

Compact relaxations for polynomial programming problems

S. Cafieri (ENAC Toulouse, F)

L. Létocart (LIPN Paris13, F)

F. Messine (ENSEEIH Toulouse, F)

P. Hansen (GERAD & HEC, CA)

L. Liberti (LIX, Ecole Polytechnique, F)

June 1, 2012

▷ Introduction

The setting

Exact reformulations

Reformulation-
Linearization
Technique

RLT literature
review

Reduced RLT

RRLT literature
review

New developments

Why bother?

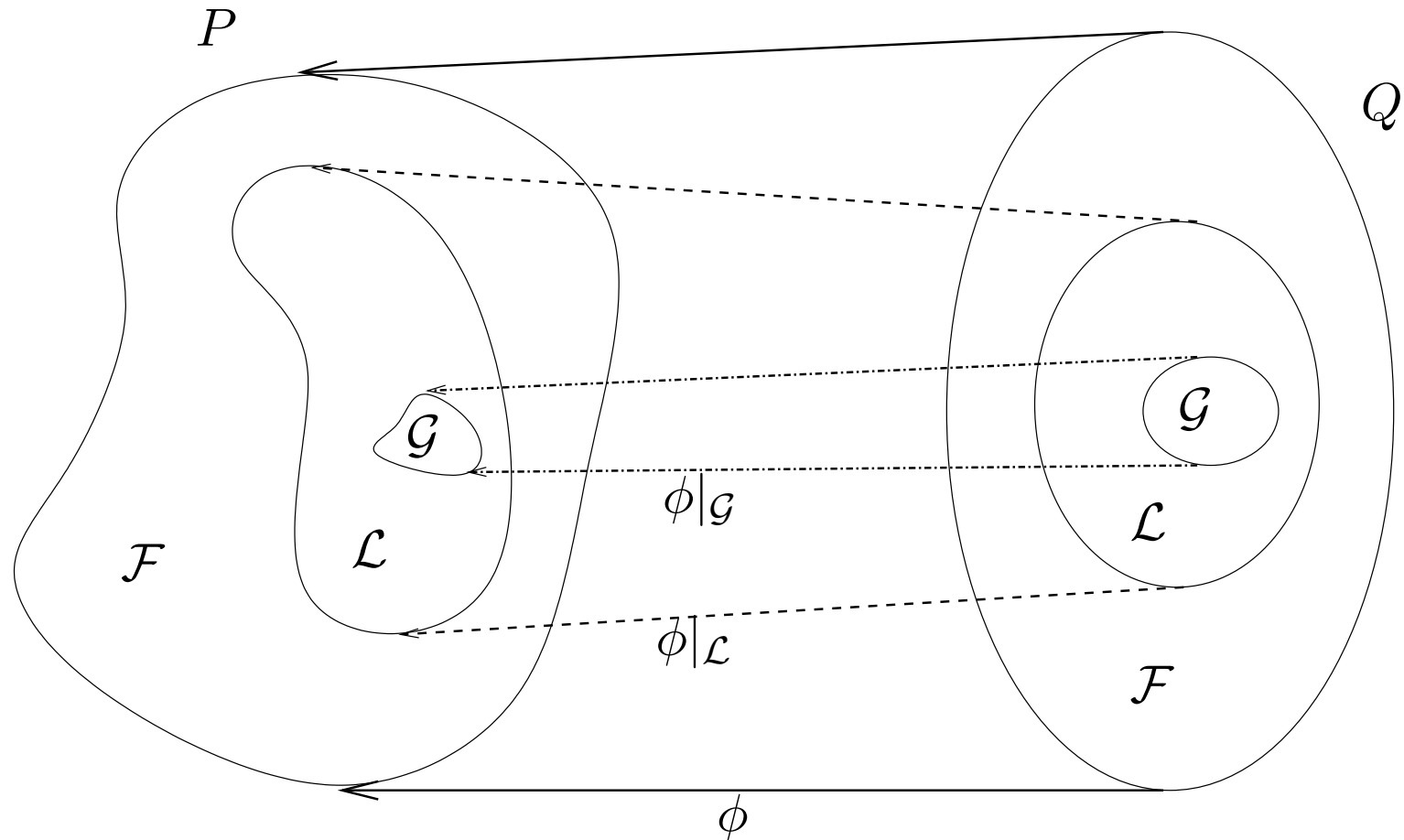
Thank you

Introduction

- Introduction
 - ▷ The setting
- Exact reformulations
- Reformulation-Linearization Technique
- RLT literature review
- Reduced RLT
- RRLT literature review
- New developments
- Why bother?
- Thank you

- ☐ Focus on (nonconvex) polynomial programming problems
- ☐ Aim to solve with sBB
- ☐ Need a tight convex (linear) relaxation at each node
- ☐ Reformulate before relaxing

Introduction
The setting
Exact
▷ reformulations
Reformulation-Linearization
Technique
RLT literature review
Reduced RLT
RRLT literature review
New developments
Why bother?
Thank you



- ☐ P harder than Q
- ☐ find optima in Q , map them back to P
- ☐ for each opt. in $P \exists$ corresponding opt. in Q

Introduction

Reformulation-
Linearization

▷ Technique

Aim

RLT constraints 1/2

RLT constraints 2/2

Linearization

Relaxation 1/2

Relaxation 2/2

RLT literature
review

Reduced RLT

RRLT literature
review

New developments

Why bother?

