
Clustering-Based Interior-Point Strategies for Stochastic Programs

Victor M. Zavala

Received: date / Accepted: date

Abstract We present interior-point strategies for convex stochastic programs in which inexpensive inexact Newton steps are computed from compressed Karush-Kuhn-Tucker (KKT) systems obtained by clustering block scenarios. Using Schur analysis, we show that the compression can be characterized as a parametric perturbation of the full-space KKT matrix. This property enables the possibility of retaining superlinear convergence without requiring matrix convergence. In addition, it enables an explicit characterization of the residual and we use this characterization to derive a clustering strategy. We demonstrate that high compression rates of 50-90% are possible and we also show that effective preconditioners can be obtained.

Keywords interior-point · stochastic · large-scale · clustering.

1 Introduction and Basic Notation

We consider convex stochastic programs of the following form

$$\min \varphi := \left(\frac{1}{2} y_0^T Q_0 y_0 + d_0^T y_0 \right) + S^{-1} \sum_{s \in \mathcal{S}} \left(\frac{1}{2} y_s^T Q_s y_s + d_s^T y_s \right) \quad (1a)$$

$$\text{s.t.} \quad W_0 y_0 = b_0, \quad (\lambda_0) \quad (1b)$$

$$T_s y_0 + W_s y_s = b_s, \quad (\lambda_s), \quad s \in \mathcal{S} \quad (1c)$$

$$y_0 \geq 0, \quad (\nu_0) \quad (1d)$$

$$y_s \geq 0, \quad (\nu_s), \quad s \in \mathcal{S}. \quad (1e)$$

Here, $\mathcal{S} := \{1..S\}$ and $S = |\mathcal{S}|$, $y_0, \nu_0 \in \mathbb{R}^{n_0}$, $y_s, \nu_s \in \mathbb{R}^{n_s}$, $\lambda_0 \in \mathbb{R}^{m_0}$, and $\lambda_s \in \mathbb{R}^{m_s}$. The total number of variables is $n := n_0 + \sum_{s \in \mathcal{S}} n_s$, of equality constraints is $m := m_0 + \sum_{s \in \mathcal{S}} m_s$, and of inequalities is n . The matrices $Q_0, Q_s, s \in \mathcal{S}$ are positive semi-definite, but the strategies derived

Preprint Number ANL/MCS-P3050-1112

Victor M. Zavala

Mathematics and Computer Science Division, Argonne National Laboratory
9700 South Cass Avenue, Argonne, IL 60439

Tel.: +1-630-252-3343

Fax: +1-630-252-5986

E-mail: vzavala@mcs.anl.gov

can also handle linear programs (LPs). In the following, we will refer to y_0 as the first-stage variables and to $y_s, s \in \mathcal{S}$ as the second-stage variables.

We assume that the scenarios are generated by sampling an underlying probability distribution such as in a sample-average approximation setting [23]. As is typical in stochastic optimization, the number of scenarios required to achieve desired accuracies can be large and limits the scope of existing off-the-shelf algorithms. In this work, we present strategies that compress or cluster the scenarios adaptively inside the solver (at the linear algebra level) in order to reduce the complexity.

To start the discussion, we derive some necessary notation and construct the linear algebra setting. The Lagrange function of (1) is given by

$$\begin{aligned} \mathcal{L}(y, \lambda, \nu) = & \frac{1}{2} y_0^T Q_0 y_0 + d_0^T y_0 + \lambda_0^T (W_0 y_0 - b_0) - \nu_0^T y_0 \\ & + S^{-1} \sum_{s \in \mathcal{S}} \left(\frac{1}{2} y_s^T Q_s y_s + d_s^T y_s + \lambda_s^T (T_s y_0 + W_s y_s - b_s) - \nu_s^T y_s \right). \end{aligned} \quad (2)$$

Here, $\mathbf{y} := [y_0^T, y_1^T, \dots, y_S^T]$, $\lambda^T := [\lambda_0^T, \lambda_1^T, \dots, \lambda_S^T]$, and $\nu^T := [\nu_0^T, \nu_1^T, \dots, \nu_S^T]$. Note that the multipliers $\lambda_s, \nu_s, s \in \mathcal{S}$ have been implicitly scaled by S^{-1} . In a typical primal-dual interior-point (IP) setting, the Karush-Kuhn-Tucker (KKT) conditions are solved by relaxing the complementarity conditions. This gives the system,

$$\nabla_{y_0} \mathcal{L} = 0 = Q_0 y_0 + d_0 + A_0^T \lambda_0 - \nu_0 + S^{-1} \sum_{s \in \mathcal{S}} T_s^T \lambda_s \quad (3a)$$

$$\nabla_{y_s} \mathcal{L} = 0 = Q_s y_s + d_s + W_s^T \lambda_s - \nu_s, \quad s \in \mathcal{S} \quad (3b)$$

$$\nabla_{\lambda_0} \mathcal{L} = 0 = W_0 y_0 - b_0 \quad (3c)$$

$$\nabla_{\lambda_s} \mathcal{L} = 0 = T_s y_0 + W_s y_s - b_s, \quad s \in \mathcal{S} \quad (3d)$$

$$0 = Y_0 V_0 e - \mu \quad (3e)$$

$$0 = Y_s V_s e - \mu, \quad s \in \mathcal{S}, \quad (3f)$$

together with the implicit condition $y_0, \nu_0, y_s, \nu_s \geq 0$. Here, $\mu \geq 0$ is the barrier parameter, e is a vector of ones of appropriate dimension, $Y_0 = \text{diag}(y_0)$, $Y_s := \text{diag}(y_s)$, $V_0 = \text{diag}(\nu_0)$, and $V_s = \text{diag}(\nu_s)$. We define $\alpha_0 := Y_0 V_0 e - \mu$ and $\alpha_s := Y_s V_s e - \mu$, $s \in \mathcal{S}$. The search step is obtained by solving the *full-space* KKT system

$$Q_0 \Delta y_0 + A_0^T \Delta \lambda_0 - \Delta \nu_0 + S^{-1} \sum_{s \in \mathcal{S}} T_s^T \Delta \lambda_s = -\nabla_{y_0} \mathcal{L} \quad (4a)$$

$$Q_s \Delta y_s + W_s^T \Delta \lambda_s - \Delta \nu_s = -\nabla_{y_s} \mathcal{L}, \quad s \in \mathcal{S} \quad (4b)$$

$$W_0 \Delta y_0 = -\nabla_{\lambda_0} \mathcal{L} \quad (4c)$$

$$T_s \Delta y_0 + W_s \Delta y_s = -\nabla_{\lambda_s} \mathcal{L}, \quad s \in \mathcal{S} \quad (4d)$$

$$Y_0 \Delta \nu_0 + V_0 \Delta y_0 = -\alpha_0 \quad (4e)$$

$$Y_s \Delta \nu_s + V_s \Delta y_s = -\alpha_s, \quad s \in \mathcal{S}. \quad (4f)$$

This system has the well-known arrowhead form,

$$\begin{bmatrix} K_1 & & & B_1 \\ & K_2 & & B_2 \\ & & \ddots & \vdots \\ & & & K_S & B_S \\ B_1^T & B_2^T & \dots & B_S^T & K_0 \end{bmatrix} \begin{bmatrix} \Delta w_1 \\ \Delta w_2 \\ \vdots \\ \Delta w_S \\ \Delta w_0 \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_S \\ r_0 \end{bmatrix}, \quad (5)$$

with $\Delta w_0^T := [\Delta y_0^T, \Delta \lambda_0^T, \Delta \nu_0^T]$, $\Delta w_s^T := [\Delta y_s^T, \Delta \lambda_s^T, \Delta \nu_s^T]$, $r_0^T := [r_{y_0}^T, r_{\lambda_0}^T, \alpha_0^T]$, $r_s^T := [r_{y_s}^T, r_{\lambda_s}^T, \alpha_s^T]$, and

$$K_0 := \begin{bmatrix} Q_0 & A_0^T & -I \\ W_0 & 0 & 0 \\ V_0 & 0 & Y_0 \end{bmatrix}, \quad K_s := \begin{bmatrix} Q_s & W_s^T & -I \\ W_s & 0 & 0 \\ V_s & 0 & Y_s \end{bmatrix}, \quad B_s := \begin{bmatrix} 0 & 0 & 0 \\ T_s & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \quad (6)$$

In compact form,

$$\Phi_w(w) \Delta w = -\Phi(w, \mu). \quad (7)$$

Here, $\Phi(\cdot, \cdot)$ are the KKT conditions (3), and $\Phi_w(\cdot) = \nabla_w \Phi(\cdot)$ is the left-hand side matrix in (5). We also define a permutation matrix Π satisfying $(\Pi w)^T = [y^T, \nu^T]$.

