

...

...

Received: date / Accepted: date

Abstract Keywords augmented Lagrangian method · proximal bundle method · nonlinear block Gauss-Seidel method · simplicial decomposition method · parallel computing

Mathematics Subject Classification (2000) 90-08, 90C06, 90C11, 90C15, 90C25, 90C26, 90C30, 90C46

1 Introduction and Background

The problem of interest has the form

$$f^* := \min_{x,y,z} \left\{ \sum_{i=1}^m p_i [c^\top x_i + d_i^\top y_i] : (x_i, y_i) \in K_i, x_i = z \text{ for } i = 1, \dots, m, z \in \mathbb{R}^{n_1} \right\}, \quad (1)$$

where K is a bounded polyhedral set intersected with mixed-integrality constraints.

We develop a branch-and-bound approach to solving (1) based on solving Lagrangian relaxations due to relaxing $Qx = z$. At each node, we use the recently developed algorithm SDM-GS-ALM to solve two equivalent characterizations of the Lagrangian dual problem. The Lagrangian dual function is defined

$$\phi(\omega) := \min_{x,y,z} \left\{ p_i [c^\top x_i + \omega_i^\top (x_i - z) + d_i^\top y_i] : (x_i, y_i) \in K_i, z \in \mathbb{R}^{n_1} \right\}.$$

Under the dual feasibility condition $\sum_{i=1}^m p_i \omega_i = 0$, we have

$$\phi(\omega) = \min_{x,y,z} \left\{ p_i \left[(c + \omega_i)^\top x_i + d_i^\top y_i \right] : (x_i, y_i) \in K_i, z \in \mathbb{R}^{n_1} \right\}$$

The Lagrangian dual problem is:

$$\phi^* := \max_{\omega \in \Omega} \phi(\omega). \quad (2)$$

Its well-known primal characterization takes the form

$$f_C^* := \min_{x, y, z} \left\{ \sum_{i=1}^m p_i [c^\top x_i + d_i^\top y_i] : (x_i, y_i) \in \text{conv}(K_i), x_i = z \text{ for } i = 1, \dots, m, z \in \mathbb{R}^{n_1} \right\}. \quad (3)$$

It is well-known [under our standing assumptions] that $f_C^* = \phi^*$, although in general, $\phi^* \leq f^*$, with $\phi^* < f^*$ being typical with the presence of mixed-integrality constraints on x and y .

Dual optimal solutions to (2), when they exist, are generically denoted by ω^* , and the set of corresponding optimal primal solutions verifying the value $\phi(\omega^*)$ is denoted by $K^*(\omega^*)$. The optimal solutions to the second characterization (3), when they exist, are denoted (x^*, y^*, z^*) . We shall denote the set $Z := \{(x, y) \in K : \exists z \in \mathbb{R}^n \text{ s.t. } x_i = z \text{ for } i = 1, \dots, m\}$.

We shall use both characterizations (2) and (3) to inform different ways of branching on variables and branching values.

1. Viewing the node subproblems as instances of problem (3), we have optimal solutions (x^*, y^*, z^*) to problem (3) that satisfy

$$(x^*, y^*, z^*) \in \{(x, y, z) : (x, y) \in \text{conv}(K), x_i = z, i = 1, \dots, m\},$$

but $(x^*, y^*) \notin K$ in the case of strict duality gap $\phi_C^* < f^*$. Thus we would project (x^*, y^*) onto K , denoting the projection by $P_K((x^*, y^*))$. (Given that K is defined by polyhedral and mixed-integer constraints, this projection can just take the form of rounding the components of (x, y) with integer restrictions to their nearest integer values.) We would consider the discrepancies $(x^*, y^*) - P_K((x^*, y^*))$ and their dispersions.

2. Viewing the node subproblems as instances of problem (2), assuming we have a dual optimal solution ω^* , we have corresponding sets of primal certificate solutions $K^*(\omega^*)$. For these solutions, we have $K^*(\omega^*) \cap Z = \emptyset$ in the case of duality gap $\phi_C^* < f^*$. We consider one of two projections:
 - (a) Take the projection $P_Z(K(\omega^*))$. Conceptually, we want to accumulate the dispersions in some set-wise or point-wise fashion. Ideally, we do not consider the dispersion of only one $\hat{x} \in K^*(\omega^*)$, but the entire set $K^*(\omega^*)$, which is contained in the highest dimensional face of $\text{conv}(K)$ for which (x^*, y^*) is in its relative interior.
 - (b) Rather than projection, we use the information from x^*, z^* and ω^* . (Or, more practically, x^k, z^k and ω^k .) We still work with the set $K^*(\omega^k)$, whose vertex description is available from computing $\hat{x}^t \in K^*(\omega^t)$ at each iteration $t = 1, \dots, k$.

