AN INERTIA-FREE FILTER LINE-SEARCH ALGORITHM FOR LARGE-SCALE NONLINEAR PROGRAMMING*

NAIYUAN CHIANG AND VICTOR M. ZAVALA†

Abstract. We present a filter line-search algorithm that does not require inertia information about the linear system to ensure global convergence. The proposed approach performs curvature tests along the search step to ensure descent. This feature permits more modularity in the linear algebra, enabling the use of a wider range of iterative and decomposition strategies. We use the inertia-free approach in an interior-point framework and provide numerical evidence that this is as robust as inertia detection via symmetric indefinite factorizations and can lead to important reductions in solution time.

Key words. inertia, nonlinear programming, filter line-search, nonconvex, large-scale

AMS subject classifications. 34B15, 34H05, 49N35, 49N90, 90C06, 90C30, 90C55, 90C59

1. Introduction.

Filter line-search strategies have proven effective at solving large-scale nonconvex nonlinear programs (NLPs) [34, 14, 11, 33, 6, 25]. These strategies monitor the inertia of the linear system to detect negative curvature and regularize the linear system when such is present. This approach ensures that the computed directions are of descent when the constraint violation is sufficiently small, a necessary condition for global convergence. The ability to handle negative curvature is also essential from a practical standpoint in order to deal with highly nonlinear and inherently ill-conditioned problems [39]. Inertia information is provided by symmetric indefinite factorization routines such as MA27, MA57, MUMPS, and Pardiso [19, 18, 2, 31]. An inertia-revealing preconditioning strategy based on incomplete factorizations has also been proposed that enables the use of iterative linear strategies [32]. Unfortunately, many other linear algebra strategies and libraries cannot be used because they do not provide inertia information. Examples include iterative techniques such as multigrid, Lagrange-Newton-Krylov, and inexact constraint preconditioning [8, 7, 5]; parallel solvers for graphics processing units (GPUs) and distributed-memory systems [1, 3]; and decomposition strategies for stochastic optimization, optimal control, and network problems widely used in convex optimization [20, 27, 29, 36, 30, 42, 23]. Consequently, although inertia information is key to enabling robust performance, it can also hinder modularity, application scope, and scalability.

Byrd, Curtis, and Nocedal recently proposed a line-search exact penalty framework that does not require inertia information [10]. 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 [15]. The strategy has also proven to be effective when used within an interior-point framework [16].

In this work, we present an inertia-free filter line-search strategy for nonconvex NLPs. The approach tests for curvature along the tangential component of the search step. We prove that global convergence can be guaranteed if the step satisfies this test and if the iteration matrix is nonsingular. These requirements are less restrictive than the standard positive definiteness assumption for the reduced Hessian used in existing implementations. We also present an inertia-free alternative that performs a curvature test directly on the full search step, and we prove that global convergence can also be guaranteed. We implement the inertia-free strategies

*Preprint ANL/MCS-P5197-0914

†Mathematics and Computer Science Division, Argonne National Laboratory, 9700 South Cass Avenue, Argonne, IL 60439, USA (nychiang@mcs.anl.gov, vzavala@mcs.anl.gov)

in an interior-point framework and perform small- and large-scale numerical experiments to demonstrate that these are as robust as inertia detection based on symmetric indefinite LBL^T factorizations in terms of iteration counts. In addition, we demonstrate that the inertia-free strategies can significant reduce the number of trial factorizations due to increased flexibility in step acceptance.

The paper is structured as follows. Section 2 presents the filter line-search algorithm of Wächter and Biegler [34, 35] in an interior-point framework and discusses regularity and inertia assumptions needed to guarantee global convergence. Section 3 presents the new inertia-free strategies and establishes global convergence. Section 4 compares the numerical performance of both strategies. Section 5 presents concluding remarks.

2. Interior-Point Framework. Consider the NLP of the form

$$\min \quad f(x) \tag{2.1a}$$
$$\text{s.t.}\ c(x) = 0 \tag{2.1b}$$
$$x \geq 0 \tag{2.1c}$$

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

$$\begin{aligned} \min \quad & \varphi^\mu(x) := f(x) - \mu \sum_{j=1}^{n} \ln x^{(j)} \\ \text{s.t.} \quad & c(x) = 0 \end{aligned} \tag{2.2}$$

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 (2.2) and drives the barrier parameter μ monotonically to zero. For adaptive barrier strategies, see [28]. To solve each barrier problem, we apply Newton's method to the optimality conditions:

$$\nabla_x \varphi^\mu(x) + \nabla_x c(x)\lambda = 0 \tag{2.3a}$$
$$c(x) = 0 \tag{2.3b}$$
$$x \geq 0. \tag{2.3c}$$

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

$$\begin{bmatrix} W_k(\delta) & J_k^T \\ J_k & \end{bmatrix} \begin{bmatrix} d_k \\ \lambda_k^+ \end{bmatrix} = - \begin{bmatrix} g_k \\ c_k \end{bmatrix}. \tag{2.4}$$

We refer to this system as the Karush-Kuhn-Tucker (KKT) system. Here, $c_k := c(x_k)$, $J_k := \nabla_x c(x_k)^T \in \Re^{m\times n}$, $g_k := \nabla_x \varphi_k^\mu$, $W_k(\delta) := H_k + \Sigma_k + \delta I$ with scalar $\delta > 0$ and I is the identity matrix, $H_k := \nabla_{xx}\mathcal{L}(x_k, \lambda_k) \in \Re^{n\times n}$, $\mathcal{L}(x_k, \lambda_k) := \varphi^\mu(x_k) + \lambda_k^T c(x_k)$, and $\Sigma_k := X_k^{-2}$. One can show that the 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 (2.1c), can be used as long as the products $x_k^{(j)} \nu_k^{(j)}$ remain bounded by μ [35, 16]. To enable compact notation, we define the matrix

$$M_k(\delta) := \begin{bmatrix} W_k(\delta) & J_k^T \\ J_k & \end{bmatrix}, \tag{2.5}$$

and we refer to this as the KKT matrix. One can compute the search directions $(d_k, \lambda_k^+ - \lambda_k)$ using the decomposition

$$d_k = n_k + t_k. \tag{2.6}$$

Here, n_k satisfies $J_k n_k = -c_k$, and t_k is computed from

$$\begin{bmatrix} W_k(\delta) & J_k^T \\ J_k & \end{bmatrix} \begin{bmatrix} t_k \\ \lambda_k^+ \end{bmatrix} = - \begin{bmatrix} g_k + W_k(\delta) n_k \\ 0 \end{bmatrix}. \tag{2.7}$$

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

$$\mathcal{F}_0 := \{(\theta, \varphi) \mid \theta \geq \theta^{max}\} \tag{2.8}$$

with a given parameter $\theta^{max} > 0$. Given a search step d_k, a line search is started from counter $l \leftarrow 0$ and $\alpha_{k,0} = \alpha_k^{max} \leq 1$ to define trial iterates $x_k(\alpha_{k,l}) := x_k + \alpha_{k,l} d_k$. We consider the following conditions to check whether a trial iterate should be accepted.

- *Filter Condition FC:*

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

- *Switching Condition SC:*

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

- *Armijo Condition AC:*

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

- *Sufficient Decrease Condition SDC:*

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

Here, $\kappa_\theta > 0$, $s_\theta > 1$, $s_\varphi \geq 1$, and $\eta_\varphi \in (0,1)$ are given constants and

$$m_k(\alpha) := \alpha g_k^T d_k \tag{2.9}$$

is a linear model of $\varphi(x_k + \alpha d_k) - \varphi(x_k)$.