Thank you

Reformulation-Linearization Technique

Introduction

Reformulation-
Linearization
Technique

▷ Aim

RLT constraints 1/2

RLT constraints 2/2

Linearization

Relaxation 1/2

Relaxation 2/2

RLT literature
review

Reduced RLT

RRLT literature
review

New developments

Why bother?

Thank you

Reformulation-Linearization Technique (RLT):
tightens the linear relaxation of mixed-integer (nonconvex)
QCQPs

$$\left. \begin{array}{ll} \min_{\mathbf{x} \in \mathbb{R}^{n_1} \times \mathbb{Z}^{n_2}} & \mathbf{c}_0 \mathbf{x} + \mathbf{x} Q_0 \mathbf{x} \\ \forall 1 \leq i \leq q & \mathbf{c}_i \mathbf{x} + \mathbf{x} Q_i \mathbf{x} \leq 0 \\ \forall 1 \leq i \leq m & \mathbf{a}_i \mathbf{x} \leq b_i \\ & \mathbf{x} \in [\mathbf{x}^L, \mathbf{x}^U]. \end{array} \right\}$$

Introduction

Reformulation-
Linearization
Technique

Aim

RLT constraints
▷ 1/2

RLT constraints 2/2

Linearization

Relaxation 1/2

Relaxation 2/2

RLT literature
review

Reduced RLT

RRLT literature
review

New developments

Why bother?

Thank you

$\forall j, \ell \leq n = n_1 + n_2, i \leq m$, all factors non-negative \Rightarrow
constraints are valid

$$(x_j - x_j^L)(x_\ell - x_\ell^L) \geq 0$$

$$(x_j - x_j^L)(x_\ell^U - x_\ell) \geq 0$$

$$(x_j^U - x_j)(x_\ell - x_\ell^L) \geq 0$$

$$(x_j^U - x_j)(x_\ell^U - x_\ell) \geq 0$$

$$(x_j - x_j^L)(b_i - \mathbf{a}_i \mathbf{x}) \geq 0$$

$$(x_j^U - x_j)(b_i - \mathbf{a}_i \mathbf{x}) \geq 0$$

Introduction

Reformulation-
Linearization
Technique

Aim

RLT constraints 1/2

RLT constraints
▷ 2/2

Linearization

Relaxation 1/2

Relaxation 2/2

RLT literature
review

Reduced RLT

RRLT literature
review

New developments

Why bother?

Thank you

$\forall j, \ell \leq n, i \leq m$, get:

$$x_j x_\ell - x_\ell^L x_j - x_j^L x_\ell + x_j^L x_\ell^L \geq 0$$

$$-x_j x_\ell + x_\ell^U x_j + x_j^L x_\ell - x_j^L x_\ell^U \geq 0$$

$$-x_j x_\ell + x_\ell^L x_j + x_j^U x_\ell - x_j^U x_\ell^L \geq 0$$

$$x_j x_\ell - x_\ell^U x_j - x_j^U x_\ell + x_j^U x_\ell^U \geq 0$$

$$-x_j \mathbf{a}_i \mathbf{x} + x_j^L \mathbf{a}_i \mathbf{x} + x_j b_i - x_j^L b_i \geq 0$$

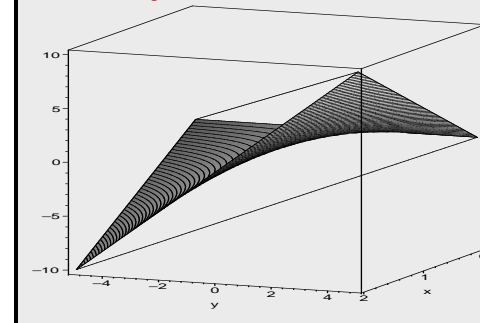
$$x_j \mathbf{a}_i \mathbf{x} - x_j^U \mathbf{a}_i \mathbf{x} - x_j b_i + x_j^U b_i \geq 0$$

