Comput Optim Appl (2016) 64:327–354
DOI 10.1007/s10589-015-9820-y

An inertia-free filter line-search algorithm for large-scale nonlinear programming

Nai-Yuan Chiang[1] · **Victor M. Zavala**[2]

Received: 28 May 2015 / Published online: 15 February 2016

Abstract We present a filter line-search algorithm that does not require inertia information of the linear system. This feature enables the use of a wide range of linear algebra strategies and libraries, which is essential to tackle large-scale problems on modern computing architectures. The proposed approach performs curvature tests along the search step to detect negative curvature and to trigger convexification. We prove that the approach is globally convergent and we implement the approach within a parallel interior-point framework to solve large-scale and highly nonlinear problems. Our numerical tests demonstrate that the inertia-free approach is as efficient as inertia detection via symmetric indefinite factorizations. We also demonstrate that the inertia-free approach can lead to reductions in solution time because it reduces the amount of convexification needed.

Keywords Inertia · Nonlinear programming · Filter line-search · Nonconvex · Large-scale

Electronic supplementary material The online version of this article (doi:10.1007/s10589-015-9820-y) contains supplementary material, which is available to authorized users.

✉ Victor M. Zavala
victor.zavala@wisc.edu

Nai-Yuan Chiang
nychiang@mcs.anl.gov

[1] Mathematics and Computer Science Division, Argonne National Laboratory, 9700 South Cass Avenue, Argonne, IL 60439, USA

[2] Department of Chemical and Biological Engineering, University of Wisconsin-Madison, 1415 Engineering Hall, Madison, WI 53706, USA

1 Introduction

Filter line-search strategies have proven effective at solving large-scale, constrained nonlinear programs (NLPs) [6,12,16,30,37]. This algorithmic approach is implemented in the widely used IPOPT package [38]. In a filter line-search setting, it is essential to detect the presence of negative curvature and to regularize the Hessian of the Lagrangian when such is present. This ensures that the computed step is a descent direction when the constraint violation is sufficiently small. This in turn prevents the filter from accepting iteration subsequences that only improve the constraint violation.

The ability to handle negative curvature is essential when dealing with highly nonlinear and inherently ill-conditioned problems [43]. In a constrained NLP setting, the presence of negative curvature is detected using inertia information of the augmented matrix. Inertia information is provided by symmetric indefinite factorization routines such as MA27, MA57, MUMPS, and PARDISO [2,20,21,35]. An inertia-revealing preconditioning strategy based on incomplete factorizations has also been proposed that enables the use of iterative linear strategies such as QMR [36]. Unfortunately, many modern and efficient linear algebra strategies and libraries are not capable of providing inertia information. Examples include iterative techniques such as multigrid (geometric and algebraic) and Lagrange–Newton–Krylov [7,8]; parallel libraries for graphics processing units (GPUs) and distributed-memory systems such as MAGMA, ELEMENTAL, Trilinos, and PETSc [1,4,29]; and decomposition strategies for stochastic optimization, optimal control, and network problems that are widely used in convex optimization [22,26,32–34,41,45]. Consequently, the need for inertia information hinders modularity, application scope, and scalability. Byrd et al. recently proposed a line-search exact penalty framework that does not require inertia information [11]. In their approach, termination tests are included to guarantee that the search step provides sufficient progress in the merit function. This approach can also deal with inexact linear algebra and has been extended to deal with rank-deficient Jacobians [17]. The strategy has also proven to be effective when used within an interior-point framework [18].

Trust-region strategies provide a natural mechanism to detect negative curvature. Modern trust-region implementations for constrained NLP decompose the search step into tangential and normal components. The tangential step is computed by approximately solving a trust-region subproblem [10,23,28]. This is typically done by using a projected conjugate gradient (PCG) scheme that detects negative curvature at the inner iterates. In the presence of a direction of negative curvature, the PCG path is continued along this direction until it reaches the trust-region boundary. This approach is guaranteed to be globally convergent because it can always improve the Cauchy step (or revert to it) [15]. It is well-known, however, that the quality of the step can be poor when the PCG procedure is terminated prematurely and this can result in excessive shrinking of the trust-region and slow progress. A Lanczos approach can be used to improve the quality of the PCG step when this hits the trust-region boundary but this requires a more expensive procedure [24]. Under a trust-region setting, one is limited in the linear algebra techniques that can be used to compute the search step. In particular, one requires schemes that are compatible with the globalization approach (i.e., improve the progress of the Cauchy step). Linear algebra strategies such as direct factorizations, GMRES, and QMR are not compatible in this sense. This is important because some of these linear algebra strategies might handle difficult linear systems more efficiently than PCG. This observation has in fact motivated the implementation of a hybrid trust-region and line-search strategy in the widely used KNITRO package [40]. In addition, trust-region settings require tailored preconditioners that project (exactly or inexactly) the iterates onto the nullspace of the constraint Jacobian [23,28]. This can be a limitation when the preconditioner does not have the required projection properties or when it involves an iterative scheme such as multigrid. A line-search setting has the practical advantage that any linear algebra scheme can be used to compute the step (as long as the computed step is a descent direction) and this provides computational flexibility. Such flexibility motivates our interest in line-search approaches. The price to pay in a line-search setting is that regularization of the Hessian matrix (convexification) might be needed and this tends to decrease the quality of the computed step and increase the number of trial step computations. Trust-region settings, on the other hand, have the advantage that no explicit regularization is needed (this is done implicitly through the trust-region constraint) and this can enable more efficient handling of ill-posed problems such as those arising in parameter estimation [3]. Motivated by this property, we are interested in deriving step acceptance tests for line-search settings that limit the amount of regularization needed and that can better handle ill-posed problems.

In this work, we present an inertia-free filter line-search strategy for nonconvex NLPs. Motivated by curvature tests used in trust-region settings, the approach performs a curvature test along the tangential component of the computed step. The curvature test guarantees descent when the constraint violation is sufficiently small. When the test is not fulfilled, it triggers regularization of the Hessian matrix. We prove that global convergence of the algorithm can be guaranteed and that the norm of the step is a consistent criticality metric if the computed step satisfies the curvature test and if the iteration matrix is nonsingular at each iteration. These requirements are significantly less restrictive than the positive definiteness assumption for the reduced Hessian used in the standard filter line-search algorithm. We implement our developments in an interior-point framework and perform extensive numerical tests which include CUTE and Schittkowski problems as well as large-scale and highly nonlinear problems arising from power grid and natural gas networks. We demonstrate that two variants of the inertia-free approach are as efficient as the standard inertia detection strategy based on symmetric indefinite LBL^T factorizations in terms of iteration counts. We also demonstrate that the inertia-free variants can reduce the number of trial factorizations and solution times because of increased flexibility. We also demonstrate that such flexibility enables us to handle ill-posed problems more efficiently.

The paper is structured as follows. Section 2 presents the filter line-search algorithm of Wächter and Biegler [38,39] in an interior-point framework and discusses assumptions needed to guarantee global convergence. Section 4 presents the new inertia-free strategies and establishes global convergence. Section 5 compares the numerical performance of both strategies. Section 6 presents concluding remarks.

2 Interior-point framework

Consider the NLP of the form

$$\min_{x\in\Re^n} \quad f(x) \tag{1a}$$

$$\text{s.t. } c(x) = 0 \tag{1b}$$

$$x \geq 0. \tag{1c}$$

Here, $x \in \Re^n$ are primal variables and the objective and constraint functions are $f : \Re^n \to \Re$ and $c : \Re^n \to \Re^m$, respectively. We use a logarithmic barrier framework with subproblems of the form

$$\min_{x\in\Re^n} \varphi^{\mu}(x) := f(x) - \mu \sum_{j=1}^{n} \ln x^{(j)} \tag{2a}$$

$$\text{s.t. } c(x) = 0 \tag{2b}$$

where $\mu > 0$ is the barrier parameter and $x^{(j)}$ is the jth entry of vector x. We consider a framework that solves a sequence of barrier problems (2) and drives the barrier parameter μ monotonically to zero.

To approximately solve each barrier problem, we apply Newton's method to its optimality conditions:

$$\nabla_x \varphi^{\mu}(x) + \nabla_x c(x)\lambda = 0 \tag{3a}$$

$$c(x) = 0 \tag{3b}$$

while enforcing $x \geq 0$ along the search. Here, $\lambda \in \Re^m$ are multipliers for equality constraints. The primal variables and multipliers at iteration k are denoted as (x_k, λ_k). Their corresponding search directions $(d_k, \lambda_k^+ - \lambda_k)$ can be obtained by solving the linear system

$$\begin{bmatrix} W_k & J_k^T \\ J_k & 0 \end{bmatrix} \begin{bmatrix} d_k \\ \lambda_k^+ \end{bmatrix} = - \begin{bmatrix} g_k \\ c_k \end{bmatrix}. \tag{4}$$

We refer to this system as the augmented system. Here, $c_k := c(x_k)$, $J_k := \nabla_x c(x_k)^T \in \Re^{m\times n}$, $g_k := \nabla_x \varphi_k^{\mu}$, $W_k := H_k + \Sigma_k$, $H_k := \nabla_{xx}\mathcal{L}(x_k, \lambda_k) \in \Re^{n\times n}$, $\mathcal{L}(x_k, \lambda_k) := \varphi^{\mu}(x_k) + \lambda_k^T c(x_k)$, and $\Sigma_k := X_k^{-2}$ with $X_k := \text{diag}(x_k)$. One can show that the primal-dual approximation $\Sigma_k \approx X_k^{-1} V_k$, where $V_k := \text{diag}(\nu_k)$ and ν_k are multiplier estimates for the bounds (1c), can be used as long as the products $x_k^{(j)} \nu_k^{(j)}$ remain proportional to μ [18,39]. To enable compact notation, we define the augmented matrix

$$M_k := \begin{bmatrix} W_k & J_k^T \\ J_k & 0 \end{bmatrix}. \tag{5}$$

We can also consider the computation of the search directions d_k using the decomposition

$$d_k = n_k + t_k. \tag{6}$$

Here, n_k is computed from

$$\begin{bmatrix} W_k & J_k^T \\ J_k & 0 \end{bmatrix} \begin{bmatrix} n_k \\ \cdot \end{bmatrix} = - \begin{bmatrix} 0 \\ c_k \end{bmatrix}, \tag{7}$$

and t_k is computed from

$$\begin{bmatrix} W_k & J_k^T \\ J_k & 0 \end{bmatrix} \begin{bmatrix} t_k \\ \lambda_k^+ \end{bmatrix} = - \begin{bmatrix} g_k + W_k n_k \\ 0 \end{bmatrix}, \tag{8}$$

where λ_k^+ is the multiplier update.