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

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

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

$$\mathcal{F}_{k+1} \leftarrow \mathcal{F}_k \cup \{(\theta, \varphi) \mid \varphi \geq \varphi(x_k) - \gamma_\varphi \theta(x_k), \quad \theta \geq (1-\gamma_\theta)\theta(x_k)\} \tag{2.10}$$

with parameters $\gamma_\varphi, \gamma_\theta \in (0,1)$; otherwise, we leave the filter unchanged (i.e., $\mathcal{F}_{k+1} \leftarrow \mathcal{F}_k$). If the trial step size $\alpha_{k,l}$ becomes smaller than α_k^{min} and the step has not been accepted in either case, then we revert to feasibility restoration, and the filter is augmented. A strategy to obtain α_k^{min} is proposed in [35]. 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, constants $\theta_{max} \in (\theta(x_0), \infty]$, γ_θ, $\gamma_\varphi \in (0,1)$, $\eta_\varphi \in (0,1)$, $\kappa_\theta > 0$, $s_\theta > 1$, $s_\varphi \geq 1$, $\eta_\varphi \in (0,1)$, and $0 < \tau_2 \leq \tau_1 < 1$.
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.
3. **Compute Search Direction.** Compute step d_k.
4. **Backtracking Line-Search.**
   4.1. **Initialize**. Set $\alpha_{k,0} \leftarrow \alpha_k^{max}$ and counter $\ell \leftarrow 0$.
   4.2. **Compute Trial Point**. If $\alpha_{k,\ell} \leq \alpha_k^{min}$ revert to feasibility restoration in Step 8. Otherwise, set trial point $x_k(\alpha_{k,\ell}) \leftarrow x_k + \alpha_{k,\ell} d_k$.
   4.3. **Check Acceptability to the Filter**. If FC does not hold, reject trial point $x_k(\alpha_{k,\ell})$, and go to Step 4.5.
   4.4. **Check Sufficient Progress**.
      4.4.1. If SC and AC hold, accept trial point $x_k(\alpha_{k,\ell})$ and go to Step 5.
      4.4.2. 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 4.5.
   4.5. **New Trial Step Size**. 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 4.2.
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 (2.10). Otherwise, leave filter unchanged.
7. **Next Iteration**. Increase iteration counter $k \leftarrow k+1$ and go to Step 3.
8. **Feasibility Restoration**. Compute an iterate x_{k+1} that satisfies FC and SDC. Augment filter using (2.10), and go to Step 7.

The global convergence analysis of the filter line-search algorithm provided in [35] assumes a step decomposition of the form

$$d_k = Y_k \bar{q}_k + Z_k \bar{p}_k \tag{2.11a}$$

$$\bar{q}_k := -(J_k Y_k)^{-1} c_k \tag{2.11b}$$

$$\bar{p}_k := (Z_k^T W_k(\delta) Z_k)^{-1} Z_k^T (g_k + W_k(\delta) Y_k \bar{q}_k), \tag{2.11c}$$

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., $J_k Z_k = 0$). The analysis also relies on the criticality measure

$$\chi_k := \|Z_k \bar{p}_k\| = \|\bar{p}_k\|, \tag{2.12}$$

where the last identity follows from the orthonormality of Z_k (i.e., $Z_k^T Z_k = I$). The analysis requires assumptions (G) [35, p. 10]. These require the reduced Hessian $Z_k^T W_k(\delta) Z_k$ to be positive definite (G3) and the Jacobian to have full row rank (G4). These are needed to guarantee that the criticality measure χ_k is well defined in the sense that as the measure converges to zero 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^*. Under a full rank assumption of the Jacobian (G4) and (2.11b) we have that $\lim_{i\to\infty}\bar{q}_{k_i} = 0$ as $\lim_{i\to\infty} x_{k_i} = x^*$. From $\lim_{i\to\infty}\chi_{k_i}$, (2.11c), and (2.12); and because (G3) guarantees nonsingularity of the reduced Hessian we have that $\lim_{i\to\infty}\|Z_{k_i}^T g_{k_i}\| = 0$.

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 [35]). As we will discuss in Section 3, this *descent lemma* is essential in order to establish global convergence.

Positive definiteness of the reduced Hessian (G3) can be ensured in a practical setting by monitoring the inertia of the KKT matrix $M(\delta)$ and correcting it (if necessary) by using the regularization parameter δ. This approach is justified from the results of Gould [21] that prove that the reduced Hessian is positive definite if and only if $M(\delta)$ has n positive, m negative, and no zero eigenvalues. We state this condition formally as

$$\text{Inertia}(M(\delta)) = \{n, m, 0\}. \tag{2.13}$$

The KKT matrix $M(\delta)$ can be decomposed as LBL^T by using symmetric indefinite linear solvers, where L is a nonsingular lower triangular matrix and B is a matrix composed of 1×1 and 2×2 diagonal blocks. By Sylvester's law of inertia we also know that the eigenvalues of $M(\delta)$ are the eigenvalues of B. Furthermore, because each 2×2 block is constructed by having one positive and one negative eigenvalue, the inertia of $M(\delta)$ can easily be estimated from the inertia of B [9].

A basic inertia-based regularization strategy is as follows. The matrix $M(\delta)$ is decomposed as LBL^T for $\delta = 0$ and the step d_k is computed using this decomposition. If the inertia is correct (i.e., condition (2.13) is satisfied), the step d_k is accepted. If not, the regularization parameter δ is increased and a new step d_k is obtained. The procedure is repeated until the matrix $M(\delta)$ 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. The inertia correction strategy implemented in the current version of IPOPT [34] is shown below.

Inertia-Based Regularization (IBR)

IBR-1 Factorize $M(\delta)$ with $\delta = 0$. If (2.13) holds, compute d_k and stop.
IBR-2 If $\delta^{last} = 0$, set $\delta \leftarrow \bar{\delta}^0$, otherwise set $\delta \leftarrow \max\{\delta^{min}, \kappa^- \delta^{last}\}$.
IBR-3 Factorize $M(\delta)$ with current δ. If (2.13) 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, $0 < \bar{\delta}_{min} < \bar{\delta}^0 < \bar{\delta}_{max}$, $0 < \kappa^- < 1 < \kappa^+ < \hat{\kappa}^+$ are given constants.

3. Inertia-Free Strategies. Estimating the inertia of $M(\delta)$ 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 (2.13) is based solely on the structural properties of $M(\delta)$ 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 (2.6), KKT system (2.7), and the criticality measure

$$\Psi_k^t = \|t_k\|. \tag{3.1}$$

As argued in Section 3.1, a step decomposition is not strictly necessary, but it is advantageous for the analysis and can be used to enable the use of projected conjugate gradient strategies [22].

For our analysis we use the following assumptions.

Assumptions (RG)

(RG1) There exists an open set $\mathcal{C} \subseteq \Re^n$ with $[x_k, x_k + d_k] \subseteq \mathcal{C}$ 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 $W_k(\delta)$ are uniformly bounded for all $k \notin \mathcal{R}_{inc}$.

(RG3) There exists $\alpha_t > 0$ such that for all $k \notin \mathcal{R}_{inc}$

$$t_k^T W_k(\delta) t_k + \max\{t_k^T W_k(\delta) n_k - g_k^T n_k, 0\} \geq \alpha_t t_k^T t_k \tag{3.2}$$

with t_k computed from (2.7) (RG3a). Furthermore, $M_k(\delta)$ is nonsingular (RG3b).

(RG4) The Jacobian J_k has full row rank for all $k \notin \mathcal{R}_{inc}$.

(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) in [35] is that we do not require positive definiteness of the reduced Hessian (G3). This requirement is replaced by our assumption (RG3a), which requires $W_k(\delta)$ to have positive curvature along the tangential direction t_k and by (RG3b), which requires nonsingularity of the KKT matrix $M_k(\delta)$. Our assumptions are therefore less restrictive.

