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

E. Dalkiran[1] H.D. Sherali[2]

[1] The Department of Industrial and Systems Engineering, Wayne State University

[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**

$$
\begin{aligned}
\textbf{PP:} \quad & \text{Minimize} \quad \phi_0(x) \\
& \text{subject to} \\
& \qquad \phi_r(x) \geq \beta_r, \forall r = 1, \ldots, R_1 \\
& \qquad \phi_r(x) = \beta_r, \forall r = R_1 + 1, \ldots, R \\
& \qquad Ax = b \\
& \qquad x \in \Omega \equiv \{x : 0 \leq l_j \leq x_j \leq u_j < \infty, \forall j \in \mathcal{N}\},
\end{aligned}
$$

where

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

Mathematical formulation: Problem **PP**

$$\textbf{PP:} \quad \text{Minimize} \quad \phi_0(x)$$
$$\text{subject to}$$
$$\phi_r(x) \geq \beta_r, \forall r = 1, \ldots, R_1$$
$$\phi_r(x) = \beta_r, \forall r = R_1 + 1, \ldots, R$$
$$Ax = b$$
$$x \in \Omega \equiv \{x : 0 \leq l_j \leq x_j \leq u_j < \infty, \forall j \in \mathcal{N}\},$$

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

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

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**

$$\begin{array}{lll} \textbf{PP:} & \text{Minimize} & \phi_0(x) \\ & \text{subject to} & \\ & & \phi_r(x) \ge \beta_r, \forall r = 1, \ldots, R_1 \\ & & \phi_r(x) = \beta_r, \forall r = R_1 + 1, \ldots, R \\ & & Ax = b \\ & & x \in \Omega \equiv \{x : 0 \le l_j \le x_j \le u_j < \infty, \forall j \in \mathcal{N}\}, \end{array}$$

where

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

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

- Bound-factors:

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

- Bound-factor constraints:

$$\prod_{j \in J_1} (x_j - l_j) \prod_{j \in J_2} (u_j - x_j) \ge 0, \forall (J_1 \cup J_2) \in \mathcal{N}^\delta.$$

Linearization Substitute a new RLT variable for each distinct monomial as given by:

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

RLT(Ω'): Minimize $[\phi_0(x)]_L$ (2a)

subject to

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

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

$$Ax = b \tag{2d}$$

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

$$x \in \Omega' \equiv \{x : 0 \leq l'_j \leq x_j \leq u'_j < \infty, \forall j \in \mathcal{N}\} \tag{2f}$$

$$X_J = \prod_{j \in J} x_j, \forall J \in \cup_{d=2}^{\delta} \mathcal{N}^d. \tag{2g}$$

$$\textbf{RLT}(\Omega'): \quad \text{Minimize} \quad [\phi_0(x)]_L \tag{2a}$$

subject to

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

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

$$Ax = b \tag{2d}$$

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

$$x \in \Omega' \equiv \{x : 0 \leq l'_j \leq x_j \leq u'_j < \infty, \forall j \in \mathcal{N}\} \tag{2f}$$

$$X_J = \prod_{j \in J} x_j, \forall J \in \cup_{d=2}^{\delta} \mathcal{N}^d. \tag{2g}$$

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?

RLT(Ω'): Minimize $[\phi_0(x)]_L$ (2a)

subject to

$$[\phi_r(x)]_L \geq \beta_r, \forall r = 1, \ldots, R_1 \quad (2b)$$

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

$$Ax = b \quad (2d)$$

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

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

$$X_J = \prod_{j \in J} x_j, \forall J \in \cup_{d=2}^{\delta} \mathcal{N}^d. \quad (2g)$$

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. ν-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:

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

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 $x_1^2 x_2$:

$$[(x_1 - l_1)(x_1 - l_1)(x_2 - l_2)]_L \geq 0, \quad [(x_1 - l_1)(u_1 - x_1)(x_2 - l_2)]_L \geq 0, \quad [(u_1 - x_1)(u_1 - x_1)(x_2 - l_2)]_L \geq 0,$$
$$[(x_1 - l_1)(x_1 - l_1)(u_2 - x_2)]_L \geq 0, \quad [(x_1 - l_1)(u_1 - x_1)(u_2 - x_2)]_L \geq 0, \quad [(u_1 - x_1)(u_1 - x_1)(u_2 - x_2)]_L \geq 0,$$

