RLT-POS: Reformulation-Linearization Technique (RLT)-based Optimization Software for Polynomial Programming Problems

E. Dalkiran\textsuperscript{1} H.D. Sherali\textsuperscript{2}

\textsuperscript{1}The Department of Industrial and Systems Engineering, Wayne State University

\textsuperscript{2}Grado Department of Industrial and Systems Engineering, Virginia Tech

MINLP 2014

Acknowledgement: This research has been supported by the National Science Foundation under Grant No. CMMI-0969169.
Mathematical formulation: Problem PP

**PP**: Minimize $\phi_0(x)$
subject to

$\phi_r(x) \geq \beta_r$, $\forall r = 1, \ldots, R_1$

$\phi_r(x) = \beta_r$, $\forall r = R_1 + 1, \ldots, R$

$Ax = b$

$x \in \Omega \equiv \{x : 0 \leq l_j \leq x_j \leq u_j < \infty, \forall j \in \mathcal{N}\}$,

where

$\phi_r(x) \equiv \sum_{t \in T_r} \alpha_{rt} \left[ \prod_{j \in J_{rt}} x_j \right]$, for $r = 0, \ldots, R$. 

Mathematical formulation: Problem PP

**PP:** Minimize $\phi_0(x)$

subject to

\[
\phi_r(x) \geq \beta_r, \ \forall r = 1, \ldots, R_1 \\
\phi_r(x) = \beta_r, \ \forall r = R_1 + 1, \ldots, R \\
Ax = b \\
x \in \Omega \equiv \{x : 0 \leq l_j \leq x_j \leq u_j < \infty, \ \forall j \in \mathcal{N}\},
\]

where

\[
\phi_r(x) \equiv \sum_{t \in T_r} \alpha_{rt} \left[ \prod_{j \in J_{rt}} x_j \right], \text{ for } r = 0, \ldots, R.
\]

**Reformulation** Generate *bound-factors* and append *bound-factor constraints*:

- **Bound-factors:**
  \[
  (x_j - l_j) \geq 0 \text{ and } (u_j - x_j) \geq 0, \ \forall j \in \mathcal{N}.
  \]
Mathematical formulation: Problem PP

**PP:** Minimize $\phi_0(x)$
subject to

$$
\phi_r(x) \geq \beta_r, \forall r = 1, \ldots, R_1
$$
$$
\phi_r(x) = \beta_r, \forall r = R_1 + 1, \ldots, R
$$
$$
Ax = b
$$

$x \in \Omega \equiv \{x : 0 \leq l_j \leq x_j \leq u_j < \infty, \forall j \in \mathcal{N}\}$,

where

$$
\phi_r(x) \equiv \sum_{t \in T_r} \alpha_{rt} \left[ \prod_{j \in J_{rt}} x_j \right], \text{ for } r = 0, \ldots, R.
$$

**Reformulation** Generate *bound-factors* and append *bound-factor constraints*:

- **Bound-factors:**

  $$(x_j - l_j) \geq 0 \text{ and } (u_j - x_j) \geq 0, \forall j \in \mathcal{N}.$$  

- **Bound-factor constraints:**

  $$
  \prod_{j \in J_1} (x_j - l_j) \prod_{j \in J_2} (u_j - x_j) \geq 0, \forall (J_1 \cup J_2) \in \mathcal{N}^\delta.
  $$
Mathematical formulation: Problem PP

\textbf{PP:} Minimize \( \phi_0(x) \)

subject to

\( \phi_r(x) \geq \beta_r, \forall r = 1, \ldots, R_1 \)

\( \phi_r(x) = \beta_r, \forall r = R_1 + 1, \ldots, R \)

\( Ax = b \)

\( x \in \Omega \equiv \{ x : 0 \leq l_j \leq x_j \leq u_j < \infty, \forall j \in \mathcal{N} \} \),

where

\( \phi_r(x) \equiv \sum_{t \in T_r} \alpha_{rt} \left[ \prod_{j \in J_{rt}} x_j \right], \text{ for } r = 0, \ldots, R. \)

\textbf{Reformulation} Generate \textit{bound-factors} and append \textit{bound-factor constraints}:

\begin{itemize}
  \item Bound-factors:
  \[(x_j - l_j) \geq 0 \text{ and } (u_j - x_j) \geq 0, \forall j \in \mathcal{N} \].

  \item Bound-factor constraints:
  \[ \prod_{j \in J_1} (x_j - l_j) \prod_{j \in J_2} (u_j - x_j) \geq 0, \forall (J_1 \cup J_2) \in \mathcal{N}^\delta. \]
\end{itemize}

\textbf{Linearization} Substitute a new RLT variable for each distinct monomial as given by:

\[ X_J = \prod_{j \in J} x_j, \forall J \in \bigcup_{d=2}^{\delta} \mathcal{N}^d. \]
Reformulation-Linearization Technique (RLT):

**Minimize** \( [\phi_0(x)]_L \) subject to

\[
[\phi_r(x)]_L \geq \beta_r, \forall r = 1, \ldots, R_1 \tag{2b}
\]

\[
[\phi_r(x)]_L = \beta_r, \forall r = R_1 + 1, \ldots, R \tag{2c}
\]

\[Ax = b\] \hspace{1cm} (2d)

\[
\left[ \prod_{j \in J_1} (x_j - l'_j) \prod_{j \in J_2} (u'_j - x_j) \right]_L \geq 0, \forall (J_1 \cup J_2) \in \mathcal{N}^\delta \tag{2e}
\]

\(x \in \Omega' \equiv \{x : 0 \leq l'_j \leq x_j \leq u'_j < \infty, \forall j \in \mathcal{N}\}\) \hspace{1cm} (2f)