The above representation is convenient for analysis. For implementation, however, an augmented system approach might be more convenient as storage requirements are reduced and symmetric linear algebra solvers can be used. After eliminating the bound multipliers from the KKT system (5), we obtain

$$\bar{Q}_0 \Delta y_0 + A_0^T \Delta \lambda_0 + S^{-1} \sum_{s \in \mathcal{S}} T_s^T \Delta \lambda_s = -r_{y_0} \quad (8a)$$

$$\bar{Q}_s \Delta y_s + W_s^T \Delta \lambda_s = -r_{y_s}, \quad s \in \mathcal{S} \quad (8b)$$

$$W_0 \Delta y_0 = -r_{\lambda_0} \quad (8c)$$

$$T_s \Delta y_0 + W_s \Delta y_s = -r_{\lambda_s}, \quad s \in \mathcal{S}, \quad (8d)$$

where $\bar{Q}_0 := Q_0 + Y_0^{-1} V_0$, $\bar{Q}_s := Q_s + Y_s^{-1} V_s$, $r_{y_0} := \nabla_{y_0} \mathcal{L} - Y_0^{-1} \alpha_0$, $r_{y_s} := \nabla_{y_s} \mathcal{L} - Y_s^{-1} \alpha_s$, $r_{\lambda_0} := \nabla_{\lambda_0} \mathcal{L}$, and $r_{\lambda_s} := \nabla_{\lambda_s} \mathcal{L}$. The bound multipliers are recovered from

$$\Delta \nu_0 = -Y_0^{-1} V_0 \Delta y_0 - Y_0^{-1} \alpha_0 \quad (9a)$$

$$\Delta \nu_s = -Y_s^{-1} V_s \Delta y_s - Y_s^{-1} \alpha_s, \quad s \in \mathcal{S}. \quad (9b)$$

The elimination yields an arrowhead system of the form (5) with $\Delta w_0^T := [\Delta y_0^T, \Delta \lambda_0^T]$, $\Delta w_s^T := [\Delta y_s^T, \Delta \lambda_s^T]$, $r_0^T := [r_{y_0}^T, r_{\lambda_0}^T]$, $r_s^T := [r_{y_s}^T, r_{\lambda_s}^T]$, and

$$K_0 := \begin{bmatrix} \bar{Q}_0 & A_0^T \\ W_0 & 0 \end{bmatrix}, \quad K_s := \begin{bmatrix} \bar{Q}_s & W_s^T \\ W_s & 0 \end{bmatrix}, \quad B_s := \begin{bmatrix} 0 & 0 \\ T_s & 0 \end{bmatrix}. \quad (10)$$

In this case, we redefine the full iterate vector obtained from the solution of the joint system (5)-(9) as $w^T \leftarrow [w^T, \nu_0^T, \nu_1^T, \dots, \nu_S^T]$. A similar derivation can be made for the normal decomposition approach which is particularly attractive for linear programs and diagonal matrices Q_s . In this case, we have an arrowhead system with

$$K_0 := W_0 \bar{Q}_0^{-1} W_0^T, \quad K_s := W_s \bar{Q}_s^{-1} W_s^T, \quad B_s := T_s \bar{Q}_0^{-1}. \quad (11)$$

2 Interior-Point Framework

We seek to reduce the complexity of the full-space KKT system (7) by allowing this to be solved inexactly. That is, we seek to generate much cheaper approximate steps satisfying

$$\Phi_w(w) \Delta w = -\Phi(w, \mu) + \delta(\Delta w), \quad (12)$$

where $\delta(\Delta w)$ is the residual vector induced by the step Δw . This will be done by creating a lower dimensional representation of the KKT system.

2.1 Inexact Step Computation

We propose to generate the compression of the KKT system by *clustering* the scenario blocks. Scenario clustering or aggregation is a strategy commonly used in stochastic optimization. Existing strategies, however, perform elimination prior to the solution of the problem [9, 16, 14, 5]. In other words, this is done outside the solver. This approach can be of great advantage because lower bounds and error bounds can be derived and exploited to refine the solution. This has been proposed in the context of stochastic LPs in [5, 2] and in a more general setting in [25, 29]. A drawback of this approach is that it can lead to significant inefficiency. The reason is that several problems might need to be solved as the clusters are refined. We seek to overcome these inefficiencies by performing clustering inside the solver. This will also enable the possibility to apply aggregation in a more general setting.

To derive the inexact step computation based on clustering, we partition the full-space scenario set \mathcal{S} into C clusters where $C \leq S$. Each cluster $i \in \mathcal{C} := \{1..C\}$ comprises a set of indexes $\mathcal{J}_i \in \mathcal{S}$ with $|\mathcal{J}_i|$ scenarios, and we have

$$\bigcup_{i \in \mathcal{C}} \mathcal{J}_i = \mathcal{S}. \quad (13)$$

Within each cluster $i \in \mathcal{C}$ we pick an index $j_i \in \mathcal{J}_i$ corresponding to the scenario representing the cluster¹, and we define $\mathcal{R} := \{j_1..j_C\}$ as the set of *remaining* or compressed scenarios. We will use the corresponding blocks $K_i, B_i, i \in \mathcal{R}$ to assemble a compressed KKT system. Note that $|\mathcal{R}| = C$, and in the limit we have $C = |\mathcal{S}| = |\mathcal{R}|$ and $\mathcal{J}_i, i \in \mathcal{C}$ become singletons. For each cluster we also define a scalar cluster weight $\omega_i = |\mathcal{J}_i|$, $i \in \mathcal{C}$. We also define the set of eliminated scenarios $\mathcal{E} := \mathcal{S} \setminus \mathcal{R}$.

Consider the following representation of the full-space KKT system (5)

$$\begin{bmatrix} K_{\mathcal{S}} & B_{\mathcal{S}} \\ B_{\mathcal{S}}^T & K_0 \end{bmatrix} \begin{bmatrix} \Delta w_{\mathcal{S}} \\ \Delta w_0 \end{bmatrix} = \begin{bmatrix} r_{\mathcal{S}} \\ r_0 \end{bmatrix}. \quad (14)$$

Here, $K_{\mathcal{S}}$ is a block diagonal matrix with entries $K_i, i \in \mathcal{S}$. The border matrix $B_{\mathcal{S}}$ stacks columnwise the matrices $B_s, s \in \mathcal{S}$, and the vectors $r_{\mathcal{S}}$ and $\Delta w_{\mathcal{S}}$ stacks the corresponding vectors.

We now derive the compressed KKT system and present approximation properties of the full-space system. The inexact step is computed from the compressed KKT system

$$\begin{bmatrix} \Omega_{\mathcal{R}}^{-1} K_{\mathcal{R}} & B_{\mathcal{R}} \\ B_{\mathcal{R}}^T & K_0 \end{bmatrix} \begin{bmatrix} \Delta w_{\mathcal{R}} \\ \Delta w_0 \end{bmatrix} = \begin{bmatrix} \Omega_{\mathcal{R}}^{-1} r_{\mathcal{R}} \\ r_0 - B_{\mathcal{E}}^T K_{\mathcal{E}}^{-1} r_{\mathcal{E}} \end{bmatrix}, \quad (15)$$

where $\Omega_{\mathcal{R}} = \text{diag}(\omega_r I \mid r \in \mathcal{R})$. The step for the eliminated scenarios is recovered from

$$\Delta w_{\mathcal{E}} = K_{\mathcal{E}}^{-1} (r_{\mathcal{E}} - K_0 \Delta w_0). \quad (16)$$