- Introduction
- Reformulation-
Linearization
Technique
- Aim
- RLT constraints 1/2
- RLT constraints 2/2
- ▷ Linearization
- Relaxation 1/2
- Relaxation 2/2
- RLT literature
review
- Reduced RLT
- RRLT literature
review
- New developments
- Why bother?
- Thank you

Replace $x_j x_\ell$ by var. $w_{j\ell}$ ($\mathbf{w}_j = (w_{j1}, \dots, w_{jn})$):

$$\left. \begin{aligned} w_{j\ell} &\geq x_\ell^L x_j + x_j^L x_\ell - x_j^L x_\ell^L \\ w_{j\ell} &\leq x_\ell^U x_j + x_j^L x_\ell - x_j^L x_\ell^U \\ w_{j\ell} &\leq x_\ell^L x_j + x_j^U x_\ell - x_j^U x_\ell^L \\ w_{j\ell} &\geq x_\ell^U x_j + x_j^U x_\ell - x_j^U x_\ell^U \end{aligned} \right\}$$

(McCormick's convex/concave envelopes for $x_j x_\ell$)



$$\left. \begin{aligned} \mathbf{a}_i \mathbf{w}_j &\leq x_j^L \mathbf{a}_i \mathbf{x} + x_j b_i - x_j^L b_i \\ \mathbf{a}_i \mathbf{w}_j &\geq x_j^U \mathbf{a}_i \mathbf{x} + x_j b_i - x_j^U b_i \end{aligned} \right\}$$

valid linear relations between \mathbf{x} and \mathbf{w}

Introduction

Reformulation-
Linearization
Technique

Aim

RLT constraints 1/2

RLT constraints 2/2

Linearization

▷ Relaxation 1/2

Relaxation 2/2

RLT literature
review

Reduced RLT

RRLT literature
review

New developments

Why bother?

Thank you

- $\forall i \leq q$, replace terms $x_i x_j$ in $\mathbf{x} Q_i \mathbf{x}$ with linearizing variables w_{ij}

$$\mathbf{w} = \begin{pmatrix} w_{11} & \dots & w_{1n} \\ \vdots & \ddots & \vdots \\ w_{n1} & \dots & w_{nn} \end{pmatrix}$$

- Get $\mathbf{x} Q_i \mathbf{x} = Q_i \mathbf{w}$
- Use IA on $[\mathbf{x}^L, \mathbf{x}^U]$ to compute ranges $[\mathbf{w}^L, \mathbf{w}^U]$ for \mathbf{w}
- Adjoin *commutativity constr.* $\forall j \leq \ell \leq n \ w_{j\ell} = w_{\ell j}$
- Adjoin *square constr.* $\forall i \leq n$ s.t. x_i is binary $w_{ii} = x_i$

Introduction

Reformulation-
Linearization
Technique

Aim

RLT constraints 1/2

RLT constraints 2/2

Linearization

Relaxation 1/2

▷ Relaxation 2/2

RLT literature
review

Reduced RLT

RRLT literature
review

New developments

Why bother?

Thank you

Relaxed MILP

$$\left. \begin{array}{ll} \min & \mathbf{c}_0 \mathbf{x} + Q_0 \mathbf{w} \\ \forall i \leq q & \mathbf{c}_i \mathbf{x} + Q_i \mathbf{w} \leq 0 \\ \forall i \leq m & \mathbf{a}_i \mathbf{x} \leq b_i \\ & (\text{RLT} + \text{Commutativity} + \text{Square constraints}) \\ & \mathbf{x} \in (\mathbb{R}^{n_1} \times \mathbb{Z}^{n_2}) \cap [\mathbf{x}^L, \mathbf{x}^U] \\ & \mathbf{w} \in \mathbb{R}^{n^2} \cap [\mathbf{w}^L, \mathbf{w}^U] \end{array} \right\}$$

Introduction

Reformulation-
Linearization
Technique

▷ RLT literature
review

The first paper
Mixed products
Relaxation of
bilinear terms
Mixed products
again
Relaxation hierarchy
Etc.

Reduced RLT

RRLT literature
review

New developments

Why bother?

Thank you

RLT literature review

- Introduction
- Reformulation-Linearization Technique
- RLT literature review
 - ▷ The first paper
- Mixed products
- Relaxation of bilinear terms
- Mixed products again
- Relaxation hierarchy
- Etc.
- Reduced RLT
- RRLT literature review
- New developments
- Why bother?
- Thank you

Seminal paper

Adams & Sherali, Mgt. Sci. 1986:

$$\mathbf{x} \in \{0, 1\}^n$$

MILP reformulation via Fortet's inequalities

$$w_{j\ell} \leq \min(x_j, x_\ell)$$

$$w_{j\ell} \geq \max(0, x_j + x_\ell - 1)$$

then continuous relaxation

MANAGEMENT SCIENCE
Vol. 32, No. 10, October 1986
Printed in U.S.A.