\(X_J = \prod_{j \in J} x_j, \forall J \in \bigcup_{d=2}^{\delta} \mathcal{N}^d\) \hspace{1cm} (2g)
\[ \text{RLT}(\Omega') : \quad \text{Minimize} \quad [\phi_0(x)]_L \]  
subject to

\[ [\phi_r(x)]_L \geq \beta_r, \forall r = 1, \ldots, R_1 \]  
\[ [\phi_r(x)]_L = \beta_r, \forall r = R_1 + 1, \ldots, R \]  
\[ Ax = b \]  
\[ \left[ \prod_{j \in J_1} (x_j - l'_j) \prod_{j \in J_2} (u'_j - x_j) \right]_L \geq 0, \forall (J_1 \cup J_2) \in \mathcal{N}^\delta \]  
\[ x \in \Omega' \equiv \{ x : 0 \leq l'_j \leq x_j \leq u'_j < \infty, \forall j \in \mathcal{N} \} \]  
\[ X_J = \prod_{j \in J} x_j, \forall J \in \cup_{d=2}^{\delta} \mathcal{N}^d. \]

1. Is there a strict subset of the fundamental bound-factor constraints for which the branch-and-bound algorithm described in Sherali and Tuncbilek [1992] would yet converge to a global optimum?

2. Is there additional valid inequalities that would tighten the RLT relaxation without sacrificing computational effort?
**Background**

**Reformulation-Linearization Technique (RLT)**

**RLT(Ω'):**

Minimize \( [\phi_0(x)]_L \) 
subject to

\[
[\phi_r(x)]_L \geq \beta_r, \forall r = 1, \ldots, R_1
\]

\[
[\phi_r(x)]_L = \beta_r, \forall r = R_1 + 1, \ldots, R
\]

\[
Ax = b
\]

\[
\left[ \prod_{j \in J_1} (x_j - l'_j) \prod_{j \in J_2} (u'_j - x_j) \right]_L \geq 0, \forall (J_1 \cup J_2) \in \mathcal{N}^\delta
\]

\[
x \in \Omega' \equiv \{x : 0 \leq l'_j \leq x_j \leq u'_j < \infty, \forall j \in \mathcal{N}\}
\]

\[
X_J = \prod_{j \in J} x_j, \forall J \in \bigcup_{d=2}^{\delta} \mathcal{N}^d.
\]

1. Is there a strict subset of the fundamental bound-factor constraints for which the branch-and-bound algorithm described in Sherali and Tuncbilek [1992] would yet converge to a global optimum?

2. Is there additional valid inequalities that would tighten the RLT relaxation without sacrificing computational effort?

**Model Enhancements:**

1. The \( J \)-set of filtered bound-factor constraints,
2. Reduced RLT representations or RLT formulations in the reduced space,
3. \( \nu \)-SDP cuts.
The $J$-set of bound-factor constraints

For a given index set $J$,
- $N_J \subseteq J$: the largest nonrepetitive set,
- $d_J$: the cardinality of $J$. 
The $J$-set of bound-factor constraints

For a given index set $J$,
- $N_J \subseteq J$: the largest nonrepetitive set,
- $d_J$: the cardinality of $J$.

Standard RLT constraints:

$$\left[ \prod_{j \in J_1} (x_j - l_j) \prod_{j \in J_2} (u_j - x_j) \right] \geq 0, \forall (J_1 \cup J_2) \in N_J^{d_J}.$$
The \( J \)-set of bound-factor constraints

For a given index set \( J \),
- \( N_J \subseteq J \): the largest nonrepetitive set,
- \( d_J \): the cardinality of \( J \).

Standard RLT constraints:

\[
\left[ \prod_{j \in J_1} (x_j - l_j) \prod_{j \in J_2} (u_j - x_j) \right]_L \geq 0, \forall (J_1 \cup J_2) \in N_J^{d_J}.
\]

Proposition

The convergence result in Sherali and Tuncbilek [1992] holds true if the following RLT bound-factor constraints are appended to the relaxation for each \( J \) such that the monomial \( \prod_{j \in J} x_j \) appears in Problem PP:

\[
\left[ \prod_{j \in J_1} (x_j - l_j) \prod_{j \in J_2} (u_j - x_j) \right]_L \geq 0, \forall (J_1 \cup J_2) = J.
\]
The $J$-set and the standard RLT

The $J$-set of bound-factor constraints for $x_1^2 x_2$:

\[
\begin{align*}
[(x_1 - l_1)(x_1 - l_1)(x_2 - l_2)]_L &\geq 0, \\
[(x_1 - l_1)(u_1 - x_1)(x_2 - l_2)]_L &\geq 0, \\
[(u_1 - x_1)(u_1 - x_1)(x_2 - l_2)]_L &\geq 0, \\
[(x_1 - l_1)(x_1 - l_1)(u_2 - x_2)]_L &\geq 0, \\
[(x_1 - l_1)(u_1 - x_1)(u_2 - x_2)]_L &\geq 0, \\
[(u_1 - x_1)(u_1 - x_1)(u_2 - x_2)]_L &\geq 0,
\end{align*}
\]
The $J$-set and the standard RLT

The $J$-set of bound-factor constraints for $x_1^2x_2$:

\[
\begin{align*}
& [(x_1 - l_1)(x_1 - l_1)(x_2 - l_2)]_L \geq 0, & & [(x_1 - l_1)(u_1 - x_1)(x_2 - l_2)]_L \geq 0, & & [(u_1 - x_1)(u_1 - x_1)(x_2 - l_2)]_L \geq 0, \\
& [(x_1 - l_1)(x_1 - l_1)(u_2 - x_2)]_L \geq 0, & & [(x_1 - l_1)(u_1 - x_1)(u_2 - x_2)]_L \geq 0, & & [(u_1 - x_1)(u_1 - x_1)(u_2 - x_2)]_L \geq 0.
\end{align*}
\]