Due to the large number of variables that would be branched on under the first approach, branching is not based on the first approach. Although we do use the dispersions from the first approach discrepancies to inform branching

based on the second approach. Branching occurs on components $x_{i,j}$ of each x_i that either violate integrality constraints, or that show dispersion from some consensus value associated with other violated constraints. The constraints added due to branching are denoted $x \in B$. Based on this constraint, define $K_B := \{(x, y) \in K : x \in B\}$, so that we define, respectively,

$$f_B^* := \min_{x,y,z} \left\{ \sum_{i=1}^m p_i [c^\top x_i + d_i^\top y_i] : (x_i, y_i) \in K_i, x_i \in B_i, x_i = z \text{ for } i = 1, \dots, m, z \in \mathbb{R}^{n_1} \right\}, \quad (4)$$

$$\phi_B^* := \max_{\omega \in \Omega} \phi_B(\omega). \quad (5)$$

with

$$\phi_B(\omega) = \min_{x,y,z} \left\{ p_i \left[(c + \omega_i)^\top x_i + d_i^\top y_i \right] : (x_i, y_i) \in K_i, x_i \in B_i, z \in \mathbb{R}^{n_1} \right\} \quad (6)$$

and

$$\phi_B^* = \min_{x,y,z} \left\{ \sum_{i=1}^m p_i [c^\top x_i + d_i^\top y_i] : \begin{array}{l} (x_i, y_i) \in \text{conv}(K_i \cap B_i), \\ x_i = z \text{ for } i = 1, \dots, m, \\ z \in \mathbb{R}^{n_1} \end{array} \right\}. \quad (7)$$

During each node processing of the branch-and-bound, an attempt is made to find a feasible solution to the node-specific instance of (4), which provides a finite upper bound, which we shall refer to as the *incumbent value* and denote as $v \in \mathbb{R}$. Furthermore, we use an iterative approach, referred to as the *oracle*, to solving each instance of (5) and (7) simultaneously, that is guaranteed to converge optimally. During the processing of each node, one of three things will happen:

1. The oracle terminates due to the objective value exceeding the incumbent value (fathoming due to bound), or due to subproblem infeasibility.
2. The oracle terminates due to optimality of (4) within pre-specified precision. That is $\|(x^*, y^*) - P_K(x^*, y^*)\| + \sum_{i=1}^m \|x_i^* - z^*\| < \epsilon$. The node is fathomed after testing whether $f_B^* < v$ and updating $v \leftarrow f_B^*$ if that is the case. (In Caroe and Schultz, it is the vertex solutions (\hat{x}, \hat{y}) that are tested only, so that the criterion is $\|\hat{x}_i - \hat{z}\| < \epsilon$, where \hat{z} is some averaging of \hat{x}_i , $i = 1, \dots, m$.)
3. Otherwise, branchings need to be determined.

Definitions:
 $\mathcal{S} := \{f, K, B\}, \mathcal{P} := \{\rho, \gamma, \epsilon, t_{max}, k_{max}\}$
 v is an incumbent value, and
Precondition:

It is assumed that the value of the current node has been checked to not exceed the incumbent value v

```

function PROCESS( $\mathcal{S}, \mathcal{P}, \omega^0, v, (\tilde{x}, \tilde{y}, \tilde{z})$ )
  ( $K^*(\omega^*), (x^*, y^*, z^*), \omega^*, \phi^*$ )  $\leftarrow$  bound( $\mathcal{S}, \mathcal{P}, \omega^0$ )
  if  $\check{\phi}^* > v$  or  $\check{\phi}^* = \infty$  then
    fathom node
     $\mathcal{B} \leftarrow \emptyset$ 
    return ( $v, \tilde{z}, \mathcal{B}$ )
  end if
   $\mathcal{B} \leftarrow$  findBranchings( $K^*(\omega^*), (x^*, y^*, z^*), \omega^*$ )
  ( $\tilde{x}, \tilde{y}, \tilde{z}, v$ )  $\leftarrow$  findFeasibleSolution( $z^*, \omega^*, \mathcal{S}, \mathcal{P}$ )
  if  $\mathcal{B} \neq \emptyset$  then
    node needs to branch
  else
    fathom node (Due to optimality)
  end if
  return ( $\tilde{x}, \tilde{y}, \tilde{z}, v, \mathcal{B}$ )
end function

function BOUND( $\mathcal{S}, \mathcal{P}, \omega^0$ )
  ( $D^0, x^0, z^0, \omega^0, \phi^0$ )  $\leftarrow$  Initialize( $\mathcal{S}, \mathcal{P}, \omega^0$ )
   $k \leftarrow 0$ , term  $\leftarrow$  false
  while  $\neg(\check{\phi}^k > v$  or term) do
    ( $K^*(\omega^{k+1}), (x^{k+1}, y^{k+1}, z^{k+1}), \omega^{k+1}, \phi^{k+1}$ , term)  $\leftarrow$ 
    PSCG( $\mathcal{S}, \mathcal{P}, D^k, x^k, z^k, \omega^k, \check{\phi}^k$ )
     $k \leftarrow k + 1$ 
  end while
  return ( $K^*(\omega^k), (x^k, y^k, z^k), (\tilde{x}, \tilde{y}, \tilde{z}), \omega^k, \phi^k, v$ )
end function

