proof See Lemma 3.1 of [5]. □

The following lemma shows that, under assumptions H1-H4, the tangential step does not fail, and provides a sufficient reduction of the Lagrangian function.

Lemma 3 *If x_c^k is not a stationary point to problem (2.1), then z^+ is eventually accepted. Furthermore, we can define constants ξ_1, ξ_2 and ξ_3 such that, for every k,*

$$\Delta L_T^k \leq -\xi_1 \left\|\zeta^k\right\| \min\{\xi_2 \left\|\zeta^k\right\|, \xi_3 \sqrt{\rho^k}, 1 - \varepsilon_\mu\}. \tag{3.22}$$

Proof The proof of this Lemma is essentially the same as the proof of Lemma 3.2 in [5]. The noteworthy difference is that the Cauchy step δ_{CP} defined in step 4 of Algorithm 4 now satisfies

$$\|\delta_{CP}\| \geq \min\left\{\frac{\|\zeta\|}{\|B\|}, \Delta, 1-\varepsilon_\mu\right\},$$

while only the first two terms inside the braces were considered in [5]. □

The next lemma establishes an upper bound to the normal variation of the Lagrangian function. Note that this variation can be positive.

Lemma 4 *There is a positive constant ξ_4 such that, for k sufficiently large,*

$$\Delta L_N^{k+1} \leq \xi_4 \rho_{\max}^k \left\|\zeta^k\right\|.$$

Proof Using the Mean Value Theorem, we have

$$\begin{aligned}
&\Delta L_N^{k+1} = \\
&= \varphi(z_c^{k+1},\mu^k) - \varphi(z^k,\mu^k) + h(z_c^{k+1})^T\lambda^{k+1} - h(z^k)^T\lambda^k + (\mu^{k+1}-\mu^k)\beta(z_c^{k+1}) \\
&= \nabla\varphi(z_\xi,\mu^k)^T(z_c^{k+1}-z^k) + h(z_c^{k+1})^T\lambda^{k+1} - h(z^k)^T\lambda^k + (\mu^{k+1}-\mu^k)\beta(z_c^{k+1}) \\
&= \nabla f(x_\xi)^T(x_c^{k+1}-x^k) + \mu^k\nabla\beta(z_\xi)^T(z_c^{k+1}-z^k) + h(z_c^{k+1})^T\lambda^{k+1} - h(z^k)^T\lambda^k \\
&\quad + (\mu^{k+1}-\mu^k)\beta(z_c^{k+1}),
\end{aligned}$$

with $z_\xi = \eta z^k + (1-\eta)z_c^{k+1}$, for some $\eta \in [0,1]$. Under hypothesis H2, there is a constant $M > 0$ such that $s_i^k, s_{c_i}^k \leq M$. Using this, (3.11), (3.5), (3.16), (3.19), (3.4), (3.6) and (3.2), we obtain

$$\begin{aligned}
\Delta L_N^{k+1} &\leq \xi_0 \left\|x_c^{k+1} - x^k\right\| + \mu^k \sum_{i=1}^{m_I} \frac{s_i^k - s_{c_i}^{k+1}}{\eta s_i^k + (1-\eta)s_{c_i}^{k+1}} + \xi_0 \left\|h(z_c^{k+1})\right\| \\
&\quad + \xi_0 \left\|h(z^k)\right\| + (\mu^k - \mu^{k+1})m_I M \\
&\leq \xi_0^2 \left\|h(z^k)\right\| + \xi_0 \left\|h(z^k)\right\| + \xi_0 \left\|h(z^k)\right\| + \mu^k m_I \left(\frac{1-\varepsilon_\mu}{\eta + (1-\eta)\varepsilon_\mu} + M\right) \\
&\leq (\xi_0^2 + 2\xi_0)2\rho^k + \rho^k m_I \left(\frac{1-\varepsilon_\mu}{\eta + (1-\eta)\varepsilon_\mu} + M\right) \leq \xi_4 \rho_{\max}^k \left\|\zeta^k\right\|,
\end{aligned}$$

where $\xi_4 = 4(\xi_0^2 + 2\xi_0) + 2m_I \dfrac{1-\varepsilon_\mu}{\eta + (1-\eta)\varepsilon_\mu} + 2M.$ □

Lemma 5 shows that, between iterations where $\rho_{\max}$ does not change, the Lagrangian function value decreases proportionally to the Lagrangian variation in the tangential steps.

Lemma 5 *If $\rho_{\max}^{k+1} = \rho_{\max}^{k+2} = \cdots = \rho_{\max}^{k+j}$, for all $j \geq 1$, then*

$$L_c^{k+j} - L_c^k = \sum_{i=k+1}^{k+j} \Delta L_c^i \leq \frac{1}{4} \sum_{i=k}^{k+j-1} \Delta L_T^i + r^k,$$

where $r^k = \frac{1}{2}[L_{ref}^k - L_c^k]$.

Proof See Lemma 3.4 of [5]. □

Finally, the next lemma ensures the existence of sufficient normal space in the trust cylinders, so that the Lagrangian function can be decreased.

The lemma is based on the fact that for large enough k, inequality (3.22) ensures that $\left|\Delta L_H^k\right|$ is greater than a fraction of $\sqrt{\rho^k}$. Besides, in the proof of Lemma 4, we saw that $\left\|\Delta L_N^k\right\| = \mathcal{O}(\rho^k)$, which means that, for a very large k, the restoration will not destroy the decrease of the Lagrangian function, preventing the algorithm from making excessive updates of $\rho_{\max}$.

Lemma 6 *If DCI generates an infinite sequence* $\{z^k\}$*, then*

(i) There are positive constants ξ_5 *and* ξ_6 *such that, if*

$$\rho_{\max}^k < \min\{\xi_5 \left\|\zeta^k\right\|, \xi_6\}, \tag{3.23}$$

then $\rho_{\max}$ *doesn't change on iteration* $k+1$.