Assumption (RG3a) can always be satisfied for sufficiently large δ and for sufficiently small α_t such that $\alpha_t \leq \lambda_{min}(Z_k^T W_k(\delta) Z_k)$. 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 $u^T Z_k^T W_k(\delta) Z_k u \geq \alpha_t u^T u$. 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 $(\max\{t_k^T W_k(\delta) n_k - g_k^T n_k, 0\})$ does not affect these properties. This is because if the argument is negative, then this is set to zero and if argument is positive, it provides additional flexibility to satisfy the curvature test.

We now prove that conditions (RG3b) and (RG4) guarantee that the reduced Hessian is nonsingular and that the proposed criticality measure Ψ_k^t is well defined.

LEMMA 3.1. *Let (RG3b) and (RG4) hold for fixed δ and define $M = M_k(\delta)$, $W = W_k(\delta)$, $Z = Z_k$, and $J = J_k$. Then (i) the reduced Hessian $Z^T W Z$ is nonsingular and (ii) the inverse of M exists and can be expressed as*

$$M_{inv} = \begin{bmatrix} W & J^T \\ J & \end{bmatrix}^{-1} = \begin{bmatrix} P & (I - PW)J^T V^{-1} \\ V^{-1} J (I - WP) & -V^{-1} J (W - WPW) J^T V^{-1} \end{bmatrix} \tag{3.3}$$

with

$$P = Z(Z^T W Z)^{-1} Z^T \tag{3.4}$$

and $V = JJ^T$.

Proof: Part (i) follows from Lemmas 3.2 and 3.4 in [21]. These results establish that if the Jacobian has full row rank (RG4) there exists a nonsingular matrix R such that

$$R^T M R = \begin{bmatrix} & & I \\ & Z^T W Z & \\ I & & \end{bmatrix}. \tag{3.5}$$

From Sylverster's law of inertia and the structure of $R^T M R$ we have that Inertia(M) = Inertia$(Z^T W Z) + \{m, m, 0\}$. Consequently, the number of zero eigenvalues of M is equal to the number of zero eigenvalues of $Z^T W Z$. Because M is nonsingular (RG3b), 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 noticing that $J^T (JJ^T)^{-1} J = I - ZZ^T$ [4, p. 20] and $JZ = 0$, we obtain

$$\begin{aligned}
A_{11} &= WP + J^T V^{-1} J(I - WP) \\
&= WP + (I - ZZ^T)(I - WP) \\
&= I - ZZ^T + ZZ^T W Z (Z^T W Z)^{-1} Z^T \\
&= I && (3.6a) \\
A_{12} &= W(I - PW) J^T (JJ^T)^{-1} - J^T (JJ^T)^{-1} J (W - WPW) J^T (JJ^T)^{-1} \\
&= ZZ^T (W - WPW) J^T (JJ^T)^{-1} \\
&= \left(ZZ^T W - ZZ^T W Z (Z^T W Z)^{-1} Z^T W \right) J^T (JJ^T)^{-1} \\
&= 0 && (3.6b) \\
A_{22} &= J(I - PW) J^T (JJ^T)^{-1} \\
&= J(I - Z(Z^T W Z)^{-1} Z^T) J^T (JJ^T)^{-1} \\
&= JJ^T (JJ^T)^{-1} - JZ (Z^T W Z)^{-1} Z^T W J^T (JJ^T)^{-1} \\
&= I. && (3.6c)
\end{aligned}$$

The proof is complete. □

We note that in contrast to the results in [22], we do not need to assume nonsingularity of the Hessian $W_k(\delta)$ to obtain an expression for the projection matrix P_k (3.4).

From the explicit form of the inverse of $M_k(\delta)$ in (3.3) and (2.7) we have that,

$$t_k = -Z_k (Z_k^T W_k(\delta) Z_k)^{-1} Z_k^T (g_k + W_k(\delta) n_k). \tag{3.7}$$

We now prove that the criticality measure (3.1) is well-defined under assumptions (RG).

Theorem 3.2. *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} *satisfy* $J_{k_i} n_{k_i} = -c_{k_i}$*, and let* t_{k_i} *solve* (2.7)*. Then*

$$\lim_{i \to \infty} \Psi_{k_i}^t = 0 \implies \lim_{i \to \infty} \| Z_{k_i}^T g_{k_i} \| = 0$$

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

Proof: Define $M = M_{k_i}(\delta)$, $W = W_{k_i}(\delta)$, $J = J_{k_i}$, and $Z = Z_{k_i}$. Because $Jn_{k_i} = -c_{k_i}$, condition $\lim_{i\to\infty} x_{k_i} = x^*$ for feasible x^* ensures $\lim_{i\to\infty} n_{k_i} = 0$. Under the assumptions of Lemma 3.1 we know that $Z^T W Z$ is nonsingular and therefore that the projection matrix P_{k_i} exists and is nonsingular. From (3.1), (3.7), and $\lim_{i\to\infty} n_{k_i} = 0$ as $\lim_{i\to\infty} x_{k_i} = x^*$ we obtain the result. □

We are now ready to establish the following *descent lemma*, which is needed to guarantee global convergence of the filter line-search algorithm.

LEMMA 3.3. *Let (RG1)-(RG4) 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 $W := W_{k_i}(\delta)$, $J := J_{k_i}$, $g := g_{k_i}$, $d := d_{k_i}$, $\Psi := \Psi^t_{k_i}$, $\theta := \theta_{k_i}$, $n := n_{k_i}$, and $t := t_{k_i}$. Multiplying the first row of (2.7) by 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 test (RG3a) 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 $\|t^T W n\| \leq c_1 \Psi \theta$ and $\|g^T n\| \leq c_2 \theta$ for $c_1, c_2 > 0$. From (3.1) we have that $\|t\| = \Psi$, and we thus have

$$\begin{aligned} g^T d &\leq -\alpha_t \Psi^2 + c_1 \Psi \theta + c_2 \theta \\ &\leq \Psi \left(-\alpha_t \epsilon + c_2 \theta + \frac{c_3}{\epsilon} \theta \right). \end{aligned}$$

Defining $\epsilon_1 := \min\left\{\theta_{inc}, \frac{\epsilon^2 \alpha_t}{2(c_1 \epsilon + c_2)}\right\}$, 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 test (RG3a) 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 $c_1 = c_2 = 0$ and for ϵ_2 defined previously. □

The descent lemma is crucial because it guarantees that the objective function will be improved at a subsequence of nonstationary iterates (i.e., those with $\Psi^t_{k_i} \geq \epsilon$) that have 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 3.4. *Let assumptions (RG) hold. The filter line-search algorithm delivers a sequence $\{x_k\}$ satisfying*

$$\mathop{lim}_{k\to\infty}\ \theta(x_k) = 0 \tag{3.9a}$$

$$\mathop{lim\,inf}_{k\to\infty}\ \Psi^t(x_k) = 0. \tag{3.9b}$$

Proof: We go through the results leading to the proof of Theorem 2 in [35] and argue that our assumptions (RG) are sufficient. Unless otherwise stated, all lemmas refer to those in [35]. Lemma 1 establishes boundedness of d_k, λ_k^+, and $|m_k(\alpha)|$. This follows from (RG1), which guarantees that the right-hand side of the KKT system is bounded, and from (RG3b), which guarantees that the inverse of $M_k(\delta)$ is bounded. Lemma 2 is replaced by the descent Lemma 3.3 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 3.3 of this work. Lemma 6 requires only (RG1). Lemma 8 requires the descent Lemma 3.3 of this work. Lemma 10 establishes that for a subsequence of nonstationary iterates the filter is eventually not augmented. This requires the descent Lemma 3.3 of this work. The result follows. □

We use the following strategy to enforce (RG3). Note that because the Jacobian is assumed to be full rank, conditions (RG3a) and (RG3b) will eventually hold for sufficiently large δ.