We define a two-dimensional filter of the form $\mathcal{F} := \{(\theta(x), \varphi(x))\}$ with $\theta(x) = \|c(x)\|$, $\varphi(x) := \varphi^{\mu}(x)$ for a fixed barrier parameter μ, where $\|\cdot\|$ is the Euclidean norm. At each value of μ, the filter is initialized as

$$\mathcal{F}_0 := \{(\theta, \varphi) \mid \theta \geq \theta^{max}\} \tag{9}$$

with a given parameter $\theta^{max} > 0$. The filter at iteration k is denoted as $\mathcal{F}_k$. Given a search step d_k, a line search is started from counter $\ell \leftarrow 0$ and $\alpha_{k,0} = \alpha_k^{max} \leq 1$ to define trial iterates $x_k(\alpha_{k,l}) := x_k + \alpha_{k,\ell} d_k$.

We define

$$m_k(\alpha) := \alpha g_k^T d_k \tag{10}$$

as the linear model of $\varphi(x_k + \alpha d_k) - \varphi(x_k)$. We note that d_k is a descent direction[1] if $m_k(\alpha) < 0$. Having constants $\kappa_\theta > 0, s_\theta > 1, s_\varphi \geq 1, \gamma_\theta \in (0, 1)$, and $\eta_\varphi \in (0, 1)$, we consider the following conditions to check whether a trial iterate should be accepted.

- *Filter Condition FC*

$$(\theta(x_k(\alpha_{k,\ell})), \varphi(x_k(\alpha_{k,\ell}))) \notin \mathcal{F}_k$$

- *Switching Condition SC*

$$-m_k(\alpha_{k,\ell}) > 0 \quad \text{and} \quad [-m_k(\alpha_{k,\ell})]^{s_\varphi}[\alpha_{k,\ell}]^{1-s_\varphi} > \kappa_\theta[\theta(x_k)]^{s_\theta}$$

- *Armijo Condition AC*

$$\varphi(x_k(\alpha_{k,\ell})) \leq \varphi(x_k) + \eta_\varphi m_k(\alpha_{k,\ell}).$$

[1] We use the expression "descent direction" with respect to the barrier function.

– *Sufficient Decrease Condition SDC*

$$\theta(x_k(\alpha_{k,\ell})) \leq (1-\gamma_\theta)\theta(x_k) \quad \text{or} \quad \varphi(x_k(\alpha_{k,\ell})) \leq \varphi(x_k) - \gamma_\varphi \theta(x_k).$$

The filter condition FC is the first requirement for accepting a trial iterate $x_k(\alpha_{k,\ell})$. If the pair $(\theta(x_k(\alpha_{k,\ell})), \varphi(x_k(\alpha_{k,\ell}))) \in \mathcal{F}_k$ (i.e., the trial iterate is contained in the filter), the step is rejected and we decrease the stepsize. If the trial iterate is not contained in the filter, we continue testing additional conditions. We have two possible cases:

– If SC holds, the step d_k is a descent direction, and we check whether AC holds. If AC holds, we accept the trial point $x_k(\alpha_{k,\ell})$. If not, we decrease the stepsize.
– If SC does not hold and SDC holds, we accept the trial iterate $x_k(\alpha_{k,\ell})$. If not, we decrease the stepsize.

If the trial iterate $x_k(\alpha_{k,\ell})$ is accepted in the second case, the filter is augmented as

$$\mathcal{F}_{k+1} \leftarrow \mathcal{F}_k \cup \{(\theta,\varphi) \mid \varphi \geq \varphi(x_k) - \gamma_\varphi\theta(x_k), \;\; \theta \geq (1-\gamma_\theta)\theta(x_k)\} \tag{11}$$

with parameters $\gamma_\varphi \in (0,1)$; otherwise, we leave the filter unchanged (i.e., $\mathcal{F}_{k+1} \leftarrow \mathcal{F}_k$). If the trial stepsize $\alpha_{k,\ell}$ becomes smaller than α_k^{min} and the step has not been accepted in either case, we revert to feasibility restoration, and the filter is augmented. A strategy to obtain α_k^{min} is proposed in [39]. We define the set $\mathcal{R}_{inc}$ as the set of iteration counters k in which feasibility restoration is called.

We refer to the first condition of SDC as SDCC to emphasize that this condition accepts the trial iterate if it improves the constraint violation. Similarly, we refer to the second condition as SDCO to emphasize that this condition accepts the trial iterate if it improves the objective function. We refer to successful iterates in which the filter is not augmented (iterates in which the switching condition SC holds) as *f-iterates*. The filter line-search algorithm is summarized below.

Filter line-search algorithm

0. **Given** starting point x_0, λ_0, constants $\theta_{max} \in (\theta(x_0), \infty]$, γ_θ, $\gamma_\varphi \in (0,1)$, $\eta_\varphi \in (0,1)$, $\kappa_\theta > 0$, $s_\theta > 1$, $s_\varphi \geq 1$ and $0 < \tau_1 \leq \tau_2 < 1$.
1. **Initialize** filter $\mathcal{F}_0 := \{(\theta,\varphi) \,:\, \theta \geq \theta_{max}\}$ and iteration counter $k \leftarrow 0$.
2. **Check Convergence.** Stop if x_k is a stationary point (i.e., satisfies (3)).
3. **Compute Search Direction.** Compute step $(d_k, \lambda_k^+ - \lambda_k)$ from (6)-(8).
4. **Backtracking Line-Search.**
   (a) **Initialize**. Set $\alpha_{k,0} \leftarrow \alpha_k^{max}$ and counter $\ell \leftarrow 0$.
   (b) **Compute Trial Point**. If $\alpha_{k,\ell} \leq \alpha_k^{min}$ revert to feasibility restoration in Step 8. Otherwise, set trial point $x_k(\alpha_{k,\ell}) \leftarrow x_k + \alpha_{k,\ell} d_k$.
   (c) **Check Acceptability to the Filter**. If FC does not hold, reject trial point $x_k(\alpha_{k,\ell})$, and go to Step 4e.
   (d) **Check Sufficient Progress**.
      i If SC and AC hold, accept trial point $x_k(\alpha_{k,\ell})$ and go to Step 5.
      ii If SC does not hold and SDC hold, accept trial point $x_k(\alpha_{k,\ell})$, and go to Step 5. Otherwise, go to Step 4e.

   (e) **New Trial stepsize**. Choose $\alpha_{k,\ell+1} \in [\tau_1\alpha_{k,\ell}, \tau_2\alpha_{k,\ell}]$, set $\ell \leftarrow \ell + 1$, and go to Step 4b.

5. **Accept Trial Point**. Set $\alpha_k \leftarrow \alpha_{k,\ell}$ and $x_{k+1} \leftarrow x_k(\alpha_{k,\ell})$.
6. **Augment Filter**. If SC is not satisfied, augment filter using (11). Otherwise, leave filter unchanged.
7. **Next Iteration**. Increase iteration counter $k \leftarrow k + 1$ and go to Step 2.
8. **Feasibility Restoration**. Compute an iterate x_{k+1} that satisfies FC and SDC. Augment filter using (11), and go to Step 7.

The above algorithm seeks to solve the barrier subproblem (2) approximately for a fixed μ. Once this problem is solved, we decrease the barrier parameter μ and we reset the filter (9). This outer loop is repeated until we find a stationary point for the original NLP (1). We highlight that the global convergence analysis of the filter line-search algorithm discussed in the next section does not require a particular stepsize rule for the multipliers (it only requires boundedness of the multiplier updates λ_k^+). Consequently, we do not provide a explicit stepsize rule in the above algorithm. This is left as an implementation issue at this point that is discussed in Sect. 5.

3 Inertia-based strategy

The global convergence analysis of the filter line-search algorithm provided in [39] assumes a step decomposition of the form

$$d_k = Y_k\bar{q}_k + Z_k\bar{p}_k \tag{12a}$$

$$\bar{q}_k := -(J_kY_k)^{-1}c_k \tag{12b}$$

$$\bar{p}_k := -(Z_k^TW_kZ_k)^{-1}Z_k^T(g_k + W_kY_k\bar{q}_k), \tag{12c}$$

where $Y_k \in \Re^{n\times m}$ and $Z_k \in \Re^{n\times(n-m)}$ are matrices such that the columns of $[Y_k\ Z_k]$ form an orthonormal basis for $\Re^n$ and the columns of Z_k are the basis of the null space of J_k (i.e., $J_kZ_k = 0$). The convergence analysis also relies on the criticality measure

$$\chi_k := \|Z_k\bar{p}_k\| = \|\bar{p}_k\|, \tag{13}$$

where the last identity follows from the orthonormality of Z_k (i.e., $Z_k^TZ_k = I$). The analysis requires assumptions (G) in [39]. To facilitate the discussion, we state the relevant assumptions here:

Assumptions (G)

- (G1) There exists an open set $\Omega \subseteq \Re^n$ with $[x_k, x_k + d_k] \subseteq \Omega$ for all $k \notin \mathcal{R}_{inc}$ in which $\varphi(\cdot)$ and $c(\cdot)$ are twice differentiable and their values and derivatives are bounded.
- (G2) The matrices W_k are uniformly bounded for all $k \notin \mathcal{R}_{inc}$.
- (G3) The matrices W_k are uniformly positive definite on the null space of the Jacobian J_k.

(G4) There exists a constant $c_A > 0$ so that for all $k \notin \mathcal{R}_{inc}$, the smallest singular value of J_k is bounded by c_A.

(G5) There exists a constant $\theta_{inc} > 0$ such that $k \notin \mathcal{R}_{inc}$ whenever $\theta(x_k) \leq \theta_{inc}$.

Assumptions (G3) and (G4) are needed to guarantee that χ_k is a valid criticality measure in the sense that it converges to zero as we approach a first-order stationary point. To see this, consider a subsequence $\{x_{k_i}\}$ with $\lim_{i\to\infty}\chi_{k_i} = 0$ and $\lim_{i\to\infty}x_{k_i} = x^*$ for some feasible point x^*. If the Jacobian is of full row rank (implied by assumption (G4)) and (12b) holds, we have that $\lim_{i\to\infty}\bar{q}_{k_i} = 0$ as $\lim_{i\to\infty}x_{k_i} = x^*$. From $\lim_{i\to\infty}\chi_{k_i}$, (12c), and (13); and because (G3) guarantees nonsingularity of the reduced Hessian we have that $\lim_{i\to\infty}\|Z_{k_i}^T g_{k_i}\| = 0$. In Sect. 4 we will observe that a decomposition of the form (12) can always be obtained when the Jacobian is of full row rank and the augmented matrix is nonsingular (these are less restrictive assumptions). Consequently, the step computed from the augmented system (8) is equivalent to that obtained with (12).