The J-set and the standard RLT

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

$$
\begin{array}{lll}
[(x_1 - l_1)(x_1 - l_1)(x_2 - l_2)]_L \geq 0, & [(x_1 - l_1)(u_1 - x_1)(x_2 - l_2)]_L \geq 0, & [(u_1 - x_1)(u_1 - x_1)(x_2 - l_2)]_L \geq 0, \\
[(x_1 - l_1)(x_1 - l_1)(u_2 - x_2)]_L \geq 0, & [(x_1 - l_1)(u_1 - x_1)(u_2 - x_2)]_L \geq 0, & [(u_1 - x_1)(u_1 - x_1)(u_2 - x_2)]_L \geq 0,
\end{array}
$$

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

$$
\begin{array}{ll}
[(x_1 - l_1)(x_1 - l_1)(x_1 - l_1)]_L \geq 0, & [(x_1 - l_1)(u_1 - x_1)(u_1 - x_1)]_L \geq 0, \\
[(x_1 - l_1)(x_1 - l_1)(u_1 - x_1)]_L \geq 0, & [(u_1 - x_1)(u_1 - x_1)(u_1 - x_1)]_L \geq 0.
\end{array}
$$

$$
\begin{array}{lll}
[(x_1 - l_1)(x_1 - l_1)(x_2 - l_2)]_L \geq 0, & [(x_1 - l_1)(u_1 - x_1)(x_2 - l_2)]_L \geq 0, & [(u_1 - x_1)(u_1 - x_1)(x_2 - l_2)]_L \geq 0, \\
[(x_1 - l_1)(x_1 - l_1)(u_2 - x_2)]_L \geq 0, & [(x_1 - l_1)(u_1 - x_1)(u_2 - x_2)]_L \geq 0, & [(u_1 - x_1)(u_1 - x_1)(u_2 - x_2)]_L \geq 0, \\
[(x_1 - l_1)(x_2 - l_2)(x_2 - l_2)]_L \geq 0, & [(x_1 - l_1)(x_2 - x_2)(u_2 - x_2)]_L \geq 0, & [(x_1 - l_1)(u_2 - x_2)(u_2 - x_2)]_L \geq 0, \\
[(u_1 - x_1)(x_2 - l_1)(x_2 - l_2)]_L \geq 0, & [(u_1 - x_1)(x_2 - l_2)(u_2 - x_2)]_L \geq 0, & [(u_1 - x_1)(u_2 - x_2)(u_2 - x_2)]_L \geq 0,
\end{array}
$$

$$
\begin{array}{ll}
[(x_2 - l_2)(x_2 - l_2)(x_2 - l_2)]_L \geq 0, & [(x_2 - l_2)(u_2 - x_2)(u_2 - x_2)]_L \geq 0, \\
[(x_2 - l_2)(x_2 - l_2)(u_2 - x_2)]_L \geq 0, & [(u_2 - x_2)(u_2 - x_2)(u_2 - x_2)]_L \geq 0.
\end{array}
$$

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

$$\prod_{j \in J} x_j = \prod_{j \in N_J} x_j^{r_j}$$

	The number of bound-factor constraints	The number of new RLT variables
J-set	$\prod_{j \in N_J} (r_j + 1)$	$\prod_{j \in N_J} (r_j + 1) - (\lvert N_J \rvert + 1)$
$\mathcal{N}^{\delta}$-set	$\binom{2n + \delta - 1}{\delta}$	$\binom{n + \delta}{\delta} - (n + 1)$

The number of RLT constraints and new RLT variables

$$\prod_{j \in J} x_j = \prod_{j \in N_J} x_j^{r_j}$$

	The number of bound-factor constraints	The number of new RLT variables
J-set	$\prod_{j \in N_J} (r_j + 1)$	$\prod_{j \in N_J} (r_j + 1) - (\lvert N_J \rvert + 1)$
$\mathcal{N}^{\delta}$-set	$\binom{2n + \delta - 1}{\delta}$	$\binom{n + \delta}{\delta} - (n + 1)$