A TIGHT LINEARIZATION AND AN ALGORITHM FOR ZERO-ONE QUADRATIC PROGRAMMING PROBLEMS*

WARREN P. ADAMS AND HANIF D. SHERALI

Department of Mathematical Sciences, Clemson University, Clemson, South Carolina 29631
Department of Industrial Engineering and Operations Research, Virginia Polytechnic Institute and State University, Blacksburg, Virginia 24061

This paper is concerned with the solution of linearly constrained zero-one quadratic programming problems. Problems of this kind arise in numerous economic, location decision, and strategic planning situations, including capital budgeting, facility location, quadratic assignment, media selection, and dynamic set covering. A new linearization technique is presented for this problem which is demonstrated to yield a tighter continuous or linear programming relaxation than is available through other methods. An implicit enumeration algorithm which uses Lagrangian relaxation, Benders' cutting planes, and local explorations is designed to exploit the strength of this linearization. Computational experience is provided to demonstrate the usefulness of the proposed linearization and algorithm.

(ZERO-ONE QUADRATIC PROGRAMMING; LINEARIZATION TECHNIQUES; IMPLICIT ENUMERATION; LAGRANGIAN RELAXATION; BENDERS' DECOMPOSITION)

1. Introduction

This paper is concerned with linearly-constrained zero-one quadratic programming problems (QP). Problems of this structure arise in numerous economic, facility location, and strategic planning situations. These include, for example, the media selection problem described by Zangwill (1965), the capital budgeting problem considered by Laughhunn (1970) and by Peterson and Laughhunn (1971), certain facility location problems given in Vaish (1974), a multitude of applications associated with quadratic assignment problems (see Sherali 1979), and dynamic facility relocation problems such as the dynamic set covering problem formulated in Chrissis *et al.* (1978).

Basically, the available solution procedures for Problem QP may be classified as attempting either to solve the problem directly or to transform Problem QP into an equivalent linear mixed-integer program, and then solve the latter problem. The direct methods involve partitioning techniques or implicit enumeration strategies. Lazimy (1982) considers a more general version of Problem QP, i.e., a mixed-integer quadratic programming problem, and demonstrates that one can transform this problem into an equivalent program without adding new variables or constraints. This renders the problem more amenable to Geoffrion's (1972) generalized Benders' technique, in that the cutting planes derived are linear in the integer variables. Balas (1969) suggests an alternative partitioning algorithm for the same class of problems, but requires the addition of an extra set of variables. Both these methods, however, are based on the use of a Dorn (1960) type of duality, and therefore require appropriate convexity assumptions in regard to the objective function. Another direct-search cutting plane algorithm is suggested by Kunzi and Oettli (1963) for all-integer quadratic programming problems.

The direct implicit enumeration procedures developed for Problem QP are more straightforward. These methods include the works of Cabot and Francis (1970), Hansen (1972), Mao and Wallingford (1968), and McBride and Yormark (1980). Recently, Carter (1984) has suggested methods for improving various direct-search schemes by

* Accepted by Alexander H. G. Rinnooy Kan; received July 1984. This paper was with the authors 4 months for 1 revision.

Introduction
Reformulation- Linearization Technique
RLT literature review
The first paper
▷ Mixed products
Relaxation of bilinear terms
Mixed products again
Relaxation hierarchy
Etc.
Reduced RLT
RRLT literature review
New developments
Why bother?
Thank you

Adams & Sherali, Op. Res. 1990:

$\mathbf{x} \in \mathbb{R}^{n_1} \times \{0, 1\}^{n_2}$ with products
involving at least one binary
variable

MILP reformulation

Theorem 1. *Problems MIQPP and MILPR (and, hence, MILP) are equivalent in the following sense. Given any (x, y) feasible to MIQPP, there exists a (w, u) such that (x, y, w, u) is feasible to MILPR with the same objective value. Conversely, given any (x, y, w, u) feasible to MILPR, the solution (x, y) is feasible to MIQPP with the same objective value.*

then continuous relaxation

LINEARIZATION STRATEGIES FOR A CLASS OF ZERO-ONE MIXED INTEGER PROGRAMMING PROBLEMS

WARREN P. ADAMS

Clemson University, Clemson, South Carolina

HANIF D. SHERALI

Virginia Polytechnic Institute and State University, Blacksburg, Virginia

(Received April 1984; revisions received July 1986, April 1988; accepted January 1989)

This paper is concerned with a new linearization strategy for a class of zero-one mixed integer programming problems that contains quadratic cross-product terms between continuous and binary variables, and between the binary variables themselves. This linearization scheme provides an equivalent mixed integer linear programming problem which yields a tighter continuous relaxation than that obtainable via the alternative linearization techniques available in the literature. Moreover, the proposed technique provides a unifying framework in the sense that all the alternate methods lead to formulations that are accessible through appropriate surrogates of the constraints of the new linearized formulation. Extensions to various other types of mixed integer nonlinear programming problems are also discussed.