```

1.1 Parallel Stabilized Column Generation (PSCG)

In applying the AL method to problem (3), the continuous master problem for fixed $\omega \in Z^\perp$ takes the form

$$\phi_\rho^{AL}(\omega) := \min_{x, y, z} \{L_\rho(x, y, z, \omega), (x, y) \in \text{conv}(K), z \in Z\} \quad (8)$$

where the augmented Lagrangian (AL) relaxes $x_i = z$, $i = 1, \dots, m$ and is defined by

$$L_\rho(x, y, z, \omega) := \sum_{i=1, \dots, m} p_i \left[(c + \omega_i)^\top x_i + d_i^\top y_i + \frac{\rho}{2} \|x_i - z\|_2^2 \right]. \quad (9)$$

In the algorithm that follows, we use the following approximation $\hat{\phi} : \mathbb{R}^q \times \mathbb{R}^n \times \mathbb{R}^q \mapsto \mathbb{R}$ of ϕ^C centered at (x^k, z^k) , $k \geq 0$, in place the cutting plane model:

$$\hat{\phi}(\omega, x^k, y^k, z^k) := L_\rho(x^k, y^k, z^k, \omega) + \frac{\rho}{2} \sum_{i=1, \dots, m} \|x_i^k - z^k\|_2^2.$$

In [?], we developed an efficiently parallelizable iterative procedure to solving problem (??). In this paper, we incorporate this procedure as part of the bounding mechanism within a branch-and-bound approach....

For review purposes, the iterative dual procedure is given as follows.

Algorithm 1 A regular iteration of PSCG.

Preconditions:

$(x^k, y^k) \in \text{conv}(K)$, $z^k \in \arg\min_z \left\{ \sum_{i=1, \dots, m} \|x_i - z\|^2 \right\}$, $\omega^k \in \Omega$,
 $\phi^k = \phi(\omega^k)$, $\{(x^k, y^k) + \alpha((\hat{x}, \hat{y}) - (x^k, y^k)) : \alpha \in [0, 1]\} \subseteq D \subseteq \text{conv}(K)$ where
 $(\hat{x}, \hat{y}) \in \arg\min_{x, y} \{\nabla_{x, y} L_\rho(x^k, y^k, z^k, \omega^k)[(x - x^k); (y - y^k)] : (x, y) \in K\}$.
 $S := \{c, d, K, B\}$, $\mathcal{P} := \{\rho, \gamma, \epsilon, t_{max}, k_{max}\}$
1: **function** PSCG($S, \mathcal{P}, D^k, x^k, y^k, z^k, \omega^k, \phi^k$)
2: **for** $k = 1, 2, \dots, k_{max}$ **do**
3: Initialize $\omega^{k+1} \leftarrow \omega^k$; $\phi^{k+1} \leftarrow \phi^k$ \triangleright (Default, null-step updates)
4: $(\hat{x}^{k+1}, (\hat{x}^{k+1}, y^{k+1}, z^{k+1}), D^{k+1}, \Gamma) \leftarrow \text{SDM-GS}(L_\rho(\cdot, \cdot, \cdot, \omega^k), K \cap B, D^k, x^k, y^k, z^k, t_{max})$
5: **if** $L_\rho(x^{k+1}, y^{k+1}, z^{k+1}, \omega^k) + \frac{\rho}{2} \sum_{i=1}^m \|x_i^{k+1} - z^{k+1}\|_2^2 - \phi^k \leq \epsilon$ **then**
6: **return** $(x^{k+1}, y^{k+1}, z^{k+1}, \omega^{k+1}, \phi^{k+1}, D^{k+1}, \text{true})$
7: **end if**
8: $\tilde{\phi} \leftarrow L_\rho(x^{k+1}, y^{k+1}, z^{k+1}, \omega^k) + \frac{\rho}{2} \sum_{i=1}^m \|x_i^{k+1} - z^{k+1}\|_2^2 - \Gamma$
9: $\tilde{\gamma} \leftarrow \frac{\tilde{\phi} - \phi^k}{L_\rho(x^{k+1}, y^{k+1}, z^{k+1}, \omega^k) + \frac{\rho}{2} \sum_{i=1}^m \|x_i^{k+1} - z^{k+1}\|_2^2 - \phi^k}$
10: **if** $\tilde{\gamma} \geq \gamma$ **then**
11: set $\omega^{k+1} \leftarrow \omega^k + \rho(x^{k+1} - z^{k+1})$, $\phi^{k+1} \leftarrow \tilde{\phi}$
12: **end if**
13: Possibly update ρ , e.g., $\rho \leftarrow \frac{1}{\min\{\max\{(2/\rho)(1-\tilde{\gamma}), 1/(10\rho), 10^{-4}\}, 10/\rho\}}$ as in [1]
14: **end for**
15: **return** $(\hat{K}(\omega^{k+1}), (x^{k+1}, y^{k+1}, z^{k+1}), \omega^{k+1}, \phi^{k+1}, D^{k+1}, \text{false})$
16: **end function**