2.2 Clustering

To see the motivation behind the proposed compressed system (15), we consider its Schur decomposition and that of the full-space system (14). For the full-space system we have

$$(K_0 - B_{\mathcal{S}}^T K_{\mathcal{S}}^{-1} B_{\mathcal{S}}) \Delta w_0^* = r_0 - B_{\mathcal{S}}^T K_{\mathcal{S}}^{-1} r_{\mathcal{S}} \quad (17)$$

¹ This can be, for instance, the closest point to the cluster centroid. An alternative and much simpler approach is to pick a random point in the cluster.

or, in compact form, $Z\Delta w_0^* = r_Z$ with $Z := K_0 - B_S^T K_S^{-1} B_S$ and $r_Z := r_0 - B_S^T K_S^{-1} r_S$. This system will be referred to as the exact Schur system, which yields the exact step Δw_0^* . Schur decomposition implicitly yields the step for the second-stage variables,

$$\Delta w_S^* = K_S^{-1}(r_S - K_0 \Delta w_0^*). \quad (18)$$

A Schur decomposition for the compressed system yields

$$\begin{aligned} (K_0 - \Omega_{\mathcal{R}} B_{\mathcal{R}}^T K_{\mathcal{R}}^{-1} B_{\mathcal{R}}) \Delta w_0 &= r_0 - B_{\mathcal{R}}^T K_{\mathcal{R}}^{-1} r_{\mathcal{R}} - B_{\mathcal{E}}^T K_{\mathcal{E}}^{-1} r_{\mathcal{E}} \\ &= r_0 - B_S^T K_S^{-1} r_S \\ &= r_Z, \end{aligned} \quad (19)$$

In compact form we have $\bar{Z}\Delta w_0 = r_Z$ with $\bar{Z} := K_0 - \Omega_{\mathcal{R}} B_{\mathcal{R}}^T K_{\mathcal{R}}^{-1} B_{\mathcal{R}}$. This system will be referred to as the inexact or compressed Schur system. Note the exact and inexact Schur system have the same right-hand side.

The Schur decomposition of the compressed system implicitly gives the step for the second-stage variables,

$$\Delta w_S = K_S^{-1}(r_S - K_0 \Delta w_0). \quad (20)$$

The inexact step Δw_0 induces a residual on the exact Schur system $Z\Delta w_0 = r_Z + \delta_Z(\Delta w_0)$. In Section 3 we analyze this perturbation in the context of the full-space KKT system (14). A direct calculation yields an explicit characterization of the residual induced by the inexact step Δw_0 on the exact Schur system. We have,

$$\begin{aligned} \delta_Z(\Delta w_0) &= Z\Delta w_0 - r_Z \\ &= (Z - \bar{Z})\Delta w_0 - r_Z + \bar{Z}\Delta w_0 \\ &= (Z - \bar{Z})\Delta w_0 \\ &= (\Omega_{\mathcal{R}} B_{\mathcal{R}}^T K_{\mathcal{R}}^{-1} B_{\mathcal{R}} - B_S^T K_S^{-1} B_S) \Delta w_0 \\ &= \sum_{i \in \mathcal{C}} \sum_{k \in \mathcal{J}_i} (B_{j_i}^T K_{j_i}^{-1} B_{j_i} - B_k^T K_k^{-1} B_k) \Delta w_0. \end{aligned} \quad (21)$$

The above relationship only holds for the definition of the weighting matrix $\Omega_{\mathcal{R}}$ corresponding to the cluster weights $\|\mathcal{J}_i\|$, $i \in \mathcal{C}$. We define

$$d_i(\Delta w_0) := \sum_{k \in \mathcal{J}_i} (B_{j_i}^T K_{j_i}^{-1} B_{j_i} - B_k^T K_k^{-1} B_k) \Delta w_0, \quad i \in \mathcal{C}. \quad (22)$$

With this, $\delta_Z(\Delta w_0) = \sum_{i \in \mathcal{C}} d_i(\Delta w_0)$. We also define the residual contributions,

$$\gamma_s(\Delta w_0) = B_s^T K_s^{-1} B_s \Delta w_0, \quad s \in \mathcal{S}. \quad (23)$$

We define the contribution corresponding to the cluster centroids as $\gamma_i^c(\cdot) := \gamma_{j_i}(\cdot)$, $i \in \mathcal{C}$. Using these definitions we have

$$d_i(\Delta w_0) = \sum_{k \in \mathcal{J}_i} (\gamma_i^c(\Delta w_0) - \gamma_k(\Delta w_0)), \quad i \in \mathcal{C}. \quad (24)$$

This quantity is a *dissimilarity metric* that measures the spread of the cluster elements. In the ideal case where, for each cluster $i \in \mathcal{C}$, all of its elements are equal we have $d_i(\Delta w_0) = 0$, $i \in \mathcal{C}$ and $(Z - \bar{Z})\Delta w_0 = 0$. Also, if the spread of the elements of the cluster is small, then the residual $\delta_Z(\Delta w_0)$ will be small. In other words, the compressed KKT system (15) gives consistent behavior.

2.2.1 Clustering Strategy

The explicit characterization of the residual (21) enables us to derive numerical strategies to compute clusters. Our objective becomes clear: *We seek to compress the scenarios in such a way that it minimizes the residual $\delta_Z(\Delta w_0)$.* This is achieved by minimizing $(Z - \bar{Z}) \cdot u$ along a given direction u . In addition, we want to perform the compression inexpensively.

We consider the following strategy. For a given direction u we compute the residual contributions $\gamma_s(u)$, $s \in \mathcal{S}$ from (23). These vectors are used as scenario *features* in performing numerical clustering. Numerical clustering strategies such as hierarchical, k -means, and fuzzy clustering can be used for this. For a review of different techniques see [15]. Clustering algorithms explicitly or implicitly seek to minimize the *distortion metric*

$$J(r_{s,i}, \bar{\gamma}_i, \gamma_s(u)) := \sum_{s \in \mathcal{S}} \sum_{i \in \mathcal{C}} r_{s,i} \|\bar{\gamma}_i - \gamma_s(u)\|^2. \quad (25)$$

Where $\|\cdot\|$ is the Euclidean norm and $r_{s,i} \in \{0, 1\}$, $s \in \mathcal{S}$, $i \in \mathcal{C}$ are indicator variables such that a value of one indicates that vector s belongs to cluster i . For instance, given a set of feature vectors $\gamma_s(\cdot)$, $s \in \mathcal{S}$, the k -means algorithm seeks for the entries $r_{s,i}$ and centroids $\bar{\gamma}_i$ that minimize the distortion measure. The minimization is only approximate as the problem is NP-hard; efficient heuristic algorithms, however, are available.

The following result establishes a connection between the residual and the distortion metric.

Theorem 1 *Assume (i) a given step u and cluster information \mathcal{R} and $\omega_{\mathcal{R}}$. In addition, assume that (ii) for each $i \in \mathcal{C}$ there exists an index $j_i \in \mathcal{R}$ satisfying $\bar{\gamma}_i = \gamma_{j_i}^c(u)$. Then, the norm of the residual $\delta_Z(u)$ and the distortion metric satisfy $\|\delta_Z(u)\|^2 \leq J(r_{s,i}, \bar{\gamma}_i, \gamma_s(u))$.*

Bounding the residual $\delta_Z(u)$ we obtain

$$\begin{aligned} \|\delta_Z(u)\|^2 &= \|(Z - \bar{Z})u\|^2 \\ &= \left\| \sum_{i \in \mathcal{C}} d_i(u) \right\|^2 \\ &\leq \sum_{i \in \mathcal{C}} \|d_i(u)\|^2 \\ &\leq \sum_{i \in \mathcal{C}} \sum_{k \in \mathcal{J}_i} \|\gamma_i^c(u) - \gamma_k(u)\|^2 \\ &= \sum_{s \in \mathcal{S}} \sum_{i \in \mathcal{C}} r_{s,i} \|\gamma_i^c(u) - \gamma_s(u)\|^2. \end{aligned} \quad (26)$$

The proof is complete by using assumption (ii) and definition (25). \square

This result implies that applying a clustering scheme to the residual contributions $\gamma_s(u)$, $s \in \mathcal{S}$ is an implicit approach to approximately minimize the residual $\delta_Z(u)$. The clustering procedure leads to the remaining set \mathcal{R} with weighting $\omega_{\mathcal{R}}$ and to the predicted residual $\delta_p(u) = Zu - r_Z$. Note that the residual can only be predicted as the vector u is not known a priori because this is given by the actual step Δw_0 computed. For direction u , we propose to use the step Δw_0 at the *previous* iteration. This choice will yield a predicted residual that we hope will yield a good representation of the residual induced by the actual inexact step Δw_0 computed. This can be expected asymptotically as we approach the solution but this will not be necessarily the case early in the search. The hope is thus that a using given direction u at least allows us to identify contributions of different magnitudes. The option of applying clustering directly on the matrices $B_s^T K_s^{-1} B_s$ so as to minimize $(Z - \bar{Z})$ is prohibitively expensive. Mixed-integer formulations are in principle also possible but we leave this as subject of future research.

