

Structured Nonconvex Optimization of Large-Scale Energy Systems Using PIPS-NLP

Naiyuan Chiang, Cosmin G. Petra, and Victor M. Zavala

Abstract—We present PIPS-NLP, a software library for the solution of large-scale structured nonconvex optimization problems on high-performance computers. We discuss the features of the implementation in the context of electrical power and gas network systems. We illustrate how different model structures arise in these domains and how these can be exploited to achieve high computational efficiency. Using computational studies from security-constrained ACOPF and line-pack dispatch in natural gas networks, we demonstrate robustness and scalability.

Index Terms—high-performance computing, optimization, nonconvex, structures, power systems, natural gas, networks

I. INTRODUCTION

Structures are pervasive in engineering optimization. Examples include multistage stochastic optimization, networks, optimal control, state and parameter estimation, and partial differential algebraic equations. The combination of these structures induces modular and heterogeneous linear algebra systems within optimization algorithms that can be exploited to achieve high computational efficiency. In this context, interior-point algorithms are particularly attractive because the structure of the linear system remains unchanged along the search, thus simplifying the implementation of decomposition and preconditioning strategies. In the context of energy systems, structure-exploiting interior-point implementations have been developed for stochastic economic dispatch [6], security-constrained AC- and DCOPF [7], [4], and natural gas optimization problems [3].

One of the main challenges arising in implementing modular linear algebra in nonconvex optimization is the need to detect and mitigate negative curvature. Negative curvature arises from nonconvexities of the constraints and objective functions. This, if not properly handled, can attract the algorithm to undesired stationary points (i.e., maxima instead of minima). To our knowledge, the only structured nonconvex optimization implementation that deals with negative curvature is the one of Laird and coworkers [5], [11]. In their implementation, inertia of arrowhead linear systems is inferred by monitoring the inertia of the diagonal blocks and of the Schur complement. This strategy, however, requires the use of specific linear algebra solvers capable of estimating inertia and can be thus restrictive. For instance, linear solvers for graphics processing units (GPUs) implemented in the MAGMA library [8] and iterative solvers implemented in the PETSc library do not provide inertia information [1]. In addition, inertia detection cannot be easily generalized to problems with heterogeneous structures.

PIPS-NLP, the implementation that we present here, uses a filter line-search interior-point algorithm coupled to a modular

linear algebra kernel that exploits heterogeneous structures. The algorithm bypasses the need for inertia information by performing a curvature test to the Hessian matrix along the computed step. This simple feature enables modular implementations because it does not impose restrictions on the linear algebra solvers that can be used. We discuss heterogeneous structures arising in energy systems and how these can be exploited by PIPS-NLP. In addition, we demonstrate that the implementation can tackle large-scale nonconvex problems and that the implementation is scalable.

The paper is structured as follows. In Section II we describe the PIPS-NLP implementation which includes the filter line-search algorithm, the curvature detection strategy, and different strategies to exploit modular linear algebra. In Section III we present computational studies for challenging security-constrained ACOPF problems and stochastic optimal control problems arising in line-pack management in natural gas networks. Conclusions and directions of future work are presented in Section IV.

II. PIPS-NLP

In this section we discuss the filter line-search interior-point algorithm implemented in PIPS-NLP and the curvature detection strategy used to deal with nonconvexities. We also describe different structures that can be exploited by the linear algebra kernel.

Consider the general nonlinear programming problem (NLP) of the form

$$\min f(x) \quad (\text{II.1a})$$

$$\text{s.t. } c(x) = 0, \quad (\lambda) \quad (\text{II.1b})$$

$$x \geq 0 \quad (\nu) \quad (\text{II.1c})$$

here, $x \in \mathbb{R}^n$ are primal variables, $\lambda \in \mathbb{R}^m$ are multipliers for equality constraints, and $\nu \in \mathbb{R}^n$ are multipliers for the bound constraints. The functions $f : \mathbb{R}^n \rightarrow \mathbb{R}$ and $c : \mathbb{R}^n \rightarrow \mathbb{R}^m$ are smooth and possibly nonconvex. To solve the NLP, we use an interior-point logarithmic barrier framework. The barrier subproblem is given by

$$\min \varphi^\mu(x) := f(x) - \mu \sum_{j=1}^n \ln x_{(j)} \quad (\text{II.2})$$

$$\text{s.t. } c(x) = 0, \quad (\lambda)$$

where $\mu > 0$ is the barrier parameter and $x_{(j)}$ is the j th entry of vector x . We consider a framework that solves a sequence of barrier problems (II.2) and drives the barrier parameter μ monotonically to zero. To solve each barrier problem, we apply

N. Chiang, C. G. Petra, and V. M. Zavala are with the Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439, USA. E-mail: {nychiang, petra,vzavala}@mcs.anl.gov.