(ii) Furthermore, if $\liminf \left\|\zeta^k\right\| > 0$, *then there is* $k_0 > 0$ *such that, for all* $k \geq k_0$,

$$\rho_{\max}^k = \rho_{\max}^{k_0}. \tag{3.24}$$

(iii) If the tangential step and the vector of multipliers satisfy

$$\left\|z^k - z_c^k\right\| = \mathcal{O}(\left\|\zeta^k\right\|), \tag{3.25}$$

$$\left\|\lambda^{k+1} - \lambda^k\right\| = \mathcal{O}(\left\|\zeta^k\right\|), \tag{3.26}$$

$$(\lambda_I^{k+1})^T (s_c^{k+1} - s^k) = \mathcal{O}(\left\|\zeta^k\right\| \rho^k), \tag{3.27}$$

then (3.24) *is satisfied, regardless of the value of* $\liminf \left\|\zeta^k\right\|$. *In other words,* $\rho_{\max}^k$ *remains sufficiently bounded away from zero.*

Proof The proof of this Lemma is similar to the proof of Lemma 3.5 in [5]. The differences are that the constant ξ_6 must satisfy $\xi_6 \leq \frac{1}{2}\xi_1(1-\varepsilon_\mu)/\xi_4$, (to compensate the differences between Lemma 3 and Lemma 3.2 in [5]), that (3.26) was modified to use only the approximation to the multipliers instead of the function λ_{LS}, and that the proof of the third part requires the additional assumption (3.27), due to the inclusion of the barrier function. □

We now present the global convergence theorem of our algorithm.

Theorem 1 *Under Hypothesis H1-H4, if method DCI generates an infinite sequence, then there is a subsequence that converges to a stationary point of* (2.1). *If conditions* (3.25), (3.26) *and* (3.27) *are also satisfied, then every convergent subsequence of* $\{x_c^k\}$ *has a limit point that is stationary for* (2.1).

Proof The proof of this Theorem is essentially the same as the one given for Theorem 3.6 of [5], using the corresponding lemmas when necessary. However, the main theorem of [5] says only that every accumulation point of $\{z_c^k\}$ is stationary, so we need a few extra steps to prove that every convergent subsequence of $\{x_c^k\}$ has a limit point that is stationary for (2.1).

Thus, let $\{k_l\}$ be a subsequence of indexes such that $\left\|\zeta^{k_l}\right\| \longrightarrow 0$. From properties (3.2)-(3.4) and (3.6), we have that $\rho^{k_l} \longrightarrow 0$, $\left\|h(z_c^{k_l})\right\| \longrightarrow 0$ and $\mu^{k_l} \longrightarrow 0$. Now, from the definition of ζ^{k_l} we conclude that $-\mu^{k_l} e - S_c^{k_l} \lambda_I^{k_l} \longrightarrow 0$, so that $S_c^{k_l} \lambda_I^{k_l} \longrightarrow 0$, and $\nabla f(x_c^{k_l}) + \nabla c(x_c^{k_l})^T \lambda^{k_l} \longrightarrow 0$. Furthermore, $\lambda_I^{k_l} \leq \alpha(\mu^{k_l})^r$, so that $\lim \lambda_I^{k_l} \leq 0$. Therefore, the limit point of this subsequence is stationary. □

3.1 Convergence to infeasible stationary points

If problem (2.1) is infeasible, our method will fail to find a point inside the trust cylinder. In this case, the algorithm stops declaring that a infeasible stationary point was found.

In this section, we show that, if the normal step can't find z_c^k such that

$$\left\|h(z_c^k)\right\| \leq \rho^k,$$

then at least it converges to a stationary point of (2.1). In order to obtain this result, we write the infeasibility measure at x as

$$\min \ \left\|c_E(x)\right\|^2 + \left\|c_I^-(x)\right\|^2, \tag{3.28}$$

where $v^- = (\min\{0, v_1\}, \min\{0, v_2\}, \ldots, \min\{0, v_n\})^T$.

Theorem 2 *$z^* = [x^*, s^*]^T$ is a stationary point for the normal problem* (2.8) *if, and only if, x^* is a stationary point for* (3.28)*.*

Proof We simply need to show that $\nabla c_I(x^*)^T[c_I(x^*) - s^* - c_I^-(x^*)] = 0$, and this is obtained through simple algebraic manipulations. □

We assume here that the normal algorithm converges to a stationary point for (2.8). The conditions for this to happen depend on the method chosen, but are usually mild. For instance, using the method described by Francisco et al. [24], this result can be achieved under the following hypotheses.

Hypothesis H5 *(H1 from [24]) The sequence $\{z_N^j\}$ generated by the normal algorithm is bounded. (This can be obtained, for example, if the level set $\{z \mid \|h(z)\| \leq \|h(z_N^0)\|\}$ is bounded.)*

Hypothesis H6 *(H2 from [24]) For every z, w in a convex, open and bounded set L containing the whole sequence generated by the algorithm and every point whose value is computed during the algorithm, we have*

$$\|\nabla h(z) - \nabla h(w)\| \leq 2\gamma_0 \|z - w\|.$$

4 Local Convergence

We now analyze the local behavior of the method assuming that a convergent sequence to a local minimizer exists.

Let $\{z^k\}$ and $\{z_c^k\}$ be sequences that converge to z^*, and assume that $\{\lambda^k\}$ converges to $\lambda^* = \lambda_{LS}(z^*, 0)$. In this case, we have

$$\begin{cases} \nabla f(x^*) + \nabla c(x^*)^T \lambda^* = 0, & \\ c_E(x^*) = 0, & c_I(x^*) \geq 0, \\ c_I(x^*)^T \lambda_I^* = 0, & \lambda_I^* \leq 0. \end{cases}$$

Let $\mathcal{A}(x) = \{i \in E \cup I : c_i(x) = 0\}$, $\mathcal{A}^* = \mathcal{A}(x^*)$ and $\nabla c_A(x) = [c_i(x)]_{i \in \mathcal{A}(x)}^T$. We restrict our attention to the case where x^* is a "good minimizer" for (2.1), in the sense that the rows of $\nabla c_A(x^*)$ are linearly independent, and

$$y^T \left[\nabla^2 f(x^*) + \sum_{i \in \mathcal{A}^*} \nabla^2 c_i(x^*) \lambda_i^* \right] y \geq \theta_1 \|y\|^2 , \tag{4.1}$$

with $\theta_1 > 0$ and $y \in T = \{w : w^T \nabla c_i(x^*) = 0 : i \in E \cup J\}$, where $J = \{i \in I : \lambda_i^* < 0\}$.

Suppose that λ_A^k and λ_A^* are the vectors that contain, respectively, the components of λ^k and λ^* corresponding to the active constraints. Since $\nabla c_A(x)$ is continuous, it has full row rank in a neighborhood of x^*, so we can define

$$\lambda_A(x) = -[\nabla c_A(x) \nabla c_A(x)^T]^{-1} \nabla c_A(x) \nabla f(x),$$
$$g_A(x) = \nabla f(x) + \nabla c_A(x)^T \lambda_A(x),$$

and

$$H_A(x, \lambda) = \nabla^2 f(x) + \sum_{i \in \mathcal{A}^*} \nabla^2 c_i(x) \lambda_i.$$

In a neighborhood V^* of x^*, we can define the orthogonal projector in the null space of $\nabla c_A(x)$ as

$$P(x) = I - \nabla c_A(x)^T [\nabla c_A(x) \nabla c_A(x)^T]^{-1} \nabla c_A(x),$$

which is Lipschitz continuous, since $c \in C^2$.