Inertia-Free Regularization (IFR)

IFR-1 Given n_k, factorize $M_k(\delta)$ with $\delta = 0$ and compute t_k. If t_k satisfies (RG3a) and $M_k(\delta)$ 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 n_k, factorize $M_k(\delta)$ with current δ and compute t_k. If t_k satisfies (RG3a) and $M_k(\delta)$ 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.

Remark: The curvature condition (3.2) 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 [35]. In either case, however, we might need to regularize the Hessian in order to enforce nonsingularity of $M_k(\delta)$ at every iteration.

Remark: A caveat of inertia-free strategies is that they cannot guarantee that the step computed via the KKT system is a minimum of the associated quadratic program (the inertia-based approach guarantees this). While enabling global convergence with enhanced flexibility is a great benefit of inertia-free strategies, the 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. The effect depends on the application, however, and numerical experimentation is needed. We provide numerical results in Section 4 and discuss this issue further.

Remark: As seen in Lemma 3.3, the term $(\max\{t_k^T W_k(\delta) n_k - g_k^T n_k, 0\})$ in (RG3a) is harmless and is included only to provide additional flexibility. Because of this, one might also consider the simpler test $t_k^T W_k(\delta) t_k \geq \alpha_t t_k^T t_k$ and still enable convergence.

3.1. Alternative Inertia-Free Strategies. Computing the normal and tangential components of the step separately can be beneficial in certain situations. For instance, the use of a projected conjugate gradient (PCG) scheme provides a mechanism to perform the curvature test

$$t_{k,j}^T W_k(\delta) t_{k,j} \geq \alpha_t t_{k,j}^T t_{k,j} \tag{3.10}$$

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 constraint preconditioner which might not be available in certain applications.

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

$$d_k^T W_k(\delta) d_k + \max\{-(\lambda_k^+)^T c_k, 0\} \geq \alpha_d d_k^T d_k, \tag{3.11}$$

with d_k computed from (2.4).

To argue that test (3.11) is consistent, we use the criticality measure

$$\Psi_k^d = \|d_k\|. \tag{3.12}$$

If (RG3b) and (RG4) hold, we have from Lemma 3.1 that

$$\begin{aligned} d_k &= -P g_k - (I - PW) J^T (JJ^T)^{-1} c_k \\ &= -Z(Z^T W Z)^{-1} Z^T (g_k - W(\delta) J^T (JJ^T)^{-1} c_k) + J^T (JJ^T)^{-1} c_k. \end{aligned} \tag{3.13}$$

If we set $Y_k = J_k^T$, we have that d_k has the same structure as the decomposition in (2.11). Moreover, we have that

$$\Psi_k^d = \Psi_k^t + O(\|c_k\|). \tag{3.14}$$

Consequently, the results of Theorem 3.2 still hold, and the criticality measure is well defined. If $-(\lambda_k^+)^T c_k < 0$, from (3.11), (RG1), (RG4), and (RG3b) we have that $\|\lambda_k^+\| \leq \kappa$ for some $\kappa > 0$ and, therefore,

$$\begin{aligned} g_k^T d_k &= -d_k^T W_k d_k + d_k^T J_k^T \lambda_k^+ \\ &= -d_k^T W_k d_k + c_k^T \lambda_k^+ \\ &\leq -\alpha_d (\Psi_k^d)^2 + \kappa \theta_k. \end{aligned} \tag{3.15}$$

Consequently, the results of Lemma 3.3 hold with an appropriate definition of constants. If $-(\lambda_k^+)^T c_k \geq 0$ the result follows with $\kappa = 0$.

The curvature condition (3.11) holds for any δ and $\alpha_d \geq \lambda_{min}(W_k(\delta))$, and we note that the term $\max\{-(\lambda_k^+)^T c_k, 0\}$ is harmless and is used only to enhance flexibility.

Assumption (RG1) can be guaranteed to hold only if all iterates x_k remain strictly in the interior of the feasible region. This condition guarantees that the barrier function $\varphi^{mu}(x_k)$ and its derivatives are bounded. One can show that Theorem 3 in [35] holds under assumptions (RG). This result establishes that the iterates x_k remain in the strict interior of the feasible region if the maximum step size α_k^{max} is determined by using the following fraction-to-the-boundary rule

$$\alpha_k^{max} := \max\{\alpha \in (0, 1] \,:\, x_k + \alpha d_k \geq (1 - \tau) x_k\}, \tag{3.16}$$

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 3.1), is sufficient to establish the result.

4. Numerical Results. In this section we benchmark the inertia-based and inertia-free strategies using the PIPS-NLP interior-point framework [13]. 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.

4.1. Implementation. Our filter line-search implementation follows along the lines of that of IPOPT [34], but we implement neither a feasibility restoration nor watchdog heuristics. If the restoration phase is reached, we terminate the algorithm. We prefer to use this approach in order to isolate the effects of inertia correction on performance. As in IPOPT, we allows steps that are very small in norm $\|d_k\|$ to be accepted even if they do not satisfy SDC or SAC. 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) IFRd: inertia-free regularization with the curvature test (3.11), and (3) IFRt: inertia-free regularization with the curvature test (3.2). We use the same parameters for the IBR and IFR to increase/decrease the regularization parameter δ. For IFRd and IFRt we set both α_t and α_d to 1×10^{-12} as default. We also scale these parameters using the barrier parameter μ, as suggested in [16].

For the IFRt strategy, we compute the normal step by solving the linear system

$$\begin{bmatrix} W_k(\delta) & J_k^T \\ J_k & \end{bmatrix} \begin{bmatrix} n_k \\ \cdot \end{bmatrix} = - \begin{bmatrix} 0 \\ c_k \end{bmatrix}, \tag{4.1}$$

by factorizing $M_k(\delta)$ using MA57 [18], and we reuse the factorization to compute the tangential component from (2.7). For IBR we estimate the inertia of the KKT matrix using MA57.

We use the optimality error described in [34, Section 2] with a convergence tolerance of 1×10^{-6}, we perform iterative refinement for the KKT system with a tolerance of 1×10^{-12}, and we set the maximum number of iterations to 1,000. If the line search cannot find an acceptable point within 20 trials, the last trial step size is used in the next iteration. We use a pivoting tolerance of 1×10^{-4} for MA57.

Some of the test problems considered have Jacobians that are nearly rank-deficient. To deal with these instances, we regularize the (2,2) block of the KKT matrix as in IPOPT whenever we detect the KKT matrix to be singular [34, Section 3.1]. We emphasize that the convergence theory of this work and the supporting theory in [35] does not hold in this case anymore. In practice, however, we still observe convergence and satisfactory performance. We leave the theoretical treatment of this case as part of future work.

4.2. Small-Scale Tests. We consider 100 CUTEr instances, all requiring regularization in at least one of the three regularization strategies. We also use additional instances from energy applications that include building optimization, security-constrained optimal power flow, optimal control of chemical reactors, and optimal control of natural gas networks. The corresponding models are reported in [37, 12, 40, 38].

The performance of the three strategies is presented in Table 4.1. Here, we report the optimal objective value (Obj) as well as the number of iterations (Iter), and regularizations (Reg). The total number of factorizations is equal to the number of iterations plus the number of regularizations. We use the term "MAX" to denote the tests reaching the limit of iterations and the term "FAIL" to denote those tests terminated because a restoration phase is required. The last five instances in the table are the energy problems.