Newton's method to the Karush-Kuhn-Tucker system:

$$\nabla_x \varphi^\mu(x) + \nabla_x c(x)\lambda = 0 \quad (\text{II.3a})$$

$$c(x) = 0 \quad (\text{II.3b})$$

$$x \geq 0. \quad (\text{II.3c})$$

We denote the iterates at counter k as (x_k, λ_k) . In the filter line-search strategy of [9], the primal search direction d_k and the multipliers λ_k^+ are obtained by solving the following linearized *augmented* system:

$$\begin{bmatrix} W_k(\delta) & J_k^T \\ J_k & \end{bmatrix} \begin{bmatrix} d_k \\ \lambda_k^+ \end{bmatrix} = - \begin{bmatrix} g_k \\ c_k \end{bmatrix}. \quad (\text{II.4})$$

Here, $c_k := c(x_k)$, $J_k := \nabla_x c(x_k)^T$, $g_k := \nabla_x \varphi_k^\mu$, $W_k(\delta) := H_k + \Sigma_k + \delta \mathbb{I}$, $H_k := \nabla_{xx} \mathcal{L}(x_k, \lambda_k)$, $\mathcal{L}(x_k, \lambda_k) := \varphi^\mu(x_k) + \lambda_k^T c(x_k)$, and $\Sigma_k := X_k^{-2}$, and $X_k := \text{diag}(x_k)$. We use the approximation $\Sigma_k \approx X_k^{-1} V_k$ with $V_k := \text{diag}(\nu_k)$ to obtain better behavior of the algorithm. We also define the *regularization parameter* $\delta \geq 0$; and, to enable compact notation, we define the augmented matrix

$$M_k(\delta) := \begin{bmatrix} W_k(\delta) & J_k^T \\ J_k & \end{bmatrix}. \quad (\text{II.5})$$

A. Filter Line-Search

We use the filter line-search framework proposed by Wächter and Biegler [9] to promote global convergence. We define a two-dimensional filter of the form $\mathcal{F} := \{\theta(x), \varphi(x)\}$ with $\theta(x) = \|c(x)\|$ and $\varphi(x) := \varphi^\mu(x)$ for fixed barrier parameter μ . The filter is initialized as

$$\mathcal{F}_0 := \{(\theta, \varphi) \mid \theta \geq \theta^{max}\} \quad (\text{II.6})$$

with a given parameter $\theta^{max} > 0$. Given a 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 $x_k(\alpha_{k,\ell}) := x_k + \alpha_{k,\ell} d_k$. 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)*:

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

- *Armijo Condition (AC)*:

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

- *Sufficient Reduction Condition (SRC)*:

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

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

$$m_k(\alpha) := \alpha g_k^T d_k \quad (\text{II.7})$$

is a linear model of $\varphi(\cdot)$ at x_k in the direction d_k . Note that the direction d_k is of descent for the objective function if $m_k(\alpha) < 0$ for $\alpha \in (0, 1]$.

Satisfying the filter condition (FC) is the first requirement to accept a trial iterate $x_k(\alpha_{k,\ell})$. We say that the trial iterate is contained in the filter if $(\theta(x_k(\alpha_{k,\ell})), \varphi(x_k(\alpha_{k,\ell}))) \in \mathcal{F}_k$. If this is the case, the trial iterate improves neither the objective nor the constraint violation. The step then is rejected, the step size $\alpha_{k,\ell}$ is reduced, and we increase counter $\ell \leftarrow \ell + 1$. If the trial iterate is not contained in the filter then it makes progress in either the constraint violation or the objective function, so we continue to test additional conditions. In particular, we check whether the switching condition (SC) holds. We have two cases:

- Case I: 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 iterate $x_k(\alpha_{k,\ell})$. If not, we decrease the step size.
- Case II: If (SC) does not hold then we check whether (SRC) holds. If (SRC) holds, then we accept the trial iterate $x_k(\alpha_{k,\ell})$. If not, we decrease the step size.

If the trial iterate $x_k(\alpha_{k,\ell})$ is accepted in Case II, then the filter is updated as

$$\mathcal{F}_{k+1} := \\ \mathcal{F}_k \cup \{(\theta, \varphi) \mid \varphi \geq \varphi(x_k) - \gamma_\varphi \theta(x_k), \theta \geq (1 - \gamma_\theta)\theta(x_k)\} \quad (\text{II.8})$$

with parameters $\gamma_\varphi, \gamma_\theta \in (0, 1)$; otherwise, we leave the filter unchanged: $\mathcal{F}_{k+1} = \mathcal{F}_k$. If the trial step size $\alpha_{k,\ell}$ becomes smaller than α_k^{min} and the step has not been accepted then the algorithm reverts to a restoration phase. A strategy to obtain α_k^{min} is proposed in [9].

The combination of the switching condition (SC) and the Armijo condition (AC) is key. These conditions enforce progress in the objective function if a direction d_k is of descent and the predicted progress for the objective is large compared with the constraint violation. In the absence of these conditions, the filter can accept points that make progress only on the constraint violation and not on the objective and thus the iterates will be attracted to maximizers instead of minimizers.

B. Negative Curvature

The missing piece in the filter line-search algorithm described is a step computation mechanism that guarantees the delivery of steps that decrease the objective function at least when the constraint violation is sufficiently small. To this end, implementations such as IPOPT [9] accept the step d_k only if the inertia of the augmented matrix $M_k(\delta)$ is correct for a given value of the regularization parameter δ . In particular, IPOPT checks that the number of positive eigenvalues is n and that the number of negative eigenvalues is m , which guarantees that the Hessian $W_k(\delta)$ projected on the null space of the Jacobian J_k is positive definite. Positive definiteness of the projected Hessian (also known as reduced Hessian) in turn guarantees that the direction d_k is of descent for a sufficiently small constraint violation. Inertia information is obtained from direct factorization procedures of symmetric indefinite linear solvers such as MA57 and Pardiso.