A commonly used technique in the development of solution procedures for nonlinear integer and mixed integer programming problems is that of linearization, that is, the construction of an equivalent linear (mixed) integer representation of the problem. Numerous authors, including Glover (1975), Glover and Woolsey (1973, 1974), Petersen (1971), Watters (1967) and Zangwill (1965) have suggested methods for obtaining linear reformulations of various classes of discrete nonlinear programming problems. The methods they propose appear to be seemingly unrelated, and the success obtained via these methods is highly dependent on the specific problem. Unfortunately, no central or unifying theory has heretofore been presented.

A crucial concern in the construction of a linear (mixed) integer representation of a nonlinear mixed integer programming problem is how well the linear formulation lends itself to existing solution strategies. As reported by many authors, for example, Bazaraa and Sherali (1980), Geoffrion and Graves (1974), Geoffrion and McBride (1979), Magnanti and Wong (1981), McDaniel and Devine (1977), Rardin and Unger (1976) and Williams (1974), an integer linear formulation whose continuous, or linear programming, relaxation closely approximates the convex hull of feasible integer solutions in the vicinity of the optimum is computationally advantageous. Hence, a linearization technique that provides a tight linear

programming relaxation, while keeping the problem computationally tractable, is highly desirable.

The purpose of this paper is to present a new linearization technique for a general class of nonlinear mixed integer programming problems. This linearization technique possesses two principle advantages. First, it provides a type of unifying theory for the various existing linearization strategies. Second, the continuous relaxation of this linearization theoretically dominates those of other linearizations found in the literature. Moreover, in the types of problems we have solved thus far, the proposed linearization has demonstrated a significant computational advantage.

This paper is organized as follows. Section 1 presents and justifies the proposed linearization scheme. Section 2 compares this linearization strategy with those of Glover (1975), Glover and Woolsey (1973, 1974), Petersen (1971), Watters (1967) and Zangwill (1965) demonstrating that these alternate schemes are all obtainable through appropriate surrogates of constraints of the proposed linearized problem. Hence, not only does the new linearization provide a tighter linear programming relaxation, but it also provides a unifying framework which ties together all these alternate schemes. An example is then given to demonstrate that the proposed linearization strictly dominates the other available linearizations with respect to the continuous relaxation. The final section extends these ideas to some other, more general, types

Subject classification: Programming; linear reformulations of nonlinear integer programs; Programming; integer, nonlinear, mixed integer programming problems.

Operations Research
Vol. 38, No. 2, March–April 1990

217

0030-354X/90/3802-0217\$01.25
© 1990 Operations Research Society of America

Introduction
Reformulation-Linearization Technique
RLT literature review
The first paper
Mixed products
Relaxation of
▷ bilinear terms
Mixed products again
Relaxation hierarchy
Etc.
Reduced RLT
RRLT literature review
New developments
Why bother?
Thank you

Sherali & Alameddine, JOGO 1992:

$\mathbf{x} \in \mathbb{R}^n$ with bilinear products

LP relaxation

under special condition, LP is reformulation

A New Reformulation-Linearization Technique for Bilinear Programming Problems*

HANIF D. SHERALI and AMINE ALAMEDDINE

Department of Industrial and Systems Engineering, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061-0118, U.S.A.

(Received: 14 December 1990; accepted: 7 November 1991)

Abstract. This paper is concerned with the development of an algorithm for general bilinear programming problems. Such problems find numerous applications in economics and game theory, location theory, nonlinear multi-commodity network flows, dynamic assignment and production, and various risk management problems. The proposed approach develops a new Reformulation-Linearization Technique (RLT) for this problem, and imbeds it within a provably convergent branch-and-bound algorithm. The method first reformulates the problem by constructing a set of nonnegative variable factors using the problem constraints, and suitably multiplies combinations of these factors with the original problem constraints to generate additional valid nonlinear constraints. The resulting nonlinear program is subsequently linearized by defining a new set of variables, one for each nonlinear term. This "RLT" process yields a linear programming problem whose optimal value provides a tight lower bound on the optimal value to the bilinear programming problem. Various implementation schemes and constraint generation procedures are investigated for the purpose of further tightening the resulting linearization. The lower bound thus produced theoretically dominates, and practically is far tighter, than that obtained by using convex envelopes over hyper-rectangles. In fact, for some special cases, this process is shown to yield an exact linear programming representation. For the associated branch-and-bound algorithm, various admissible branching schemes are discussed, including one in which branching is performed by partitioning the intervals for only one set of variables x or y , whichever are fewer in number. Computational experience is provided to demonstrate the viability of the algorithm. For a large number of test problems from the literature, the initial bounding linear program itself solves the underlying bilinear programming problem.