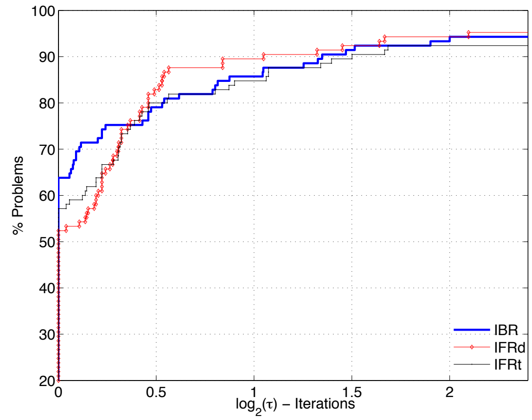

Fig. 4.1. *Number of iterations*

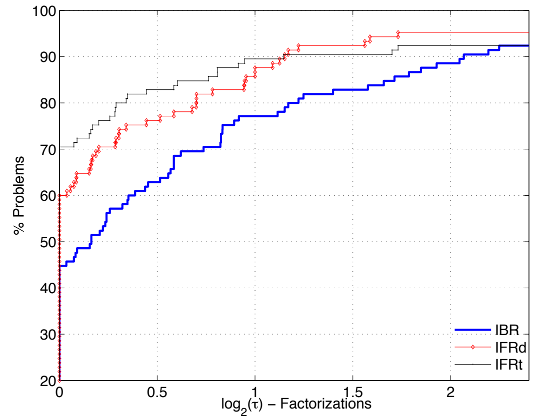

Fig. 4.2. *Number of factorizations*

As seen in Table 4.1, four test problems fail to converge with IBR as well as with IFRd. In three of those instances both strategies fail. With IFRt, six test problems fail, with two also failing for IBR and IFRd. We conclude that the robustness of the inertia-free strategies is competitive. In Figures 4.1 and 4.2 we present Dolan-Moré profiles [17] for the total number of iterations and factorizations reported in Table 4.1. In Figure 4.1 we can see that IBR requires fewer iterations to converge in 60% of the instances, but the differences are not dramatic. The robustness of IFRd is the same as that of IBR in 90% of the problems. In Figure 4.2 we can see that IFRd and IFRt significantly outperform IBR in terms of the total number of factorizations. From the numbers in Table 4.1 we have calculated that IBR requires, on average, 0.77 regularizations per iteration while IFRd and IFRt require 0.34 and 0.24 regularizations per iteration, respectively. These are relative improvements of over 50% in the total number of factorizations. We thus conclude that IFR strategies provide significantly more flexibility to accept steps.

Among all the instances that can be solved with the three strategies, IFRd and IFRt yield the same objective values as IBR in more than 90% of the instances. The performance is remarkable. IBR yields better objectives in eight instances (`biggs5`, `biggs6`, `hatlfde`, `humps`, `s267`, `s272`, `s393`, and `s394`). The names of these instances are highlighted in boldface in Table 4.1. We attribute the tendency of IBR to reach better objectives to the fact that the solutions of the KKT 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 expect that steps computed with IBR should yield improvements in the objective more often. To verify this claim, 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 step size 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 4.2 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. From this row we can see that the IBR algorithm accepts 12% of the steps due to SAC. In

TABLE 4.1
Performance of inertia-based and inertia-free strategies on small-scale instances.

	IBR			IFRd			IFRt		
Problem	**Obj**	**Iter**	**Reg**	**Obj**	**Iter**	**Reg**	**Obj**	**Iter**	**Reg**
avion2	9.47E+07	42	47	9.47E+07	59	10	9.47E+07	55	10
biggs3	9.99E-14	8	5	9.99E-14	8	5	9.99E-14	8	5
biggs5	1.08E-19	20	19	3.06E-01	19	3	3.06E-01	19	3
biggs6	8.91E-15	33	26	3.06E-01	18	0	3.06E-01	18	0
disc2	1.56E+00	79	84	1.56E+00	91	40	2.30E+00	86	20
dixmaang	1.00E+00	16	17	1.00E+00	176	6	1.00E+00	176	6
expfit	2.41E-01	8	8	2.41E-01	11	7	2.41E-01	11	7
expquad	-3.62E+06	23	17	-3.62E+06	34	0	-3.62E+06	34	0
fminsurf	1.00E+00	31	10	1.00E+00	40	0	1.00E+00	40	0
foo	2.03E+00	19	5	2.03E+00	34	5	2.03E+00	40	2
goffin	4.54E-06	3	2	4.54E-06	3	0	4.54E-06	3	0
grasp_nonc	4.91E-01	37	18	4.91E-01	42	4	4.91E-01	51	4
growth	1.00E+00	72	12	1.00E+00	74	1	1.00E+00	74	1
gulf	2.14E-16	21	11	5.01E-22	26	13	5.01E-22	26	13
hatlfde	4.43E-07	25	9	1.53E+01	18	0	1.53E+01	18	0
humps	8.84E-13	509	753	1.11E+03	203	220	1.11E+03	203	220
hvycrash		FAIL			FAIL			FAIL	
hypcir	0.00E+00	5	0	0.00E+00	5	3	0.00E+00	5	0
kissing		FAIL			FAIL		1.00E+00	657	15
kowosb	3.08E-04	8	5	3.08E-04	10	1	3.08E-04	10	1
mexhat	-4.01E-02	3	2	-4.01E-02	4	0	-4.01E-02	4	0
minc44	2.57E-03	15	10	2.57E-03	20	6	2.57E-03	28	10
minmaxbd	1.16E+02	66	20	1.16E+02	49	17	1.16E+02	56	2
minsurf_socp		FAIL			MAX			MAX	
mistake	-1.00E+00	17	8	-1.00E+00	54	29	-1.00E+00	54	29
nb_L1_eps	1.30E+01	253	144	1.30E+01	789	404	1.30E+01	463	210
nb_L2	2.65E+00	38	7	2.65E+00	36	3	2.65E+00	36	3
ngone	-6.37E-01	47	36	-6.37E-01	201	44	-6.42E-01	133	27
nonconvqp	-2.50E+05	10	10	-2.50E+05	10	9	-2.50E+05	10	10
osborne1	5.46E-05	46	10	5.46E-05	44	0	5.46E-05	44	0
price	2.92E-13	44	5	4.93E-13	44	0	4.93E-13	44	0
robotarm3	2.03E+00	19	5	2.03E+00	34	5	2.03E+00	40	2
s205	2.15E-21	11	7	2.91E-25	11	6	2.91E-25	11	6
s212	1.16E-22	11	6	1.16E-22	11	6	1.16E-22	11	6
s219	-1.00E+00	60	28	-1.00E+00	21	0	-1.00E+00	21	0
s221	-1.00E+00	61	37	-1.00E+00	35	20	-1.00E+00	61	9
s232	-1.00E+00	14	13	-1.00E+00	17	13	-1.00E+00	17	13
s236	-8.20E+00	12	2	-8.20E+00	12	2	-8.20E+00	12	2
s237	-5.89E+01	26	7	-5.89E+01	25	6	-5.89E+01	26	7
s238	-8.20E+00	71	57	-8.20E+00	19	9	-5.89E+01	50	12
s239	-8.20E+00	13	5	-8.20E+00	14	8	-8.20E+00	13	5
s242	5.40E-08	18	8	4.56E-08	21	5	4.56E-08	21	5
s245	1.59E-14	11	8	4.95E-13	8	0	4.95E-13	8	0
s247	1.72E-16	7	6	1.72E-16	6	0	1.72E-16	6	0
s248	-8.00E-01	15	5	-8.00E-01	15	5	-8.00E-01	15	5
s250	-3.30E+03	12	10	-3.30E+03	12	10	-3.30E+03	12	10
s251	-3.46E+03	12	9	-3.46E+03	12	9	-3.46E+03	12	9
s252	4.00E-02	19	0	4.00E-02	19	1	4.00E-02	19	0
s254	-1.73E+00	7	7	-1.73E+00	8	6	-1.73E+00	7	7
s257	7.54E-17	12	5	7.54E-17	14	5	7.54E-17	14	5
s258	2.74E-18	40	5	2.74E-18	40	5	2.74E-18	40	5
s260	2.74E-18	40	5	2.74E-18	40	5	2.74E-18	40	5
s265	1.90E+00	6	7	1.90E+00	6	7	1.90E+00	6	7
s267	1.42E-18	33	32	1.50E-02	31	4	1.50E-02	31	4
s270	9.03E-08	19	20	9.14E-08	21	1	9.14E-08	21	1
s272	1.48E-13	76	83	5.66E-03	19	6	5.66E-03	19	6
s282	1.94E-18	60	2	1.94E-18	60	2	1.94E-18	60	2
s287	1.37E-17	40	5	1.37E-17	40	5	1.37E-17	40	5
s289	0.00E+00	8	6	0.00E+00	8	6	0.00E+00	8	6
s294	3.97E+00	19	3	3.97E+00	19	3	3.97E+00	19	3
s295	3.99E+00	29	7	1.54E-16	39	10	1.54E-16	39	10
s296	3.99E+00	38	7	3.91E-19	47	10	3.91E-19	47	10
s311	5.80E-25	6	5	3.18E-17	7	4	3.18E-17	7	4
s315	-8.00E-01	13	8	-8.00E-01	12	6	-8.00E-01	12	6
s316	3.34E+02	7	7	3.34E+02	7	7	3.34E+02	6	0
s319	4.52E+02	11	7	4.52E+02	11	7	4.52E+02	8	0
s320	4.86E+02	13	7	4.86E+02	13	7	4.86E+02	9	0
s321	4.96E+02	16	13	4.96E+02	16	8	4.96E+02	11	0
s322	5.00E+02	33	43	5.00E+02	22	9	5.00E+02	16	0
s327	3.06E-02	18	2	3.06E-02	23	0	3.06E-02	23	0
s329	-6.96E+03	13	7	-6.96E+03	13	7	-6.96E+03	13	7
s333	4.33E-02	7	3	4.33E-02	7	0	4.33E-02	7	0
s334	8.21E-03	7	5	8.21E-03	8	0	8.21E-03	8	0
s336	-3.38E-01	17	11	-3.38E-01	17	11	-3.38E-01	25	0
s338	-7.21E+00	13	17	-7.21E+00	13	17	-7.21E+00	11	9
s340	-5.40E-02	9	7	-5.40E-02	10	7		FAIL	
s341	-2.26E+01	7	0	-2.26E+01	10	4	-2.26E+01	7	0
s350	3.08E-04	8	5	3.08E-04	10	1	3.08E-04	10	1
s351	3.19E+02	10	7	3.19E+02	11	2	3.19E+02	11	2
s353	-3.99E+01	7	0	-3.99E+01	7	0	-3.99E+01	7	0
s355	6.97E+01	34	34	6.97E+01	93	38	6.97E+01	71	31
s356	1.88E+00	14	12		MAX		1.88E+00	45	4
s358	5.46E-05	23	7	5.46E-05	20	0	5.46E-05	20	0
s365	5.21E+01	20	8	5.21E+01	24	16		FAIL	
s366	1.23E+03	20	3	1.23E+03	20	3	1.23E+03	20	3
s367	-3.74E+01	36	39	-3.74E+01	16	9	-3.74E+01	13	0
s368	0.00E+00	46	71	3.55E-15	5	0	3.55E-15	5	0
s374	2.33E-01	51	67	2.33E-01	60	6	2.91E-01	29	2
s375	-1.52E+01	19	18	-1.52E+01	19	19	-1.52E+01	19	18
s378	-4.78E+01	22	12	-4.74E+01	55	13		FAIL	
s380	3.17E+00	69	13	3.17E+00	45	1	3.17E+00	56	0
s387	-8.25E+03	17	10	-8.25E+03	17	10	-8.25E+03	21	12
s388	-5.82E+03	62	76	-5.82E+03	30	10	-5.82E+03	76	54
s389	-5.81E+03	84	110	-5.81E+03	40	18	-5.81E+03	33	14
s393	8.63E-01	31	23	1.03E+00	13	2	1.03E+00	13	2
s394	1.92E+00	14	9	4.97E+00	13	0	4.97E+00	13	0
s395	1.92E+00	14	11	1.92E+00	14	3	1.92E+00	14	3
static3		FAIL		-1.53E+03	23	0	-1.53E+03	23	0
tre	1.72E-46	9	7	1.72E-46	9	7	1.72E-46	9	7
woods	6.85E-15	40	5	6.85E-15	40	5	6.85E-15	40	5
buildings	1.75E+03	183	161	1.75E+03	153	24	1.75E+03	106	3
IEEE57_opf	5.84E+00	32	38	5.84e+00	29	29	5.84e+00	29	29
IEEE162_opf	1.64E+00	145	183	1.64E+00	23	0	1.64E+00	23	0
IEEE300_opf	1.17E-02	33	9	1.17E-02	31	0	1.17E-02	31	0
reactor	8.92E+00	74	32	8.93E+00	74	9		FAIL	