The inertia detection strategy is used to determine the regularization parameter δ as follows: If the inertia of the augmented matrix for $\delta = 0$ is not correct, then δ increased and the

augmented matrix is factorized again. The factorization procedure is repeated until the inertia is correct. The number of trial factorizations can be significant and, because the factorization of the augmented matrix is the most expensive step in the algorithm, computational performance can be greatly affected. Note also that if the regularization parameter δ is too large the quality of the step can degrade because the regularized Hessian becomes a poor of the true Hessian, thus deteriorating progress.

While the inertia detection strategy is theoretically sound, it focuses on the properties of the augmented matrix $M_k(\delta)$ alone and can thus discard directions d_k that are of descent but for which the inertia of the augmented matrix is not correct. Note also that inertia estimates can vary depending on the implementation and algorithmic parameters of different linear solvers (e.g., MA57 vs. Pardiso). Furthermore, if the augmented matrix has modular structures or if we use iterative or unsymmetric solvers, inertia information will not be available. This situation is important because many libraries targeted to hybrid high-performance computing architectures are actively being developed.

To bypass the need for inertia information, PIPS-NLP performs the following curvature test,

$$d_k^T W_k(\delta) d_k \geq \kappa d_k^T d_k, \quad (\text{II.9})$$

for a parameter $\kappa > 0$. If this test is satisfied, then we accept the step d_k ; if this is not satisfied, we increase the regularization parameter δ and compute a new step d_k (using the new regularized augmented matrix). The procedure is repeated until the curvature test is passed. We can justify the curvature test by analyzing the augmented system (II.4). If we multiply the first row by d_k^T , we obtain

$$d_k^T W_k(\delta) d_k - c_k^T \lambda_k^+ = -d_k^T g_k, \quad (\text{II.10})$$

where we use the second row to deduce that $d_k^T A_k^T = -c_k^T$. When the constraint violation $\theta(x_k) = \|c(x_k)\|$ is sufficiently small and the curvature test holds, we have that $-d_k^T g_k \approx d_k^T W_k(\delta) d_k \geq \kappa d_k^T d_k \geq 0$. Consequently, the direction d_k is of descent and the switching (SC) and Armijo condition (AC) will hold for some step size $\alpha_{k,\ell}$. This feature is remarkable and it highlights the fact that the filter mechanism provides great flexibility. In particular, if the constraint violation is large, the curvature test is not relevant and the algorithm can make progress by accepting steps that decrease the constraint violation. Once it reaches a certain threshold, however, the curvature test will guarantee descent and the algorithm will accept only those steps for which progress in the objective function is achieved.

The curvature test presented here requires only an inner-outer product of the step and the Hessian of the form $d_k^T W_k(\delta) d_k$ to check whether the direction is of descent and this can be computed inexpensively. Because this test does not require inertia information, one can use any type of solver available for the augmented system. This includes decomposition approaches, iterative solvers, and dense solvers. In addition, we avoid the need for additional factorizations in highly nonconvex problems and ill-posed problems, as we demonstrate in Section III.

C. Linear Algebra Structures

We now focus our attention on how structures arising in optimization models permeate down to the augmented system (II.4). Consider the NLP problem of the form

$$\min f_0(x_0) + \sum_{p \in \mathcal{P}} f_p(x_p, x_0) \quad (\text{II.11a})$$

$$\text{s.t.} \quad c_0(x_0) = 0 \quad (\lambda_0) \quad (\text{II.11b})$$

$$c_p(x_p, x_0) = 0, \quad p \in \mathcal{P} \quad (\lambda_p) \quad (\text{II.11c})$$

$$x_0 \geq 0 \quad (\nu_0) \quad (\text{II.11d})$$

$$x_p \geq 0, \quad p \in \mathcal{P} \quad (\nu_p). \quad (\text{II.11e})$$

Here, $\mathcal{P} := \{1 \dots P\}$ is a set of partitions with variables $x_p \in \mathbb{R}^{n_p}$ and the interface or coupling variables are $x_0 \in \mathbb{R}^{n_0}$. A similar notation is used for the multipliers. The linear algebra system of this problem can be permuted to the following block-bordered-diagonal (BBD) form,

$$\begin{bmatrix} K_0 & B_1^T & B_2^T & \dots & B_P^T \\ B_1 & K_1 & & & \\ B_2 & & K_2 & & \\ \vdots & & & \ddots & \\ B_P & & & & K_P \end{bmatrix} \begin{bmatrix} \Delta w_0 \\ \Delta w_1 \\ \Delta w_2 \\ \vdots \\ \Delta w_P \end{bmatrix} = - \begin{bmatrix} r_0 \\ r_1 \\ r_2 \\ \vdots \\ r_P \end{bmatrix} \quad (\text{II.12})$$

where $\Delta w_0 = (\Delta x_0, \Delta \lambda_0)$, $\Delta w_p = (\Delta x_p, \Delta \lambda_p)$,

$$K_0 = \begin{bmatrix} W_0(\delta) & J_0^T \\ J_0 & \end{bmatrix}, \quad K_p = \begin{bmatrix} W_p(\delta) & J_p^T \\ J_p & \end{bmatrix} \\ B_p = \begin{bmatrix} Q_p \\ T_p \end{bmatrix}, \quad (\text{II.13a})$$