Key words. Bilinear programming, nonconvex programming, global optimization, branch-and-bound, reformulation-linearization technique.

1. Introduction

This paper is concerned with the solution of bilinear programming problems of the form

$$BLP \text{ Minimize } \Phi(x, y) = c'x + d'y + x'Gy \quad (1.1)$$

$$\text{subject to } (x, y) \in Z = \left\{ (x, y): \begin{array}{l} A_1x + D_1y \leq b_1 \\ A_2x + D_2y = b_2 \end{array} \right\} \quad (1.2a)$$

$$(x, y) \in \Omega \equiv \{(x, y): 0 \leq x \leq u < \infty, 0 \leq y \leq U < \infty\} \quad (1.3)$$

*This paper was presented at the II. IIASA Workshop on Global Optimization, Sopron (Hungary), December 9–14, 1990.

Journal of Global Optimization 2: 379–410, 1992.

© 1992 Kluwer Academic Publishers. Printed in the Netherlands.

- Introduction
- Reformulation-
Linearization
Technique
- RLT literature
review
- The first paper
- Mixed products
- Relaxation of
bilinear terms
 - ▷ Mixed products
- Relaxation hierarchy
- Etc.
- Reduced RLT
- RRLT literature
review
- New developments
- Why bother?
- Thank you

*Adams & Sherali,
Math. Prog. 1993:*

$\mathbf{x} \in \mathbb{R}^{n_1} \times \{0, 1\}^{n_2}$ specialized to
bilinear products involving one
continuous and one binary
variable

MILP reformulation

then continuous relaxation

Mathematical Programming 59 (1993) 279-305
North-Holland

279

Mixed-integer bilinear programming problems

Warren P. Adams

Department of Mathematical Sciences, Clemson University, Clemson, SC, USA

Hanif D. Sherali

Department of Industrial and Systems Engineering, Virginia Polytechnic Institute and State University, Blacksburg, VA, USA

Received 22 June 1987

Revised manuscript received 5 March 1992

This paper addresses a class of problems called mixed-integer bilinear programming problems. These problems are identical to the well known bilinear programming problems with the exception that one set of variables is restricted to be binary valued, and they arise in various production, location-allocation, and distribution application contexts. We first identify some special cases of this problem which are relatively more readily solvable, even though their continuous relaxations are still nonconvex. For the more general case, we employ a linearization technique and design a composite Lagrangian relaxation-implicit enumeration-cutting plane algorithm. Extensive computational experience is provided to test the efficacy of various algorithmic strategies and the effects of problem data on the computational effort of the proposed algorithm.

Key words: Bilinear program, linearization, cutting planes, Lagrangian relaxation.

1. Introduction

In this paper we study a class of mathematical programs referred to as *mixed-integer bilinear programming problems* (MIBLP). These problems may be stated as follows:

MIBLP: minimize $\{c^T x + d^T y + x^T C y : x \in X, y \in Y, y \text{ binary}\}$ (1.1)

where X and Y are nonempty, bounded polyhedral sets given respectively as

$$X = \left\{ x \in \mathbb{R}^m : \sum_{i=1}^m a_{ki} x_i = b_k \text{ for } k = 1, \dots, K, x \geq 0 \right\}$$

and

$$Y = Y_1 \cap Y_2 \cap Y_3,$$

with

$$Y_1 = \left\{ y \in \mathbb{R}^s : \sum_{j=1}^s G_{lj} y_j \geq g_l \text{ for } l = 1, \dots, L \right\},$$

$$Y_2 = \left\{ y \in \mathbb{R}^n : \sum_{j=1}^n H_{pj} y_j = h_p \text{ for } p = 1, \dots, P \right\},$$

Correspondence to: Prof. Warren P. Adams, Department of Mathematical Sciences, Clemson University, Martin Hall, Clemson, SC 29634-1907, USA.