In addition to H1-H4, our local convergence analysis requires the following hypotheses.

Hypothesis H7

$$\left\|\lambda^{k+1} - \lambda^k\right\| = \mathcal{O}(\left\|\zeta^k\right\|),$$
$$(\lambda_I^{k+1})^T (s_c^{k+1} - s^k) = \mathcal{O}(\left\|\zeta^k\right\| \rho^k).$$

Hypothesis H8 *B_x^k is asymptotically uniformly positive definite in $\mathcal{N}(\nabla c_A(x_c^k))$, that is, we can redefine the constant θ_1 of (4.1) in a way that, for k sufficiently large,*

$$\theta_1 \|y\|^2 \leq y^T B_x^k y \leq \xi_0 \|y\|^2 , \tag{4.2}$$

for $y \in \mathcal{N}(\nabla c_A(x_c^k))$.

Hypothesis H9 *Let Z_A^k be a matrix whose columns are a orthonormal basis for the null space of $\nabla c_A(x_c^k)$. Define*

$$\delta_x^k = -Z_A^k[(Z_A^k)^T B_x^k Z_A^k]^{-1}(Z_A^k)^T g_A(x_c^k), \tag{4.3}$$

$$\delta_{s_i}^k = \frac{1}{s_{c_i}^k}\nabla c_{I_i}(x_c^k)^T\delta_x^k, \tag{4.4}$$

and $\delta_A^k = [(\delta_x^k)^T \ (\delta_s^k)^T]^T$. We assume that δ_A^k is the first step tried by the algorithm if $\|\delta_A^k\| \le \Delta$ and $s_c^k + S_c^k\delta_s^k \ge \varepsilon_\mu s_c^k$. In addition, we assume that

$$P(x_c^k)[B_x^k - H_A(x^*,\lambda^*)]\delta_x^k = o(\|\delta_x^k\|), \tag{4.5}$$

and that

$$(\delta_s^k)^T[B_s^k - \mu I]\delta_s^k = o(\|\delta_s^k\|^2). \tag{4.6}$$

Note that, if $s_{c_i}^k \longrightarrow 0$, then $i \in \mathcal{A}^*$, i.e. $\nabla c_{I_i}(x_c^k)^T$ is one of the rows of $\nabla c_A(x_c^k)$, which implies that $\nabla c_{I_i}(x_c^k)^T Z_A^k = 0$, so $\delta_{s_i}^k = 0$. Otherwise, $s_{c_i}^k$ remains bounded away from zero, which means that there is a constant $s_{\min} > 0$ such that $s_{c_i}^k \ge s_{\min}$ for $i \notin \mathcal{A}^*$.

Hypothesis H10 *There are positive constants θ_A and θ_p, such that, for k sufficiently large,*

$$\frac{1}{\theta_p}\|\zeta^k\| \le \|g_A(x_c^k)\| \le \theta_A\|\zeta^k\|,$$

$$\|c_A(x_c^k)\| = \Theta(\|h(z_c^k)\|), \qquad \|c_A(x^k)\| = \Theta(\|h(z^k)\|),$$

$$\|x_c^{k+1} - x^k\| = \mathcal{O}(\|c_A(x^k)\|).$$

Since $\nabla c_A(x)$ and $H_A(x,\lambda)$ are continuous and $\nabla c_A(x^*)$ has full rank, our hypotheses imply that there exist $\theta_3 > 0$ and a neighborhood V^* of x^* such that, for x, $x_c^k \in V^*$,

$$\|\nabla h(z)^T\lambda\| \ge \theta_3\|\lambda\|, \qquad \lambda \in \mathbb{R}^{n+m_I}, \ s \in \mathbb{R}_+^{m_I}. \tag{4.7}$$

Besides, using (4.5), the fact that $H_A(x^*,\lambda^*) = W_x(x^*,\lambda^*)$ and the continuity of W_x, we obtain

$$P(x_c^k)[B_x^k - W_x(x_c^k,\lambda^k)]\delta_x^k = o(\|\delta_x^k\|). \tag{4.8}$$

Now, dividing $q^k(\delta) = q_x^k(\delta_x) + q_s^k(\delta_s)$, where

$$q_x^k(\delta_x) = \frac{1}{2}\delta_x^T B_x^k\delta_x + \delta_x^T\nabla f(x_c^k)$$

and

$$q_s^k(\delta_s) = \frac{1}{2}\delta_s^T B_s^k\delta_s + \delta_s^T(-\mu e),$$

we have that $\delta_x^k = Z_A^k\nu^k \in \mathcal{N}(\nabla c_A(x_c^k))$, where ν^k is the minimizer of

$$\min \ \overline{q}_x^k(\nu) = q_x^k(Z_A^k\nu) = \frac{1}{2}\nu^T(Z_A^k)^T B_x^k Z_A^k\nu + \nu^T(Z_A^k)^T\nabla f(x_c^k).$$

Therefore, since $(Z_A^k)^T \nabla f(x_c^k) = (Z_A^k)^T g_A(x_c^k)$, we have

$$(Z_A^k)^T [B_x^k Z_A^k \nu^k + g_A(x_c^k)] = 0. \tag{4.9}$$

Moreover, from (4.2) and the fact that $(Z_A^k)^T Z_A^k = I$, matrix $[(Z_A^k)^T B_x^k Z_A^k]^{-1}$ satisfies, in the neighbourhood V^*, for every $u \in \mathbb{R}^{\overline{n}-m_A}$,

$$\frac{1}{\gamma_2} \|u\|^2 \leq u^T [(Z_A^k)^T B_x^k Z_A^k]^{-1} u \leq \frac{1}{\gamma_1} \|u\|^2 . \tag{4.10}$$

In the next lemma, we show that δ_A^k is eventually accepted by the algorithm.