$J_0 = \nabla_{x_0} c_0(x_0)$, $J_p = \nabla_{x_p} c_p(x_0, x_p)$, $T_p = \nabla_{x_0} c_p(x_0, x_p)$, $W_0(\delta) = \nabla_{x_0, x_0} \mathcal{L} + X_0^{-1} V_0 + \delta \mathbb{I}$, $W_p(\delta) = \nabla_{x_p, x_p} \mathcal{L} + X_p^{-1} V_p + \delta \mathbb{I}$, and $Q_p = \nabla_{x_0, x_p} \mathcal{L}$. The BBD system can be solved in parallel by forming the Schur complement and then solving the Schur system to obtain a step for the coupling variables Δw_0 . The step for the rest of the variables can then be recovered by solving block systems in parallel.

The BBD structure is typically exploited in security-constrained ACOPF and two-stage stochastic economic dispatch where each partition corresponds to a contingency or scenario and the coupling variables are the so-called first-stage, design, or here-and-now variables. This structure can also be induced by partitioning the network in deterministic ACOPF or economic dispatch models. One should bear in mind, however, that the computational efficiency obtained by exploiting the BBD structure is limited by the number of the coupling variables. If coupling is large, then Schur decomposition will not be efficient compared with direct factorization of the augmented matrix.

The BBD structure can be induced recursively. For instance, consider that each partition $p \in \mathcal{P}$ can be further partitioned into P_p partitions with associated sets \mathcal{P}_p , $p \in \mathcal{P}$. This approach creates the variable partitions $x_p = (x_{p,0}, x_{p,1}, \dots, x_{p,P_p})$, $p \in \mathcal{P}$ and partitions for objective and constraints as

$$f_p(x_0, x_p) = f_{p,0}(x_0, x_{p,0}) + \sum_{j \in \mathcal{P}_p} f_{p,j}(x_0, x_{p,0}, x_{p,j}), \quad p \in \mathcal{P} \quad (\text{II.14a})$$

$$c_p(x_0, x_p) = \begin{cases} c_{p,0}(x_0, x_{p,0}) & p \in \mathcal{P} \\ c_{p,j}(x_0, x_{p,0}, x_{p,j}), \quad j \in \mathcal{P}_p, \quad p \in \mathcal{P} \end{cases} \quad (\text{II.14b})$$

We thus obtain an NLP of the form

$$\min f_0(x_0) + \sum_{p \in \mathcal{P}} \left(f_{p,0}(x_0, x_{p,0}) + \sum_{j \in \mathcal{P}_p} f_{p,j}(x_0, x_{p,0}, x_{p,j}) \right) \quad (\text{II.15a})$$

$$\text{s.t.} \quad c_0(x_0) = 0 \quad (\lambda_0) \quad (\text{II.15b})$$

$$c_{p,0}(x_0, x_{p,0}) = 0, \quad p \in \mathcal{P} \quad (\lambda_{p,0}) \quad (\text{II.15c})$$

$$c_{p,j}(x_0, x_{p,0}, x_{p,j}) = 0, \quad j \in \mathcal{P}_p, \quad p \in \mathcal{P} \quad (\lambda_{p,j}) \quad (\text{II.15d})$$

$$x_0 \geq 0 \quad (\nu_0) \quad (\text{II.15e})$$

$$x_{p,0} \geq 0, \quad p \in \mathcal{P} \quad (\nu_{p,0}) \quad (\text{II.15f})$$

$$x_{p,j} \geq 0, \quad j \in \mathcal{P}_p, \quad p \in \mathcal{P} \quad (\nu_{p,j}). \quad (\text{II.15g})$$

One can show that this problem yields an augmented system of the form (II.12) but in which each diagonal block K_p has a BBD structure of the form:

$$K_p = \left[\begin{array}{c|ccc} K_{p,0} & B_{p,1}^T & B_{p,2}^T & \cdots & B_{p,P_p}^T \\ \hline B_{p,1} & K_{p,1} & & & \\ B_{p,2} & & K_{p,2} & & \\ \vdots & & & \ddots & \\ B_{p,P_p} & & & & K_{p,P_p} \end{array} \right], \quad p \in \mathcal{P}. \quad (\text{II.16})$$

Problems with recursive BBD structures arise in multistage stochastic optimization. In these problems, each scenario is further partitioned in multiple scenarios. Recursive BBD structures also arise in security-constrained ACOPT and stochastic economic dispatch if we partition the network in each contingency/scenario or in deterministic variants if we partition a subnetwork recursively into additional subnetworks.

