Noname manuscript No.
(will be inserted by the editor)

...

...

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

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

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

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

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:

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

Its well-known primal characterization takes the form

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

It is well-known [under our standing assumptions] that $f_C^* = \phi^*$, although in general, $\phi^* \leqslant f^*$, 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 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 $Z := \{(x, y) \in K : \exists\ z \in \mathbb{R}^n \text{ s.t. } x_i = z \text{ for } i = 1, \ldots, m\}$.

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

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

   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 $\text{P}_K((x^*, y^*))$. (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 $(x^*, y^*) - \text{P}_K((x^*, y^*))$ 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 $K^*(\omega^*) \cap Z = \varnothing$ in the case of duality gap $\phi_C^* < f^*$. We consider one of two projections:
   (a) Take the projection $\text{P}_Z(K(\omega^*))$. 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 $\widehat{x} \in K^*(\omega^*)$, 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 $\widehat{x}^t \in K^*(\omega^t)$ at each iteration $t = 1, \ldots, k$.

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,

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

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

with

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

and

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

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 $\upsilon \in \mathbb{R}$. 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 $\|(x^*, y^*) - \mathrm{P}_K(x^*, y^*)\| + \sum_{i=1}^{m} \|x_i^* - z^*\| < \epsilon$. The node is fathomed after testing whether $f_B^* < \upsilon$ and updating $\upsilon \leftarrow f_B^*$ if that is the case. (In Caroe and Schultz, it is the vertex solutions $(\widehat{x}, \widehat{y})$ that are tested only, so that the criterion is $\|\widehat{x}_i - \widehat{z}\| < \epsilon$, where $\widehat{z}$ is some averaging of $\widehat{x}_i$, $i = 1, \ldots, m$.)
3. Otherwise, branchings need to be determined.

Definitions:
$\mathcal{S} := \{f, K, B\}$, $\mathcal{P} := \{\rho, \gamma, \epsilon, t_{max}, k_{max}\}$
υ 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 υ
function PROCESS($\mathcal{S}$, $\mathcal{P}$, ω^0, υ, $(\widetilde{x}, \widetilde{y}, \widetilde{z})$)
$(K^*(\omega^*), (x^*, y^*, z^*), \omega^*, \phi^*) \leftarrow$ **bound**($\mathcal{S}$, $\mathcal{P}$ ω^0)
if $\breve{\phi}^* > \upsilon$ **or** $\breve{\phi}^* = \infty$ **then**
fathom node
$\mathcal{B} \leftarrow \varnothing$
return $(\upsilon, \widetilde{z}, \mathcal{B})$
end if
$\mathcal{B} \leftarrow$ **findBranchings**$(K^*(\omega^*), (x^*, y^*, z^*), \omega^*)$
$(\widetilde{x}, \widetilde{y}, \widetilde{z}, \upsilon) \leftarrow$ **findFeasibleSolution**$(z^*, \omega^*, \mathcal{S}, \mathcal{P})$
if $\mathcal{B} \neq \varnothing$ **then**
node needs to branch
else
fathom node (Due to optimality)
end if
return $(\widetilde{x}, \widetilde{y}, \widetilde{z}, \upsilon), \mathcal{B})$
end function
function BOUND($\mathcal{S}$, $\mathcal{P}$ ω^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 $\neg(\breve{\phi}^k > \upsilon$ **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$(\mathcal{S}, \mathcal{P}, D^k, x^k, z^k, \omega^k, \breve{\phi}^k)$
$k \leftarrow k + 1$
end while
return $\left(K^*(\omega^k), (x^k, y^k, z^k), (\widetilde{x}, \widetilde{y}, \widetilde{z}), \omega^k, \phi^k, \upsilon\right)$
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

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

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

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

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

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

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:
$(x^k, y^k) \in \text{conv}(K)$, $z^k \in \text{argmin}_z \left\{\sum_{i=1,\ldots,m} \|x_i - z\|^2\right\}$, $\omega^k \in \Omega$,
$\phi^k = \phi(\omega^k)$, $\{(x^k, y^k) + \alpha((\hat{x}, \hat{y}) - (x^k, y^k)) : \alpha \in [0, 1]\} \subseteq D \subseteq \text{conv}(K)$ where
$(\hat{x}, \hat{y}) \in \text{argmin}_{x,y} \{\nabla_{x,y} L_\rho(x^k, y^k, z^k, \omega^k)[(x - x^k); (y - y^k)] : (x, y) \in K\}$.
$\mathcal{S} := \{c, d, K, B\}$, $\mathcal{P} := \{\rho, \gamma, \epsilon, t_{max}, k_{max}\}$