Positive definiteness of the reduced Hessian (G3) also guarantees that the search direction is of descent when the constraint violation is sufficiently small and the criticality measure is nonzero (see Lemma 2 in [39]). As we will discuss in Sect. 4, this *descent lemma* is essential for establishing global convergence. Assumption (G3) can be enforced in a practical setting by monitoring the inertia (number of positive, negative, and zero eigenvalues) of the augmented matrix M_k and correcting it (if necessary) by regularizing the Hessian matrix as $W_k \leftarrow W_k + \delta I$ for $\delta \geq 0$. This *convexification* approach is justified from the results of Gould [25], which show that the reduced Hessian is positive definite if and only if the augmented matrix M_k has n positive, m negative, and no zero eigenvalues. We state this condition formally as

$$\text{Inertia}(M_k) = \{n, m, 0\}. \tag{14}$$

The augmented matrix M_k can be decomposed as LBL^T by using symmetric indefinite factorizations where L is a unit lower triangular matrix and B is a block diagonal matrix composed of 1×1 and 2×2 diagonal blocks. From Sylvester's law of inertia we know that the eigenvalues of M_k are the eigenvalues of B. Furthermore, because each 2×2 block is constructed by having one positive and one negative eigenvalue, the inertia of M_k can be estimated from the inertia of B [9].

We can enforce assumption (G3) using the following procedure. The matrix M_k is decomposed as LBL^T for $\delta = 0$ and the search direction d_k is computed using this decomposition. If the inertia is correct (i.e., condition (14) is satisfied), the search direction d_k is used as trial step in the line-search procedure. If the inertia is not correct, the regularization parameter δ is increased, the augmented matrix is refactorized and a new direction d_k is obtained. The procedure is repeated until the matrix M_k has the correct inertia. Heuristics are incorporated to accelerate the rate of increase/decrease of δ in order to ensure that the number of trial factorizations is not too large (because each factorization is computationally expensive). The inertia-based regularization strategy implemented in the current version of IPOPT [38] is shown below.

Inertia-based regularization (IBR)

IBR-1 Factorize M_k with $\delta = 0$. If (14) holds, compute d_k and stop.
IBR-2 If $\delta^{last} = 0$, set $\delta \leftarrow \bar{\delta}^0$, otherwise set $\delta \leftarrow \max\{\bar{\delta}^{min}, \kappa^- \delta^{last}\}$.
IBR-3 Factorize M_k with current δ. If (14) holds, set $\delta^{last} \leftarrow \delta$, compute d_k and stop.
IBR-4 If $\delta^{last} = 0$, set $\delta \leftarrow \hat{\kappa}^+ \delta$, otherwise set $\delta \leftarrow \kappa^+ \delta$ and go to IBR-3.

Here, $0 < \bar{\delta}^{min} < \bar{\delta}^0$, $0 < \kappa^- < 1 < \kappa^+ < \hat{\kappa}^+$ are given constants.

4 Inertia-free strategies

Estimating the inertia of M_k can be complicated or impossible when a decomposition of the form LBL^T is not available. As we noted in the introduction, this situation limits our options for computing the search step and motivates us to consider inertia-free strategies. If we take one step back we realize that the primary practical intention of inertia correction is to guarantee that the direction d_k is of descent when the constraint violation is sufficiently small. This approach, however, introduces a disconnect between the regularization procedure (IBR) and the filter line-search globalization procedure. In particular, the inertia test (14) is based solely on the structural properties of M_k and not on the computed direction d_k. Hence, the regularization procedure IBR can implicitly discard productive descent directions in attempting to enforce a correct inertia. We thus consider another route to enforce descent.

We first discuss a inertia-free strategy that uses the step decomposition (6), system (7)–(8), and the criticality measure

$$\Psi_k^t := \|t_k\|. \tag{15}$$

As we will argue in Sect. 4.1, a step decomposition is not strictly necessary, but it is advantageous for the analysis and can be used to enable the use of PCG strategies [23].

For our analysis we make the following assumptions.

Assumptions (RG)

(RG1) There exists an open set $\Omega \subseteq \Re^n$ with $[x_k, x_k + d_k] \subseteq \Omega$ for all $k \notin \mathcal{R}_{inc}$ in which $\varphi(\cdot)$ and $c(\cdot)$ are twice differentiable and their values and derivatives are bounded.

(RG2) The matrices W_k are uniformly bounded for all $k \notin \mathcal{R}_{inc}$.

(RG3) There exists a constant $\alpha_t > 0$ such that the step components n_k, t_k satisfy the following *curvature condition* (RG3a):

$$t_k^T W_k t_k + \max\{t_k^T W_k n_k - g_k^T n_k, 0\} \geq \alpha_t t_k^T t_k. \tag{16}$$

Furthermore, the augmented matrix M_k is nonsingular and its inverse is bounded for all $k \notin \mathcal{R}_{inc}$ (RG3b).

(RG4) There exists a constant $c_A > 0$ so that for all $k \notin \mathcal{R}_{inc}$, the smallest singular value of J_k is bounded by c_A.

(RG5) There exists a constant $\theta_{inc} > 0$ such that $k \notin \mathcal{R}_{inc}$ whenever $\theta(x_k) \leq \theta_{inc}$.

The key difference between assumptions (RG) and assumptions (G) used in the standard filter line-search algorithm is that we do not require positive definiteness of the reduced Hessian (G3). This requirement is replaced by assumptions (RG3) and we now show that these are less restrictive and sufficient to guarantee global convergence. We begin by showing that conditions (RG3b) and (RG4) are sufficient to guarantee that the reduced Hessian is nonsingular and that (15) is a valid criticality measure.

Lemma 1 *Let (RG3b) and (RG4) hold and define $M = M_k$, $W = W_k$, $Z = Z_k$, and $J = J_k$. Then (i) the reduced Hessian $Z^T W Z$ is nonsingular and (ii) the inverse of M admits a decomposition of the form*

$$M_{inv} = \begin{bmatrix} W & J^T \\ J & 0 \end{bmatrix}^{-1} = \begin{bmatrix} P & (I - PW)J^T V^{-1} \\ V^{-1}J(I - WP) & -V^{-1}J(W - WPW)J^T V^{-1} \end{bmatrix} \tag{17}$$

with

$$P = Z(Z^T W Z)^{-1} Z^T \tag{18}$$

and $V = JJ^T$.

Proof Part (i) follows from Lemmas 3.2 and 3.4 in [25]. These results establish that if the Jacobian is of full row rank (RG4) there exists a nonsingular matrix R such that

$$RMR^T = \begin{bmatrix} & & I \\ & Z^T W Z & \\ I & & \end{bmatrix}. \tag{19}$$

From Sylvester's law of inertia and the structure of RMR^T we have that

$$\text{Inertia}(M) = \text{Inertia}(Z^T W Z) + \{m, m, 0\}. \tag{20}$$

Consequently, the number of zero eigenvalues of M is equal to the number of zero eigenvalues of $Z^T W Z$. By assumption (RG3b) we know that M is nonsingular and thus we know that it does not have zero eigenvalues. Consequently, $Z^T W Z$ does not have zero eigenvalues either and therefore is nonsingular.

To prove (ii), we first note that (RG3b) and (RG4) guarantee that P exists. We denote the blocks of $M_{inv}M$ as A_{11}, $A_{12} = A_{21}^T$, A_{22}, and we seek to prove that $M_{inv}M = I$. By direct calculation and by noticing that $J^T(JJ^T)^{-1}J = I - ZZ^T$ [5, p. 20] and $JZ = 0$, we obtain

$$\begin{aligned} A_{11} &= WP + J^T V^{-1} J(I - WP) \\ &= WP + (I - ZZ^T)(I - WP) \\ &= I - ZZ^T + ZZ^T WZ(Z^T WZ)^{-1} Z^T \\ &= I \end{aligned} \tag{21a}$$

$$\begin{aligned} A_{12} &= W(I - PW)J^T(JJ^T)^{-1} - J^T(JJ^T)^{-1}J(W - WPW)J^T(JJ^T)^{-1} \\ &= ZZ^T(W - WPW)J^T(JJ^T)^{-1} \\ &= \left(ZZ^T W - ZZ^T WZ(Z^T WZ)^{-1} Z^T W\right) J^T(JJ^T)^{-1} \\ &= 0 \end{aligned} \tag{21b}$$

$$\begin{aligned} A_{22} &= J(I - PW)J^T(JJ^T)^{-1} \\ &= J(I - Z(Z^T WZ)^{-1} Z^T W)J^T(JJ^T)^{-1} \\ &= JJ^T(JJ^T)^{-1} - JZ(Z^T WZ)^{-1} Z^T WJ^T(JJ^T)^{-1} \\ &= I. \end{aligned} \tag{21c}$$

The proof is complete. □

From (7), (8), and the explicit form of the inverse of M_k in (17) we have that,

$$n_k = -(I - Z_k(Z_k^T W_k Z_k)^{-1} Z_k^T W_k) J_k^T (J_k J_k^T)^{-1} c_k, \tag{22}$$

and

$$t_k = -Z_k(Z_k^T W_k Z_k)^{-1} Z_k^T (g_k + W_k n_k). \tag{23}$$

We thus see that the tangential step is equivalent to the tangential component $Z_k \bar{p}_k$ obtained from the decomposition (12). Consequently, Lemma 1 implies that an expression for the tangential step of the form in (12) can always be obtained if the Jacobian is of full row rank and the augmented matrix is nonsingular. We also note that the expression for the normal component in (12) is not equivalent to that in (22) but this is not necessary as long as n_k obtained from (7) is bounded and decays to zero as we approach a feasible point (equivalence can be achieved by replacing W_k with I in (7)). These observations allow us to prove that the measure (15) is a valid criticality measure under assumptions (RG), as stated in the next result.

Theorem 1 *Consider a subsequence $\{x_{k_i}\}$ with $lim_{i\to\infty} x_{k_i} = x^*$ for a feasible x^*, let (RG3b) and (RG4) hold, let n_{k_i} solve (7), and let t_{k_i} solve (8). Then*

$$\lim_{i\to\infty} \Psi^t_{k_i} = 0 \implies \lim_{i\to\infty} \left\| Z_{k_i}^T g_{k_i} \right\| = 0$$

for Z_{k_i} spanning the null space of J_{k_i}.

Proof Define $M := M_{k_i}$, $W := W_{k_i}$, $J := J_{k_i}$, and $Z := Z_{k_i}$. Boundedness of n_k follows from (RG1), which guarantees that the right-hand side of the augmented system (7) is bounded and from (RG3b) which guarantees that the inverse of M_k is bounded. Boundedness of n_k together with (RG1) and (RG2) guarantee that the right hand side of (8) is bounded. This fact, together with (RG3b), guarantee that t_k is bounded. We thus have that condition $\lim_{i\to\infty} x_{k_i} = x^*$ for feasible x^* ensures $\lim_{i\to\infty} n_{k_i} = 0$. From Lemma 1 we know that $Z^T W Z$ is nonsingular and therefore the projection matrix P_{k_i} exists and is nonsingular. From (15), (23), and $\lim_{i\to\infty} n_{k_i} = 0$ as $\lim_{i\to\infty} x_{k_i} = x^*$ we obtain the result. □

We now show that the curvature condition of assumption (RG3a) is sufficient to ensure that the step d_k is a descent direction.