Consider now the structure

$$\min f(x, u) \quad (\text{II.17a})$$

$$\text{s.t.} \quad c(x, u) = 0, \quad (\lambda) \quad (\text{II.17b})$$

$$x \geq 0 \quad (\nu_x) \quad (\text{II.17c})$$

$$u \geq 0, \quad (\nu_u). \quad (\text{II.17d})$$

In this structure, u is assumed to be of the same dimension as the number of degrees of freedom of the problem. In other words, if u is fixed, then $c(x, u) = 0$ is a square system of equations. The augmented system of this problem has the form,

$$\begin{bmatrix} W_{xx}(\delta) & W_{xu} & J_x^T \\ W_{ux} & W_{uu}(\delta) & J_u^T \\ J_x & J_u & \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta u \\ \Delta \lambda \end{bmatrix} = - \begin{bmatrix} r_x \\ r_u \\ r_\lambda \end{bmatrix}. \quad (\text{II.18})$$

By construction, the Jacobian $J_x = \nabla_x c(x, u)$ is square; and, if it is nonsingular, then we can construct the following null-space matrix:

$$Z = \begin{bmatrix} -J_x^{-1} J_u \\ I \end{bmatrix}. \quad (\text{II.19})$$

The step for u can then be obtained by solving a reduced system of the form $Z^T W(\delta) Z \Delta u = r_Z$ where r_Z is an appropriate right-hand side vector and $Z^T W(\delta) Z$ is the reduced Hessian. Having the step for u , we compute the step for x from $\Delta x = -J_x^{-1}(c_k + J_u \Delta u)$. Note that this approach requires factorizations of J_x and of the reduced Hessian $Z^T W(\delta) Z$ instead of factorizations of the entire augmented matrix $M_k(\delta)$.

Because of this, this approach can yield significant speed-ups when the number of degrees of freedom u is small.

The reduced space structure arises in PDE-constrained optimization in which $c(x, u) = 0$ is a set of discretized PDEs and u are the controls. We will see that this structure arises in optimal control of natural gas networks. This structure is encountered in DAE-constrained optimization problems such as those arising in predictive control for automatic generation control (AGC). In addition, note that most problems can be cast in reduced space form as long as the constraints are square and the Jacobian with respect to x is nonsingular. For instance, in ACOPT, we can choose active and reactive generation as controls u . If the network is connected, the power flow problem $c(x, u) = 0$, is square and well-posed and the Jacobian is nonsingular for fixed u .

We can now consider cases with embedded structures. As an example, consider problems arising in stochastic optimization of PDEs, stochastic predictive control, and security-constrained ACOPT. In these cases, the controls are coupling variables and also degrees of freedom. These problems have the following structure:

$$\min f_0(u_0) + \sum_{p \in \mathcal{P}} f_p(x_p, u_0) \quad (\text{II.20a})$$

$$\text{s.t.} \quad c_0(u_0) = 0 \quad (\lambda_0) \quad (\text{II.20b})$$

$$c_p(x_p, u_0) = 0, \quad p \in \mathcal{P} \quad (\lambda_p) \quad (\text{II.20c})$$

$$u_0 \geq 0 \quad (\nu_0) \quad (\text{II.20d})$$

$$x_p \geq 0, \quad p \in \mathcal{P} \quad (\nu_p). \quad (\text{II.20e})$$

Here, the Jacobian $\nabla_{x_p} c_p(x_p, u_0)$ is assumed to be nonsingular. In this case, each block in the BBD system (II.12) has the form in (II.18).

Many structures can be envisioned by combining the basic constructs presented here. For instance, one might consider security-constrained predictive control for AGC in which the network is partitioned in each contingency. One might also consider co-optimizing a natural gas and a power grid network in which each subnetwork is a partition with its own controls. We could consider adding a stochastic structure on top of these structures.

The ability to communicate structures to the optimization solver has other advantages beyond computational efficiency. In particular, it can enable a more modular model construction. For instance, if two or more modelers have access only to the data of their local network, they can express the problem independently and identify which are the coupling variables to solve a centralized problem. Note also that this approach is beneficial because optimization modeling languages (e.g., AMPL) can take a significant amount of time to process large-scale models. By specifying structures, a modeler also becomes more disciplined in the sense that one can validate that the model is well-posed. For instance, one can validate that the number of degrees of freedom is correct or that the square simulation model $c(x, u) = 0$ is well-posed. Such validation is important in complex models where the choice of degrees of freedom or partitioning of the network is not unique, as is the case in natural gas networks [10].

III. COMPUTATIONAL STUDIES

In this section we illustrate the performance of PIPS-NLP in challenging large-scale nonconvex problems. We first analyze performance of the curvature test strategy against inertia correction using security-constrained ACOPF problems. We demonstrate that inertia information is not needed to ensure convergence of the algorithm. We also demonstrate that, in some difficult cases, using inertia information can in fact lead to inefficiencies.

A. Security-Constrained ACOPF

We use a traditional security-constrained ACOPF formulation with polar constraints. We also consider two different objectives: total active generation cost and total voltage magnitude. Numerical studies are built based on the 6-, 24-, 57-, 118-, and 300-bus IEEE test systems from MATPOWER [12].