Lemma 7 *For k sufficiently large, the step δ_A^k is accepted by Algorithm 1.*

Proof Since $g_A(x_c^k) \in \mathcal{N}(\nabla c_A(x_c^k))$, there is ν_p^k such that $g_A(x_c^k) = Z_A^k \nu_p^k$ with $\left\|g_A(x_c^k)\right\| = \left\|\nu_p^k\right\|$. Combining (4.3) and (4.10) we have

$$\left\|\delta_x^k\right\| \leq \frac{1}{\gamma_1} \left\|g_A(x_c^k)\right\| . \tag{4.11}$$

Furthermore, for $i \notin \mathcal{A}^*$,

$$\left|\delta_{A_i}^k\right| = \left|\frac{\nabla c_i(x_c^k)^T \delta_x^k}{s_i}\right| \leq \frac{\xi_0}{s_{\min}} \left\|\delta_x^k\right\| \leq \frac{\xi_0}{\gamma_1 s_{\min}} \left\|g_A(x_c^k)\right\| ,$$

so

$$\left\|\delta_s^k\right\| \leq \frac{m_A \xi_0}{\gamma_1 s_{\min}} \left\|g_A(x_c^k)\right\| . \tag{4.12}$$

Therefore, for k sufficiently large,

$$\left\|\delta_A^k\right\| \leq \theta_4 \left\|g_A(x_c^k)\right\| , \tag{4.13}$$

where $\theta_4 = \sqrt{1 + (m_A \xi_0 / s_{\min})^2} / \gamma_1$. Since $g_A(x_c^k) \longrightarrow 0$, we have $\left\|\delta_A^k\right\| < \Delta_{\min}$ and $\delta_s^k \geq (\varepsilon_\mu - 1)e$ for k sufficiently large. Thus, from Hypothesis H9, δ_A^k will be the first step tried by the algorithm, in some neighbourhood V^*.

For k sufficiently large, $\overline{q}_x(\nu)$ is a positive definite quadratic function. Therefore, it's minimum, $\overline{q}_x(\nu^k)$, satisfies

$$q_x(\delta_x^k) \leq -\frac{1}{2\gamma_2} \left\|g_A(x_c^k)\right\|^2 , \tag{4.14}$$

where the last inequality comes from (4.10). Besides, using (4.4), (3.15), (3.9), (3.6), (3.2), (4.11) and Hypothesis H10, we have

$$\begin{aligned}
q_s(\delta_s^k) &= \frac{1}{2}\delta_s^T B_s \delta_s + \delta_s^T(-\mu e) \le \frac{1}{2}\|B_s\| \sum_{i\notin\mathcal{A}^*} \delta_{s_i}^2 - \mu \sum_{i\notin\mathcal{A}^*} \delta_{s_i} \\
&= \frac{1}{2}\|B_s\| \sum_{i\notin\mathcal{A}^*} \frac{1}{s_i^2}[\nabla c_i(x_c^k)^T\delta_x]^2 - \mu \sum_{i\notin\mathcal{A}^*} \frac{\nabla c_i(x_c^k)^T\delta_x}{s_i} \\
&\le \frac{1}{2} m \frac{\xi_0^3}{s_{\min}^2}\|\delta_x\|^2 + \alpha_\rho \rho^k m \frac{\xi_0}{s_{\min}}\|\delta_x\| \\
&\le \frac{m\xi_0^3}{2s_{\min}}\|\delta_x\|^2 + \frac{\alpha_\rho \rho_{\max}^{k-1} m}{s_{\min}}\left\|\zeta^k\right\|\|\delta_x\| \\
&\le \frac{m\xi_0^3}{2s_{\min}\gamma_1^2}\left\|g_A(x_c^k)\right\|^2 + \frac{\alpha_\rho \rho_{\max}^0 m\theta_p}{s_{\min}\gamma_1}\left\|g_A(x_c^k)\right\|^2. \qquad (4.15)
\end{aligned}$$

Combining (4.14) with (4.15), we obtain

$$q(\delta_A) = q_x(\delta_x) + q_s(\delta_s) \le \gamma_3 \left\|g_A(x_c^k)\right\|^2, \qquad (4.16)$$

where $\gamma_3 = -\frac{1}{2\gamma_2} + \frac{m\xi_0^3}{2s_{\min}\gamma_1^2} + \frac{\alpha_\rho\rho_{\max}^0 m\theta_p}{s_{\min}\gamma_1}$.

Now, using a Taylor expansion, the fact that $A(z_c^k)\delta_A^k = 0$ and $\delta_x^k = P(x_c^k)\delta_x^k$, (4.8), (4.6), (4.11), (4.12) and (4.13), we have

$$\begin{aligned}
\Delta L_T^+ &= L(z_c^k + \Lambda_c^k\delta_A^k, \lambda^k, \mu^k) - L(z_c^k, \lambda^k, \mu^k) \\
&= \nabla_z L(z_c^k, \lambda^k, \mu^k)^T \Lambda_c^k \delta_A^k + \frac{1}{2}(\delta_A^k)^T \Lambda_c^k \nabla_{zz}^2 L(z_c^k, \lambda^k, \mu^k)\Lambda_c^k\delta_A^k + o(\left\|\Lambda_c^k\delta_A^k\right\|^2) \\
&= (\zeta^k)^T\delta_A^k + \frac{1}{2}(\delta_A^k)^T W(z_c^k, \lambda^k, \mu^k)\delta_A^k + o(\left\|\delta_A^k\right\|^2) \\
&= q(\delta_A^k) + \frac{1}{2}(\delta_A^k)^T[W(z_c^k, \lambda^k, \mu^k) - B^k]\delta_A^k + o(\left\|\delta_A^k\right\|^2) \\
&= q(\delta_A^k) + o(\left\|g_A(x_c^k)\right\|^2). \qquad (4.17)
\end{aligned}$$

From (4.16), (4.17) and the fact that $\eta_1 \in (0,1)$, it follows that, for k sufficiently large,

$$r = \frac{\Delta L_T^+}{q(\delta_A^k)} = 1 + \frac{o(\left\|g_A(x_c^k)\right\|^2)}{q(\delta_A^k)} \ge 1 + \frac{1}{\gamma_3}\frac{o(\left\|g_A(x_c^k)\right\|^2)}{\left\|g_A(x_c^k)\right\|^2} \ge 1 - \frac{1}{\gamma_3}\gamma_3(1-\eta_1) = \eta_1,$$

which implies that one of the conditions of step 3 of Algorithm 1 is satisfied.