Lemma 2 *Let Assumptions (RG1)–(RG5) hold. If x_{k_i} is a subsequence of iterates for which $\Psi^t_{k_i} \geq \epsilon$ with a constant ϵ independent of i, then there exist positive constants ϵ_1, ϵ_2 such that*

$$\theta_{k_i} \leq \epsilon_1 \implies \frac{m_{k_i}(\alpha)}{\alpha} \leq -\epsilon_2.$$

Proof Define $W := W_{k_i}$, $J := J_{k_i}$, $g := g_{k_i}$, $d := d_{k_i}$, $\Psi := \Psi^t_{k_i}$, $\theta := \theta_{k_i}$, $n := n_{k_i}$, and $t := t_{k_i}$. Multiplying the first row of (8) by t^T and recalling that $Jt = 0$, we obtain

$$t^T W t = -g^T t - t^T W n.$$

We know that $g^T d = g^T n + g^T t$. Thus, combining terms, we obtain

$$-g^T d = t^T W t + t^T W n - g^T n.$$

We consider two cases. In the first case we have that $t^T W n - g^T n < 0$, and the curvature condition (16) guarantees that $t^T W t \geq \alpha_t t^T t$ and $-g^T d \geq \alpha_t t^T t + t^T W n - g^T n$. From (RG1) we can obtain the bounds $|t^T W n| \leq \bar{c}_1 \Psi \theta$ and $|g^T n| \leq \bar{c}_2 \theta$ for $\bar{c}_1, \bar{c}_2 > 0$. From (15) we have that $\|t\| = \Psi$, and we thus have

$$\begin{aligned} g^T d &\leq -\alpha_t \Psi^2 + \bar{c}_1 \Psi \theta + \bar{c}_2 \theta \\ &\leq \Psi \left(-\alpha_t \epsilon + \bar{c}_1 \theta + \frac{\bar{c}_2}{\epsilon} \theta \right). \end{aligned}$$

Defining $\epsilon_1 := \min \left\{ \theta_{inc}, \frac{\epsilon^2 \alpha_t}{2(\bar{c}_1 \epsilon + \bar{c}_2)} \right\}$ with θ_{inc} from (RG5), it follows that for all $\theta \leq \epsilon_1$ we have $m(\alpha) \leq -\alpha \epsilon_2$ with $\epsilon_2 := \frac{\epsilon^2 \alpha_t}{2}$. In the second case, we have that $t^T W n - g^T n \geq 0$ and the curvature condition guarantees that $t^T W t + t^T W n - g^T n \geq \alpha_t t^T t$ and $-g^T d \geq \alpha_t t^T t$. In this case the result follows with $\epsilon_1 := \theta_{inc}$ because $\bar{c}_1 = \bar{c}_2 = 0$ and for ϵ_2 defined previously. □

The descent lemma guarantees that the objective function will be improved at a subsequence of nonstationary iterates (i.e., those with $\Psi^t_{k_i} \geq \epsilon$) having a sufficiently small constraint violation θ_{k_i}. This implies that f-iterates will eventually be accepted and the filter is eventually not augmented. This in turn implies that an infinite subsequence of nonstationary iterates cannot exist. We now prove that assumptions (RG) guarantee global convergence of the filter line-search algorithm.

Theorem 2 *Let Assumptions (RG) hold. The filter line-search algorithm delivers a sequence $\{x_k\}$ satisfying*

$$\lim_{k\to\infty} \theta(x_k) = 0 \tag{25a}$$

$$\liminf_{k\to\infty} \Psi^t(x_k) = 0. \tag{25b}$$

Proof We go through the results leading to the proof of Theorem 2 in [39] and argue that our assumptions (RG) are sufficient. Unless otherwise stated, all lemmas refer to those in [39]. Boundedness of d_k follows from boundedness of its components n_k and t_k which in turn follow from (RG1), (RG2), and (RG3b), as was argued in the proof of Theorem 1. We can also use similar arguments to prove that boundedness of λ_k^+ follows from boundedness of n_k, and from (RG1), (RG2) and (RG3b). Boundedness of $|m_k(\alpha)|$ follows from (RG1). Lemma 2 is replaced by the descent Lemma 2 of this work. Lemma 3 are standard bounding results that follow from Taylor's theorem and require only (RG1). Lemma 4 follows from the descent Lemma 2 of this work. Lemma 6 requires only (RG1). Lemma 8 requires the descent Lemma 2 of this work. Lemma 10 establishes that for a subsequence of nonstationary iterates the filter is eventually not augmented. This requires the descent Lemma 2 of this work. The result follows. □

The curvature condition (16) of assumption (RG3a) can hold even if the reduced Hessian matrix is not positive definite, as required by assumption (G3) in the standard filter line-search algorithm. Consequently, the curvature condition is less restrictive. If the curvature condition does not hold for the computed step, we can enforce it by regularizing the Hessian matrix $W_k \leftarrow W_k + \delta I$. Specifically, the curvature condition (16) can always be satisfied for sufficiently large δ and sufficiently small α_t satisfying $\alpha_t \leq \lambda_{min}(Z_k^T W_k Z_k)$. The reason is that t_k lies on the null space of J_k and, consequently, can always be expressed as $t_k = Z_k u$ for a given nonzero vector u and the curvature condition then implies that $u^T Z_k^T W_k Z_k u \geq \alpha_t u^T u$. We also know that an appropriate α_t exists for any δ because $\lambda_{min}(Z_k^T W_k(\delta) Z_k)$ is an increasing function of δ. Note also that the term $(\max\{t_k^T W_k n_k - g_k^T n_k, 0\})$ does not affect these properties. This is because, if the argument is negative, then this is set to zero; and, if the argument is positive, it provides additional flexibility to satisfy the curvature condition.

We can enforce nonsingularity of M_k and boundedness of its inverse, as required by assumption (RG3b), by regularizing the Hessian W_k as well while making sure that (RG2) holds. This is necessary to guarantee that both n_k and t_k are bounded. Note also that nonsingularity of M_k can be monitored using a linear solver and boundedness of n_k and t_k can be monitored directly. These observations lead to the following inertia-free regularization (IFR) procedure:

Inertia-free regularization (IFR)

IFR-1 Given constant $\alpha_t > 0$, factorize M_k with $\delta = 0$ and compute n_k, t_k from (7) and (8). If t_k satisfies the curvature condition (16) and M_k satisfies (RG3b), set $d_k = t_k + n_k$, and terminate.

IFR-2 If $\delta^{last} = 0$, set $\delta \leftarrow \bar{\delta}^0$, otherwise set $\delta \leftarrow \max\{\delta^{min}, \kappa^- \delta^{last}\}$.

IFR-3 Given constant $\alpha_t > 0$, factorize M_k with current δ and compute n_k, t_k from (7) and (8). If t_k satisfies (16) and M_k satisfies (RG3b), set $d_k \leftarrow n_k + t_k$, and terminate.

IFR-4 If $\delta^{last} = 0$, set $\delta \leftarrow \hat{\kappa}^+ \delta$, otherwise set $\delta \leftarrow \kappa^+ \delta$ and go to IFR-3.

We make the following remarks:

– The curvature condition (16) is enforced at every iteration. In principle, however, one can enforce it only at iterations in which the constraint violation is less than a certain small threshold value θ_{sml}. This approach is consistent with the observation that the switching condition SC needs to be checked only at iterations with small constraint violation [39]. In either case, however, we might need to regularize the Hessian in order to enforce nonsingularity of M_k at every iteration.

– A potential caveat of the inertia-free strategy is that it cannot guarantee that the step computed via the augmented system is a minimum of the associated quadratic program (the inertia-based approach guarantees this). Consequently, while enabling global convergence with enhanced flexibility is a great benefit of inertia-free strategy, the potential price to pay is the possibility of having a larger proportion of steps that are accepted because of improvements on constraint violation and not on the objective function. This situation might ultimately manifest as a tendency to get attracted to first-order stationary points with larger objective values than those obtained with the inertia-based strategy. We provide numerical results in Sect. 5 to discuss this issue further.

– As seen in Lemma 2, the term $(\max\{t_k^T W_k n_k - g_k^T n_k, 0\})$ in the curvature condition is harmless and is included only to provide additional flexibility. Because of this, one might also consider the simpler test $t_k^T W_k t_k \geq \alpha_t t_k^T t_k$ and still guarantee convergence.

– The normal component n_k can also be computed from system (7) by replacing W_k with I. This, in fact, is a more natural choice than our approach because the normal and tangential components correspond to the decomposition (12) for $Y_k = J_k^T$. Moreover, this approach naturally guarantees that the corresponding augmented matrix is bounded and that the normal step is bounded. This, approach, however, would require the solution of a linear system with a different coefficient matrix as the one used to compute the tangential step.

– Assumption (RG3b) is needed to guarantee that the tangential step is bounded. In practice, however, this condition might be unnecessarily strong if the normal step is guaranteed to be bounded. This is because, in principle, one could still compute a productive tangential step that satisfies (RG3a) even if (RG3b) does not hold. Establishing convergence in this case, however, would require a more sophisticated analysis.

– Assumption (RG1) can be guaranteed to hold only if all iterates x_k remain strictly in the interior of the nonnegative orthant. This condition guarantees that the barrier function $\varphi^{\mu}(x_k)$ and its derivatives are bounded. One can show that Theorem 3 in [39] holds under assumptions (RG). This result establishes that the iterates x_k remain in the strict interior of the feasible region if the maximum stepsize α_k^{max} is determined by using the following fraction-to-the-boundary rule

$$\alpha_k^{max} := \max\{\alpha \in (0, 1] \, : \, x_k + \alpha d_k \geq (1 - \tau)x_k\}, \tag{26}$$

for a fixed parameter $\tau \in (0, 1)$. The full row rank assumption of J_k (RG4), together with the assumption that its rows and the corresponding rows of the active bounds of x_k are linearly independent as well as the nonsingularity of the reduced Hessian (which follows from (RG3b) and Lemma 1), is sufficient for establishing the result.

4.1 Alternative inertia-free tests

Computing the normal and tangential components of the step separately can be beneficial in certain situations. For instance, the use of a PCG scheme provides a mechanism to perform the curvature test

$$t_{k,j}^T W_k t_{k,j} \geq \alpha_t t_{k,j}^T t_{k,j} \tag{27}$$

on the fly at each PCG iteration j and thus terminate early and save some work if the test does not hold for the current regularization parameter δ. This approach can also be beneficial because more test directions are used to identify negative curvature. This approach, however, requires a preconditioner that projects the iterations onto the null space of the Jacobian, which might not be available in certain applications.