1) *Curvature Detection*: We first compare the performance of inertia detection (IC) with that of curvature detection (dWd). We get inertia information from the linear solver MA57 [13]. All results reported in this section are run on a standard computer with 10 GB of RAM, 2.7 GHz Intel Core i7 processor, running 64-bit Linux. Table I shows the number of iterations and number of regularization trials required for solving the test problems. For each case study, we test using real power generation cost (Gen) and total voltage magnitude (Vol) as objectives. If a test case does not require inertia correction or curvature correction we put a zero in the table to indicate that both strategies behave the same way. This case indicates that, while the problem is nonconvex, the solver does not encounter negative curvature along the search. When the value is non-zero, the fraction a/b denotes the number of regularizations necessary a and the number of interior-point (IPM) iterations b . As can be seen, the use of the total voltage objective induces more regularization. This indicates that this objective leads to ill-posedness (multiple solutions have similar objective values). The curvature detection strategy in some cases does not require regularization at all, whereas the inertia detection strategy does. Note, however, that in some cases curvature detection requires more iterations. While we cannot conclude that a particular strategy is superior, we can conclude that the curvature detection strategy remains robust even in the absence of inertia information, which is the key feature sought to enable structure exploitation.

TABLE I
NUMBER OF REGULARIZATIONS AND NUMBER OF IPM ITERATIONS FOR INERTIA AND CURVATURE DETECTION.

Problem	Contingencies	IC		dWd	
		Gen	Vol	Gen	Vol
IEEE-6	11	0	0	0	0
IEEE-24	37	0	0	0	0
IEEE-57	79	0	6/28	0	0/26
IEEE-118	177	11/47	6/43	3/52	13/63
IEEE-118	159	8/46	6/32	3/51	0/43

For the IEEE-300 system, we found that the inertia estimates obtained from MA57 varied significantly under different pivoting tolerances, whereas the curvature detection strategy remains robust. This situation occurs even when considering the traditional deterministic ACOPF problem. Results are shown in

Table II. These results are relevant because we can see that inertia information in some cases can be unreliable. In addition, the results indicate that the curvature detection strategy can enable faster solution times by relaxing the pivoting tolerance without compromising the convergence of the algorithm.

TABLE II
NUMBER OF IPM ITERATIONS AND EXTRA FACTORIZATIONS DUE TO REGULARIZATION (EXTRA FACT) AS A FUNCTION OF PIVOTING TOLERANCES FOR MA57 FOR IEEE-300.

Pivoting Tolerance	IC		dWd	
	Iter	Extra Fact	Iter	Extra Fact
1×10^{-8}	24	9	24	0
1×10^{-6}	24	15	24	0
1×10^{-4}	25	8	24	0
1×10^{-2}	24	11	24	0

2) *Scalability*: To demonstrate the scalability of PIPS-NLP, we parallelize the security-constrained ACOPF problem on the Fusion Cluster at Argonne National Laboratory by exploiting the BBD structure of the augmented system. We use the IEEE-118 problem with 159 contingencies as the case study. The scalability results are shown in Table III. We observe strong scaling, which indicates that the computational time decreases as we increase the number of processors by the same factor. Moreover, while the curvature detection strategy requires more iterations compared with inertia detection, it also requires less extra factorizations. For this case, we estimated the inertia of the BBD system using Haynsworth formula as proposed in [5].

TABLE III
SCALABILITY OF SECURITY-CONSTRAINED ACOPF FOR IEEE-118.

MPI Proc	dWd			IC		
	Iter	Extra Fact	Time(s)	Iter	Extra Fact	Time(s)
8	43	0	32.97	32	6	28.00
16	43	0	17.65	32	6	14.63
32	43	0	9.32	32	6	7.65
40	43	0	7.34	32	6	6.27
80	43	0	4.56	32	6	3.78
160	43	0	2.97	32	6	2.46

We also solved the IEEE-300 system with 271 contingencies. This problem contains 995,770 variables and 832,326 equality constraints. The results are presented in Table IV. We found that the Jacobian matrix of this problem is nearly rank-deficient and the algorithm thus requires significantly more iterations. We can see, however, that strong scaling is retained. We can also see that curvature detection requires more iterations but the solution time is less than that of inertia detection because significantly fewer regularizations are needed. In a parallel computing context this result is relevant because curvature detection reduces the number of times that the Schur system needs to be assembled and solved, which is the parallelization bottleneck (serial component).

B. Line-Pack Gas Dispatch

We now present scalability results for stochastic line-pack dispatch problems for natural gas networks. Here, the gas demands from power plants is assumed to be uncertain. These demands can be driven, for instance, by uncertainty in wind power capacity which in turn drives fast-ramping power plants.

TABLE IV
SCALABILITY OF SECURITY-CONSTRAINED ACOFP FOR IEEE-300.

MPI Proc	dWd			IC		
	Iter	Extra Fact	Time(s)	Iter	Extra Fact	Time(s)
16	185	23	607.17	100	145	641.32
136	162	10	72.19	100	144	89.10
272	155	16	41.57	100	143	63.18