2.2.2 Comparison with Scenario Elimination

An interpretation of the clustering procedure is that it seeks to eliminate blocks $B_s^T K_s^{-1} B_s$ that generate similar contributions to the Schur complement. In other words, the blocks have similar actions along the direction of the first-stage step Δw_0 .

Another related observation is that some scenarios have significantly more impact than others. This is typically observed close to the solution as activity in different scenarios reveals and it is manifested through the magnitudes of contributions $B_s^T K_s^{-1} B_s$. Consequently, it makes sense to try to eliminate scenarios that lead to small contributions. In particular, when a given scenario has a larger share of active inequalities, then the corresponding block will dominate the Schur complement. To understand this behavior, consider the normal decomposition approach. In this case, we have $K_s = W_s(Q_s + Y_s^{-1}V_s)^{-1}W_s^T$. If there is little activity in this scenario then $B_s^T K_s^{-1} B_s$ tends to be small because some or most entries of V_s tend to zero. This property has been exploited in a more general setting for constraint elimination in interior point solvers [26] and in the context of stochastic programs for scenario elimination [6]. Scenario elimination from the KKT system based on random sampling has also been recently proposed [22,3]. In these approaches, scenarios are *eliminated* (dropped) from the KKT matrix if the value of the primal or slack variable tends to be below a given threshold.

The proposed clustering approach is more general in the sense that clustering leads to elimination of both scenarios with small and large contributions. For instance, consider the case in which all the contributions $\gamma_s, s \in \mathcal{S}$ are all the same and large. Elimination by thresholds will be limited in how many scenarios it can drop and it can introduce a large residual. Clustering, on the other hand, can yield an exact step with zero residual using a single scenario. In particular, note that the residual induced by scenario elimination by thresholds is

$$\delta_Z^\mathcal{E}(u) = B_\mathcal{E}^T K_\mathcal{E}^{-1} B_\mathcal{E} u. \quad (27)$$

If the eliminated scenarios \mathcal{E} have large contributions $\gamma_s(\cdot)$ then they cannot be eliminated because they would lead to an excessively inexact step. In addition, this compromises convergence to the solution of the original problem (1).

The clustering approach is based on the residual characterization and not on the value of the primal and dual variables as proposed in [26,6]. This gives much more flexibility, particularly early in the search where activity has not been fully revealed. Also, as opposed to out of the solver clustering, the approach does not require the scenarios to be close. It only requires the scenarios to induce similar actions on the search direction. In addition, note that we do not compute weighted averages of the matrices to form the compressed KKT system as in [2,25] which incurs high computational costs.

A key property of the clustering approach is that the residual $\delta(\Delta w_0) = (Z - \bar{Z})\Delta w_0$ can converge to zero even if the full-space and compressed Schur matrices Z, \bar{Z} are not the same. In other words, the requirement is only that the two matrices are *similar* along the direction Δw_0 . As we will discuss in the next section, this leads to a Dennis-Moré [11] condition and enables the possibility of retaining full-space superlinear convergence using the compressed KKT system representation.

3 Full-Space Analysis

In this section we discuss asymptotic convergence and we explore the possibility of deriving full-space preconditioners using compressed KKT systems. A technical question that arises is *How can we characterize the residual introduced by the inexact step on the full-space KKT system?* As

we will see, the compressed system introduces a matrix perturbation localized in the first-stage block of the KKT system. This property will be exploited in the subsequent analysis.

3.1 Asymptotic Convergence

From the Schur analysis of the previous section we have that clustering introduces the perturbation matrix

$$\begin{aligned} P_{\mathcal{R}} &:= \Omega_{\mathcal{R}} B_{\mathcal{R}}^T K_{\mathcal{R}}^{-1} B_{\mathcal{R}} - B_{\mathcal{S}}^T K_{\mathcal{S}}^{-1} B_{\mathcal{S}} \\ &= (\bar{Z} - Z). \end{aligned} \quad (28)$$

This is the error between the exact and inexact Schur complements. The Schur analysis also motivates the introduction of the term $\Omega_{\mathcal{R}}^{-1} r_{\mathcal{R}}$ in (15) to retain the gradient information and make the right-hand sides of the Schur systems coincide.

To establish the connection with the full-space KKT system, we consider the perturbed system

$$\begin{bmatrix} K_{\mathcal{S}} & B_{\mathcal{S}} \\ B_{\mathcal{S}}^T & K_0 + P_{\mathcal{R}} \end{bmatrix} \begin{bmatrix} \Delta w_{\mathcal{S}} \\ \Delta w_0 \end{bmatrix} = \begin{bmatrix} r_{\mathcal{S}} \\ r_0 \end{bmatrix}. \quad (29)$$

In compact form,

$$\bar{\Phi}_w(w) \Delta w = -\Phi(w, \mu). \quad (30)$$

This is a perturbed variant of (7). We have the following result, establishing the equivalence between the compressed KKT system (15), the compressed Schur System (19), and the perturbed full-space KKT system (29).

Theorem 2 *Systems (15), (19), and (29) deliver the same first-stage Δw_0 step.*

Proof: The equivalence results from Schur decomposition. The decomposition for (29) leads to

$$(K_0 + P_{\mathcal{R}} - B_{\mathcal{S}}^T K_{\mathcal{S}}^{-1} B_{\mathcal{S}}) \Delta w_0 = (K_0 - \Omega_{\mathcal{R}} B_{\mathcal{R}}^T K_{\mathcal{R}}^{-1} B_{\mathcal{R}}) \Delta w_0.$$

This is equivalent to the left-hand side of the Schur system (19) which results from the decomposition of (15). The right-hand side for the three systems is $r_0 - B_{\mathcal{S}}^T K_{\mathcal{S}}^{-1} r_{\mathcal{S}}$. Consequently, the systems deliver the same step Δw_0 . \square

Equivalence of the step Δw_0 also implies equivalence of the entire step Δw because the compressed KKT system (15) yields the same step for the remaining scenarios $\Delta w_{\mathcal{R}}$ as that obtained from the perturbed system (29) and because $\Delta w_{\mathcal{E}}$ recovered from (16) is the same as that obtained from the perturbed system (29).

The perturbed full-space system (29) enables a characterization of the clustering approach as a full-space inexact Newton method. In addition, it provides a mechanism for implementation using the sparse compressed system (15) instead of a Schur decomposition framework as will be discussed in Section 3.2.

We now characterize the residual induced by Δw on the full-space system.

Theorem 3 *The inexact step Δw applied to the exact full-space system (14) generates a full-space residual $\delta(\Delta w)$ satisfying*

$$\|\delta(\Delta w)\| = \|\delta_Z(\Delta w_0)\| = \|P_{\mathcal{R}} \Delta w_0\|. \quad (31)$$

In addition, the residual is localized in the first stage so that $\|K_0 \Delta w_0 + B_{\mathcal{S}}^T \Delta w_{\mathcal{S}} - r_0\| = \|P_{\mathcal{R}} w_0\|$ and $\|K_{\mathcal{S}} \Delta w_{\mathcal{S}} + B_{\mathcal{S}} \Delta w_0 - r_{\mathcal{S}}\| = 0$.