The bound-factor constraints for $x_1^2x_2$ with the standard RLT:

\[
\begin{align*}
& [(x_1 - l_1)(x_1 - l_1)(x_1 - l_1)]_L \geq 0, & & [(x_1 - l_1)(u_1 - x_1)(u_1 - x_1)]_L \geq 0, \\
& [(x_1 - l_1)(x_1 - l_1)(u_1 - x_1)]_L \geq 0, & & [(u_1 - x_1)(u_1 - x_1)(u_1 - x_1)]_L \geq 0.
\end{align*}
\]

\[
\begin{align*}
& [(x_1 - l_1)(x_1 - l_1)(x_2 - l_2)]_L \geq 0, & & [(x_1 - l_1)(u_1 - x_1)(x_2 - l_2)]_L \geq 0, & & [(u_1 - x_1)(u_1 - x_1)(x_2 - l_2)]_L \geq 0, \\
& [(x_1 - l_1)(x_1 - l_1)(u_2 - x_2)]_L \geq 0, & & [(x_1 - l_1)(u_1 - x_1)(u_2 - x_2)]_L \geq 0, & & [(u_1 - x_1)(u_1 - x_1)(u_2 - x_2)]_L \geq 0, \\
& [(x_1 - l_1)(x_2 - l_2)(x_2 - l_2)]_L \geq 0, & & [(x_1 - l_1)(x_2 - x_2)(x_2 - x_2)]_L \geq 0, & & [(x_1 - l_1)(u_2 - x_2)(x_2 - x_2)]_L \geq 0, \\
& [(u_1 - x_1)(x_2 - l_2)(x_2 - l_2)]_L \geq 0, & & [(u_1 - x_1)(x_2 - x_2)(x_2 - x_2)]_L \geq 0, & & [(u_1 - x_1)(u_2 - x_2)(x_2 - x_2)]_L \geq 0, \\
& [(x_2 - l_2)(x_2 - l_2)(x_2 - l_2)]_L \geq 0, & & [(x_2 - l_2)(x_2 - x_2)(x_2 - x_2)]_L \geq 0, & & [(u_2 - x_2)(u_2 - x_2)(x_2 - x_2)]_L \geq 0, \\
& [(x_2 - l_2)(x_2 - l_2)(u_2 - x_2)]_L \geq 0, & & [(x_2 - l_2)(x_2 - x_2)(u_2 - x_2)]_L \geq 0, & & [(u_2 - x_2)(u_2 - x_2)(u_2 - x_2)]_L \geq 0.
\end{align*}
\]
The number of RLT constraints and new RLT variables

\[ \prod_{j \in J} x_j = \prod_{j \in N_j} x_j^{r_j} \]
The number of RLT constraints and new RLT variables

\[ \prod_{j \in J} x_j = \prod_{j \in N_j} x_j^{r_j} \]

**J-set**
- The number of bound-factor constraints: \( \prod_{j \in N_J} (r_j + 1) \)
- The number of new RLT variables: \( \prod_{j \in N_J} (r_j + 1) - (|N_J| + 1) \)

**N-δ-set**
- \( (2n + \delta - 1) \)
- \( \binom{n + \delta}{\delta} - (n + 1) \)
The number of RLT constraints and new RLT variables

\[ \prod_{j \in J} x_j = \prod_{j \in N_j} x_{j}^{r_j} \]

\( J \)-set: The number of bound-factor constraints
\[ \prod_{j \in N_j} (r_j + 1) \]
\( N^\delta \)-set: The number of new RLT variables
\[ \left( \begin{array}{c} 2n + \delta - 1 \\ \delta \end{array} \right) \]

Example: For a PP involving only \( x_1^3 x_2 x_3^2 \) and \( x_4^2 x_5 x_6^3 \) as nonlinear terms, the \( J \)-set and the \( N^\delta \)-set respectively generate in total:
- 48 and 12376 bound-factor constraints,
- 40 and 917 new RLT variables.
Reduced RLT representations

Linear Equality Subsystem: \( Ax = b \)
Constraint-based RLT restrictions:

\[
[(Ax = b) \times \prod_{j \in J} x_j]_L, \text{ yielding } AX_{(J)} = bX_J, \ \forall J \subseteq \mathcal{N}^d, d = 1, \ldots, \delta - 1. \tag{3}
\]
Reduced RLT representations

Linear Equality Subsystem: \( Ax = b \)
Constraint-based RLT restrictions:

\[
[(Ax = b) \times \prod_{j \in J} x_j]_L, \text{ yielding } AX(J) = bX_J, \ \forall J \subseteq \mathcal{N}^d, d = 1, \ldots, \delta - 1. \tag{3}
\]

Given a basis \( B \) of \( A \),

- \( Ax = b \Rightarrow Bx_B + Nx_N = b, \)
- \( AX(.,J) = bX_J, \Rightarrow BX(BJ) + NX(NJ) = bX_J, \ \forall J \subseteq \mathcal{N}^d, d = 1, \ldots, \delta - 1. \)
Reduced RLT representations

Linear Equality Subsystem: $Ax = b$
Constraint-based RLT restrictions:

$$[(Ax = b) \times \prod_{j \in J} x_j]_L, \text{ yielding } AX_{(., J)} = bX_J, \forall J \subseteq \mathcal{N}^d, d = 1, \ldots, \delta - 1. \quad (3)$$