Introduction
Reformulation-Linearization Technique
RLT literature review
The first paper
Mixed products
Relaxation of bilinear terms
Mixed products again
Relaxation hierarchy
Etc.
Reduced RLT
RRLT literature review
New developments
Why bother?
Thank you

Sherali & Adams, DAM 1994:

$$\mathbf{x} \in \mathbb{R}^{n_1} \times \{0, 1\}^{n_2}$$

Convex hull of MILP feasible region obtained through hierarchy of RLT constraints

Level d RLT: let

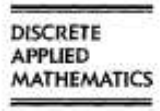
$$J_1, J_2 \subseteq \{1, \dots, n_2\} \text{ with } |J_1 \cup J_2| = d \text{ and}$$

$$F_d(J_1, J_2) = \prod_{j \in J_1} x_j \prod_{j \in J_2} (1 - x_j);$$

multiply RLT- $(d - 1)$ constraints by all $F_d(J_1, J_2)$ the linearize to obtain RLT- d



Discrete Applied Mathematics 52 (1994) 83–106



A hierarchy of relaxations and convex hull characterizations for mixed-integer zero-one programming problems

Hamid D. Sherali^{a,*}, Warren P. Adams^b

^aDepartment of Industrial and Systems Engineering, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061-0118, USA

^bDepartment of Math Sciences, Clemson University, Clemson, SC 29634-1007, USA

(Received 21 October 1991; revised 18 August 1992)

Abstract

This paper is concerned with the generation of tight equivalent representations for mixed-integer zero-one programming problems. For the linear case, we propose a technique which first converts the problem into a nonlinear, polynomial mixed-integer zero-one problem by multiplying the constraints with some suitable d -degree polynomial factors involving the n binary variables, for any given $d \in \{0, \dots, n\}$, and subsequently linearizes the resulting problem through appropriate variable transformations. As d varies from zero to n , we obtain a hierarchy of relaxations spanning from the ordinary linear programming relaxation to the convex hull of feasible solutions. The facets of the convex hull of feasible solutions in terms of the original problem variables are available through a standard projection operation. We also suggest an alternate scheme for applying this technique which gives a similar hierarchy of relaxations, but involving fewer "complicating" constraints. Techniques for tightening intermediate level relaxations, and insights and interpretations within a disjunctive programming framework are also presented. The methodology readily extends to multilinear mixed-integer zero-one polynomial programming problems in which the continuous variables appear linearly in the problem.

Key words: Mixed-integer zero-one problems; Tight relaxations; Convex hull representations; Facetial inequalities; Disjunctive programming

1. Introduction

Recently, Sherali and Adams [7] have proposed a new technique for generating a hierarchy of relaxations for linear and polynomial zero-one programming problems, spanning the spectrum from the continuous relaxation to the convex hull representation. The present paper provides an extension of this approach to the

* Corresponding author.

Introduction

Reformulation-
Linearization
Technique

RLT literature
review

The first paper
Mixed products
Relaxation of
bilinear terms
Mixed products
again

Relaxation hierarchy
▷ Etc.

Reduced RLT

RRLT literature
review

New developments

Why bother?

Thank you

... and many more!

extensions (polynomial and signomial programming, SDP, convex MINLP, disjunctive cuts) and applications

Introduction

Reformulation-
Linearization
Technique

RLT literature
review

▷ Reduced RLT

The quadratic case
RRLT-1 constraints
Main result
Geometric
interpretation

RRLT literature
review

New developments

Why bother?

Thank you

Reduced RLT

Introduction

Reformulation-
Linearization
Technique

RLT literature
review

Reduced RLT
▷ The quadratic
case

RRLT-1 constraints

Main result
Geometric
interpretation

RRLT literature
review

New developments

Why bother?

Thank you

Consider mixed-integer QCQP subject to linear equality constraints $A\mathbf{x} = \mathbf{b}$ (A has full rank)

$$\left. \begin{array}{ll} \min_{\mathbf{x} \in \mathbb{R}^{n_1} \times \mathbb{Z}^{n_2}} & \mathbf{c}_0\mathbf{x} + \mathbf{x}Q_0\mathbf{x} \\ \forall 1 \leq i \leq q & \mathbf{c}_i\mathbf{x} + \mathbf{x}Q_i\mathbf{x} \leq 0 \\ & A\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \in \mathcal{X} \cap [\mathbf{x}^L, \mathbf{x}^U]. \end{array} \right\}$$

\mathcal{X} is a polyhedron

Introduction

Reformulation-
Linearization
Technique

RLT literature
review

Reduced RLT

The quadratic case
RRLT-1
▷ constraints

Main result
Geometric
interpretation

RRLT literature
review

New developments

Why bother?