TABLE 4.2
Percentage of steps accepted by different criteria in the filter.

Problem	IBR SAC	IBR SRCO	IBR SRCC	IFRd SAC	IFRd SRCO	IFRd SRCC	IFRt SAC	IFRt SRCO	IFRt SRCC
avion2	40	33	26	34	47	19	27	51	20
disc2	11	34	52	1	54	44	0	49	50
foo	0	53	47	0	56	44	0	53	48
goffin	67	33	0	67	33	0	67	33	0
grasp_nonconvex	30	40	30	24	24	52	20	31	49
minc44	0	67	33	0	50	50	0	71	29
minmaxbd	0	33	67	0	27	73	0	29	71
mistake	0	71	29	0	59	41	0	59	41
nb_L1_eps	40	44	16	11	51	38	21	49	31
nb_L2	5	71	24	6	78	17	6	78	17
ngone	0	79	21	0	62	38	0	71	29
nonconvqp	90	10	0	80	10	10	90	10	0
robotarm3	0	53	47	0	56	44	0	53	48
s219	0	35	65	0	5	95	0	5	95
s221	43	13	43	51	40	9	44	13	43
s236	0	92	8	0	92	8	0	92	8
s237	4	88	8	4	88	8	4	88	8
s238	3	70	27	0	84	16	2	66	32
s239	8	77	15	7	64	29	8	77	15
s247	43	57	0	17	83	0	17	83	0
s248	0	47	53	0	47	53	0	47	53
s252	5	63	32	5	68	26	5	63	32
s254	0	14	86	0	25	75	0	14	86
s265	33	0	17	33	0	17	50	0	17
s270	5	89	5	5	86	10	5	86	10
s315	8	77	15	8	83	8	8	83	8
s316	0	14	86	0	14	86	0	17	83
s319	18	9	73	18	9	73	0	13	88
s320	8	15	77	8	15	77	0	11	89
s321	6	13	81	13	6	81	0	9	91
s322	3	27	67	5	14	82	0	6	94
s327	6	83	11	4	87	9	4	87	9
s336	6	12	82	6	12	82	0	24	76
s338	8	23	69	8	23	69	0	36	64
s341	14	43	43	10	40	50	14	43	43
s355	3	94	3	1	85	14	0	90	10
s367	14	58	25	0	88	12	0	85	15
s374	0	76	24	2	60	38	0	90	10
s380	1	42	57	0	42	58	2	48	50
s387	6	53	41	6	53	41	5	57	38
s388	2	63	35	3	50	47	1	63	36
s389	11	40	49	3	53	45	3	58	39
s393	6	94	0	15	85	0	15	85	0
s394	7	93	0	0	100	0	0	100	0
s395	0	93	7	0	100	0	0	100	0
buildings	12	34	54	7	20	73	9	13	77
IEEE162_opf	3	94	2	0	96	4	0	96	4
IEEE300_opf	0	64	36	0	61	39	0	61	39
Average	12	52	35	10	52	38	9	53	37

contrast, the percentages are only 10% and 9% for IFRd and IFRt, respectively. If we add the total percentages in which the steps are accepted because of improvements in the objective (add SAC and SDCO), we have that the percentages are 64%, 62%, and 62% for IBR, IFRd, and IFRt, respectively. The IBR strategy accepts more steps that decrease the objective. An interesting observation is that IFRt and IFRd accept the same percentage of steps. We note, however, that we cannot draw general conclusions about the tendency of the strategies to provide better objective values because of the presence of multiple local minima. In particular, we note that IFRt and IFRd yield better objective values than does IBR in three instances (`s238,s295,s296`).