The objective is to satisfy these uncertain gas demands for each possible scenario by building up storage (line-pack) inside the pipeline in such a way that it minimizes compression power. This problem can be cast as a stochastic optimal control problem in which the dynamic model is given by a complex set of partial differential equations (PDEs) describing the transport of gas in each pipeline in the network. We present here only the basic features of the problem. For a detailed description, we refer the reader to [10]. The dynamic equations for each link (pipeline) have the form:

$$\frac{\partial p_\ell}{\partial t} + \frac{1}{A_\ell} \frac{p_\ell}{\rho_\ell} \frac{\partial f_\ell}{\partial x} = 0 \quad (\text{III.21a})$$

$$\frac{1}{A_\ell} \frac{\partial f_\ell}{\partial t} + \frac{\partial p_\ell}{\partial x} + \frac{8\lambda_\ell}{\pi^2 D_\ell^5} \frac{f_\ell |f_\ell|}{\rho_\ell} = 0 \quad (\text{III.21b})$$

for all $\ell \in \mathcal{L}$ with boundary conditions,

$$f_\ell|_{x=0} = f_\ell^{in} \quad (\text{III.22a})$$

$$f_\ell|_{x=L_\ell} = f_\ell^{out} \quad (\text{III.22b})$$

$$p_\ell|_{x=L_\ell} = \theta_{rec(\ell)} \quad (\text{III.22c})$$

$$p_\ell|_{x=0} = \theta_{snd(\ell)}, \ell \in \mathcal{L}_p \quad (\text{III.22d})$$

$$p_\ell|_{x=0} = \theta_{snd(\ell)} + \Delta\theta_\ell, \ell \in \mathcal{L}_a. \quad (\text{III.22e})$$

Here, $p_\ell, f_\ell, \rho_\ell$ are the link pressures, flows, and densities, respectively. These variables are function of space and time. The set of links \mathcal{L} is split into passive \mathcal{L}_p (with no compression) and active links \mathcal{L}_a (with compression). The inlet and outlet flows for each link are f_ℓ^{in} and f_ℓ^{out} . The node pressures are given by θ_n . The flows of the pipelines are connected through a network balance of the form:

$$\sum_{\ell:rec(\ell)=n} f_\ell^{out} + \sum_{i:sup(i)=n} s_i - \sum_{\ell:snd(\ell)=n} f_\ell^{in} - \sum_{j:dem(j)=n} d_j(\omega) = 0. \quad (\text{III.23})$$

Here, s_i are supply flows and $d_j(\omega)$ are the stochastic demand flows in scenario ω . The power consumed by each compressor is given by:

$$P_\ell = f_\ell^{in} c_p T \left(\left(\frac{\theta_{snd(\ell)} + \Delta\theta_\ell}{\theta_{snd(\ell)}} \right)^{\frac{\gamma-1}{\gamma}} - 1 \right). \quad (\text{III.24})$$

As can be seen, this problem is nonlinear and computationally intensive because of the large number of PDEs. This problem has the key property, however, that the number of degrees of freedom (controls) is small compared with the complexity of the PDEs. The controls are the boost pressures $\Delta\theta_\ell$, $\ell \in \mathcal{L}_a$. In addition, it has the natural BBD structure induced by stochastic optimization formulations in which the boost pressures are the

here-and-now variables coupling all the scenarios (we make one decision for the controls for all scenarios). In the stochastic case we replicate the PDE equations for each scenario. The gas line-pack problem also presents network structure that can be exploited by partitioning.

To demonstrate the performance of PIPS-NLP, we first consider a system with 13 nodes and 12 pipelines and 10 compressors. The system spans 1,600 km and the pipelines have diameters of 0.9 m. We present computational results for this problem under two settings: in the first case we exploit only the stochastic structure (Table V) while in the second case we exploit the stochastic and the reduced space structure of the problem (Table VI). This problem has 96 scenarios and a total of 1,930,752 variables. Strong scaling is observed in both cases and exploiting the reduced space structure decreases the computational time by a factor of nearly 3. We also note that despite the nonlinearity of the equations, the algorithm did not require regularization in this case.

TABLE V
SCALABILITY OF PIPS-NLP EXPLOITING STOCHASTIC STRUCTURE.

Scenarios	n	Obj	Iter	Time(hh:mm:ss)	MPI Proc
96	1,930,752	1.39×10^2	42	01:13:16	8
96	1,930,752	1.39×10^2	42	00:38:18	16
96	1,930,752	1.39×10^2	42	00:24:55	24
96	1,930,752	1.39×10^2	42	00:19:23	32
96	1,930,752	1.39×10^2	42	00:12:42	48
96	1,930,752	1.39×10^2	42	00:06:48	96

TABLE VI
SCALABILITY OF PIPS-NLP EXPLOITING STOCHASTIC AND REDUCED SPACE STRUCTURE.

Scenarios	n	Obj	Iter	Time(hh:mm:ss)	MPI Proc
96	1,930,752	1.39×10^2	42	00:29:54	8
96	1,930,752	1.39×10^2	42	00:14:45	16
96	1,930,752	1.39×10^2	42	00:10:00	24
96	1,930,752	1.39×10^2	42	00:07:36	32
96	1,930,752	1.39×10^2	42	00:05:14	48
96	1,930,752	1.39×10^2	42	00:02:54	96

We now illustrate the performance of PIPS-NLP in a deterministic line-pack dispatch problem in which we partition the network. This study is done for the Belgium natural gas network system presented in Figure 1 [2]. This problem is significantly more complex but, as can be seen, the network can be partitioned trivially at the Perennes node and the resulting subnetworks are connected. The largest problem solved has around 1.2 million variables per partition (2.4 million total) for the deterministic case. We study the performance of PIPS-NLP as we increase the resolution of the discretization of the PDEs. We denote as N_x as the number of discretization points. We compare the performance of the direct factorization of MA57 with that of Schur decomposition. The computational times are presented in Table VII. As can be seen, for low discretization resolutions both strategies have essentially the same computational times. When the discretization resolution is high, however, the factorization time of MA57 becomes significant, and network partitioning becomes beneficial. Note that increasing N_x has an effect similar to that of increasing the size of the network. This indicates that partitioning can become valuable in tackling very large networks.