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 $\mathcal{N}^{\delta}$-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:

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

Given a basis B of A,

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

Reduced RLT representations

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

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

Given a basis B of A,

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

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:

$$BX_{(BJ)} + NX_{(NJ)} = bX_J, \forall J \subseteq \mathcal{N}^d, d = 1, \ldots, \delta - 1$$
$$\left[\prod_{j \in J_1} (x_j - l_j) \prod_{j \in J_2} (u_j - x_j)\right]_L \geq 0, \forall (J_1 \cup J_2) \subseteq \mathcal{N}^\delta$$
$$l \leq x \leq u$$
$$X_J = \prod_{j \in J} x_j, \forall J \subseteq \mathcal{N}^d, d = 2, \ldots, \delta.$$

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:

$$BX_{(BJ)} + NX_{(NJ)} = bX_J, \forall J \subseteq \mathcal{N}^d, d = 1, \ldots, \delta - 1$$
$$\left[\prod_{j \in J_1} (x_j - l_j) \prod_{j \in J_2} (u_j - x_j)\right]_L \geq 0, \forall (J_1 \cup J_2) \subseteq J_N^\delta$$
$$l \leq x \leq u, \text{ and } \prod_{j \in J} l_j \leq X_J \leq \prod_{j \in J} u_j, \forall J \subseteq \mathcal{N}^d; d = 2, \ldots, \delta, |J_B \bar{\cap} J| \geq 1$$
$$X_J = \prod_{j \in J} x_j, \forall J \subseteq J_N^d, d = 2, \ldots, \delta.$$

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_{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(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$_{\text{SDP}}$(PP2) vs. RLT$_{\text{SDP}}$(PP1) in $\mathbb{R}^{n-m}$

- The size of the LP relaxations,
- The quality of the lower bounds at the root node.

RLT$_{\text{SDP}}$(Hybrid)

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

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

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 $\mu = \frac{\text{GAP}_1}{\text{GAP}_2} \times \frac{\text{N}_1}{\text{N}_2}$

If $(\mu < 1)$
    implement RLT_{SDP}(PP2)
else
    implement RLT_{SDP}(PP1) in $\mathbb{R}^{n-m}$.

$$
\textbf{PP:} \quad
\begin{aligned}
\text{Minimize} \quad & x_1 x_2 x_3 x_5^2 \\
\text{subject to} \quad & x_1 + 0.5x_3 + x_4 = 3 \\
& x_2 + x_5 = 6 \\
& x_3 x_5 - x_4^2 \geq 1.5 \\
& l_i \leq x_i \leq u_i, i = 1, 2, 3, 4, 5.
\end{aligned}
$$

PP:

$$\begin{aligned} \text{Minimize} \quad & x_1 x_2 x_3 x_5^2 \\ \text{subject to} \quad & x_1 + 0.5x_3 + x_4 = 3 \\ & x_2 + x_5 = 6 \\ & x_3 x_5 - x_4^2 \geq 1.5 \\ & l_i \leq x_i \leq u_i, i = 1, 2, 3, 4, 5. \end{aligned}$$

$$\begin{aligned} & X_{12355} + X_{13555} - 6X_{1355} = 0 \\ & X_{13555} + 0.5X_{33555} + X_{34555} - 3X_{3555} = 0 \\ & X_{1355} + 0.5X_{3355} + X_{3455} - 3X_{355} = 0. \end{aligned}$$

PP: Minimize $x_1 x_2 x_3 x_5^2$

subject to

$$x_1 + 0.5x_3 + x_4 = 3$$
$$x_2 + x_5 = 6$$
$$x_3 x_5 - x_4^2 \geq 1.5$$
$$l_i \leq x_i \leq u_i, i = 1, 2, 3, 4, 5.$$

$$X_{12355} + X_{13555} - 6X_{1355} = 0$$
$$X_{13555} + 0.5X_{33555} + X_{34555} - 3X_{3555} = 0$$
$$X_{1355} + 0.5X_{3355} + X_{3455} - 3X_{355} = 0.$$

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

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.

$$
\textbf{PP:} \quad \begin{array}{ll} \text{Minimize} & x_1 x_2 x_3 x_5^2 \\ \text{subject to} & x_1 + 0.5x_3 + x_4 = 3 \\ & x_2 + x_5 = 6 \\ & x_3 x_5 - x_4^2 \geq 1.5 \\ & l_i \leq x_i \leq u_i, i = 1, 2, 3, 4, 5. \end{array}
$$

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

where $\mathcal{L} = \{\{4, 4\}, \{3, 3, 5, 5, 5\}, \{3, 4, 5, 5, 5\}\}$.