Proof: Define $\bar{\Phi}_w := \bar{\Phi}_w(w)$, $\Phi_w := \Phi_w(w)$, and $\Phi := \Phi(w, \mu)$. We have that $\bar{\Phi}_w \Delta w - \Phi = 0$. Following a similar procedure as in (21), we obtain

$$\begin{aligned}
\delta(\Delta w) &= \bar{\Phi}_w \Delta w + \Phi \\
&= \begin{bmatrix} K_S \Delta w_S + B_S \Delta w_0 - r_S \\ K_0 \Delta w_0 + B_S^T \Delta w_S - r_0 \end{bmatrix} \\
&= (\Phi_w - \bar{\Phi}_w) \Delta w - \Phi + \bar{\Phi}_w \Delta w \\
&= (\Phi_w - \bar{\Phi}_w) \Delta w \\
&= \begin{bmatrix} 0 & 0 \\ 0 & -P_{\mathcal{R}} \end{bmatrix} \begin{bmatrix} \Delta w_S \\ \Delta w_0 \end{bmatrix} \\
&= \begin{bmatrix} 0 \\ -P_{\mathcal{R}} \Delta w_0 \end{bmatrix}. \tag{32}
\end{aligned}$$

The result follows by taking norms. \square

We highlight that it is not necessary to form the Schur complement to evaluate the residual. All that we need is $\|K_0 \Delta w_0 + B_S^T \Delta w_S - r_0\|$. This above result is also important because it implies that the quality of the full-space step is the same as the quality of the first-stage step.

The compressed KKT matrix does not need to exactly match the full-space matrix in order to achieve superlinear convergence. Define $\delta^k := \delta(\Delta w^k)$. The requirement is that $\|\delta^k\| = \|\bar{\Phi}(w^k) - \bar{\Phi}(w^k)\| = o(\|\Delta w^k\|)$. This in turn requires the cluster sequence \mathcal{R}^k to satisfy $\|\delta^k\| = \|P_{\mathcal{R}^k} \Delta w_0^k\| = o(\|\Delta w^k\|)$. In practice, this can be enforced in a number of ways such as to require that $\|\delta^k\|$ is a fraction of the right-hand side $\|\delta^k\| \leq \ell^k \|\bar{\Phi}(w^k, \mu^{k+1})\|$ for $\{\ell^k\}$ converging to zero [10]. This can be achieved by setting $\ell^k = \min(0.5, \|\bar{\Phi}(w^k, \mu^{k+1})\|)$. The refinement can be done by increasing the number of clusters since in the ideal case where $|\mathcal{C}^k| = |\mathcal{S}| = |\mathcal{R}^k|$ we obtain exact Newton steps with $\delta^k = 0$. Of course, this discussion assumes that the sequence $\{\mu^k\}$ is updated at least superlinearly [24, 27, 1].

3.2 Preconditioning and Scalability

Because compression introduces a matrix perturbation, the quality of the delivered steps is expected to degrade as the KKT system becomes larger and ill-conditioned. The effect of ill-conditioning can be ameliorated by using iterative refinement or by using the compressed system to precondition the full-space system under an iterative linear algebra setting. In an iterative linear algebra setting, the solver will pass a vector u to which we apply the preconditioner $y = \bar{\Phi}_w^{-1} u$. If the preconditioner is exact we have $y^* = \bar{\Phi}_w^{-1} u$. From Theorem 3 we have that for a given vector u the inexact system delivers a vector $y^T = [y_S^T, y_0^T]$ satisfying

$$\bar{\Phi}_w y = u + (\bar{\Phi}_w - \bar{\Phi}_w) y = u + P y. \tag{33}$$

Here, $P = \text{diag}(0, -P_{\mathcal{R}})$ and $\bar{\Phi}_w = (\bar{\Phi}_w - \bar{\Phi}_w)$. We also have $P y = [0 \ -P_{\mathcal{R}} y_0]$ and

$$\begin{aligned}
y - y^* &= \bar{\Phi}_w^{-1} P y \\
&= \bar{\Phi}_w^{-1} (\bar{\Phi}_w - \bar{\Phi}_w) y \\
&= (I - \bar{\Phi}_w^{-1} \bar{\Phi}_w) y. \tag{34}
\end{aligned}$$

From the first relationship we have that $y^* = (I - \bar{\Phi}_w^{-1} P) y$. This condition is typically found in an iterative refinement analysis which states that y converges to y^* if the spectrum of $\bar{\Phi}_w^{-1} P$

is less than one. In other words, it is possible to refine the inexact step for mild matrix perturbations. From the last relation we have that the error $y - y^*$ induced by the inexact step will be small if the eigenvalues of $\Phi_w^{-1}\bar{\Phi}_w$ cluster around one, as expected. This last equation also provides a connection between the eigenvalues of the inexact and exact matrices and the matrix perturbation. We have that $\Phi_w(y - y^*) = (\Phi_w - \bar{\Phi}_w)y$ and from the Wielandt-Hoffman Theorem [12] we have that,

$$\begin{aligned} \sum_k (\lambda_k(\Phi_w) - \lambda_k(\bar{\Phi}_w))^2 &\leq \|\Phi_w - \bar{\Phi}_w\|_F^2 \\ &= \|P\|_F^2 = \|P_{\mathcal{R}}\|_F^2. \end{aligned} \quad (35)$$

Here, $\|\cdot\|_F$ is the Frobenius norm, and k is the eigenvalue index. We thus have that the preconditioner quality improves as the number of clusters increases.

A clear benefit of compression arises in the case where the full-space system cannot be factorized in memory but compression enables this option. This is particularly helpful in problems with a large dimensionality in the first-stage where Schur decomposition is not scalable. This is true for, instance, in problems with first-stage dimensionality beyond $O(10^3)$ where the serial factorization of the Schur complement is not possible or too expensive [18]. If the sparse factorization of the compressed system is possible then we can use this as preconditioner for the full-space KKT system within an iterative framework and bypass the Schur factorization bottleneck.

4 A Clustering-Based IP Algorithm

Given a full-space inexact step Δw^k we compute a steplength α ensuring that $\Pi(w^k + \alpha\Delta w^k) \geq (1 - \tau)w^k$, where $\tau \in (0, 1)$ is a fraction-to-the-boundary parameter. To accommodate inexact steps, we use the following merit function to measure progress toward the solution [21,8]:

$$\Xi(w^k) := \|\nabla_y \mathcal{L}(w^k)\| + \|\nabla_\lambda \mathcal{L}(w^k)\| + \|Y^k V^k e\|. \quad (36)$$

We define a reduction ratio for the merit function $\eta \in (0, 1]$. At each iteration k we start with minimum number of clusters C^{min} and try to compute a cheap step. If the step reduces the merit function and if the residual is not too large, then we proceed. If not, we update the number of clusters $C \leftarrow \min\{C + \Delta C, S\}$ where $\Delta C > 0$ is a cluster number update rate. The algorithm is summarized below.

Algorithm IP-CLUSTER. Given $\epsilon \geq 0$, η , τ , C^{min} , and ΔC , DO:

1. Compute $\Xi(w^k)$ and ℓ^k .
2. IF $\|\Xi(w^k)\| \leq \epsilon$ TERMINATE. Otherwise, CONTINUE.
3. Set number of clusters $C^k := C^{min}$.
4. Compute cluster information \mathcal{R}^k and $\omega_{\mathcal{R}}^k$ using Δw^{k-1} .
5. Compute step Δw^k using compressed system (15) and (16).
6. Determine maximum step size $\alpha \geq (0, 1]$ satisfying $\Pi(w^k + \alpha\Delta w^k) \geq (1 - \tau)w^k$ and compute trial iterate $w_+^k \leftarrow w^k + \alpha\Delta w^k$.
7. IF $\Xi(w_+^k) \leq \eta \Xi(w^k)$ AND $\|\delta^k\| \leq \ell^k \|\bar{\Phi}(w^k, \mu^{k+1})\|$ ACCEPT trial step $w^{k+1} \leftarrow w_+^k$, UPDATE $\mu^{k+1} \leftarrow \mu^k + \Delta\mu^k$, SET $k \leftarrow k + 1$, and RETURN TO 1). Otherwise, RETURN to 3), and set $C^k \leftarrow \min\{S, C^k + \Delta C\}$.

At each iteration, the algorithm reverts to an exact step using all scenarios if sufficient progress is not achieved. If an iterative linear algebra setting is used in step 5) we have the following strategy. If a maximum number of linear algebra iterations is reached we TERMINATE, set $C^k \leftarrow \min \{S, C^k + \Delta C\}$ and we RETURN TO step 3). We do this in order to refine the preconditioner by increasing the number of clusters.