Given a basis $B$ of $A$,

- $Ax = b \Rightarrow Bx_B + Nx_N = b$,
- $AX_{(., J)} = bX_J, \Rightarrow BX_{(BJ)} + NX_{(NJ)} = bX_J, \forall J \subseteq \mathcal{N}^d, d = 1, \ldots, \delta - 1.$

Proposition

Let the equality system $Ax = b$ be partitioned as $Bx_B + Nx_N = b$ for any basis $B$ of $A$, and define

$$Z = \left\{ (x, X) : Ax = b, (3) \text{ and } X_J = \prod_{j \in J} x_j, \forall J \subseteq J_N^d, \text{ for } d = 2, \ldots, \delta \right\}.$$

Then, we have $X_J = \prod_{j \in J} x_j, \forall J \subseteq \mathcal{N}^d, \text{ for } d = 2, \ldots, \delta \}.$
Equivalent formulations

**PP1:**

\[ BX_{(BJ)} + NX_{(NJ)} = bX, \forall J \subseteq \mathcal{N}_d, d = 1, \ldots, \delta - 1 \]

\[ \left[ \prod_{j \in J_1} (x_j - l_j) \prod_{j \in J_2} (u_j - x_j) \right]_L \geq 0, \forall (J_1 \cup J_2) \subseteq \mathcal{N}_\delta \]

\[ l \leq x \leq u \]

\[ X_J = \prod_{j \in J} x_j, \forall J \subseteq \mathcal{N}_d, d = 2, \ldots, \delta. \]
Equivalent formulations

**PP1:**

\[
BX_{(BJ)} + NX_{(NJ)} = bX_J, \quad \forall J \subseteq \mathcal{N}^d, \quad d = 1, \ldots, \delta - 1
\]

\[
\left[ \prod_{j \in J_1} (x_j - l_j) \prod_{j \in J_2} (u_j - x_j) \right]_L \geq 0, \quad \forall (J_1 \cup J_2) \subseteq \mathcal{N}^\delta
\]

\[
l \leq x \leq u
\]

\[
X_J = \prod_{j \in J} x_j, \quad \forall J \subseteq \mathcal{N}^d, \quad d = 2, \ldots, \delta.
\]

**PP2:**

\[
BX_{(BJ)} + NX_{(NJ)} = bX_J, \quad \forall J \subseteq \mathcal{N}^d, \quad d = 1, \ldots, \delta - 1
\]

\[
\left[ \prod_{j \in J_1} (x_j - l_j) \prod_{j \in J_2} (u_j - x_j) \right]_L \geq 0, \quad \forall (J_1 \cup J_2) \subseteq \mathcal{N}^\delta
\]

\[
l \leq x \leq u, \quad \text{and} \quad \prod_{j \in J} l_j \leq X_J \leq \prod_{j \in J} u_j, \quad \forall J \subseteq \mathcal{N}^d, \quad d = 2, \ldots, \delta, \quad |J_B \cap J| \geq 1
\]

\[
X_J = \prod_{j \in J} x_j, \quad \forall J \subseteq \mathcal{J}_N^d, \quad d = 2, \ldots, \delta.
\]
Equivalent formulations

**PP1:**

\[ BX_{(BJ)} + NX_{(NJ)} = bX_J, \forall J \subseteq \mathcal{N}^d, d = 1, \ldots, \delta - 1 \]

\[
\left[ \prod_{j \in J_1} (x_j - l_j) \prod_{j \in J_2} (u_j - x_j) \right]_L \geq 0, \forall (J_1 \cup J_2) \subseteq \mathcal{N}^\delta
\]

\[ l \leq x \leq u \]

\[ X_J = \prod_{j \in J} x_j, \forall J \subseteq \mathcal{N}^d, d = 2, \ldots, \delta. \]

**PP2:**

\[ BX_{(BJ)} + NX_{(NJ)} = bX_J, \forall J \subseteq \mathcal{N}^d, d = 1, \ldots, \delta - 1 \]

\[
\left[ \prod_{j \in J_1} (x_j - l_j) \prod_{j \in J_2} (u_j - x_j) \right]_L \geq 0, \forall (J_1 \cup J_2) \subseteq J_N^\delta
\]

\[ l \leq x \leq u, \text{ and } \prod_{j \in J} l_j \leq X_J \leq \prod_{j \in J} u_j, \forall J \subseteq \mathcal{N}^d, d = 2, \ldots, \delta, |J_B \cap J| \geq 1 \]

\[ X_J = \prod_{j \in J} x_j, \forall J \subseteq J_N^d, d = 2, \ldots, \delta. \]

**Enhancing Algorithm RLT(PP2):** Identify key bound-factor constraints, positive associated dual variables, at the root node and append within Problem PP2.
**RLT(PP1) in** $\mathbb{R}^{n-m}$ **and Hybrid algorithm**

**RLT(PP1) in** $\mathbb{R}^{n-m}$: Eliminate the basic variables $x_B$ for the basis $B$ via the substitution. Then, implement regular RLT process in the space of the $(n - m)$ nonbasic variables.
**RLT(PP1) in \( \mathbb{R}^{n-m} \) and Hybrid algorithm**

**RLT(PP1) in \( \mathbb{R}^{n-m} \):** Eliminate the basic variables \( x_B \) for the basis \( B \) via the substitution. Then, implement regular RLT process in the space of the \( (n - m) \) nonbasic variables.