J-set in $\mathbb{R}^{n-m}$:

$$
\begin{aligned}
\text{Minimize} \quad & 18X_{355} - 3X_{3355} - 6X_{3455} - 3X_{3555} + 0.5X_{33555} + X_{34555} \\
\text{subject to} \quad & X_{35} - X_{44} \geq 1.5 \\
& l_1 \leq 3 - 0.5x_3 - x_4 \leq u_1 \\
& l_2 \leq 6 - x_5 \leq u_2 \\
& \left[\prod_{j \in J_1} (x_j - l_j) \prod_{j \in J_2} (u_j - x_j) \right]_L \geq 0, \forall (J_1 \cup J_2) \in \mathcal{L} \\
& l_i \leq x_i \leq u_i, i = 3, 4, 5.
\end{aligned}
$$

Table : Relaxation sizes and optimality gaps.

	# of equalities	# of bound-factor constraints	% optimality gap
RLT	2	2002	65.2
RLT-E	250	2002	2.6
RRLT	250	252	87
RRLT+	250	252+55	2.6
J-set	2	27	81.6
J-RRLT+	5	31+20	52.3
J-NB	0	31	1279

Table : The number of problems solved within the minimum CPU time among the *J*-set, *J*-RRLT+, and the *J*-NB.

Degree	*J*-set	*J*-RRLT+	*J*-NB	*J*-Hybrid (Offline)
2	13	6	0	13
3	11	9	3	16
4	11	9	4	20
5	8	7	9	17
6	5	3	16	16
7	7	3	14	20
Total	55	37	46	102

Table : The number of problems solved within the minimum CPU time among the J-set, J-RRLT+, and the J-NB.

Degree	J-set	J-RRLT+	J-NB	J-Hybrid (Offline)
2	13	6	0	13
3	11	9	3	16
4	11	9	4	20
5	8	7	9	17
6	5	3	16	16
7	7	3	14	20
Total	55	37	46	102

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

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

ν-SDP cuts

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

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:

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

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:

$x_{(1)} = \begin{bmatrix} 1 \\ x \end{bmatrix}$, 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.

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

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:

$x_{(1)} = \begin{bmatrix} 1 \\ x \end{bmatrix}$, 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.

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

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 $(\bar{x}, \bar{X})$.

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:
$x_{(1)} = \begin{bmatrix} 1 \\ x \end{bmatrix}$, 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.

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

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 $\bar{\alpha}^T M\bar{\alpha} = [(\bar{\alpha}^T\nu)^2]_L \geq 0$, and go to Step 1.

ν-vectors

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

ν-vectors

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

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

where

$$J^* \in \arg\max_{J \subseteq \bar{\mathcal{N}}} \Bigg\{ \sum_{\substack{r \in \{0, 1, \ldots, R\} : \\ J_{rt} = J \text{ for some } t \in T_r}} \left| \alpha_{rt} \left[\bar{X}_{J_{rt}} - \prod_{j \in J_{rt}} \bar{x}_j \right] \right| \Bigg\}.$$

ν-vectors

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

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

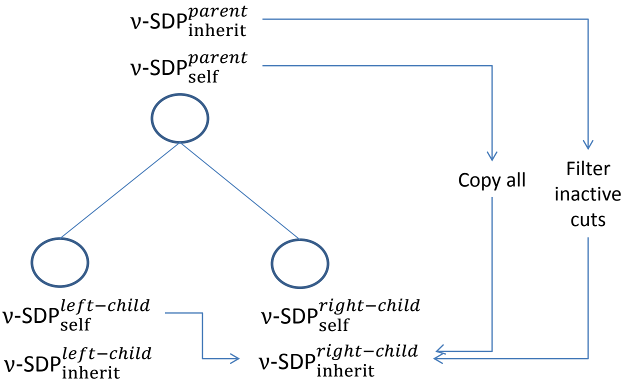
where

$$J^* \in \underset{J \subseteq \bar{\mathcal{N}}}{\arg\max} \left\{ \sum_{\substack{r \in \{0,1,\ldots,R\}: \\ J_{rt} = J \text{ for some } t \in T_r}} \left| \alpha_{rt} \left[\bar{X}_{J_{rt}} - \prod_{j \in J_{rt}} \bar{x}_j \right] \right| \right\}.$$