Finally, from (3.18), (4.13), and Hypothesis H10, we have

$$\left\|\Lambda_c^k\delta_A^k\right\| \le \left\|\Lambda_c^k\right\|\left\|\delta_A^k\right\| \le \xi_0\theta_4\left\|g_A(x_c^k)\right\| \le \xi_0\theta_4\theta_A\left\|\zeta^k\right\|.$$

Combining (4) with Hypothesis H7 we ensure that the hypotheses of the third part of Lemma 6 are satisfied, so there exists a sufficiently large k_0 such that $\rho_{\max}^k = \rho_{\max}^{k_0}$ for $k \ge k_0$. Therefore, using (3.3) and (3.13), we obtain

$$\left\|\zeta^k\right\| \le 10^4\rho^k \frac{\left\|g(z_c^k, \mu^k)\right\| + 1}{\rho_{\max}^k} \le 10^4\rho^k\frac{\xi_0+1}{\rho_{\max}^{k_0}} = \beta\rho^k,$$

where $\beta = 10^4(1+\xi_0)/\rho_{\max}^{k_0}$. Along with (2), (4.13), Hypotheses H10 and (3.4), this implies that, for k sufficiently large,

$$\begin{aligned}\left\|h(z_c^k + \Lambda_c^k \delta_A^k)\right\| &\leq \left\|h(z_c^k)\right\| + \left\|h(z_c^k + \Lambda_c^k \delta_A^k) - h(z_c^k)\right\| \leq \left\|h(z_c^k)\right\| + \bar{\xi}_0 \left\|\delta_A^k\right\|^2 \\ &\leq \left\|h(z_c^k)\right\| + \bar{\xi}_0 \theta_4^2 \left\|g_A(x_c^k)\right\|^2 \leq \left\|h(z_c^k)\right\| + \bar{\xi}_0 \theta_4^2 \theta_A^2 \left\|\zeta^k\right\|^2 \\ &\leq \rho^k + \beta \bar{\xi}_0 \theta_4^2 \theta_A^2 \left\|\zeta^k\right\| \rho^k = \rho^k (1 + \beta \bar{\xi}_0 \theta_4^2 \theta_A^2 \left\|\zeta^k\right\|).\end{aligned}$$

Since, for k sufficiently large, $\beta \bar{\xi}_0 \theta_4^2 \theta_A^2 \left\|\zeta^k\right\| < 1$, step δ_A^k is eventually accepted. □

From now on, we will also suppose that the normal step satisfies the following hypothesis.

Hypothesis H11 *For k sufficiently large, the normal step $\delta_N^{k+1} = z_c^{k+1} - z^k$ is obtained taking one or more steps of the form*

$$\delta_N^+ = -J^+ h(z_c) = -J^T (JJ^T)^{-1} h(z_c), \tag{4.18}$$

where J satisfies

$$\|J - \nabla h(z_c)\| = \mathcal{O}(\left\|\zeta^k\right\|). \tag{4.19}$$

Note that, using (4.7), we can redefine θ_3 in a way that $\left\|J^T \lambda\right\| \geq \theta_3 \|\lambda\|$ for k sufficiently large.

Using a Taylor expansion, (4.7), (4.18), (4.19), (3.19) and the continuity of $A(z)$, it is easy to show that, if $z_c^{k+1} \neq z^k$ for k sufficiently large, then the first normal step of iteration $k+1$, say δ_N^+, satisfies

$$\left\|\delta_N^+\right\| = \left\|J^+ h(z_c)\right\| \leq \frac{1}{\theta_3} \|h(z_c)\| = \mathcal{O}(\|h(z_c)\|), \tag{4.20}$$

and

$$\left\|h(z_c^{k+1})\right\| = o(\left\|h(z^k)\right\|). \tag{4.21}$$

In the next lemma, we show that $\|g_A(x)\|$ and $\|c_A(x)\|$ define an optimality measurement, i.e. they can be used to measure how close a point x is to x^*.

Lemma 8 *In a neighborhood V^* of x^*, we have*

$$\|x - x^*\| = \Theta(\|c_A(x)\| + \|g_A(x)\|).$$

Proof Since $g_A(x)$ and $c_A(x)$ are Lipschitz continuous, we have

$$\|c_A(x)\| + \|g_A(x)\| = \|c_A(x) - c_A(x^*)\| + \|g_A(x) - g_A(x^*)\| = \mathcal{O}(\|x - x^*\|)$$

Now, let us consider the Lagrangian function associated to the active constrains, i.e. $L_A(x, \lambda) = f(x) + c_A(x)^T \lambda$, whose derivatives are

$$\nabla L_A(x, \lambda) = \begin{bmatrix} \nabla f(x) + \nabla c_A(x)^T \lambda \\ c_A(x) \end{bmatrix}$$

and

$$\nabla^2 L_A(x,\lambda) = \begin{bmatrix} H_A(x,\lambda) & \nabla c_A(x)^T \\ \nabla c_A(x) & 0 \end{bmatrix}.$$

Notice that $\nabla L_A(x^*,\lambda^*) = 0$ and $\nabla^2 L_A(x^*,\lambda^*)$ is invertible.

Let us define $e_x = x^* - x$, $e_\lambda = \lambda^* - \lambda$ and $e = [e_x^T \; e_\lambda^T]^T$. Since $\nabla^2 L_A(x^*,\lambda^*)$ is invertible, we have that

$$\left\| \nabla^2 L_A(x^*,\lambda^*) v \right\| \geq \sigma \left\| v \right\|,$$

for every $v \in \mathbb{R}^{n+m_A}$. Moreover, since ∇L_A is differentiable, there are neighborhoods of x^* and λ^* such that, for x and λ in these neighborhoods,

$$\left\| \nabla L_A(x,\lambda) \right\| = \left\| \nabla L_A(x,\lambda) - \nabla L_A(x^*,\lambda^*) \right\| \geq \frac{\sigma}{2} \left\| \begin{bmatrix} x - x^* \\ \lambda - \lambda^* \end{bmatrix} \right\| \geq \frac{\sigma}{2} \left\| e_x \right\|.$$

In particular, in a neighbourhood of x^* we can choose $\lambda = \lambda_A(x)$, so that

$$\|x - x^*\| \leq \frac{2}{\sigma} \left\| \nabla L_A(x, \lambda_A(x)) \right\| = \mathcal{O}(\|g_A(x)\| + \|c_A(x)\|).$$

□

To conclude this section, we present the local convergence theorem of our method, showing that it has the same convergence rate as the equality constrained algorithm proposed in [5].