The proposed framework is general and is only intended to provide a mechanism to incorporate the clustering procedure within an inexact interior-point framework. The framework allows for multiple specializations including barrier parameter update [19,13,26,1,21,24], steplength determination [8,20], strategies to promote local and global convergence [27,4,21], and more general termination criteria [28]. We do not consider these refinements here because we focus on the quality of the compressed system. For implementation details of an inexact interior point algorithm we refer the reader to [7].

5 Numerical Studies

To illustrate the performance of the clustering approach, we consider stochastic variants of problems obtained from the CUTER library, the benchmark problems collected by Linderoth, Wright, and Shapiro (LWS) reported in [17], and network expansion problems.

The implementation follows the IP-CLUSTER algorithm. We use Mehrotra's heuristic to update the barrier parameter adaptively [19]. We also use equal steplengths for primal and dual variables. We use a single-link hierarchical clustering algorithm to perform clustering. While the k -means algorithm is theoretically sound, we found it to be unreliable and unstable in large problems as its performance strongly depends on the initialization procedure. The inexact IP algorithm is implemented in Matlab and we use this for demonstration purposes in small to medium problems. We have deactivated the step acceptance tests of step 7) in order to achieve a more systematic analysis of the effect of inexactness. By activating the tests the number of iterations vary somewhere in between the inexact and exact results presented here and the results are less informative.

5.1 CUTER and LWS Problems

The deterministic CUTER QP problems have the form

$$\min \frac{1}{2}y^T Qy + d^T y, \text{ s.t. } Ay = b, y \geq 0. \quad (37)$$

We generate a stochastic version of this problem by defining b as a random vector. We create scenarios for this vector $b_s, s \in \mathcal{S}$ using the nominal value b as mean and a standard deviation $\pm\sigma = 0.5b$. We then formulate the two-stage problem

$$\min e^T y_0 + S^{-1} \sum_{s \in \mathcal{S}} \frac{1}{2} y_s^T Q y_s + d^T y_s \quad (38a)$$

$$\text{s.t. } Ay_s = b_s, s \in \mathcal{S} \quad (38b)$$

$$y_s + y_0 \geq 0, s \in \mathcal{S} \quad (38c)$$

$$y_0 \geq 0. \quad (38d)$$

We transform this problem into form (1) by adding slack variables. We also consider the two-stage LPs GDB, LANDS, 20TERM, and SSN from the LWS benchmark [17]. We did not consider the STORM problem as this is too large for the existing implementation.

Table 1 Convergence history of inexact and exact interior-point algorithms for LOTSCH problem with 100 scenarios.

k	Exact			Inexact					
	φ^k	Φ^k	$ \mathcal{R} ^k$	φ^k	Φ^k	$\ \delta^k\ $	$\ \delta_p^k\ $	$\ P_{\mathcal{R}}^k\ _F$	$ \mathcal{R} ^k$
1	1.29E+02	5.03E+04	100	1.29E+02	5.03E+04	4.69E-11	4.03E-12	1.01E-13	25
5	6.78E+03	1.02E+01	100	6.62E+03	9.90E+00	6.69E-01	2.63E+00	2.89E-02	25
10	4.94E+03	4.34E-01	100	4.92E+03	9.93E-01	1.00E+00	1.09E+00	3.68E-01	25
15	4.77E+03	4.75E-05	100	4.77E+03	4.42E-02	4.39E-03	6.48E-03	8.98E-01	25
16	4.77E+03	2.31E-06	100	4.77E+03	4.97E-03	7.34E-05	5.98E-04	6.35E-02	25
17	4.77E+03	1.81E-08		4.77E+03	8.25E-05	1.26E-05	4.21E-05	1.28E-01	25
18				4.77E+03	1.26E-05	1.27E-06	1.84E-06	1.95E-01	25
19				4.77E+03	1.27E-06	2.39E-08	1.47E-08	1.39E-02	25
20				4.77E+03	2.39E-08				

We first consider a test using the stochastic LOTSCH problem to illustrate the behavior of the algorithm. In this case we use 100 randomly generated scenarios. The convergence history of the inexact and exact algorithms is presented in Table 1. The exact algorithm uses the entire set of scenarios and converges in 17 iterations. The inexact version uses only 25 scenarios to form the compressed system per iteration and converges in 20 iterations. Note that the actual residual $\|\delta^k\|$, the predicted residual $\|\delta_p^k\|$, and the matrix perturbation $\|P_{\mathcal{R}}^k\|_F$ are negligible in the first iteration. The reason is that all the variables were initialized at the same value (all problem variables were initialized at a value of 100) and thus all the block scenarios are the same. This demonstrates that clustering can identify this case and take exact steps. We also note that the Schur complement error does not converge to zero. Superlinear convergence, however, is achieved.

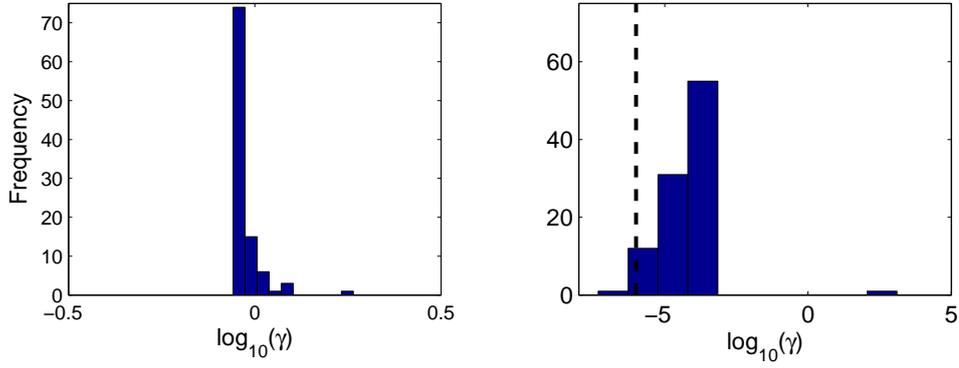
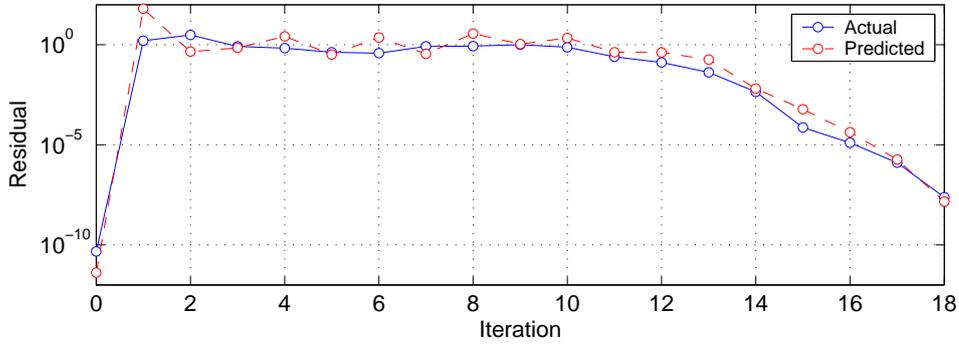
In Figure 1 we present the distribution of the contributions $\|\gamma_s(\cdot)\|$, $s \in \mathcal{S}$ at iteration 5 and at the final iteration. Note that the distribution changes significantly throughout the search. Consequently, the clusters should be updated adaptively at each iteration. The dashed line indicates the final convergence tolerance of 1×10^{-7} . Note that even if most of the contributions are well above the tolerance, most of them can be compressed and we can achieve an accurate solution. This is a key advantage over scenario elimination procedures.

In Figure 2 we present the history for the actual and predicted residuals. As can be seen, the clustering strategy does a good job at predicting the magnitude of the actual residual. This indicates that obtaining clusters by using the direction $u = \Delta w_0^{k-1}$ is a reasonable strategy.

We ran the inexact and exact algorithms for some problems of the CUTER family and a couple from the LWS benchmark. Our tests use 100 scenarios in the small and medium cases and 50 scenarios in the large cases. In all cases we compress the KKT system by 50%. The number of iterations is presented in Table 2. The inexact algorithm has similar performance as the exact counterpart in all instances. For the larger problems 20TERM and SSN we noted that compression rates of only 10-20% were possible due to ill-conditioning. We analyze these two problems in the preconditioning section 5.3.