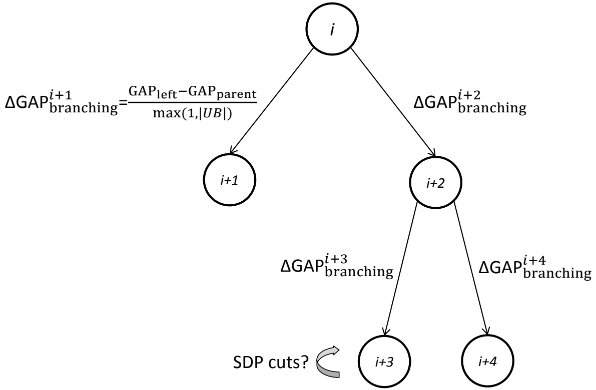
For a polynomial constraint $\phi_r(x) \geq \beta_r$ of order δ_r, define $\Delta_r \equiv \lfloor \frac{\delta}{2} - \frac{\delta_r}{2} \rfloor$. If $\Delta_r \geq 1$, let $\nu^{(3)} = [1, \text{all monomials of order } \Delta_r \text{ using } x_j, j \in \mathcal{N}]^T$. Then, we impose the following:

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

ν-SDP cut inheritance



Cut generation vs. branching



Expected $(\Delta\mathrm{GAP}^{i+3}_{\mathrm{SDP}}) > \frac{\Delta\mathrm{GAP}^{i+3}_{\mathrm{branching}}}{2}$

Table : Performances of SDP cut generation routines.

	Average CPU time (in seconds)					Number of unsolved problems				
Degree	No SDP	Routine 1	Routine 2	Routine 3	Routine 4	No SDP	Routine 1	Routine 2	Routine 3	Routine
2	122.3	106.5	105.6	106.7	106.7	6	6	6	6	6
3	152.0	119.1	114.0	126.4	125.5	6	3	3	4	4
4	174.0	128.4	128.0	144.1	141.2	5	3	3	3	3
5	124.7	62.9	62.6	75.0	69.0	3	0	0	1	1
6	76.1	45.3	44.8	48.6	47.2	2	1	1	1	1
7	103.9	77.0	71.5	83.9	76.6	1	0	0	0	0

- 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).

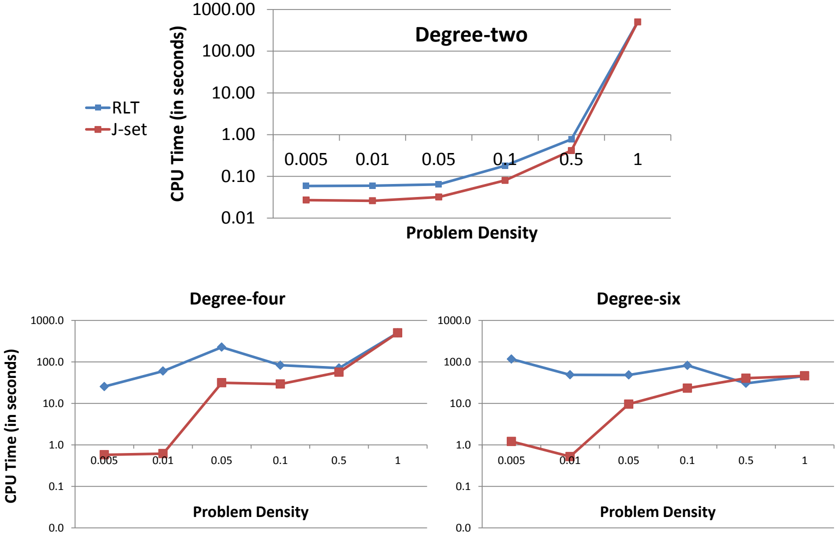
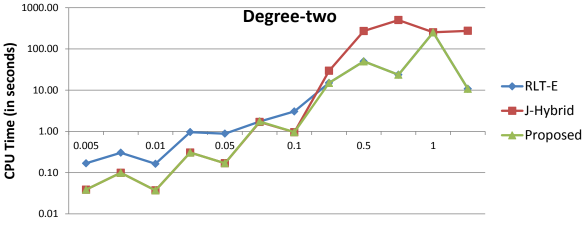