Preconditions: $(\tilde{x}, \tilde{y}) \in \text{conv}(K)$, $\tilde{z} \in \arg\min_z \{F(\tilde{x}, \tilde{y}, z) : z \in Z\}$, $D \subseteq \text{conv}(K)$

1: **function** SDM-GS($F, X, Z, D, \tilde{x}, \tilde{z}, t_{max}$)
2: **for** $t = 1, \dots, t_{max}$ **do**
3: $\hat{x} \leftarrow \arg\min_x \{F(x, y, \tilde{z}) : (x, y) \in D\}$
4: $\hat{z} \leftarrow \arg\min_z \{F(\tilde{x}, \tilde{y}, z) : z \in Z\}$
5: **end for**
6: $(\hat{x}, \hat{y}) \in \arg\min_{x, y} \{\nabla_{x, y} F(\tilde{x}, \tilde{y}, \tilde{z})[(x - \tilde{x}); (y - \tilde{y})] : (x, y) \in K\}$
7: Reconstruct D to be any set such that
8: $\{(\hat{x}, \hat{y}) + \alpha((\hat{x}, \hat{y}) - (\tilde{x}, \tilde{y})) : \alpha \in [0, 1]\} \subseteq D \subseteq \text{conv}(K)$
9: Set $\Gamma \leftarrow -\nabla_{x, y} F(\tilde{x}, \tilde{y}, \tilde{z})[(\hat{x} - \tilde{x}); (\hat{y} - \tilde{y})]$
10: **return** $(\hat{K}, \hat{x}, \hat{y}, \hat{z}, D, \Gamma)$
11: **end function**

Proposition 1 Let $\{(x^k, y^k, z^k, \omega^k)\}$ be a sequence generated by Algorithm 1 applied to problem (1) with X compact, Z a linear subspace, $\omega^0 \in \Omega$, B closed and convex, $\rho > 0$, $\gamma \in (0, 1)$, $\epsilon = 0$ and $k_{max} = \infty$. If there exists a dual optimal solution ω^* to the dual problem (2), then either

1. $\omega^k = \bar{\omega}$ is fixed and optimal for (2) for $k \geq \bar{k}$ for some finite \bar{k} ; or

2. ω^k is never optimal for (2) for any finite $k \geq 1$, but $\lim_{k \rightarrow \infty} \omega^k = \bar{\omega}$ is optimal,
and the sequence $\{(x^k, y^k, z^k)\}$ has limit points $(\bar{x}, \bar{y}, \bar{z})$, each of which are optimal for problem (3).

2 Branch-and-bound with SCG-GS-ALM as bounding procedure

In order to inform branching decisions, we need a sense of dispersion based on some violation of feasibility, and a value with which to bound.