function PSCG($\mathcal{S}$, $\mathcal{P}$, D^k, x^k, y^k, z^k, ω^k, ϕ^k)
  for $k = 1, 2, \ldots, k_{max}$ **do**
    Initialize $\omega^{k+1} \leftarrow \omega^k$, $\phi^{k+1} \leftarrow \phi^k$ ▷ (Default, null-step updates)
    $(\hat{x}^{k+1}, (x^{k+1}, y^{k+1}, z^{k+1}), D^{k+1}, \Gamma) \leftarrow$ SDM-GS($L_\rho(\cdot, \cdot, \cdot, \omega^k)$, $K \cap B$, D^k, x^k, y^k, z^k, t_{max})
    if $L_\rho(x^{k+1}, y^{k+1}, z^{k+1}, \omega^k) + \frac{\rho}{2} \sum_{i=1}^m \left\|x_i^{k+1} - z^{k+1}\right\|_2^2 - \phi^k \leqslant \epsilon$ **then**
      return $(x^{k+1}, y^{k+1}, z^{k+1}, \omega^{k+1}, \phi^{k+1}, D^{k+1}, \textbf{true})$
    end if
    $\tilde{\phi} \leftarrow L_\rho(x^{k+1}, y^{k+1}, z^{k+1}, \omega^k) + \frac{\rho}{2} \sum_{i=1}^m \left\|x_i^{k+1} - z^{k+1}\right\|_2^2 - \Gamma$
    $\tilde{\gamma} \leftarrow \frac{\tilde{\phi} - \phi^k}{L_\rho(x^{k+1}, y^{k+1}, z^{k+1}, \omega^k) + \frac{\rho}{2} \sum_{i=1}^m \left|x_i^{k+1} - z^{k+1}\right|_2^2 - \phi^k}$
    if $\tilde{\gamma} \geqslant \gamma$ **then**
      set $\omega^{k+1} \leftarrow \omega^k + \rho(x^{k+1} - z^{k+1})$, $\phi^{k+1} \leftarrow \tilde{\phi}$
    end if
    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]
  end for
  return $(\widehat{K}(\omega^{k+1}), (x^{k+1}, y^{k+1}, z^{k+1}), \omega^{k+1}, \phi^{k+1}, D^{k+1}, \textbf{false})$
end function

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

function SDM-GS(F, X, Z, D, $\tilde{x}$, $\tilde{z}$, t_{max})
  for $t = 1, \ldots, t_{max}$ **do**
    $\tilde{x} \leftarrow \text{argmin}_x \{F(x, y, \tilde{z}) : (x, y) \in D\}$
    $\tilde{z} \leftarrow \text{argmin}_z \{F(\tilde{x}, \tilde{y}, z) : z \in Z\}$
  end for
  $(\hat{x}, \hat{y}) \in \text{argmin}_{x,y} \{\nabla_{x,y} F(\tilde{x}, \tilde{y}, \tilde{z})[(x - \tilde{x}); (y - \tilde{y})] : (x, y) \in K\}$
  Reconstruct D to be any set such that
    $\{(\tilde{x}, \tilde{y}) + \alpha((\hat{x}, \hat{y}) - (\tilde{x}, \tilde{y})) : \alpha \in [0, 1]\} \subseteq D \subseteq \text{conv}(K)$
  Set $\Gamma \leftarrow -\nabla_{x,y} F(\tilde{x}, \tilde{y}, \tilde{z})[(\hat{x} - \tilde{x}); (\hat{y} - \tilde{y})]$
  return $\left(\widehat{K}, \tilde{x}, \tilde{y}, \tilde{z}, D, \Gamma\right)$
end function

Proposition 1 *Let* $\left\{(x^k, y^k, z^k, \omega^k)\right\}$ *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. $\omega^k = \overline{\omega}$ *is fixed and optimal for* (2) *for* $k \geqslant \bar{k}$ *for some finite* $\bar{k}$; *or*

2. ω^k *is never optimal for* (2) *for any finite* $k \geqslant 1$*, but* $\lim_{k\to\infty} \omega^k = \overline{\omega}$ *is optimal,*

and the sequence $\left\{(x^k, y^k, z^k)\right\}$ *has limit points* $(\overline{x}, \overline{y}, \overline{z})$*, 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

In the baseline DD branch-and-bound, a criterion for fathoming by optimality may be given by

$$\sqrt{\sum_{i=1}^{m} \|\widehat{x}_i - \widehat{z}\|_2^2} < \epsilon, \tag{10}$$

for some suitably small tolerance $\epsilon > 0$, where only the deviations from satisfaction of $\widehat{x}_i = \widehat{x}$, $i = 1, \ldots, m$, are taken into account since the integrality constraints on $\widehat{x}$ are satisfied by construction. We define for convenience the vector in $\mathbb{R}^n$:

$$\sigma_Z := \left[\sum_{i=1}^{m} \|\widehat{x}_{i,j} - \widehat{z}_j\|\right]_{j=1,\ldots,n}.$$