**RLT\textsubscript{SDP}(PP2) vs. RLT\textsubscript{SDP}(PP1) in \( \mathbb{R}^{n-m} \)**

- The size of the LP relaxations,
- The quality of the lower bounds at the root node.
RLT(PP1) in $\mathbb{R}^{n-m}$ and Hybrid algorithm

**RLT(PP1) in $\mathbb{R}^{n-m}$**: Eliminate the basic variables $x_B$ for the basis $B$ via the substitution. Then, implement regular RLT process in the space of the $(n - m)$ nonbasic variables.

**RLT$_{SDP}$(PP2) vs. RLT$_{SDP}$(PP1) in $\mathbb{R}^{n-m}$**
- The size of the LP relaxations,
- The quality of the lower bounds at the root node.

**RLT$_{SDP}$(Hybrid)**
- The swiftness of RLT$_{SDP}$(PP1) in $\mathbb{R}^{n-m}$,
- The robustness of RLT$_{SDP}$(PP2).

Compute $\mu = \frac{\text{GAP}_1}{\text{GAP}_2} \times \frac{N_1}{N_2}$
**RLT(PP1) in** $\mathbb{R}^{n-m}$ **and Hybrid algorithm**

**RLT(PP1) in** $\mathbb{R}^{n-m}$: Eliminate the basic variables $x_B$ for the basis $B$ via the substitution. Then, implement regular RLT process in the space of the $(n-m)$ nonbasic variables.

**RLT$_{SDP}$(PP2) vs. RLT$_{SDP}$(PP1) in** $\mathbb{R}^{n-m}$
- The size of the LP relaxations,
- The quality of the lower bounds at the root node.

**RLT$_{SDP}$(Hybrid)**
- The swiftness of RLT$_{SDP}$(PP1) in $\mathbb{R}^{n-m}$,
- The robustness of RLT$_{SDP}$(PP2).

Compute $\mu = \frac{\text{GAP}_1}{\text{GAP}_2} \times \frac{N_1}{N_2}$

If ($\mu < 1$)
- implement RLT$_{SDP}$(PP2)
else
- implement RLT$_{SDP}$(PP1) in $\mathbb{R}^{n-m}$. 
**PP:** Minimize \( x_1 x_2 x_3 x_5^2 \)
subject to
\[
\begin{align*}
& x_1 + 0.5 x_3 + x_4 = 3 \\
& x_2 + x_5 = 6 \\
& x_3 x_5 - x_4^2 \geq 1.5 \\
& l_i \leq x_i \leq u_i, \ i = 1, 2, 3, 4, 5.
\end{align*}
\]
**PP:** Minimize \( x_1 x_2 x_3 x_5^2 \)

subject to

\[
\begin{align*}
    x_1 + 0.5x_3 + x_4 &= 3 \\
    x_2 + x_5 &= 6 \\
    x_3 x_5 - x_4^2 &\geq 1.5 \\
    l_i &\leq x_i \leq u_i, \; i = 1, 2, 3, 4, 5.
\end{align*}
\]

\[
\begin{align*}
    x_{12355} + x_{1355} - 6x_{1355} &= 0 \\
    x_{1355} + 0.5x_{3355} + x_{3455} - 3x_{355} &= 0 \\
    x_{1355} + 0.5x_{3355} + x_{3455} - 3x_{355} &= 0.
\end{align*}
\]
**PP:** Minimize \( x_1 x_2 x_3 x_5^2 \)
subject to
\[
\begin{align*}
&x_1 + 0.5x_3 + x_4 = 3 \\
&x_2 + x_5 = 6 \\
&x_3 x_5 - x_4^2 \geq 1.5 \\
&l_i \leq x_i \leq u_i, i = 1, 2, 3, 4, 5.
\end{align*}
\]
\[
X_{12355} + X_{1355} - 6X_{135} = 0 \\
X_{1355} + 0.5X_{33555} + X_{34555} - 3X_{3555} = 0 \\
X_{135} + 0.5X_{3355} + X_{3455} - 3X_{355} = 0.
\]

or
\[
X_{12355} + 0.5X_{23355} + X_{23455} - 3X_{2355} = 0 \\
X_{23355} + X_{33555} - 6X_{3355} = 0 \\
X_{23455} + X_{34555} - 6X_{3455} = 0 \\
X_{2355} + X_{3555} - 6X_{355} = 0.
\]
Reduced RLT Routine for Sparse Problems:

Initialization: Define $\mathcal{K} \equiv \{ J : X_J \text{ appears in (2a) - (2c)} \text{ and } |J \cap J_B| \geq 1 \}$, $\mathcal{K'} = \emptyset$, and $\mathcal{L} \equiv \{ J : X_J \text{ appears within (2a) - (2c)} \text{ and } J \cap J_B = \emptyset \}$.

Step 1: If $\mathcal{K} = \emptyset$, go to Step 4. Else, select an index set $J \in \mathcal{K}$ that involves the maximum number of basic variables, delete it from the list $\mathcal{K}$ and add it to the list $\mathcal{K'}$.

Step 2: Let $B_j \in J$ be a randomly selected basic variable index. Multiply the representation of the basic variable $x_{B_j}$ in terms of the nonbasic variables by $\prod_{J - \{B_j\}} x_j$, and linearize and append this to the relaxation.

Step 3: If $J - \{B_j\}$ involves any basic variable, the multiplication at Step 2 generates monomials involving basic variables. Include these monomials within the list $\mathcal{K}$. Otherwise, if $J - \{B_j\}$ does not involve any basic variable, then include the resulting monomials within the set $\mathcal{L}$. Continue with Step 1.