In some applications it might be desirable to operate directly with the full step d_k. In this case, we can impose a curvature condition of the form

$$d_k^T W_k d_k + \max\{-(\lambda_k^+)^T c_k, 0\} \geq \alpha_d d_k^T d_k, \tag{28}$$

with d_k computed from (4).

To argue that test (28) is consistent, we use the criticality measure

$$\Psi_k^d := \|d_k\|. \tag{29}$$

If (RG3b) and (RG4) hold, we have from Lemma 1 and equation (4) that

$$\begin{aligned} d_k &= -P_k g_k - (I - P_k W_k) J_k^T (J_k J_k^T)^{-1} c_k \\ &= -Z_k (Z_k^T W_k Z_k)^{-1} Z_k^T (g_k - W_k J_k^T (J_k J_k^T)^{-1} c_k) - J_k^T (J_k J_k^T)^{-1} c_k. \end{aligned} \tag{30}$$

If we set $Y_k = J_k^T$, we have that d_k has the same structure as the step decomposition in (12). Moreover, we have that

$$\Psi_k^d = \Psi_k^t + O(\|c_k\|). \tag{31}$$

Consequently, the results of Theorem 1 still hold and Ψ_k^d is a valid criticality measure. If $-(\lambda_k^+)^T c_k < 0$, from (28), (RG1), (RG4), and (RG3b) we have that $\|\lambda_k^+\| \leq \kappa$ for some $\kappa > 0$ and, therefore,

$$\begin{aligned} g_k^T d_k &= -d_k^T W_k d_k + d_k^T J_k^T \lambda_k^+ \\ &= -d_k^T W_k d_k + c_k^T \lambda_k^+ \\ &\leq -\alpha_d (\Psi_k^d)^2 + \kappa \theta_k. \end{aligned} \tag{32}$$

Consequently, the results of Lemma 2 hold with appropriate constants. If $-(\lambda_k^+)^T c_k \geq 0$ holds the result follows with $\kappa = 0$.

The curvature condition (28) holds for any $\alpha_d \leq \lambda_{min}(W_k)$, and we note that the term $\max\{-(\lambda_k^+)^T c_k, 0\}$ is harmless and is used only to enhance flexibility.

5 Numerical results

In this section we benchmark the inertia-based and inertia-free strategies using the PIPS-NLP interior-point framework [14]. We first describe the implementation used to perform the benchmarks. We then present results for small-scale problems and large-scale problems arising from different applications.

5.1 Implementation

PIPS-NLP is an object-oriented interior-point framework implemented in C++ that facilitates the communication of model structures and the development of linear algebra strategies. These capabilities enable us to solve large-scale problems on parallel computing architectures. The algorithmic framework of PIPS-NLP follows along the lines of that of IPOPT [38] but we do not implement a feasibility restoration phase and watchdog heuristics. If the restoration phase is reached, we terminate the algorithm. We prefer to use this approach to isolate the effects of inertia correction on performance. As in IPOPT [38], we allow steps that are very small in norm $\|d_k\|$ to be accepted even if they do not satisfy SDC or SAC, we use the same stepsize for both primal and equality constraint multipliers, use a primal-dual Hessian, and use a different stepsize for the bound multipliers. We have validated the performance of our implementation by comparing it with that of IPOPT, and we have obtained nearly identical results.

We compare three regularization strategies: (1) IBR: inertia-based regularization, (2) IFRt: inertia-free regularization with the curvature test (16), and (3) IFRd: inertia-free regularization with the curvature test (28). We use the same parameters for IBR and IFR to increase/decrease the regularization parameter δ. For IFRd and IFRt we set both α_t and α_d to 10^{-12} as default. We also scale these parameters using the barrier parameter μ, as suggested in [18]. Strategy IFRd has been implemented in the latest version of IPOPT https://projects.coin-or.org/Ipopt.

For the IFRt strategy, we compute the normal step by solving the linear system (7) by factorizing M_k using MA57 [20], and we reuse the factorization to compute the tangential component from (8). For IBR we estimate the inertia of the augmented matrix using MA57.

We use the optimality error described in [38, Sect. 2] with a convergence tolerance of 10^{-6}, we perform iterative refinement for the augmented system with a tolerance of 10^{-12}, and we set the maximum number of iterations to 1000. If the line search cannot find an acceptable point within 50 trials, the last trial stepsize is used in the next iteration. We use a pivoting tolerance of 10^{-4} for MA57.

Of the entire set of test problems studied in Sect. 5.2, 13 % have Jacobians that are nearly rank-deficient. To deal with these instances, we regularize the (2,2) block of the augmented matrix as in IPOPT whenever we detect the augmented matrix to be singular [38, Sect. 3.1]. We emphasize that the convergence theory of this work and the supporting theory in [39] do not hold in this case. In particular, the tangential step computed in IFRt no longer lies exactly on the null-space of the Jacobian and the inverse expression of the augmented matrix of Lemma 1 does not hold. The regularization parameter of the (2,2) block is, however, very small (10^{-8}–10^{-11}) and this is often sufficient to avoid singularity issues. The numerical results obtained indicate that the performance of IFRd and IFRt are not strongly affected by this regularization.

5.2 Small-scale tests

We consider 929 test problems: 738 CUTE instances http://orfe.princeton.edu/~rvdb/ampl/nlmodels/cute/index.html, 188 Schittkowski instances http://orfe.princeton.edu/~rvdb/ampl/nlmodels/cute/index.html, and three additional optimal power flow instances based on IEEE data. The power flow models are accessible at http://zavalab.engr.wisc.edu/data/testinertia. Out of the 929 test problems, PIPS-NLP can solve 828 problems with at least one of the regularization strategies implemented (89 % of all the tests). This demonstrates the robustness of our implementation. The rest of the problems cannot be solved with any of the strategies considered because the solver either: (1) reaches the limit of iterations; (2) requires feasibility restoration; (3) encounters an evaluation error in the problem functions; (4) requires regularization that reaches the maximum limit of 1×10^{12}. We only compare the performance of inertia-based and inertia-free strategies for the 828 problems for which at least one strategy was successful.

The performance of the three strategies is presented in Table A1 in the Appendix. Here, we report the optimal objective (OBJ), the number of iterations (Iter), number of regularizations (Reg), and solution time in seconds. The total number of factorizations equals the number of iterations plus the number of regularizations. We use "-" to denote the tests that cannot be solved within the limit of iterations. The last three instances in the table are the energy problems.

We present the numerical results in Table A1. From these results we have that 802, 796, and 794 test problems can be solved with IBR, IFRd and IFRt, respectively. Moreover, there are several instances that can only be solved with the inertia-free strategies (e.g., `fltcher`, and `steenbrf`). We can thus conclude that the inertia-free strategies are competitive. In Figs. 1 and 2 we present Dolan–Moré profiles [19] for the total number of iterations and the average number of factorizations per iteration.

Fig. 1 Number of iterations

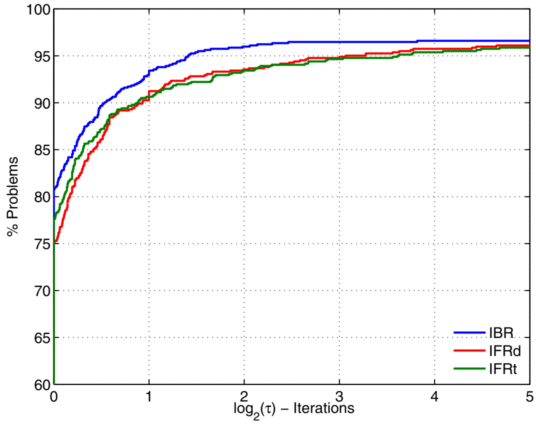
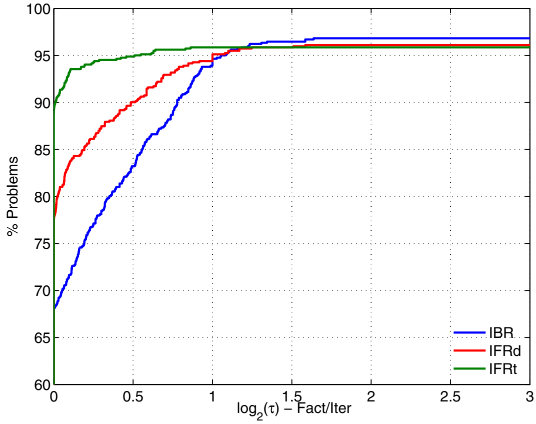


Fig. 2 Number of factorizations per iteration



In Fig. 1 we can see that IBR requires fewer iterations to converge, but the differences are negligible. In Fig. 2 we can see that IFRd and IFRt significantly outperform IBR in terms of the average number of factorizations required per iteration. From the tests reported in Table A1 we also estimate the average number of regularizations per iteration. We have that IBR requires 0.61 regularizations per iteration while IFRd and IFRt require 0.37 and 0.43 regularizations per iteration, respectively. We thus conclude that IFR strategies significantly reduce the amount of regularization needed.

The inertia-free strategies yield the same final objective value as IBR in 85 % of the instances. The performance in this respect is rather surprising. IBR yields better final objective values than IFRd and IFRt in 81 instances (e.g., instances `hatfldd` and `hatflde`), while IFRd and IFRt yield better objective values than IBR in 40 and 41 instances (e.g., instances `s295` and `s296`), respectively. We could, in principle, attribute the tendency of IBR to reach better final objective values to the fact that the solutions of the augmented system are actual minimizers of the associated quadratic program at each iteration, whereas the solutions obtained with the inertia-free strategies are not. Consequently, we would expect that steps computed with IBR should yield improvements in the objective more often. Interestingly, we have found this not necessarily to be the case. To demonstrate this, we compared the percentages of steps accepted by the filter for the three strategies as a result of improvements in the objective function and in the constraint violation. We recall that a trial stepsize can be accepted under three cases: (1) SAC: both SC and AC hold, (2) SDCO: SDC holds because of sufficient decrease in the objective, and (3) SDCC: SDC holds because of sufficient reduction in the constraint violation. In Table A2 we present the percentage of successful trial steps obtained for each case. We note that the percentages do not add to 100 % in some cases because we allow the line search to accept very small steps and because we round the percentages to the nearest integer. To perform this comparison, we consider only problems in which all strategies are successful, and we consider only problems with constraints (the unconstrained instances have a percentage of acceptance for SAC of 100 %). The last row presents the average percentages for all problems and from this we can see that IBR accepts 46 % of the steps due to SAC. The corresponding percentages are 49 and 51 % for IFRd and IFRt, respectively. If we add the total percentages in which the steps are accepted because of improvements in the objective (i.e., add SAC and SDCO), we have that the percentages are 76, 75, and 79 % for IBR, IFRd, and IFRt, respectively. This result indicates that both inertia-free strategies are competitive with inertia-based strategy in achieving productive steps for the objective value. Moreover, we recall that IFRd and IFRt yield better final objective values than IBR in 40 and 41 instances. Because of this, we cannot draw general conclusions on the tendency of the strategies to provide better final objective values. This can be attributed to the presence of multiple local minima.