Theorem 3 *Under hypotheses H1-H11, x^k and x_c^k are 2-step superlinearly convergent to x^*. If a normal step is calculated every iteration, then x^k converges superlinearly to x^*.*

Proof The proof of this theorem is similar to the one presented for Theorem 4.3 of [5], taking into account only the active constraints. □

5 Numerical Experiments

In this section we present a comparison of our `C++` implementation of the algorithm, that is called `DCICPP`, with the other well-known algorithms `ALGENCAN` version 3.0.0-beta [2, 3] and `IPOPT` version 3.12.4 [44] with MA57 solver from HSL [1]. To analyze the performance of our algorithm, we used the CUTEst [29] repository of problems. A computer with an Intel i5-4440 3.1 GHz processor and 8Gb of RAM was used for the tests. The code can be obtained in [27]. Other packages used in our implementation include OpenBLAS [43, 45], Metis [32], SuiteSparse's CHOLMOD [9, 14–17], base_matrices [40] and nope [39].

Our comparison is based on the performance profiles proposed by Dolan and Moré [20]. The profiles were generated with the software `perprof-py` [41]. Given a set of problems P and a set of algorithms S, we define $t_{s,p}$ as the time algorithm $s \in S$ takes to solve a problem $p \in P$. Then, for each algorithm $s \in S$ that solves problem $p \in P$, we define the performance ratio

$$r_{s,p} = \frac{t_{s,p}}{\min\{t_{a,p} : a \in S, a \text{ solves } p\}}.$$

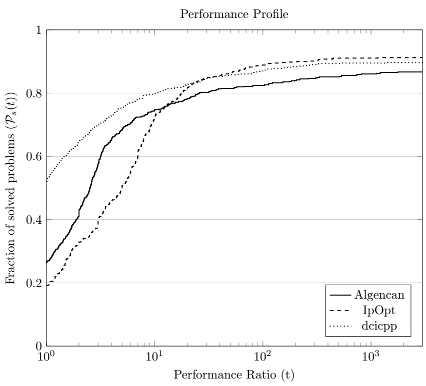
If algorithm s does not solve p, then we simply set $r_{s,p} = +\infty$.

Finally, we define the performance function of an algorithm s as

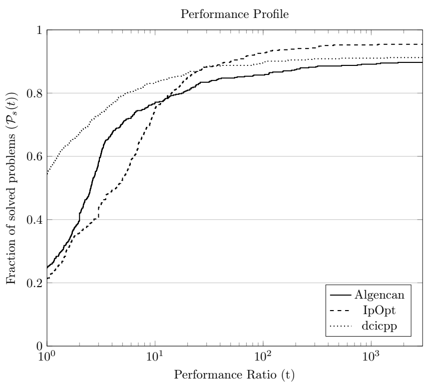
$$\mathcal{P}_s(t) = \frac{|\{p \in P : r_{s,p} \leq t\}|}{N_P},$$

where N_P is the number of elements in the set P. To compare the algorithms, we plot their functions on a finite interval $[1, r_f]$.

Our algorithm uses the Cholesky factorization to solve the linear systems in the normal step and to compute the Lagrange multipliers. As such, it is better suited for small and medium sized problems, with full rank Jacobians. Thus, in our tests we consider the 638 problems for which all of the Jacobians have full rank, and with no more than 5000 variables or constraints. The results are shown on Figure 2a. As it may be seen, `DCICPP` has a very good performance, surpassing the other two algorithms in efficiency, and also attaining a good robustness. Furthermore, it is the best algorithm for the problems in which no method is more than 20 times slower than the worst algorithm.



(a)



(b)

Fig. 2: Performance profiles (a) For the 638 problems for which Cholesky doesn't fail, and with no more than 5000 variables or 5000 constraints; (b) For the 525 problems in (a) for which the converging algorithms have close function value.

In addition to this comparison, we consider also the problems for which all converging algorithms seem to find the same solution, using the criteria given in [44]. Let $f_{s,p}$ be the objective function value obtained by solver s for problem p. We say that the converging algorithms have not found the same solution for problem p if

$$\frac{f_p^{\max} - f_p^{\min}}{1 + \max\{|f_p^{\max}|, |f_p^{\min}|\}} > 10^{-1},$$

where $f_p^{\max}$ and $f_p^{\min}$ are, respectively, the maximum and minimum of $\{f_{s,p} \mid s \in \mathcal{S}, s \text{ converges}\}$. Using this definition, there are 525 problems for which all algorithms seem to converge to the same solution. The resulting performance profile is shown of Figure 2b. Notice that the same analysis apply, despite the small loss on comparative robustness.

5.1 Efficiency of the restoration

One concern about the new algorithm is the efficiency of the infeasibility reduction scheme. The main purpose of using trust cylinders is to avoid computing a normal step per iteration, as done by SQP algorithms, for example. However, in the worst case scenario, the algorithm may need to make not only one, but several restorations in order to obtain $\|h(z_c)\| \leq \rho$.

Fortunately, Figure 3 shows that, in practice, the normal step of the algorithm is quite efficient. This figure, built from a set of 458 constrained problems for which dcicpp takes more than one iteration to converge, relates the percentage of iterations without restorations to the percentage of iterations with just one restoration step. The diameter of each circumference shown in the figure is proportional to the number of problems with the same coordinates.

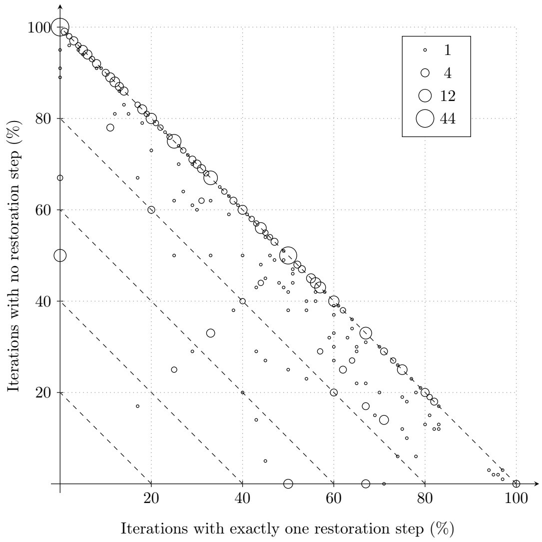


Fig. 3: Percentage of iterations with one or no restoration for 458 problems.

Diagonal lines link problems with the same percentage of iterations with no more that one restoration. The line segment that connects $(100, 0)$ and $(0, 100)$, for example, shows all of the 315 problems (69% of the total) for which no iteration takes more than one restoration. In fact, 238 of these 315 problems are located on the upper half of the segment, which means that no restoration was done on at least 50% of the iterations for these problems. Moreover, only 49 problems are below the segment that corresponds to 80%, i.e., for 89% of the problems the iterations with more than one restoration do not exceed 20% of the total number of iterations. Considering all of the 50528 iterations taken by DCICPP to solve these 458 problems, no restoration was made on 73% of them, just one restoration was necessary on 26%, and only 1% required more than 1 restoration.