Step 4: Apply the $J$-set Routine to the set $\mathcal{L}$ in order to generate the proposed set of filtered bound-factor restrictions for reformulating the model.
Model enhancements
Coordination between constraint filtering and reduced basis techniques

**PP:** Minimize \( x_1 x_2 x_3 x_5^2 \)
subject to \( x_1 + 0.5x_3 + x_4 = 3 \)
\( x_2 + x_5 = 6 \)
\( x_3 x_5 - x_4^2 \geq 1.5 \)
\( l_i \leq x_i \leq u_i, i = 1, 2, 3, 4, 5. \)

**J-RRLT:** Minimize \( X_{12355} \)
subject to \( x_1 + 0.5x_3 + x_4 = 3 \)
\( X_{13555} + 0.5X_{33555} + X_{34555} - 3X_{3555} = 0 \)
\( X_{13555} + 0.5X_{3355} + X_{3455} - 3X_{355} = 0 \)
\( x_2 + x_5 = 6 \)
\( X_{12355} + X_{13555} - 6X_{1355} = 0 \)
\( X_{35} - X_{44} \geq 1.5 \)
\[ \left[ \prod_{j \in J_1} (x_j - l_j) \prod_{j \in J_2} (u_j - x_j) \right] \geq 0, \forall (J_1 \cup J_2) \in \mathcal{L} \]
\( l_i \leq x_i \leq u_i, i = 1, 2, 3, 4, 5, \)

where \( \mathcal{L} = \{\{4, 4\}, \{3, 3, 5, 5, 5\}, \{3, 4, 5, 5, 5\}\}. \)
$J$-set in $\mathbb{R}^{n-m}$: Minimize

$$18X_{355} - 3X_{3355} - 6X_{3455} - 3X_{3555} + 0.5X_{33555} + X_{34555}$$

subject to

$$X_{35} - X_{44} \geq 1.5$$

$$l_1 \leq 3 - 0.5x_3 - x_4 \leq u_1$$

$$l_2 \leq 6 - x_5 \leq u_2$$

$$\left[ \prod_{j \in J_1} (x_j - l_j) \prod_{j \in J_2} (u_j - x_j) \right] \geq 0, \forall (J_1 \cup J_2) \in \mathcal{L}$$

$$l_i \leq x_i \leq u_i, i = 3, 4, 5.$$
Table: The number of problems solved within the minimum CPU time among the \( J \)-set, \( J \)-RRLT+, and the \( J \)-NB.

<table>
<thead>
<tr>
<th>Degree</th>
<th>( J )-set</th>
<th>( J )-RRLT+</th>
<th>( J )-NB</th>
<th>( J )-Hybrid (Offline)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>13</td>
<td>6</td>
<td>0</td>
<td>13</td>
</tr>
<tr>
<td>3</td>
<td>11</td>
<td>9</td>
<td>3</td>
<td>16</td>
</tr>
<tr>
<td>4</td>
<td>11</td>
<td>9</td>
<td>4</td>
<td>20</td>
</tr>
<tr>
<td>5</td>
<td>8</td>
<td>7</td>
<td>9</td>
<td>17</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>3</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>3</td>
<td>14</td>
<td>20</td>
</tr>
<tr>
<td>Total</td>
<td>55</td>
<td>37</td>
<td>46</td>
<td>102</td>
</tr>
</tbody>
</table>
**Table:** The number of problems solved within the minimum CPU time among the \( J \)-set, \( J \)-RRLT+, and the \( J \)-NB.

<table>
<thead>
<tr>
<th>Degree</th>
<th>( J )-set</th>
<th>( J )-RRLT+</th>
<th>( J )-NB</th>
<th>( J )-Hybrid (Offline)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>13</td>
<td>6</td>
<td>0</td>
<td>13</td>
</tr>
<tr>
<td>3</td>
<td>11</td>
<td>9</td>
<td>3</td>
<td>16</td>
</tr>
<tr>
<td>4</td>
<td>11</td>
<td>9</td>
<td>4</td>
<td>20</td>
</tr>
<tr>
<td>5</td>
<td>8</td>
<td>7</td>
<td>9</td>
<td>17</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>3</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>3</td>
<td>14</td>
<td>20</td>
</tr>
<tr>
<td>Total</td>
<td>55</td>
<td>37</td>
<td>46</td>
<td>102</td>
</tr>
</tbody>
</table>

**Table:** Average CPU time (in seconds) with the reduced basis techniques for sparse problems.

<table>
<thead>
<tr>
<th>Degree</th>
<th>( J )-set</th>
<th>( J )-RRLT+</th>
<th>( J )-NB</th>
<th>( J )-Hybrid (Offline)</th>
<th>Minimum</th>
<th>( J )-Hybrid</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>136.4</td>
<td>134.8</td>
<td>246.0</td>
<td>111.4</td>
<td>94.4</td>
<td>112.2</td>
</tr>
<tr>
<td>3</td>
<td>182.0</td>
<td>166.0</td>
<td>266.6</td>
<td>122.6</td>
<td>80.3</td>
<td>122.6</td>
</tr>
<tr>
<td>4</td>
<td>230.1</td>
<td>226.3</td>
<td>318.5</td>
<td>122.2</td>
<td>109.2</td>
<td>134.8</td>
</tr>
<tr>
<td>5</td>
<td>170.2</td>
<td>125.7</td>
<td>200.6</td>
<td>62.9</td>
<td>46.8</td>
<td>69.0</td>
</tr>
<tr>
<td>6</td>
<td>97.0</td>
<td>42.0</td>
<td>98.7</td>
<td>42.3</td>
<td>23.6</td>
<td>51.1</td>
</tr>
<tr>
<td>7</td>
<td>132.9</td>
<td>75.1</td>
<td>181.2</td>
<td>62.5</td>
<td>38.0</td>
<td>72.7</td>
</tr>
</tbody>
</table>
\( \nu \)-SDP cuts

\[ xx^T \] is symmetric and positive semidefinite \( \Rightarrow M_0 = [xx^T]_L \succeq 0. \]
$\nu$-SDP cuts

$[xx^T]$ is symmetric and positive semidefinite $\Rightarrow M_0 = [xx^T]_L \succeq 0$.