We elaborate on the behavior of the strategies in instance `IEEE_162` which is a highly nonlinear optimal power flow problem. From Table 1 we can see that IFRt and IFRd do not require regularization and converge in 23 iterations whereas IBR requires

Table 1 Performance of inertia-based and inertia-free strategies for `IEEE_162` under different pivoting tolerances

Pivoting tolerance	Strategy	Iter	Reg
1×10^{-2}	IBR	138	182
	IFRd	23	0
	IFRt	23	0
1×10^{-4}	IBR	109	132
	IFRd	23	0
	IFRt	23	0
1×10^{-6}	IBR	178	236
	IFRd	23	0
	IFRt	23	0
1×10^{-8}	IBR	592	883
	IFRd	23	0
	IFRt	23	0

145 iterations and 183 regularizations. This instance is an ill-posed problem that does not seem to have an isolated local minimum (inertia is not correct at the solution). In this instance, significant regularization is observed for IBR during the entire search. This degrades the quality of the steps and results in slow convergence. On the other hand, from the behavior of IFR we can see that productive steps can be achieved without regularizing the system. We observed similar behavior in other instances (e.g., `static3`, `s368`, and `s389`). For the `IEEE_162` instance we performed an additional experiment in which we change the pivoting tolerance of MA57. In Table 1 we compare the performance of IFRd, IFRt and IBR. We note that the number of iterations and regularizations for IBR vary quite drastically as we change the pivoting tolerance. This is an indication that the inertia estimates provided by MA57 become unreliable. This parasitic behavior is confirmed when we compare against the performance of the inertia-free approaches, which require the same number of iterations in all cases. As can be seen, inertia-free strategies can be beneficial in solving ill-posed problems. In this respect, inertia-free strategies inherit some of the desirable features of trust-region settings.

We highlight that IFRt requires two backsolves to compute the search step (one for the normal component and one for the tangential component) while IFRd requires only one. The factorization of the augmented matrix is reused. The overhead of the additional backsolve is often one to two orders of magnitude smaller than the factorization overhead. We can see this from Table A1 if one considers large instances that do not require regularization and take the same number of iterations for IFRt and IFRd. For example, `cvxqp3` requires one more second for IFRt than for IFRd out of 371 seconds of total time. For `cvxqp2`, IFRt requires 12 more seconds than IFRd while the total time is 123 seconds.

5.3 Large-scale tests

We now demonstrate that the inertia-free strategies remain efficient in large-scale problems. We use large-scale stochastic optimization problems arising from security-constrained optimal power flow and stochastic optimal control of natural gas networks [13,42]. Because of the large dimensionality of these instances we solve them using a distributed-memory Schur decomposition strategy implemented in PIPS-NLP.

5.3.1 Structured NLPs and Schur decomposition

We are interested in solving NLPs with the following structure:

$$\min \quad f_0(x_0) + \sum_{\omega\in\Omega} f_\omega(x_\omega, x_0) \tag{33a}$$

$$\text{s.t.} \quad c_0(x_0) = 0 \qquad (\lambda_0) \tag{33b}$$

$$c_\omega(x_\omega, x_0) = 0,\ \ \omega \in \Omega \qquad (\lambda_\omega) \tag{33c}$$

$$x_0 \geq 0 \qquad (\nu_0) \tag{33d}$$

$$x_\omega \geq 0,\ \ \omega \in \Omega \qquad (\nu_\omega) \tag{33e}$$

The augmented matrix (4) of the structured NLP (33) can be permuted into the following block-bordered diagonal form:

$$\hat{M} = \begin{bmatrix} \hat{M}_1 & & & & B_1^T \\ & \hat{M}_2 & & & B_2^T \\ & & \ddots & & \vdots \\ & & & \hat{M}_{|\Omega|} & B_{|\Omega|}^T \\ B_1 & B_2 & \cdots & B_{|\Omega|} & \hat{M}_0 \end{bmatrix}, \tag{34}$$

where $\Omega := \{0, 1, 2, ..., |\Omega|\}$ is the scenario set [22,32,44]. The zero index corresponds to coupling variables. We refer to this coupling scenario as the zero scenario. We use $\hat{M}$ to denote the permuted form of the augmented matrix M_k. The matrices

$$\hat{M}_\omega = \begin{bmatrix} W_\omega & J_\omega^T \\ J_\omega & 0 \end{bmatrix} \tag{35}$$

for $\omega \in \Omega$ have a saddle-point structure. Here, W_ω and J_ω are the corresponding Hessian and Jacobian contributions of each scenario and the border matrices B_ω define coupling between scenarios and the zero scenario.

The permuted augmented system can be represented as

$$\hat{M}\hat{w} = \hat{r}, \tag{36}$$

where $\hat{w} = (\hat{w}_1, \ldots, \hat{w}_{|\Omega|}, \hat{w}_0)$ are the permuted search directions for primal variables and multipliers, respectively, and $\hat{r} = (\hat{r}_1, \ldots, \hat{r}_{|\Omega|}, \hat{r}_0)$ are the permuted right-hand sides. To solve the structured augmented system (36) in parallel, we use Schur decomposition. The solution of (36) can be obtained from

$$\hat{z}_\omega = \hat{M}_\omega^{-1}\hat{r}_\omega \quad \omega \in \Omega \setminus \{0\}, \tag{37a}$$

$$\hat{w}_0 = C(\delta)^{-1}\left(\hat{r}_0 - \sum_{\omega\in\Omega} B_\omega \hat{z}_\omega\right), \tag{37b}$$

$$\hat{w}_\omega = \hat{z}_\omega - \hat{M}_\omega^{-1} B_\omega^T \hat{w}_0, \quad \omega \in \Omega \setminus \{0\}, \tag{37c}$$

where

$$C = \hat{M}_0 - \sum_{\omega\in\Omega} B_\omega \hat{M}_\omega^{-1} B_\omega^T, \tag{38}$$

is the Schur complement. Each slave processor is allocated with the information of certain blocks ω, and performs step (37a) by factorizing the local blocks $\hat{M}_\omega$ in parallel. A master processor gathers the contributions of each worker to assemble the Schur complement in (38) and computes the stepsize for the coupling variables using (37b). Having the coupling step $\hat{w}_0$, the slave processors compute the local steps $\hat{w}_\omega$ in parallel using (37c).

Because an LBL^T factorization of the entire matrix $\hat{M}$ is not available, its inertia can be inferred by using Haynsworth's inertia additivity formula [27]:

$$\text{Inertia}(\hat{M}) = \text{Inertia}(C) + \sum_{\omega \in \Omega} \text{Inertia}(\hat{M}_\omega). \tag{39}$$

We perform the factorization of the subblocks and of the Schur complement and check that the addition of their inertias satisfies the inertia condition $\text{Inertia}(\hat{M}) = \{n, m, 0\}$. If this is not the case, all the Hessian terms W_ω are regularized by using a common parameter δ until $\hat{M}$ has the correct inertia. We note that obtaining the inertia of the Schur complement C by factorizing it with a sparse symmetric indefinite routine such as MA57 is not efficient because this matrix tends to be dense. More efficient parallel codes such as MAGMA or ELEMENTAL can be used but these are based on dense factorizations schemes that do not provide inertia information [1,31]. This situation illustrates a practical complication that can be encountered when inertia information is required and motivates the development of inertia-free strategies. In our implementation we use MA57 to factorize the Schur complement (even if this is not the most efficient choice) because we seek to compare the performance of IFR with that of IBR.

We note that the serial bottleneck of the Schur complement is associated to the formation and factorization of the Schur complement. Because the Schur complement must be re-assembled and factorized whenever the regularization term δ is adjusted, regularization can increase not only total work per iteration but also parallel performance.

5.3.2 Natural gas and power grid problems

The stochastic optimal control problem for gas networks has the form presented in (40). This problem determines compressor policies $\Delta\theta_{\omega,\ell}(\tau)$ up to a preparation time T_d that build up enough gas in the network to sustain demand profiles $d_j^{target}(\omega, \tau)$ under multiple scenarios $\omega \in \Omega$. The objective function is the expected cost of compression plus an error term between the actual delivered demands and the targets. The PDEs (40b)–(40d) are transport equations that describe flow, pressure, and temperature variations inside the pipelines comprising the network. These equations also capture the dynamic response of the gas stored inside the pipelines when gas is added/withdrawn at network nodes. The boundary conditions and the network constraints (40f)-(40i) couple the pipelines. The constraint (40j) is used to compute the power consumed by each compressor station and the constraints (40l) bound the discharge and suction pressures at network nodes. The constraint (40k) enforces periodicity in the average flow of the system. Constraint (40m) are the so-called non-anticipativity constraints that require the compressor policies to be the same for each scenario. For more details in the problem formulation the reader is referred to [42]. We discretize the PDEs in space and time using finite differences and implicit Euler schemes, respectively. After discretization, we obtain an NLP of the form (33) with 128 scenarios and that contains a total of 1,024,651 variables and 1,023,104 constraints. We refer to this problem instance as `STOCH_GAS`.

The security-constrained optimal power flow (SCOPF) problem has the form shown in (41). The SCOPF problem seeks to minimize the power generation cost under a base condition (zero scenario) while guaranteeing that the operation of the network is feasible under predetermined contingencies caused by the loss of different transmission lines. Here, Ω is the set of contingencies (scenarios) and each contingency $\omega \in \Omega$ gives rise to a different network topology (reflected in the sets of links $\mathcal{L}_\omega$). The voltage angle differences between nodes i and j are defined as $\delta_{\omega,ij} := \delta_{\omega,i} - \delta_{\omega,j}$. Constraints (41b)–(41e) are derived from Kirchhoff's laws, which establish the conservation of energy in the power network. Constraints (41b) and (41c) assert that the sum of incoming power flows and power generation at any particular node (bus) must be equal to the sum of outgoing power flow and power consumption. Constraints (41d) and (41e) are power balances at each node. Constraint (41f) forces the phase angle at a reference bus b_0 to be zero. Constraints (41g)–(41j) are physical limits of the system. Constraints (41l)–(41k) are non-anticipativity constraints and state that the voltage level and real power generation at the PV buses (control buses) should remain at their base condition. The only correction action after a failure occurs in the real power generation at the reference bus, which is used to refill the power transmission loss occurring in each contingency. We use the IEEE 300 bus system with 239 contingencies as case study (we refer to the resulting problem as IEEE_300). This gives a structured NLP of the form (1) with 878,650 variables and 734,406 constraints. The gas and power flow instances are accessible at http://zavalab.engr.wisc.edu/data/testinertia.