Table 2 Number of iterations for stochastic variants of CUTER set and LWS problems with 50% of scenarios eliminated.

Problem	S	n	Exact	Inexact
QPTEST	100	505	17	17
ZECEVIC	100	606	20	20
HS76	100	707	19	19
GBD	100	1017	19	19
HS53	100	1110	14	14
LANDS	100	1204	23	24
LOTSCH	100	1212	17	20
QAFIRO	100	5151	34	38
HS118	100	5959	37	38
DUAL1	50	8670	17	21
DUAL2	50	9792	16	18

**Fig. 1** Distribution of residual contributions $\delta_s, s \in \mathcal{S}$ for LOTSCH problem. Left panel is iteration 5 and right panel is final iteration. Dashed line indicates the convergence tolerance.**Fig. 2** History of predicted and actual residuals for LOTSCH problem.

5.2 Network Expansion

In this problem family, we determine the optimal capacities for generators and branches that satisfy the random demands across the network. The problem has the following structure:

$$\min d_g^T \Delta g + d_p^T \Delta p + S^{-1} \sum_{s \in \mathcal{S}} c_g^T g_s \quad (39a)$$

$$\text{s.t. } Pp_s + Gg_s = Dd_s, \quad s \in \mathcal{S} \quad (39b)$$

$$p_L \leq p_s + \Delta p \leq p_U, \quad s \in \mathcal{S} \quad (39c)$$

$$g_L \leq g_s + \Delta g \leq g_U, \quad s \in \mathcal{S} \quad (39d)$$

$$\Delta g \geq 0 \quad (39e)$$

$$\Delta p \geq 0. \quad (39f)$$

Here, Δg and Δp are the expansion capacities for generators and branches, respectively. The corresponding costs are d_g and d_p . The generation levels in each scenario are given by g_s with costs c_g . The random demands are d_s . The matrices P, G and D are incidence matrices for the branches, generators, and demands, respectively. The branch capacities satisfy $p_L = -p_U$ because we allow for bidirectional flows. We have created a family of network problems of increasing size. The networks are radial with n_R nodes. Networks of $n_R = 5, 6$, and 10 radial nodes are depicted in Figure 3. The internal branches are represented by the dashed lines. The red circles are demands, and the blue circles are generators.

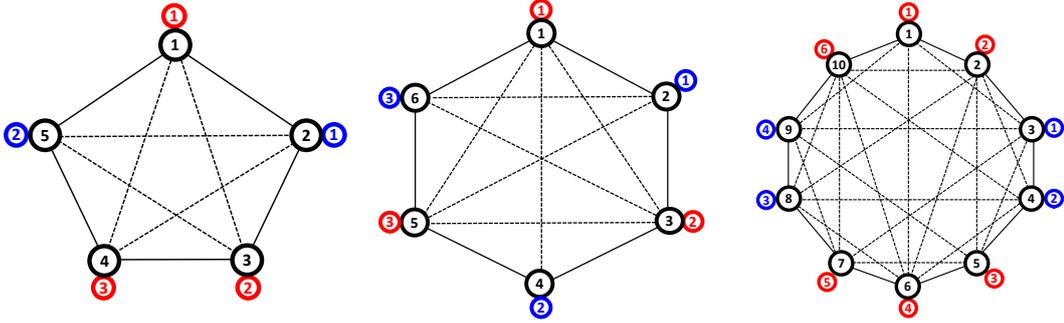


Fig. 3 Topology of radial network expansion problems.

The distribution of contributions $\|\gamma_s(\cdot)\|$ at the solution for the three network problems is presented in Figure 4. Note that most of the contributions are well above the final tolerance and there are strong dominating clusters. We have observed that it is not the magnitude of the contributions but the spread of $\gamma_s(\cdot)$ that tends to influence clustering performance more strongly. In Table 3 we compare the number of iterations for the inexact algorithm as a function of the number of eliminated scenarios $|\mathcal{E}|$. Performance is consistent for the two smaller networks and compression rates of up to 90% can be achieved. For the largest network we can compress up to 80% of the scenarios and achieve convergence.

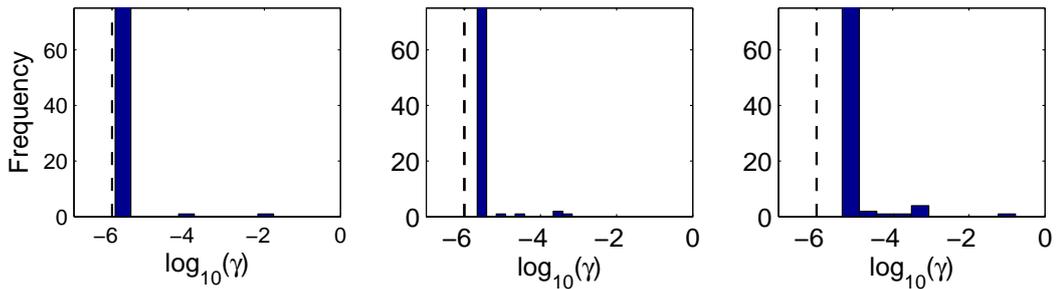


Fig. 4 Distribution of residual contributions $\|\gamma_s(\cdot)\|$, $s \in \mathcal{S}$ at final iteration for problems with 5 (left), 6 (center), and 10 (right) radial nodes. Dashed lines indicate convergence tolerance.

Table 3 Number of iterations as function of $|\mathcal{E}|$ for radial network problems with $S = 100$ scenarios.

$ \mathcal{E} $	$n_R = 5$	$n_R = 6$	$n_R = 10$
50	13	13	16
60	14	14	17
70	13	15	18
80	13	15	24
90	16	17	-

Table 4 Total number of QMR iterations for DUAL1 with low variance, SSN, and 20TERM. $S = 50$ scenarios. Avg. QMR/IP denotes average number of QMR iterations per interior-point iteration.

Problem	Strategy	n	$ \mathcal{E} $	IP Iterations	QMR Iterations	Avg. QMR/IP
DUAL1	Unpreconditioned	8,670	-	17	13,462	791.9
DUAL1	Elimination	8,670	25	17	108	6.3
DUAL1	Clustering	8,670	25	17	46	2.7
DUAL1	Elimination	8,670	35	17	162	9.5
DUAL1	Clustering	8,670	35	17	58	3.4
20TERM	Clustering	35,389	25	33	156	4.7
SSN	Clustering	38,263	25	54	287	5.3

5.3 Preconditioning

To demonstrate the preconditioning capabilities of the clustering approach in the larger problems, we applied a quasi-minimum residual (QMR) algorithm with constant tolerance of 1×10^{-6} and a termination tolerance of 1×10^{-5} . We use problems DUAL1, 20TERM, and SSN using $S = 50$ scenarios. We compare the total number of QMR iterations required for convergence for unpreconditioned and cluster preconditioning variants. In addition, we compare the performance of a preconditioner generated by eliminating (dropping) the block scenarios with the $|\mathcal{E}|$ largest contributions $\|\gamma_s(\cdot)\|$. This comparison demonstrates that clustering gives more accurate compressed KKT matrices than scenario elimination. The results are presented in Table 4. The number of QMR iterations is reduced almost by a factor of 2 for the $|\mathcal{E}| = 25$ case and by a factor of 3 for the $|\mathcal{E}| = 35$ case which corresponds to an elimination of 70% of the scenarios. Higher percentages lead to unacceptable QMR performance. In particular, we require that less than 100 QMR iterations per interior-point iteration. We can also see that the preconditioner performance for SSN and 20TERM is acceptable for a compression rate of 50%. These are the largest problems considered which include 38,263 and 35,389 variables, respectively.

To demonstrate the robustness of the clustering approach, we increase the number of scenarios for DUAL1 to $S = 100$ and we increase the variance from $\pm\sigma = 0.5b$ to $\pm\sigma = b$. This problem has a total of 17,170 variables and inequality constraints and 8,600 equality constraints. The increase in variance drastically increases activity and makes the KKT matrix much more difficult to approximate. The results are summarized in Figure 5 where we present the QMR iteration histories for the cases with $|\mathcal{E}| = 30$ and $|\mathcal{E}| = 50$. The number of iterations is consistently below 50, and the number of iterations is minimal at the beginning of the search. Interestingly, the number of iterations decreases as we approach the solution. This is an unexpected result because ill-conditioning tends to affect the performance of iterative solvers as the barrier parameter converges to zero. This demonstrates that the clustering approach has the potential of leading to effective preconditioners.