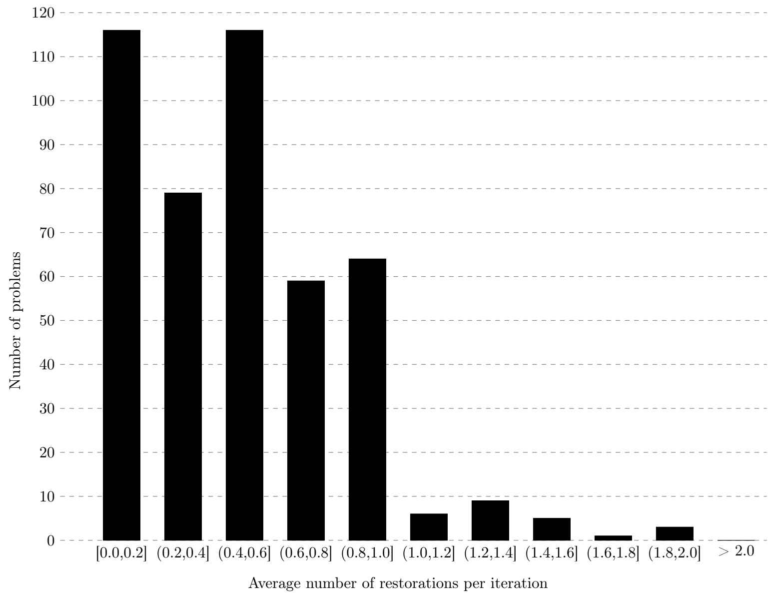


Fig. 4: Distribution of the problems according to the average number of restorations per iteration.

Figure 4 shows an histogram detailing the distribution of the problems according to the average number of restorations per iteration. From this figure, it is clear that making more than one restoration per iteration is a rare event. Besides, the median of the number of restorations per iteration is just 0.5.

6 Conclusion

In this paper we show how to extend the DCI method proposed by Bielschowsky and Gomes [5] to handle inequality constraints. The new algorithm maintains the good convergence properties of the original method, showing superlinear local convergence under reasonable assumptions. A preliminary implementation of the new algorithm shows that it is competitive to well-known nonlinear programming codes. The code is already available online and can be used with the CUTEst collection.

As a future work, we plan to combine the method with an iterative solver in order to handle very large problems, and to devise some strategy to handle rank deficient Jacobians.

References

1. HSL. a collection of Fortran codes for large scale scientific computation. URL `http://www.hsl.rl.ac.uk/`
2. Andreani, R., Birgin, E.G., Martínez, J.M., Schuverdt, M.L.: On augmented lagrangian methods with general lower-level constraints. SIAM Journal on Optimization **18**, 1286–1309 (2007). URL `http://dx.doi.org/10.1137/060654797`
3. Andreani, R., Birgin, E.G., Martínez, J.M., Schuverdt, M.L.: Augmented lagrangian methods under the constant positive linear dependence constraint qualification. Mathematical Programming **111**, 5–32 (2008). URL `http://dx.doi.org/10.1007/s10107-006-0077-1`
4. Biegler, L.T., Nocedal, J., Schmid, C.: A reduced hessian method for large-scale constrained optimization. SIAM Journal of Optimization **5**, 314–347 (1995). URL `http://dx.doi.org/10.1137/0805017`
5. Bielschowsky, R.H., Gomes, F.A.M.: Dynamic control of infeasibility in equality constrained optimization. SIAM Journal on Optimization **19**(3), 1299–1325 (2008). URL `http://dx.doi.org/10.1137/070679557`
6. Bueno, L.F., Haeser, G., Martínez, J.M.: A flexible inexact-restoration method for constrained optimization. Journal of Optimization Theory and Applications **165**(1), 188–208 (2015). URL `http://dx.doi.org/10.1007/s10957-014-0572-0`
7. Byrd, H., Gilbert, J.C., Nocedal, J.: A trust region method based on interior point techniques for nonlinear programming. Mathematical Programming **89**(1), 149–185 (2000). URL `http://dx.doi.org/10.1007/PL00011391`
8. Byrd, H., Hribar, M.E., Nocedal, J.: An interior point algorithm for large scale nonlinear programming. SIAM Journal on Optimization **9**(4), 877–900 (1999). URL `http://dx.doi.org/10.1137/S1052623497325107`
9. Chen, Y., Davis, T.A., Hager, W.W., Raamanickam, S.: Algorithm 887: Cholmod, supernodal sparse Cholesky factorization and update/downdate. ACM Transactions on Mathematical Software **35**(3) (2009). URL `http://dx.doi.org/10.1145/1391989.1391995`
10. Conn, A.R., Gould, N.I.M., Toint, P.L.: Trust Region Methods. Springer (2000). URL `http://dx.doi.org/10.1137/1.9780898719857`

11. Curtis, F.E., Gould, N.I.M., Robinson, D.P., Toint, P.L.: An interior-point trust-funnel algorithm for nonlinear optimization. Tech. Rep. 13T-010, COR@L Laboratory, Department of ISE, Lehigh University (2013). URL http://coral.ie.lehigh.edu/~frankecurtis/wp-content/papers/CurtGoulRobiToin13.pdf
12. Curtis, F.E., Nocedal, J., Wächter, A.: A matrix-free algorithm for equality constrained optimization problems with rank-deficient Jacobians. SIAM Journal on Optimization **20**(3), 1224–1249 (2009). URL http://dx.doi.org/10.1137/08072471X
13. Curtis, F.E., Schenk, O., Wächter, A.: An interior-point algorithm for large-scale nonlinear optimization with inexact step computations. SIAM Journal on Scientific Computing **32**(6), 3447–3475 (2010). URL http://dx.doi.org/10.1137/090747634
14. Davis, T.A., Hager, W.W.: Modifying a sparse Cholesky factorization. SIAM Journal on Matrix Analysis and Applications **20**(3), 606–627 (1999). URL http://dx.doi.org/10.1137/S0895479897321076
15. Davis, T.A., Hager, W.W.: Multiple-rank modifications of a sparse Cholesky factorization. SIAM Journal on Matrix Analysis and Applications **22**(4), 997–1013 (2001). URL http://dx.doi.org/10.1137/S0895479899357346
16. Davis, T.A., Hager, W.W.: Row modifications of a sparse Cholesky factorization. SIAM Journal on Matrix Analysis and Applications **26**(3), 621–639 (2005). URL http://dx.doi.org/10.1137/S089547980343641X
17. Davis, T.A., Hager, W.W.: Dynamic supernodes in sparse Cholesky update/downdate and triangular solves. ACM Transactions on Mathematical Software **35**(4) (2009). URL http://dx.doi.org/10.1145/1462173.1462176
18. Dennis., J.E., El-Alem, M., Maciel, M.C.: A global convergence theory for general trust-region-based algorithms for equality constrained optimization. SIAM Journal on Optimization **7**(1), 177–207 (1997). URL http://dx.doi.org/10.1137/S1052623492238881
19. Dennis, J.E., Vicente, L.N.: On the convergence theory of trust-region-based algorithms for equality-constrained optimization. SIAM Journal on Optimization **7**(4), 927–950 (1997). URL http://dx.doi.org/10.1137/S1052623494276026
20. Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. Mathematical Programming **91**(2), 201–213 (2002). URL http://dx.doi.org/10.1007/s101070100263
21. El-Alem, M.: A global convergence theory for Dennis, El-Alem, and Maciel's class of trust-region algorithms for constrained optimization without assuming regularity. SIAM Journal on Optimization **9**(4), 965–990 (1999). URL http://dx.doi.org/10.1137/S1052623497331762
22. Fletcher, R., Leyffer, S.: Nonlinear programming without a penalty function. Mathematical Programming **91**(2), 239–269 (2002). URL http://dx.doi.org/10.1007/s101070100244