A stronger implication in this same vein is:

$x(1) = \begin{bmatrix} 1 \\ x \end{bmatrix}$, and defining the matrix $M_1 \equiv [x(1)x(1)^T]_L = \begin{bmatrix} 1 & x^T \\ x & M_0 \end{bmatrix} \succeq 0$. 

\( [xx^T] \) is symmetric and positive semidefinite \( \Rightarrow M_0 = [xx^T] \succeq 0. \)

A stronger implication in this same vein is:
\[
x(1) = \begin{bmatrix} 1 \\ x \end{bmatrix}, \text{ and defining the matrix } M_1 \equiv [x(1)x(1)^T]_L = \begin{bmatrix} 1 & x^T \\ x & M_0 \end{bmatrix} \succeq 0.
\]

Two main approaches:
1. SDP relaxations,
2. SDP-induced valid inequalities.

\[
M = [\nu \nu^T] \succeq 0 \iff \alpha^T M \alpha = [(\alpha^T \nu)^2]_L \succeq 0, \forall \alpha \in \mathbb{R}^n (\text{or } \mathbb{R}^{n+1}), \|\alpha\| = 1.
\]
**ν-SDP cuts**

\[[xx^T]\] is symmetric and positive semidefinite \( \Rightarrow M_0 = [xx^T]_L \succeq 0.\)

A stronger implication in this same vein is:

\[
x(1) = \begin{bmatrix} 1 \\ x \end{bmatrix}, \text{ and defining the matrix } M_1 \equiv [x(1)x^T(1)]_L = \begin{bmatrix} 1 & x^T \\ x & M_0 \end{bmatrix} \succeq 0.
\]

Two main approaches:

1. SDP relaxations,
2. SDP-induced valid inequalities.

\[
M = [\nu \nu^T] \succeq 0 \iff \alpha^T M \alpha = [(\alpha^T \nu)^2]_L \geq 0, \forall \alpha \in \mathbb{R}^n (\text{or } \mathbb{R}^{n+1}), \|\alpha\| = 1.
\]

1. Let \((\bar{x}, \bar{X})\) be a solution to the RLT relaxation.
2. Check if \(\bar{M} \succeq 0\), where \(\bar{M}\) evaluates \(M\) at the solution \((\bar{x}, \bar{X})\).
\( \nu \)-SDP cuts

\[ xx^T \] is symmetric and positive semidefinite \( \Rightarrow M_0 = [xx^T]_L \succeq 0. \)

A stronger implication in this same vein is:

\[ x_{(1)} = \begin{bmatrix} 1 \\ x \end{bmatrix}, \text{ and defining the matrix } M_1 \equiv [x_{(1)}x_{(1)}^T]_L = \begin{bmatrix} 1 & x^T \\ x & M_0 \end{bmatrix} \succeq 0. \]

Two main approaches:

1. SDP relaxations,
2. SDP-induced valid inequalities.

\[ M = [\nu \nu^T] \succeq 0 \iff \alpha^T M \alpha = [(\alpha^T \nu)^2]_L \geq 0, \forall \alpha \in \mathbb{R}^n (\text{or } \mathbb{R}^{n+1}), \|\alpha\| = 1. \]

1. Let \((\bar{x}, \bar{X})\) be a solution to the RLT relaxation.
2. Check if \(\bar{M} \succeq 0\), where \(\bar{M}\) evaluates \(M\) at the solution \((\bar{x}, \bar{X})\).
   - If not, we have an \(\bar{\alpha} \in \mathbb{R}^{n+1}\) such that \(\bar{\alpha}^T \bar{M} \bar{\alpha} < 0\).
   - Append the SDP cut \(\bar{\alpha}^T \bar{M} \bar{\alpha} = [(\bar{\alpha}^T \nu)^2]_L \geq 0\), and go to Step 1.
**ν-vectors**

\[ \nu^{(1)} = \left[ 1, \{ x_j, j \in \mathcal{N} \}, \{ \text{all quadratic monomials using } x_j, j \in \mathcal{N} \}, \ldots, \{ \text{all monomials of order } \Delta \text{ using } x_j, j \in \mathcal{N} \} \right]^T \in \mathbb{R}^{(n+\Delta)} \]
\( \nu \)-vectors

\( \nu(1) = \begin{bmatrix} 1, \{ x_j, j \in N \}, \{ \text{all quadratic monomials using } x_j, j \in N \}, \ldots, \{ \text{all monomials of order } \Delta \text{ using } x_j, j \in N \} \end{bmatrix}^T \in \mathbb{R}^{(n+\Delta)\Delta}. \)