All parallel numerical tests were performed on the Fusion computing cluster at Argonne National Laboratory. Fusion contains 320 computing nodes, and each node has two quad-core Nehalem 2.6 GHz CPUs. The results are presented in Table 2. Here, #MPI denotes the number of MPI processes used for parallelization. All strategies converge to the same objective values; consequently, we report only one value. For these problem instances we have used parameter values $\alpha_t = 10^{-10}, \alpha_d = 10^{-10}$ (scaled by μ) because the default values of 10^{-12} resulted in high variability in the number of iterations for STOCH_GAS. We have observed that increasing α_t, α_d enhances efficiency. As expected, however, this comes at the expense of additional regularizations

$$\min\ \mathbb{E}\left[\int_0^T \left(\sum_{\ell\in\mathcal{L}_a} \alpha_\ell^P P_{\omega,\ell}(\tau) + \sum_{j\in\mathcal{D}} \alpha_j^d \left(d_{\omega,j}(\tau) - d_{\omega,j}^{target}(\tau)\right)^2\right) d\tau\right] \tag{40a}$$

s.t.

$$\frac{\partial p_{\omega,\ell}(x,\tau)}{\partial\tau} + \frac{ZRT_{\omega,\ell}(x,\tau)}{A_\ell}\frac{\partial f_{\omega,\ell}(x,\tau)}{\partial x} = 0,\ \ \ell\in\mathcal{L}, \omega\in\Omega, x\in\mathcal{X}_\ell, \tau\in\mathcal{T} \tag{40b}$$

$$\frac{1}{A_\ell}\frac{\partial f_{\omega,\ell}(x,\tau)}{\partial\tau} + \frac{\partial p_{\omega,\ell}(x,\tau)}{\partial x} + \frac{8\lambda_\ell}{\pi^2 D_\ell^5}\frac{f_{\omega,\ell}(x,\tau)|f_{\omega,\ell}(x,\tau)|}{\rho_{\omega,\ell}(x,\tau)} = 0,$$
$$\ell\in\mathcal{L}, \omega\in\Omega, x\in\mathcal{X}_\ell, \tau\in\mathcal{T} \tag{40c}$$

$$\rho_{\omega,\ell}(x,\tau)\, c_p \left(\frac{\partial T_{\omega,\ell}(x,\tau)}{\partial \tau} + \nu_{\omega,\ell}(x,\tau)\frac{\partial T_{\omega,\ell}(x,\tau)}{\partial x}\right) - \left(\frac{\partial p_{\omega,\ell}(x,\tau)}{\partial \tau} + \nu_{\omega,\ell}(x,\tau)\frac{\partial p_{\omega,\ell}(x,\tau)}{\partial x}\right) + \frac{\pi D_\ell U_\ell}{A_\ell}\left(T_{\omega,\ell}(x,\tau) - T_\omega^{amb}(x,\tau)\right) = 0,\ \ell \in \mathcal{L}, \omega \in \Omega, x \in \mathcal{X}_\ell, \tau \in \mathcal{T} \tag{40d}$$

$$\frac{p_{\omega,\ell}(x,\tau)}{\rho_{\omega,\ell}(x,\tau)} = ZRT_{\omega,\ell}(x,\tau),\ \ell \in \mathcal{L}, \omega \in \Omega, x \in \mathcal{X}_\ell, \tau \in \mathcal{T} \tag{40e}$$

$$p_{\omega,\ell}(L_\ell,\tau) = \theta_{\omega,rec(\ell)}(\tau),\ \ell \in \mathcal{L}, \omega \in \Omega, \tau \in \mathcal{T} \tag{40f}$$

$$p_{\omega,\ell}(0,\tau) = \theta_{\omega,snd(\ell)}(\tau),\ \ell \in \mathcal{L}_p, \omega \in \Omega, \tau \in \mathcal{T} \tag{40g}$$

$$p_{\omega,\ell}(0,\tau) = \theta_{\omega,snd(\ell)}(\tau) + \Delta\theta_{\omega,\ell}(\tau),\ \ell \in \mathcal{L}_a, \omega \in \Omega, \tau \in \mathcal{T} \tag{40h}$$

$$\sum_{\ell \in \mathcal{L}_n^{rec}} f_{\omega,\ell}(L_\ell,\tau) - \sum_{\ell \in \mathcal{L}_n^{snd}} f_{\omega,\ell}(0,\tau) + \sum_{i \in \mathcal{S}_n} s_{\omega,i}(\tau) - \sum_{j \in \mathcal{D}_n} d_{\omega,j}(\tau) = 0,\ n \in \mathcal{N}, \omega \in \Omega, \tau \in \mathcal{T} \tag{40i}$$

$$P_{\omega,\ell}(\tau) = c_p \cdot T_{\omega,\ell}(0,\tau) \cdot f_{\omega,\ell}(0,\tau)\left(\left(\frac{\theta_{\omega,snd(\ell)}(\tau) + \Delta\theta_{\omega,\ell}(\tau)}{\theta_{\omega,snd(\ell)}(\tau)}\right)^{\frac{\gamma-1}{\gamma}} - 1\right),\quad \ell \in \mathcal{L}_a, \omega \in \Omega, \tau \in \mathcal{T} \tag{40j}$$

$$\int_0^{L_\ell} f_{\omega,\ell}(x,T)dx \geq \int_0^{L_\ell} f_{\omega,\ell}(x,0)dx,\quad \ell \in \mathcal{L}_a, \omega \in \Omega \tag{40k}$$

$$\underline{\theta}_n \leq \theta_{\omega,n}(\tau) \leq \overline{\theta}_n,\ n \in \mathcal{N}, \omega \in \Omega, \tau \in \mathcal{T} \tag{40l}$$

$$\Delta\theta_{\omega,\ell}(\tau) = \Delta\theta_{0,\ell}(\tau),\ \ell \in \mathcal{L}_a,\ \omega \in \Omega \setminus \{0\}, \tau \in [0, T_d]. \tag{40m}$$

We can see that, despite the high nonlinearity of the large-scale instances, the inertia-free approaches IFRd and IFRt converge in all cases. This provides evidence that the tests can scale. In general, IFRd and IFRt require more iterations than IBR does but the number of factorizations is reduced, resulting in faster solutions. These problem instances are highly ill-conditioned, particularly STOCH_GAS as is evident from the variability of the number of iterations as we increase the number of MPI processes. This is the result of linear system errors introduced by Schur decomposition. The performance, however, is satisfactory in all cases. For IEEE_300 we note that the number of iterations for IBR does not vary as we add MPI processors whereas those of IFRd and IFRt do. While it is difficult to isolate a specific source of such behavior, we attribute this behavior to the stabilizing effect that additional regularizations of IBR provide on the linear system.

$$\min \sum_{g \in G} \left(c_{0g} + c_{1g} p_{base,g} + c_{2g} p_{base,g}^2\right) \tag{41a}$$

s.t.

$$\sum_{g | o_g = b} p_{\omega,g} - d_b^P = \sum_{(i,j) \in \mathcal{L}_\omega} f_{\omega,(i,j)}^P, \; b \in \mathcal{B}, \omega \in \Omega \tag{41b}$$

$$\sum_{g | o_g = b} q_{\omega,g} - d_b^Q = \sum_{(i,j) \in \mathcal{L}_\omega} f_{\omega,(i,j)}^Q - \frac{(V_b)^2}{2} \sum_{(i,j) \in \mathcal{L}_\omega} \gamma_{(i,j)}, \; b \in \mathcal{B}, \omega \in \Omega \tag{41c}$$

$$f_{\omega,(i,j)}^P = \begin{cases} \alpha_l V_{\omega,i}^2 - V_{\omega,i} V_{\omega,j} [\alpha_l \cos(\delta_{\omega,ij}) \\ \quad + \beta_l \sin(\delta_{\omega,ij})], & \text{otherwise} \\ \alpha_l \left(\frac{V_{\omega,i}}{\tau_t}\right)^2 - \frac{V_{\omega,i} V_{\omega,j}}{\tau_t} \\ \quad [\alpha_t \cos(\delta_{\omega,ij}) + \beta_t \sin(\delta_{\omega,ij})], & i \text{ tapped} \\ \alpha_l V_{\omega,i}^2 - \frac{V_{\omega,i} V_{\omega,j}}{\tau_t} [\alpha_t \cos(\delta_{\omega,ij}) \\ \quad + \beta_t \sin(\delta_{\omega,ij})], & j \text{ tapped} \end{cases}, \; (i,j) \in \mathcal{L}_\omega, \omega \in \Omega \tag{41d}$$

$$f_{\omega,(i,j)}^Q = \begin{cases} -\beta_l V_{\omega,i}^2 - V_{\omega,i} V_{\omega,j} [\alpha_l \sin(\delta_{\omega,ij}) \\ \quad - \beta_l \cos(\delta_{\omega,ij})], & \text{otherwise} \\ -\beta_l \left(\frac{V_{\omega,i}}{\tau_t}\right)^2 - \frac{V_{\omega,i} V_{\omega,j}}{\tau_t} \\ \quad [\alpha_l \sin(\delta_{\omega,ij}) - \beta_l \cos(\delta_{\omega,ij})], & i \text{ tapped} \\ -\beta_l V_{\omega,i}^2 - \frac{V_{\omega,i} V_{\omega,j}}{\tau_t} [\alpha_l \sin(\delta_{\omega,ij}) \\ \quad - \beta_l \cos(\delta_{\omega,ij})], & j \text{ tapped} \end{cases}, \; (i,j) \in \mathcal{L}_\omega, \omega \in \Omega \tag{41e}$$

$$\delta_{\omega,b_0} = 0, \omega \in \Omega \tag{41f}$$

$$(f_{\omega,(i,j)}^P)^2 + (f_{\omega,(i,j)}^Q)^2 \leq (f_{\omega,(i,j)}^+)^2, \; (i,j) \in \mathcal{L}, \omega \in \Omega \tag{41g}$$

$$p_{\omega,g}^- \leq p_{\omega,g} \leq p_{\omega,g}^+, \; g \in \mathcal{G}, \omega \in \Omega \tag{41h}$$

$$q_{\omega,g}^- \leq q_{\omega,g} \leq q_{\omega,g}^+, \; g \in \mathcal{G}, \omega \in \Omega \tag{41i}$$

$$V_{\omega,b}^- \leq V_{\omega,b} \leq V_{\omega,b}^+, \; b \in \mathcal{B}, \omega \in \Omega \tag{41j}$$

$$V_{\omega,b} = V_{base,b}, \; b \in \mathcal{B}_{\mathcal{PV}}, \omega \in \Omega \setminus \{base\} \tag{41k}$$

$$p_{\omega,g} = p_{base,g}, \; g \in \{g | o_g \in \mathcal{B}_{\mathcal{PV}} \setminus \{b_0\}\}. \tag{41l}$$