On the other hand, testing feasibility based on (x^*, y^*), we take

$$\sqrt{\sum_{i=1}^{m} \|(x^*, y^*) - \mathrm{P}_K((x^*, y^*))\|_2^2} < \epsilon, \tag{11}$$

for some suitably small tolerance $\epsilon > 0$, where it is assumed that the satisfaction of $x_i^* = z^*$ for each $i = 1, \ldots, m$ can be realized through the underlying dual iterative solution process at each node. We define for convenience the vector in $\mathbb{R}^m$:

$$\sigma_K := [\|(x_i^*, y_i^*) - \mathrm{P}_{K_i}((x_i^*, y_i^*))\|]_{i=1,\ldots,m}.$$

If either of the feasibility criteria (10) or (11) are not met, then branching must occur, and the branching should be informed by the dispersions measured in the respective norm terms of (10) and (11).

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 $\widehat{x}^k \in \widehat{x}(\omega^k)$, setting $\widehat{x}^k = \left[\widehat{x}^k_i\right]_{i=1,\ldots,m}$, we take an average $\widehat{z}^k = \mathrm{P}_Z(Q\widehat{x}^k)$. Based on this average, we compute a component-wise dispersion

$$\sigma^k_j := \sum_{i=1,\ldots,m} \left| \left[Q_i \widehat{x}^k_i - \widehat{z}^k_i \right]_j \right|$$

for each component index $j = 1, \ldots, n_x$, and denote $\sigma := \left[\sigma_j\right]_{j=1,\ldots,n_x}$. 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 $\widehat{x} \in \widehat{x}(\omega^k)$ may also satisfy $\widehat{x} \in \widehat{\omega}^*$. This is especially true in the typically satisfied case that $c \in Z$. Also, the approach does not use information from the dispersions $\|x^* - \mathrm{P}_X(x^*)\|$. The latter issue is addressed with the following modification to σ^k_j:

$$\sigma^k_j := \left\| \left[([\sigma_X]_i + \epsilon)[Q_i \widehat{x}^k_i) - \widehat{z}^k_i]_j \right]_{i=1,\ldots,m} \right\| + \left\| Qx^k - z^k \right\|$$

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

$$\sigma_j := \sum_{i=1,\ldots,m} \beta_i \sum_{t=1,\ldots,k} \alpha_t \left| Q_i \widehat{x}^t_{i,j} - z^k_{i,j} \right|$$

$$\sigma_j := \frac{1}{\sum_{i=1} [\sigma_X]_i + \epsilon} \sum_{i=1,\ldots,m} [\sigma_X + \epsilon]_i \sum_{t=1,\ldots,k} \alpha_t \left| Q_i \left(\widehat{x}^t_{i,j} - x^k_{i,j} \right) \right| + \rho_{i,j} \left| Q_i x^k_{i,j} - z^k_j \right|,$$

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

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

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

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 (using 32 processors)

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	1737.51795	1737.52526	35
10 min	1737.49820	1740.64181	79	10 min	1737.52047	1737.52069	64
20 min	1737.50103	1740.64181	102	20 min*	1737.52069	1737.52069	157
40 min	1737.50154	1737.53164	141	40 min*	*********	*********	***

DCAP243500 (using 32 processors)

Baseline DD				Improved DD			
Time	ϕ_{best}	f_{best}	Nodes	Time	ϕ_{best}	f_{best}	Nodes
5 min	2167.05580	2176.36500	44	5 min	2167.31697	2169.47892	31
10 min	2167.14788	2171.72801	57	10 min	2167.33071	2167.36434	42
20 min	2167.15808	2171.30679	74	20 min	2167.34274	2167.35855	65
40 min	2167.19891	2169.36526	89	40 min	2167.34750	2167.35373	104

DCAP332500 (using 32 processors)

Baseline DD				Improved DD			
Time	ϕ_{best}	f_{best}	Nodes	Time	ϕ_{best}	f_{best}	Nodes
5 min	1587.39480	1681.65011	50	5 min	1588.08901	1616.57914	32
10 min	1587.79126	1638.79564	71	10 min	1588.40129	1591.17931	55
20 min	1587.93833	1594.89727	97	20 min	1588.60162	1590.96571	91
40 min	1588.18939	1593.11529	158	40 min	1588.67017	1590.77970	128

DCAP342500 (using 32 processors)

Baseline DD				Improved DD			
Time	ϕ_{best}	f_{best}	Nodes	Time	ϕ_{best}	f_{best}	Nodes
5 min	1904.33558	1922.89408	55	5 min	1904.30712	1912.75878	43
10 min	1904.43416	1910.50410	104	10 min	1904.49445	1908.06074	64
20 min	1904.44938	1910.50410	140	20 min	1904.63236	1905.78144	112
40 min	1904.46666	1904.70152	206	40 min	1904.65008	1904.67035	166

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)