Figure : Performance of the RLT algorithms for solving **quadratic and cubic problems with equality constraints** (in CPU seconds).



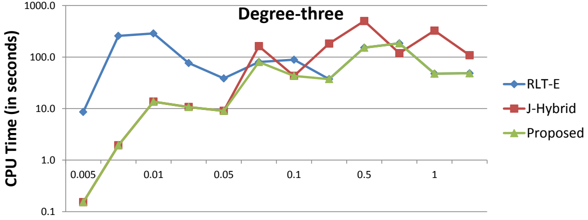
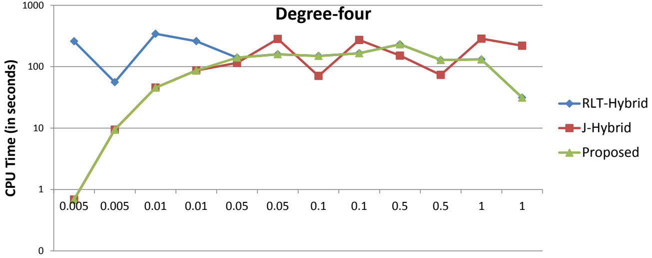
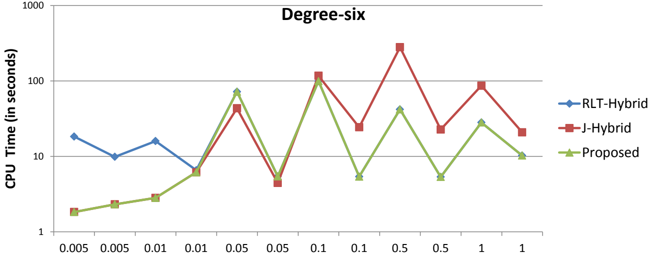
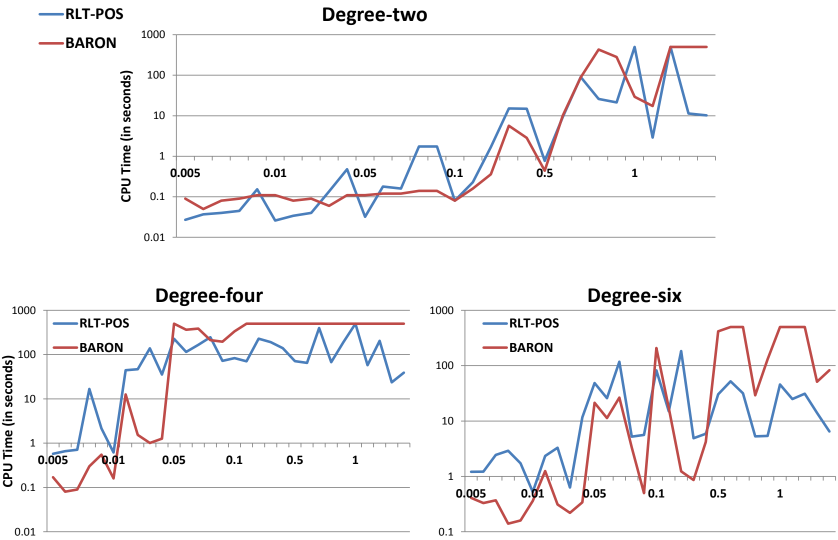


Figure : Performance of the RLT Hybrid algorithms for solving **degree-four, -five, -six, and -seven problems with equality constraints** (in CPU seconds).





RLT-POS vs. BARON



- 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.

- 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!