We did not explore larger numbers of eliminated scenarios as the performance of QMR became unacceptable. Consequently, the largest percentage of eliminated scenarios leading to acceptable performance was 50%. We performed another study for DUAL1 with $S = 30$ total sce-

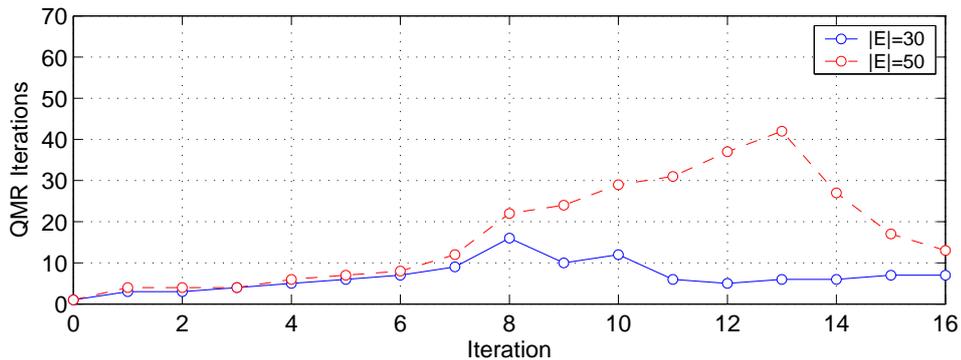


Fig. 5 History of QMR iterations for DUAL1 problem with high variance and $S = 100$ scenarios.

narios. The largest percentage of eliminated scenarios with acceptable QMR performance was 33%. This indicates that the benefits of clustering become more pronounced as the number of scenarios is increased, a result that was corroborated in other problem instances. Unfortunately, the existing implementation does not allow us to explore these regimes.

6 Conclusions and Future Work

We have presented an inexact interior point framework for stochastic convex programs. The framework computes inexpensive steps by performing clustering on the block scenarios forming the arrowhead Karush-Kuhn-Tucker (KKT) system. We showed that the compression can be characterized as a parametric perturbation of the KKT matrix. This allows for the possibility to achieve superlinear convergence without requiring matrix convergence. We derive a clustering strategy using an explicit characterization of the residual and we demonstrate that this works well in practice. We also demonstrate that clustering can yield large compression rates and can work as an effective preconditioner. In a family of small to medium test problems we have found that compression rates of 50-90% are possible. For the larger test problems preconditioning performance is acceptable with compression rates of up to 50%. As part of future work, we will develop parallel implementations to explore clustering potential in regimes with large numbers of scenarios and we will extend the clustering approach to a more general nonlinear programming setting.

Acknowledgements This work was supported by the U.S. Department of Energy, under Contract No. DE-AC02-06CH11357. Funding from the Office of Science under the Early Career program is greatly acknowledged. The author would like to thank Jorge Nocedal for several discussions on stochastic Newton methods and Cosmin Petra for providing feedback on an earlier version of the manuscript and for providing an interface to the LWS benchmark problems.

References

1. P. Armand, J. Benoist, and D. Orban. Dynamic updates of the barrier parameter in primal-dual methods for non-linear programming. *Computational Optimization and Applications*, 41:1–25, 2008.
2. John Birge. Aggregation bounds in stochastic linear programming. *Mathematical Programming*, 31:25–41, 1985.
3. R. Byrd, G. Chin, W. Neveitt, and J. Nocedal. On the use of stochastic hessian information in optimization methods for machine learning. *SIAM Journal on Optimization*, 21(3):977–995, 2011.

4. R. H. Byrd, J. Ch. Gilbert, and J. Nocedal. A trust-region method based on interior-point techniques for nonlinear programming. *Math. Programm.*, 89:149–185, 2000.
5. M.S. Casey and S. Sen. The scenario generation algorithm for multistage stochastic linear programming. *Mathematics of Operations Research*, 30(3):615–631, 2005.
6. N. Chiang and A. Grothey. Solving security constrained optimal power flow problems by a structure exploiting interior point method. *Submitted For Publication*, 2012.
7. F. Curtis, J. Huber, O. Schenk, and A. Wächter. A note on the implementation of an interior-point algorithm for nonlinear optimization with inexact step computations. *Mathematical Programming*, pages 1–19, 2012.
8. F. Curtis and J. Nocedal. Steplength selection in interior-point methods for quadratic programming. *Applied Mathematics Letters*, 20(5):516 – 523, 2007.
9. W. L. de Oliveira, C. Sagastizábal, D. Penna, M. Maceira, and J. M. Damázio. Optimal scenario tree reduction for stochastic streamflows in power generation planning problems. *Optimization Methods and Software*, 25(6):917–936, 2010.
10. R. S. Dembo, S. C. Eisenstat, and T. Steihaug. Inexact Newton methods. *SIAM Journal on Numerical Analysis*, 19(2), 1982.
11. J. E. Dennis and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. SIAM, Philadelphia, PA, 1996.
12. G.H. Golub and C.F. Van Loan. *Matrix computations*, volume 3. Johns Hopkins University Press, 1996.
13. J. Gondzio. Multiple centrality corrections in a primal-dual method for linear programming. *Computational Optimization and Applications*, 6:137–156, 1996.
14. H. Heitsch and W. Römisch. Scenario tree reduction for multistage stochastic programs. *Computational Management Science*, 6:117–133, 2009.
15. A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Comput. Surv.*, 31(3):264–323, September 1999.
16. J. M. Latorre, S. Cerisola, and A. Ramos. Clustering algorithms for scenario tree generation: Application to natural hydro inflows. *European Journal of Operational Research*, 181(3):1339 – 1353, 2007.
17. J. Linderoth, A. Shapiro, and S. Wright. The empirical behavior of sampling methods for stochastic programming. *Annals of Operations Research*, 142(1):215–241, 2006.
18. 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.
19. S. Mehrotra. On the implementation of a primal-dual interior point method. *SIAM Journal on Optimization*, 2:575–601, 1992.
20. C. Mészáros. Steplengths in interior-point algorithms of quadratic programming. *Operations Research Letters*, 25(1):39 – 45, 1999.
21. J. Nocedal, A. Wächter, and R. Waltz. Adaptive barrier update strategies for nonlinear interior methods. *SIAM Journal on Optimization*, 19(4):1674–1693, 2009.
22. C. Petra and M. Anitescu. A preconditioning technique for schur complement systems arising in stochastic optimization. *Computational Optimization and Applications*, 52:315–344, 2012.
23. A. Ruszczyński and A. Shapiro. *Stochastic Programming (Handbooks in Operations Research and Management Series)*. Elsevier Science BV, Amsterdam, 2003.
24. M. Salahi, J. Peng, and T. Terlaky. On mehrotra-type predictor-corrector algorithms. *SIAM Journal on Optimization*, 18(4):1377–1397, 2008.
25. C. M. Shetty and R. W. Taylor. Solving large-scale linear programs by aggregation. *Computers & Operations Research*, 14(5):385 – 393, 1987.
26. A. Tits, P. Absil, and W. Woessner. Constraint reduction for linear programs with many inequality constraints. *SIAM Journal on Optimization*, 17(1):119–146, 2006.
27. 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.
28. S.J. Wright. *Primal-dual interior-point methods*, volume 54. Society for Industrial Mathematics, 1987.
29. P.H. Zipkin. Bounds for row-aggregation in linear programming. *Operations Research*, 28(4):903–916, 1980.

The submitted manuscript has been created by the University of Chicago as Operator of Argonne National Laboratory (“Argonne”) under Contract No. DE-AC02-06CH11357 with the U.S. Department of Energy. The U.S. Government retains for itself, and others acting on its behalf, a paid-up, nonexclusive, irrevocable worldwide license in said article to reproduce, prepare derivative works, distribute copies to the public, and perform publicly and display publicly, by or on behalf of the Government.