\( \nu(2) = \begin{bmatrix} 1, \{ x_j, j \in N_{J*} \}, \{ \text{all quadratic monomials using } x_j, j \in N_{J*} \}, \ldots, \{ \text{all monomials of order } \Delta \text{ using } x_j, j \in N_{J*} \} \end{bmatrix}^T \)

where

\[ J^* \in \arg \max_{J \subseteq \bar{N}} \left\{ \sum_{r \in \{0, 1, \ldots, R\}} \left| \alpha_{rt} \left[ \bar{X}_{Jrt} - \prod_{j \in J_{rt}} \bar{x}_j \right] \right| \right\} \]
\( \nu \)-vectors

\[
\nu^{(1)} = \begin{bmatrix} 1, \{x_j, j \in \mathcal{N}\}, \{\text{all quadratic monomials using } x_j, j \in \mathcal{N}\}, \ldots, \{\text{all monomials of order } \Delta \text{ using } x_j, j \in \mathcal{N}\} \end{bmatrix}^T \in \mathbb{R}^{(n+\Delta)}.
\]

\[
\nu^{(2)} = \begin{bmatrix} 1, \{x_j, j \in \mathcal{N}_J^*\}, \{\text{all quadratic monomials using } x_j, j \in \mathcal{N}_J^*\}, \ldots, \{\text{all monomials of order } \Delta \text{ using } x_j, j \in \mathcal{N}_J^*\} \end{bmatrix}^T
\]

where

\[
J^* \in \arg \max_{J \subseteq \bar{\mathcal{N}}} \left\{ \sum_{r \in \{0, 1, \ldots, R\}} |\alpha_{rt} [\bar{X}_{J_{rt}} - \prod_{j \in J_{rt}} \bar{x}_j]| \right\}.
\]

For a polynomial constraint \( \phi_r(x) \geq \beta_r \) of order \( \delta_r \), define \( \Delta_r \equiv \lceil \frac{\delta}{2} - \frac{\delta_r}{2} \rceil \). If \( \Delta_r \geq 1 \), let

\[
\nu^{(3)} = \begin{bmatrix} 1, \text{all monomials of order } \Delta_r \text{ using } x_j, j \in \mathcal{N} \end{bmatrix}^T.
\]

Then, we impose the following:

\[
\left\{ \left[ \phi_r(x) - \beta_r \right] \nu^{(3)} (\nu^{(3)})^T \right\}_L \succeq 0.
\]
\( \nu \)-SDP cut inheritance

- \( \nu \)-SDP parent
- \( \nu \)-SDP self
- \( \nu \)-SDP left-child
- \( \nu \)-SDP right-child

Copy all

Filter inactive cuts
Cut generation vs. branching

\[ \Delta \text{GAP}_{\text{branching}}^{i+1} = \frac{\text{GAP}_{\text{left}} - \text{GAP}_{\text{parent}}}{\max(1,|UB|)} \]

\[ \Delta \text{GAP}_{\text{branching}}^{i+2} \]

\[ \Delta \text{GAP}_{\text{branching}}^{i+3} \]

\[ \Delta \text{GAP}_{\text{branching}}^{i+4} \]

SDP cuts? 

Expected \( (\Delta \text{GAP}_{\text{SDP}}^{i+3}) > \frac{\Delta \text{GAP}_{\text{branching}}^{i+3}}{2} \)
Table: Performances of SDP cut generation routines.

<table>
<thead>
<tr>
<th>Degree</th>
<th>No SDP</th>
<th>Routine 1</th>
<th>Routine 2</th>
<th>Routine 3</th>
<th>Routine 4</th>
<th>No SDP</th>
<th>Routine 1</th>
<th>Routine 2</th>
<th>Routine 3</th>
<th>Routine 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>122.3</td>
<td>106.5</td>
<td>105.6</td>
<td>106.7</td>
<td>106.7</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>152.0</td>
<td>119.1</td>
<td>114.0</td>
<td>126.4</td>
<td>125.5</td>
<td>6</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>174.0</td>
<td>128.4</td>
<td>128.0</td>
<td>144.1</td>
<td>141.2</td>
<td>5</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>124.7</td>
<td>62.9</td>
<td>62.6</td>
<td>75.0</td>
<td>69.0</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>76.1</td>
<td>45.3</td>
<td>44.8</td>
<td>48.6</td>
<td>47.2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>103.9</td>
<td>77.0</td>
<td>71.5</td>
<td>83.9</td>
<td>76.6</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

- Routine 1: Generate SDP cuts and re-optimize the relaxation if they perform well. Else, generate and store SDP cuts for inheritance, if they perform well. Else, don’t generate.
- Routine 2: Generate and store SDP cuts for inheritance, if they perform well. Else, don’t generate.
- Routine 3: Generate SDP cuts and re-optimize the relaxation.
- Routine 4: Generate and store SDP cuts for inheritance.
**Figure**: Performance of the RLT algorithms for solving polynomial problems *without equality constraints* (in CPU seconds).

- **Degree-two**
- **Degree-four**
- **Degree-six**

The graphs show the CPU time (in seconds) for different problem densities and degrees, comparing the RLT and J-set algorithms.
Figure: Performance of the RLT algorithms for solving \textit{quadratic and cubic problems with equality constraints} (in CPU seconds).
Computational Results

Problems with equality constraints

Figure: Performance of the RLT Hybrid algorithms for solving degree-four, -five, -six, and -seven problems with equality constraints (in CPU seconds).
**RLT-POS vs. BARON**

### Degree-two

![Graph showing computational results for Degree-two problems with equality constraints]

### Degree-four

![Graph showing computational results for Degree-four problems with equality constraints]

### Degree-six

![Graph showing computational results for Degree-six problems with equality constraints]
- Coordination between constraint filtering and reduced basis techniques.
- SDP cut generation routine for sparse problems.
- The $J$-Hybrid algorithm.
- RLT-based open-source optimization software.
Conclusions and future research directions

- Coordination between constraint filtering and reduced basis techniques.
- SDP cut generation routine for sparse problems.
- The $J$-Hybrid algorithm.
- RLT-based open-source optimization software.

- Nonlinear equality constraints.
- Tighten the relaxation in the reduced subspace.
- Stability of $J$-set of relaxations: Barrier and dual optimizer of CPLEX.
- Factorable programming problems and nonlinear integer programming problems.
Thank you!