23. Forsgren, A., Gill, P.E., Wright, M.H.: Interior methods for nonlinear optimization. SIAM review **44**(4), 525–597 (2002). URL http://dx.doi.org/10.1137/S0036144502414942
24. Francisco, J.B., Krejić, N., Martínez, J.M.: An interior-point method for solving box-constrained underdetermined nonlinear system. Journal of Computational and Applied Mathematics **177**, 67–88 (2005). URL http://dx.doi.org/10.1016/j.cam.2004.08.013
25. Gomes, F.A.M.: A sequential quadratic programming algorithm that combines merit function and filter ideas. Computational and Applied Mathematics **26**(3), 337–379 (2007). URL http://dx.doi.org/10.1590/S0101-82052007000300003
26. Gomes, F.A.M., Maciel, M.C., Martínez, J.M.: Nonlinear programming algorithms using trust regions and augmented Lagrangians with nonmonotone penalty parameters. Mathematical Programming **84**(1), 161–200 (1999). URL http://dx.doi.org/10.1007/s10107980014a
27. Gomes, F.A.M., Siqueira, A.S.: Dcicpp. http://www.github.com/abelsiqueira/dcicpp (2009)
28. Gonzaga, C.C., Karas, E., Vanti, M.: A globally convergent filter method for nonlinear programming. SIAM Journal on Optimization **14**(3), 646–669 (2003). URL http://dx.doi.org/10.1137/S1052623401399320
29. Gould, N.I.M., Orban, D., Toint, P.L.: CUTEst: a constrained and unconstrained testing environment with safe threads for mathematical optimization. Computational Optimization and Applications **60**(3), 545–557 (2015). URL http://dx.doi.org/10.1007/s10589-014-9687-3
30. Gould, N.I.M., Toint, P.L.: Nonlinear programming without a penalty function or a filter. Mathematical Programming **122**(1), 155–196 (2010). URL http://dx.doi.org/10.1007/s10107-008-0244-7
31. Karas, E.W., Gonzaga, C.C., Ribeiro, A.A.: Local convergence of filter methods for equality constrained nonlinear programming. Optimization **59**(8), 1153–1171 (2010). URL http://dx.doi.org/10.1080/02331930903085342
32. Karypis Lab: Metis - serial graph partitioning and fill-reducing matrix ordering. http://www-users.cs.umn.edi/ karypis/metis
33. Lalee, M., Nocedal, J., Plantenga, T.: On the implementation of an algorithm for large-scale equality constrained optimization. SIAM Journal on Optimization **8**(3), 682–706 (1998). URL http://dx.doi.org/10.1137/S1052623493262993
34. Nocedal, J., Wright, S.J.: Numerical Optimization, 2 edn. Springer (2006). URL http://dx.doi.org/10.1007/978-0-387-40065-5
35. Periçaro, G.A., Ribeiro, A.A., Karas, E.W.: Global convergence of a general filter algorithm based on an efficiency condition of the step. Applied Mathematics and Computation **219**(17), 9581–9597 (2013). URL http://dx.doi.org/10.1016/j.amc.2013.03.012
36. Powell, M.J., Yuan, Y.: A trust region algorithm for equality constrained optimization. Mathematical Programming **49**, 189–211 (1991). URL http://dx.doi.org/10.1007/BF01588787

37. Ribeiro, A.A., Karas, E., Gonzaga, C.C.: Global convergence of filter methods for nonlinear programming. SIAM Journal on Optimization **19**(3), 1231–1249 (2008). URL http://dx.doi.org/10.1137/060672285
38. Shanno, D.F., Vanderbei, R.J.: An interior-point algorithm for nonconvex nonlinear programming. Computational Optimization and Applications **13**, 231–252 (1999). URL http://dx.doi.org/10.1023/A:1008677427361
39. Siqueira, A.S.: NOPE - Nonlinear Optimization Preprocessing tEchniques. URL https://github.com/abelsiqueira/nope
40. Siqueira, A.S.: Base matrices. http://www.github.com/abelsiqueira/base_matrices (2009)
41. Siqueira, A.S., da Silva, R.G.C., Santos, L.R.: Perprof-py: A python package for performance profile of mathematical optimization software. Journal of Open Research Software **4**(1), e12 (2016). DOI 10.5334/jors.81. URL http://doi.org/10.5334/jors.81
42. Steihaug, T.: The conjugate gradient method and trust regions in large scale optimization. SIAM Journal on Numerical Analysis **20**(3), 626–637 (1983). URL http://dx.doi.org/10.1137/0720042
43. Wang, Q., Zhang, X., Zhang, Y., Yi, Q.: AUGEM: automatically generate high performance dense linear algebra kernels on x86 CPUs. In: Proceedings of SC13: International Conference for High Performance Computing, p. 25. ACM (2013)
44. Wächter, A., Biegler, L.T.: On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Mathematical Programming **106**(1), 25–57 (2006). URL http://dx.doi.org/10.1007/s10107-004-0559-y
45. Xianyi, Z., Qian, W., Yunquan, Z.: Model-driven level 3 BLAS performance optimization of Loongson 3A processor. In: Proceedings of the 2012 IEEE 18th international conference on parallel and distributed systems, pp. 684–691. IEEE computer society (2012)