Table 2 Performance of inertia-based and inertia-free strategies on large-scale instances

Problem	#MPI	IBR				IFRd			IFRt		
		Obj	Iter	Fact	Time(s)	Iter	Fact	Time(s)	Iter	Fact	Time(s)
IEEE_300	16	1.36E+03	112	274	627	173	190	476	209	241	651
IEEE_300	24	1.36E+03	112	274	424	208	238	403	232	255	475
IEEE_300	40	1.36E+03	112	274	263	160	169	181	168	178	206
IEEE_300	80	1.36E+03	112	274	174	183	203	119	177	190	127
IEEE_300	120	1.36E+03	112	274	126	192	224	91	201	227	103
IEEE_300	240	1.36E+03	113	274	65	170	185	47	219	240	80
STOCH_GAS	8	1.26E−02	153	278	832	122	144	621	93	106	491
STOCH_GAS	16	1.26E−02	136	251	363	245	277	789	109	122	315
STOCH_GAS	32	1.26E−02	146	274	209	211	250	301	99	112	143
STOCH_GAS	64	1.26E−02	157	286	123	112	137	74	101	114	79
STOCH_GAS	128	1.26E−02	145	275	64	127	158	52	109	125	52

6 Conclusions

We have presented a new filter line-search algorithm that does not require inertia information. This inertia-free approach performs curvature tests along computed directions to guarantee descent when the constraint violation is sufficiently small. We proved that the approach yields global convergence and is competitive with the standard inertia-based strategy based on symmetric indefinite factorizations. Moreover, we demonstrate that the inertia-free approach can significantly reduce the amount of regularization needed. The availability of inertia-free strategies opens the possibility of using different types of linear algebra strategies and libraries and thus can enhance modularity of implementations. This was demonstrated through a distributed-memory Schur decomposition setting.

Acknowledgments This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Applied Mathematics program under Contract No. DE-AC02-06CH11357. We thank Frank Curtis and Jorge Nocedal for technical discussions. Victor M. Zavala acknowledges funding from the DOE Office of Science under the Early Career program. We also acknowledge the computing resources provided by the Laboratory Computing Resource Center at Argonne National Laboratory.

References

1. Agullo, E., Demmel, J., Dongarra, J., Hadri, B., Kurzak, J., Langou, J., Ltaief, H., Luszczek, P., Tomov, S.: Numerical linear algebra on emerging architectures: the PLASMA and MAGMA projects. J. Phys. Conf. Ser. **180**, 012037 (2009)
2. Amestoy, P.R., Guermouche, A., LExcellent, J.Y., Pralet, S.: Hybrid scheduling for the parallel solution of linear systems. Parallel Comput. **32**(2), 136–156 (2006)
3. Arora, Nikhil, Biegler, Lorenz T.: A trust region SQP algorithm for equality constrained parameter estimation with simple parameter bounds. Comput. Optim. Appl. **28**(1), 51–86 (2004)

4. Balay, S., Brown, J., Buschelman, K., Eijkhout, V., Gropp, W., Kaushik, D., Knepley, M., McInnes, L., Smith, B., Zhang, H.: PETSc Users Manual Revision 3.4 (2013)
5. Benzi, M., Golub, G.H., Liesen, J.: Numerical solution of saddle point problems. Acta Numer. **14**, 1–137 (2005)
6. Biegler, L.T., Zavala, V.M.: Large-scale nonlinear programming using IPOPT: an integrating framework for enterprise-wide dynamic optimization. Comput. Chem. Eng. **33**(3), 575–582 (2009)
7. Biros, G., Ghattas, O.: Parallel Lagrange–Newton–Krylov–Schur methods for PDE-constrained optimization. Part I: The Krylov–Schur solver. SIAM J. Sci. Comput. **27**(2), 687–713 (2005)
8. Borzì, A., Schulz, V.: Multigrid methods for PDE optimization. SIAM Rev. **51**(2), 361–395 (2009)
9. Bunch, J.R., Kaufman, L.: Some stable methods for calculating inertia and solving symmetric linear systems. Math. Comput. **31**, 163–179 (1977)
10. Byrd, R.H., Gilbert, J.C., Nocedal, J.: A trust-region method based on interior-point techniques for nonlinear programming. Math. Program. **89**, 149–185 (2000)
11. Byrd, R.H., Curtis, F.E., Nocedal, J.: An inexact Newton method for nonconvex equality constrained optimization. Math. Program. **122**(2), 273–299 (2010)
12. Cervantes, A.M., Wächter, A., Tütüncü, R.H., Biegler, L.T.: A reduced space interior point strategy for optimization of differential algebraic systems. Comput. Chem. Eng. **24**(1), 39–51 (2000)
13. Chiang, N., Grothey, A.: Solving security constrained optimal power flow problems by a structure exploiting interior point method. Optim. Eng. **16**, 49–71 (2012)
14. Chiang, N., Petra, C.G., Zavala, V.M.: Structured nonconvex optimization of large-scale energy systems using PIPS-NLP. In: Proceedings of the 18th Power Systems Computation Conference (PSCC). Wroclaw, Poland (2014)
15. Conn, A.R., Gould, N.I.M., Toint, P.L.: Trust Region Methods, vol. 1. SIAM, Philadelphia (2000)
16. Costa, M.P., Fernandes, E.M.G.P.: Assessing the potential of interior point barrier filter line search methods: nonmonotone versus monotone approach. Optimization **60**(10–11), 1251–1268 (2011)
17. Curtis, F.E., Nocedal, J., Wächter, A.: A matrix-free algorithm for equality constrained optimization problems with rank-deficient Jacobians. SIAM J. Optim. **20**(3), 1224–1249 (2009)
18. Curtis, F.E., Schenk, O., Wächter, A.: An interior-point algorithm for large-scale nonlinear optimization with inexact step computations. SIAM J. Sci. Comput. **32**(6), 3447–3475 (2010)
19. Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. Math. Program. **91**(2), 201–213 (2002)
20. Duff, I.S.: Ma57—a code for the solution of sparse symmetric definite and indefinite systems. ACM Trans. Math. Softw. **30**, 118–144 (2004)
21. Duff, I.S., Reid, J.K.: MA27—A Set of Fortran Subroutines for Solving Sparse Symmetric Sets of Linear Equations. UKAEA Atomic Energy Research Establishment (1982)
22. Gondzio, J., Sarkissian, R.: Parallel interior-point solver for structured linear programs. Math. Program. **96**(3), 561–584 (2003)
23. Gould, N.I.M., Hribar, M.E., Nocedal, J.: On the solution of equality constrained quadratic programming problems arising in optimization. SIAM J. Sci. Comput. **23**(4), 1376–1395 (2001)
24. Gould, N.I.M., Lucidi, S., Roma, M., Toint, P.L.: Solving the trust-region subproblem using the Lanczos method. SIAM J. Optim. **9**(2), 504–525 (1999)
25. Gould, N.I.M.: On practical conditions for the existence and uniqueness of solutions to the general equality quadratic programming problem. Math. Program. **32**(1), 90–99 (1985)
26. Haverbeke, N., Diehl, M., De Moor, B.: A structure exploiting interior-point method for moving horizon estimation. In: Proceedings of the 48th IEEE Conference on Decision and Control, pp. 1273–1278. IEEE (2009)
27. Haynsworth, E.V.: Determination of the inertia of a partitioned Hermitian matrix. Linear Algebra Appl. **1**(1), 73–81 (1968)
28. Heinkenschloss, M., Ridzal, D.: A matrix-free trust-region SQP method for equality constrained optimization. SIAM J. Optim. **24**(3), 1507–1541 (2014)
29. Heroux, M.A., Willenbring, J.M.: Trilinos Users Guide. Citeseer (2003)
30. Kawajiri, Y., Biegler, L.T.: Optimization strategies for simulated moving bed and powerfeed processes. AIChE J. **52**(4), 1343–1350 (2006)
31. Lubin, M., Petra, C.G., Anitescu, M.: The parallel solution of dense saddle-point linear systems arising in stochastic programming. Optim. Methods Softw. **27**(4–5), 845–864 (2012)

32. Lubin, M., Petra, C.G., Anitescu, M., Zavala, V.M.: Scalable stochastic optimization of complex energy systems. In: International Conference for High Performance Computing, Networking, Storage and Analysis (SC), 2011, pp. 1–10. IEEE (2011)
33. Petra, C.G., Schenk, O., Lubin, M., Gaertner, K.: An augmented incomplete factorization approach for computing the Schur complement in stochastic optimization. SIAM J. Sci. Comput. **36**(2), C139–C162 (2014)
34. Rao, C.V., Wright, S.J., Rawlings, J.B.: Application of interior-point methods to model predictive control. J. Optim. Theory Appl. **99**(3), 723–757 (1998)
35. Schenk, O., Wächter, A., Hagemann, M.: Matching-based preprocessing algorithms to the solution of saddle-point problems in large-scale nonconvex interior-point optimization. Comput. Optim. Appl. **36**, 321–341 (2007)
36. Schenk, O., Wächter, A., Weiser, M.: Inertia-revealing preconditioning for large-scale nonconvex constrained optimization. SIAM J. Sci. Comput. **31**(2), 939–960 (2008)
37. Soler, M., Olivares, A., Staffetti, E.: Hybrid optimal control approach to commercial aircraft trajectory planning. J. Guid. Control Dyn. **33**(3), 985–991 (2010)
38. Wächter, A., Biegler, L.T.: On the implementation of a primal–dual interior point filter line search algorithm for large-scale nonlinear programming. Math. Program. **106**, 25–57 (2006)
39. Wächter, A., Biegler, L.T.: Line search filter methods for nonlinear programming: motivation and global convergence. SIAM J. Optim. **16**(1), 1–31 (2005)
40. Waltz, R.A., Morales, J.L., Nocedal, J., Orban, D.: An interior algorithm for nonlinear optimization that combines line search and trust region steps. Math. Program. **107**(3), 391–408 (2006)
41. Wang, Y., Boyd, S.: Fast model predictive control using online optimization. IEEE Trans. Control Syst. Technol. **18**(2), 267–278 (2010)
42. Zavala, V.M.: Stochastic optimal control model for natural gas networks. Comput. Chem. Eng. **64**, 103–113 (2014)
43. Zavala, V.M., Biegler, L.T.: Large-scale parameter estimation in low-density polyethylene tubular reactors. Ind. Eng. Chem. Res. **45**(23), 7867–7881 (2006)
44. Zavala, V.M., Laird, C.D., Biegler, L.T.: Interior-point decomposition approaches for parallel solution of large-scale nonlinear parameter estimation problems. Chem. Eng. Sci. **63**(19), 4834–4845 (2008)
45. Zenios, S., Lasken, R.: Nonlinear network optimization on a massively parallel connection machine. Ann. Oper. Res. **14**(1), 147–165 (1988)