2.1 Measure of feasibility

A criterion for fathoming by optimality may be given by

$$\|x^* - P_X(x^*)\| + \|Qx^* - z^*\| < \epsilon. \quad (10)$$

for some suitably small tolerance $\epsilon > 0$. If the feasibility criterion (10) is not met, then branching must occur, and the branching should be informed by the dispersions measured in the two norm terms of (10). It will be convenient to refine the dispersions, where the vector

$$\sigma_X := [\|x_i^* - P_{X_i}(x_i^*)\|]_{i=1, \dots, m}$$

is defined. The diagonal matrix whose diagonal entries are taken from σ_X will be denoted Σ_X .

2.2 Branching decisions

We have the following approaches at measuring dispersion from the constraint $Qx = z$ at iteration k :

1. Given a dual feasible solution ω^k , for some $\hat{x}^k \in \hat{x}(\omega^k)$, setting $\hat{x}^k = [\hat{x}_i^k]_{i=1, \dots, m}$, we take an average $\hat{z}^k = P_Z(Q\hat{x}^k)$. Based on this average, we compute a component-wise dispersion

$$\sigma_j^k := \sum_{i=1, \dots, m} |[Q_i \hat{x}_i^k - \hat{z}_i^k]_j|$$

for each component index $j = 1, \dots, n_x$, and denote $\sigma := [\sigma_j]_{j=1, \dots, n_x}$. This is the approach [in Caroe and Schultz]. One quirk with this approach is that, at the optimal dual solution ω^* , any of the previously generated solutions $\hat{x} \in \hat{x}(\omega^k)$ may also satisfy $\hat{x} \in \hat{\omega}^*$. This is especially true in the typically satisfied case that $c \in Z$. Also, the approach does not use information from the dispersions $\|x^* - P_X(x^*)\|$. The latter issue is addressed with the following modification to σ_j^k :

$$\sigma_j^k := \left\| \left(([\sigma_X]_i + \epsilon) [Q_i \hat{x}_i^k - \hat{z}_i^k] \right)_{i=1, \dots, m} \right\| + \|Qx^k - z^k\|$$

2. Let $\hat{x}^t \in \hat{x}(\omega^t)$ for $t = 0, 1, \dots, k$. We have for each $i = 1, \dots, m$ that $x_i^k = \sum_t \alpha^t \hat{x}^t$. Then another form of dispersion is given by:

$$\sigma_j := \sum_{i=1, \dots, m} \beta_i \sum_{t=1, \dots, k} \alpha_t |Q_i \hat{x}_{i,j}^t - z_{i,j}^k|$$

$$\sigma_j := \frac{1}{\sum_{i=1} [\sigma_X]_i + \epsilon} \sum_{i=1, \dots, m} [\sigma_X + \epsilon]_i \sum_{t=1, \dots, k} \alpha_t |Q_i (\hat{x}_{i,j}^t - x_{i,j}^k)| + \rho_{i,j} |Q_i x_{i,j}^k - z_j^k|,$$

where $0 \leq \alpha_t \leq 1$ for $t = 1, \dots, T_i$ and $\sum_{t=1, \dots, T_i} \alpha_t = 1$ are the convex multipliers used to form a convex combination $x^k = \sum_{t=1, \dots, k} \alpha_t \hat{x}^t$.

Remark 1 We also considered using a measure of the dispersion of the dual solutions ω^k , taking

$$\sigma_j := \left\| [\omega_{i,j}]_{i=1, \dots, m} \right\|.$$

This approach is (somewhat) sensible for when restricting the branching to variables with integrality restrictions, or when the disjunctions are otherwise strictly separated. It is otherwise an ill-defined approach, in that optimal dual solutions are typically non-unique, especially when bounding constraints are being added. Any further thoughts on this matter must take into account the set of dual optimal solutions.

3 Computational results

DCAP233500							
Baseline DD				Improved DD			
Time	ϕ_{best}	f_{best}	Nodes	Time	ϕ_{best}	f_{best}	Nodes
5 min	1737.48989	1740.64181	56	5 min	1737.50611	1737.64638	32
10 min	1737.49820	1740.64181	79	10 min	1737.52050	1737.52632	59
20 min	1737.50103	1740.64181	102	20 min	1737.52071	1737.52194	88
40 min	1737.50154	1737.53164	141	40 min	1737.52081	1737.52115	148

4 Conclusion and future work

References

1. Kiwiel, K.C.: Approximations in proximal bundle methods and decomposition of convex programs. *Journal of Optimization Theory and Applications* **84**(3), 529–548 (1995)