We remark on the behavior of the strategies in instance `IEEE_162_opf`. From Table 4.1 we can see that IFRt and IFRd do not require regularization and converge in 23 iterations whereas IBR requires 145 iterations and 183 regularizations. This instance is an ill-conditioned optimal power flow 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 also observed this difference in performance in instances `s368`, `static3`, and `buildings`.

4.3. Large-Scale Tests. We now demonstrate that the inertia-free strategies remain robust in large-scale problems. We use two-stage stochastic optimization problems arising from security-constrained optimal power flow (SCOPF) and stochastic optimal control of natural gas networks [12, 38]. Because of the large dimensionality of these instances we solve them using a distributed-memory decomposition strategy.

The KKT matrix (2.4) of stochastic optimization problems can be permuted into the following block-bordered diagonal form:

$$\hat{M}(\delta) = \begin{bmatrix} \hat{M}_1(\delta) & & & & B_1^T \\ & \hat{M}_2(\delta) & & & B_2^T \\ & & \ddots & & \vdots \\ & & & \hat{M}_S(\delta) & B_S^T \\ B_1 & B_2 & \cdots & B_S & \hat{M}_0(\delta) \end{bmatrix}, \tag{4.2}$$

where $i = 1, 2, ..., S$ are the scenario indexes and S is the number of scenarios [20, 41, 27]. The zero index corresponds to coupling variables, and we refer to this as the zero scenario. We use $\hat{M}(\delta)$ to denote the permuted form of the KKT matrix $M(\delta)$. The diagonal matrices

$$\hat{M}_i(\delta) = \begin{bmatrix} W_i(\delta) & J_i^T \\ J_i & 0 \end{bmatrix} \tag{4.3}$$

for $i = 0, ..., S$ have a saddle-point structure, where $W_i(\delta)$ and J_i are the corresponding Hessian and Jacobian contributions of each scenario and the border matrices B_i define coupling between scenarios and the zero scenario.

The permuted KKT system can be represented as

$$\hat{M}(\delta)\hat{w} = \hat{r}, \tag{4.4}$$

where $\hat{w} = (\hat{w}_1, \ldots, \hat{w}_S, \hat{w}_0)$ are the permuted search directions for primal variables and multipliers, respectively, and $\hat{r} = (\hat{r}_1, \ldots, \hat{r}_S, \hat{r}_0)$ are the permuted right-hand sides. To solve the structured KKT system (4.4) in parallel, we use Schur decomposition. The solution of (4.4) can be obtained from

$$\hat{z}_i = \hat{M}_i(\delta)^{-1}\hat{r}_i, \quad i = 1, \ldots, S, \tag{4.5a}$$

$$\hat{w}_0 = C(\delta)^{-1}(\hat{r}_0 - \sum_{i=1}^{S} B_i \hat{z}_i), \tag{4.5b}$$

$$\hat{w}_i = \hat{z}_i - \hat{M}_i(\delta)^{-1} B_i^T \hat{w}_0, \quad i = 1, \ldots, S, \tag{4.5c}$$

where

$$C(\delta) = \hat{M}_0(\delta) - \sum_{i=1}^{S} B_i \hat{M}_i(\delta)^{-1} B_i^T, \tag{4.6}$$

is the Schur complement. Each slave processor is allocated with the information of certain blocks i, and performs step (4.5a) by factorizing the local blocks $\hat{M}_i(\delta)$ in parallel. A master processor gathers the contributions of each worker to assemble the Schur complement in (4.6) and computes the step size for the coupling variables using (4.5b). Having the coupling step $\hat{w}_0$, the slave processors compute the local steps $\hat{w}_i$ in parallel using (4.5c). We note that the serial bottleneck in this procedure is the assembly of the Schur complement. Because the Schur complement must be re-assembled whenever the regularization term δ is adjusted, regularization can increase not only total work per iteration but also the parallel performance.

Because an LBL^T factorization of the entire matrix $\hat{M}(\delta)$ is not available, its inertia can be inferred by using Haynsworth's inertia additivity formula [24]:

$$\text{Inertia}(\hat{M}(\delta)) = \text{Inertia}(C(\delta)) + \sum_{i=1}^{S} \text{Inertia}(\hat{M}_i(\delta)). \tag{4.7}$$

We perform the factorization of the subblocks and of the Schur complement and check that the addition of their inertias gives the necessary inertia $\text{Inertia}(\hat{M}(\delta)) = \{n, m, 0\}$. If this is not the case, all the Hessian terms $W_i(\delta), i = 0, ..., S$ are regularized by using a common parameter δ until the $\hat{M}(\delta)$ has the correct inertia. We note that obtaining the inertia of the Schur complement $C(\delta)$ by factorizing it with a sparse symmetric indefinite routine such as MA57 is not efficient because this matrix tends to be dense. More efficient codes such as MAGMA or ELEMENTAL can be used but these are based on dense factorizations schemes that do not provide inertia information [1, 26]. This situation illustrates a complication that can be encountered when inertia information is required. In our implementation we use MA57 to factorize the Schur complement (even if it is not the best choice) because we seek to compare the performance of IFR with that of IBR using a compatible setting.

All tests in this section 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.

We solve a security-constrained optimal power flow instance (`IEEE_300`) that contains 878,650 variables and 734,406 constraints and a stochastic optimal control problem instance (`STOCH_GAS`) that contains 1,024,651 variables and 1,023,104 constraints. The results are presented in Table 4.3. 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 $\alpha_t = 1 \times 10^{-10}, \alpha_d = 1 \times 10^{-10}$ (scaled by μ) because the default values of 1×10^{-12} resulted in high variability in the number of iterations for `STOCH_GAS`. We have, in general, observed that increasing α_t, α_d enhances robustness. As expected, however, this comes at the expense of additional regularizations.

TABLE 4.3
Performance of inertia-based and inertia-free strategies on large-scale instances.

		IBR				IFRd			IFRt		
Problem	#MPI	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

We can see that, in general, IFRd and IFRt require more iterations than does IBR; 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 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.

5. Conclusions. We have presented new inertia-free strategies for filter line-search algorithms. The strategies perform curvature tests along computed directions that guarantee descent when the constraint violation is sufficiently small. We have proved that the strategies yield global convergence and are competitive with inertia-based strategies. 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.

Acknowledgments. We thank Frank Curtis and Jorge Nocedal for technical discussions. 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 also acknowledge the computing resources provided by the Laboratory Computing Resource Center at Argonne National Laboratory.