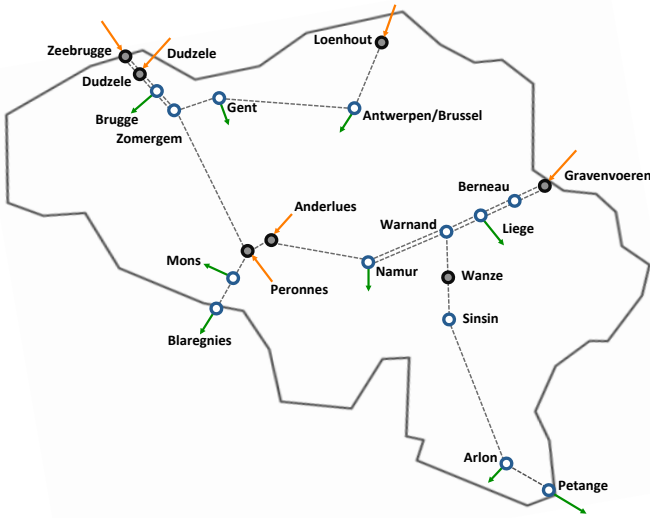


Fig. 1. Belgium natural gas network. Dark dots are compression stations.

TABLE VII
COMPUTATIONAL TIMES FOR LINE-PACK DISPATCH OF BELGIUM NATURAL GAS NETWORK.

N_x	MA57 Time (hr:mm:ss)	Schur Time (hh:mm:ss)
50	00:00:08	00:00:08
100	00:00:19	00:00:19
500	00:03:24	00:02:46
1000	00:13:17	00:07:33
3000	01:50:08	00:53:32

IV. CONCLUSIONS AND FUTURE WORK

We have presented PIPS-NLP, a scalable solver that exploits modular linear algebra structures in nonconvex optimization problems. We have provided computational studies arising in energy systems to demonstrate that the solver is robust and scalable. A crucial feature of PIPS-NLP is that it does not require inertia information, which enables the use of different linear algebra libraries. As part of future work we will seek to demonstrate scalability in multilevel implementations including multistage stochastic problems and recursive network partitioning.

ACKNOWLEDGMENTS

This work was supported by the U.S. Department of Energy, under Contract No. DE-AC02-06CH11357.

REFERENCES

- [1] S BALAY, J BROWN, K BUSCHELMAN, V EIJKHOUT, W GROPP, D KAUSHIK, M KNEPLEY, L CURFMAN MCINNES, B SMITH, AND H ZHANG, *PETSc users manual revision 3.4*, (2013).
- [2] D.L DE WOLF AND Y. SMEERS, *The gas transmission problem solved by an extension of the simplex algorithm*, *Management Science*, 46 (2000), pp. 1454–1465.
- [3] K. EHRHARDT AND M. C. STEINBACH, *Nonlinear optimization in gas networks*, Springer, 2005.
- [4] K. KAROUI, L. PLATBROOD, H. CRISCIU, AND R.A. WALTZ, *New large-scale security constrained optimal power flow program using a new interior point algorithm*, in 5th International Conference on European Electricity Market, 2008, pp. 1–6.
- [5] C. D LAIRD AND L. T. BIEGLER, *Large-scale nonlinear programming for multi-scenario optimization*, in Modeling, Simulation and Optimization of Complex Processes, Springer, 2008, pp. 323–336.
- [6] M. LUBIN, C. G. PETRA, M. ANITESCU, AND V. M. ZAVALA, *Scalable stochastic optimization of complex energy systems*, in Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis, SC '11, New York, USA, 2011, ACM, pp. 64:1–64:64.
- [7] W. QIU, A. J. FLUECK, AND F. TU, *A new parallel algorithm for security constrained optimal power flow with a nonlinear interior point method*, in IEEE Power Engineering Society General Meeting, 2005, pp. 447–453.
- [8] F. SONG, S. TOMOV, AND J. DONGARRA, *Enabling and scaling matrix computations on heterogeneous multi-core and multi-gpu systems*, in 6th ACM International Conference on Supercomputing, (ICS 2012), New York, NY, USA, 2012, ACM.
- [9] A. WÄCHTER AND L. T. BIEGLER, *On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming*, *Mathematical programming*, 106 (2006), pp. 25–57.
- [10] V. M. ZAVALA, *Stochastic optimal control model for natural gas networks*, *Computers & Chemical Engineering*, 64 (2014), pp. 103–113.
- [11] 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 (2008), pp. 4834–4845.
- [12] R.D. ZIMMERMAN, C.E. MURILLO-SANCHEZ, AND R.J. THOMAS, *Matpower: Steady-state operations, planning, and analysis tools for power systems research and education*, *Power Systems*, IEEE Transactions on, 26 (2011), pp. 12–19.
- [13] *HSL: A collection of Fortran codes for large scale scientific computation*. <http://www.hsl.rl.ac.uk>, 2011.