REFERENCES

[1] E. Agullo, J. Demmel, J. Dongarra, B. Hadri, J. Kurzak, J. Langou, H. Ltaief, P. Luszczek, and S. Tomov. Numerical linear algebra on emerging architectures: The PLASMA and MAGMA projects. In *Journal of Physics: Conference Series*, volume 180, page 012037. IOP Publishing, 2009.
[2] P. R. Amestoy, A. Guermouche, J.Y. LExcellent, and S. Pralet. Hybrid scheduling for the parallel solution of linear systems. *Parallel Computing*, 32(2):136–156, 2006.
[3] S. Balay, J. Brown, K. Buschelman, V. Eijkhout, W. Gropp, D. Kaushik, M. Knepley, L. McInnes, B. Smith, and H. Zhang. PETSc users manual revision 3.4, 2013.
[4] M. Benzi, G.H. Golub, and J. Liesen. Numerical solution of saddle point problems. *Acta Numerica*, 14:1–137, 2005.
[5] L. Bergamaschi, J. Gondzio, M. Venturin, and G. Zilli. Inexact constraint preconditioners for linear systems arising in interior point methods. *Computational Optimization and Applications*, 36(2-3):137–147, 2007.
[6] L.T. Biegler and V.M. Zavala. Large-scale nonlinear programming using IPOPT: An integrating framework for enterprise-wide dynamic optimization. *Computers & Chemical Engineering*, 33(3):575–582, 2009.
[7] G. Biros and O. Ghattas. Parallel Lagrange–Newton–Krylov–Schur methods for PDE-constrained optimization. Part I: The Krylov–Schur solver. *SIAM Journal on Scientific Computing*, 27(2):687–713, 2005.
[8] A. Borzì and V. Schulz. Multigrid methods for pde optimization. *SIAM Review*, 51(2):361–395, 2009.
[9] J.R. Bunch and L. Kaufman. Some stable methods for calculating inertia and solving symmetric linear systems. *Mathematics of computation*, pages 163–179, 1977.
[10] R.H. Byrd, F.E. Curtis, and J. Nocedal. An inexact newton method for nonconvex equality constrained optimization. *Mathematical rogramming, volume=122, number=2, pages=273–299, year=2010, publisher=Springer.*
[11] A.M. Cervantes, A. Wächter, R.H. Tütüncü, and L.T. Biegler. A reduced space interior point strategy for optimization of differential algebraic systems. *Computers & Chemical Engineering*, 24(1):39–51, 2000.
[12] N. Chiang and A. Grothey. Solving security constrained optimal power flow problems by a structure exploiting interior point method. *Optimization and Engineering*, pages 1–23, 2012.
[13] N. Chiang, C.G. Petra, and V.M. Zavala. Structured nonconvex optimization of large-scale energy systems using PIPS-NLP. In *Proceedings of the 18th Power Systems Computation Conference (PSCC), Wroclaw, Poland*, 2014.
[14] M.P. Costa and E. MGP Fernandes. Assessing the potential of interior point barrier filter line search methods: nonmonotone versus monotone approach. *Optimization*, 60(10-11):1251–1268, 2011.
[15] F.E. Curtis, J. Nocedal, and A. Wächter. A matrix-free algorithm for equality constrained optimization problems with rank-deficient Jacobians. *SIAM Journal on Optimization*, 20(3):1224–1249, 2009.
[16] F.E. Curtis, O. Schenk, and A. Wächter. An interior-point algorithm for large-scale nonlinear optimization with inexact step computations. *SIAM Journal on Scientific Computing*, 32(6):3447–3475, 2010.
[17] E.D. Dolan and J.J. Moré. Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91(2):201–213, 2002.

[18] I. S. Duff. Ma57 - a code for the solution of sparse symmetric definite and indefinite systems. *ACM Transactions on Mathematical Software*, 30:118–144, 2004.
[19] I.S. Duff and J.K. Reid. *MA27-a set of Fortran subroutines for solving sparse symmetric sets of linear equations*. UKAEA Atomic Energy Research Establishment, 1982.
[20] J. Gondzio and R. Sarkissian. Parallel interior-point solver for structured linear programs. *Mathematical Programming*, 96(3):561–584, 2003.
[21] N.I.M. Gould. On practical conditions for the existence and uniqueness of solutions to the general equality quadratic programming problem. *Mathematical Programming*, 32(1):90–99, 1985.
[22] N.I.M. Gould, M.E. Hribar, and J. Nocedal. On the solution of equality constrained quadratic programming problems arising in optimization. *SIAM Journal on Scientific Computing*, 23(4):1376–1395, 2001.
[23] N. Haverbeke, M. Diehl, and B. De Moor. A structure exploiting interior-point method for moving horizon estimation. In *Proceedings of the 48th IEEE Conference on Decision and Control*, pages 1273–1278. IEEE, 2009.
[24] E.V. Haynsworth. Determination of the inertia of a partitioned Hermitian matrix. *Linear Algebra and Its Applications*, 1(1):73 – 81, 1968.
[25] Y. Kawajiri and L.T. Biegler. Optimization strategies for simulated moving bed and powerfeed processes. *AIChE Journal*, 52(4):1343–1350, 2006.
[26] M. Lubin, C.G. Petra, and M. Anitescu. The parallel solution of dense saddle-point linear systems arising in stochastic programming. *Optimization Methods and Software*, 27(4-5):845–864, 2012.
[27] M. Lubin, C.G. Petra, M. Anitescu, and V.M. Zavala. Scalable stochastic optimization of complex energy systems. In *High Performance Computing, Networking, Storage and Analysis (SC), 2011 International Conference for*, pages 1–10. IEEE, 2011.
[28] J. Nocedal, A. Wächter, and R.A. Waltz. Adaptive barrier update strategies for nonlinear interior methods. *SIAM Journal on Optimization*, 19(4):1674–1693, 2009.
[29] C.G. Petra, O. Schenk, M. Lubin, and K. Gaertner. An augmented incomplete factorization approach for computing the schur complement in stochastic optimization. *SIAM Journal on Scientific Computing*, 36(2):C139–C162, 2014.
[30] C.V. Rao, S.J. Wright, and J.B. Rawlings. Application of interior-point methods to model predictive control. *Journal of optimization theory and applications*, 99(3):723–757, 1998.
[31] O. Schenk, A. Wächter, and M. Hagemann. Matching-based preprocessing algorithms to the solution of saddle-point problems in large-scale nonconvex interior-point optimization. *Computational Optimization and Applications*, 36:321–341, 2007.
[32] O. Schenk, A. Wächter, and M. Weiser. Inertia-revealing preconditioning for large-scale nonconvex constrained optimization. *SIAM Journal on Scientific Computing*, 31(2):939–960, 2008.
[33] M. Soler, A. Olivares, and E. Staffetti. Hybrid optimal control approach to commercial aircraft trajectory planning. *Journal of Guidance, Control, and Dynamics*, 33(3):985–991, 2010.
[34] A. Wächter and L. T. Biegler. On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106:25–57, 2006.
[35] A. Wächter and L.T. Biegler. Line search filter methods for nonlinear programming: Motivation and global convergence. *SIAM Journal on Optimization*, 16(1):1–31, 2005.
[36] Y. Wang and S. Boyd. Fast model predictive control using online optimization. *EEE Transactions on Control Systems Technology*, 18(2):267–278, 2010.
[37] V.M. Zavala. Real-time optimization strategies for building systems. *Industrial & Engineering Chemistry Research*, 52(9):3137–3150, 2012.
[38] V.M. Zavala. Stochastic optimal control model for natural gas networks. *Computers & Chemical Engineering*, 64:103–113, 2014.
[39] V.M. Zavala and L.T. Biegler. Large-scale parameter estimation in low-density polyethylene tubular reactors. *Industrial & Engineering Chemistry Research*, 45(23):7867–7881, 2006.
[40] V.M. Zavala and A. Flores-Tlacuahuac. Stability of multiobjective predictive control: A utopia-tracking approach. *Automatica*, 48(10):2627–2632, 2012.
[41] V.M. Zavala, C.D. Laird, and L.T. Biegler. Interior-point decomposition approaches for parallel solution of large-scale nonlinear parameter estimation problems. *Chemical Engineering Science*, 63(19):4834–4845, 2008.
[42] S. Zenios and R. Lasken. Nonlinear network optimization on a massively parallel connection machine. *Annals of Operations Research*, 14(1):147–165, 1988.