Thank you

- Generate RLT constraints from $A\mathbf{x} = \mathbf{b}$

$$A\mathbf{x} = \mathbf{b} \Rightarrow$$

$$\forall \ell \leq n \quad x_\ell A\mathbf{x} = x_\ell \mathbf{b} \Rightarrow$$

$$\forall \ell \leq n \quad A\mathbf{w}_\ell = x_\ell \mathbf{b}$$

- Consider homogeneous system $\forall \ell \leq n \ (A\mathbf{w}_\ell = \mathbf{0})$ and a set N of nonbasic variable index pairs (j, ℓ) ; let:

$$C = \{(\mathbf{x}, \mathbf{w}) \mid A\mathbf{x} = \mathbf{b} \wedge \forall j, \ell \leq n \ (w_{j\ell} = x_j x_\ell)\}$$

$$R_N = \{(\mathbf{x}, \mathbf{w}) \mid A\mathbf{x} = \mathbf{b} \wedge \\ \forall \ell \leq n \ (A\mathbf{w}_\ell = \mathbf{b} x_\ell) \wedge \forall (j, \ell) \in N \ (w_{j\ell} = x_j x_\ell)\}$$

- Introduction
- Reformulation-Linearization Technique
- RRLT literature review
- Reduced RLT
- The quadratic case
- RRLT-1 constraints
- ▷ Main result
- Geometric interpretation
- RRLT literature review
- New developments
- Why bother?
- Thank you

Thm.

Let $[n] = \{1, \dots, n\}$; $\exists N \subseteq [n] \times [n]$ $C = R_N$

Proof

$$\begin{aligned}
 & \text{(RRLT system)} \quad \forall \ell \leq n \quad A\mathbf{w}_\ell - x_\ell \mathbf{b} = 0 \Rightarrow \\
 & \text{(replace } b \text{ by } Ax) \quad \forall \ell \leq n \quad A\mathbf{w}_\ell - x_\ell A\mathbf{x} = 0 \Rightarrow \\
 & (\mathbf{z} = \mathbf{w}_\ell - x_\ell \mathbf{x} \text{ are vars. of hom. sys.}) \quad \forall \ell \leq n \quad A(\mathbf{w}_\ell - x_\ell \mathbf{x}) = 0. (1)
 \end{aligned}$$

$$\left[\begin{array}{l} (1) \text{ is homogeneous} \\ N \subseteq [n] \times [n]: \text{ nonbasic of (1)} \\ \forall (j, \ell) \in N \quad w_{j\ell} = x_j x_\ell \end{array} \right] \Rightarrow \left\{ \begin{array}{l} \text{get square nonsing. subsyst.} \\ A'\mathbf{z} = 0 \text{ of (1)} \\ \text{corresponding to basic cols.} \\ B = [n] \times [n] \setminus N \end{array} \right.$$

by basic linear algebra, $A'\mathbf{z} = 0$ implies $\forall (j, \ell) \in B \quad (w_{j\ell} = x_j x_\ell)$.

Cor.

RRLT constraints \Rightarrow exact ref. with fewer quadratic terms

Proof

Only need quadratic terms indexed by N , RRLT implies those in B

Geometric interpretation

Introduction

Reformulation-
Linearization
Technique

RLT literature
review

Reduced RLT

The quadratic case

RRLT-1 constraints

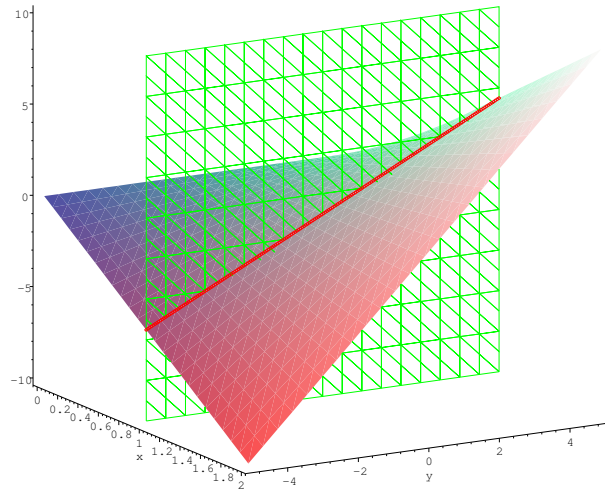
Main result
Geometric
▷ interpretation

RRLT literature
review

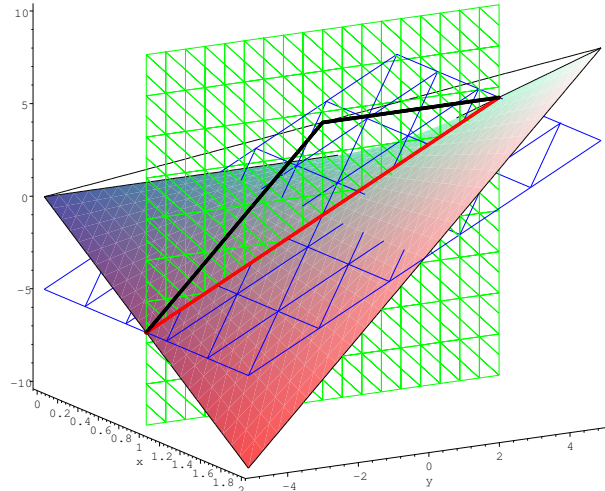
New developments

Why bother?

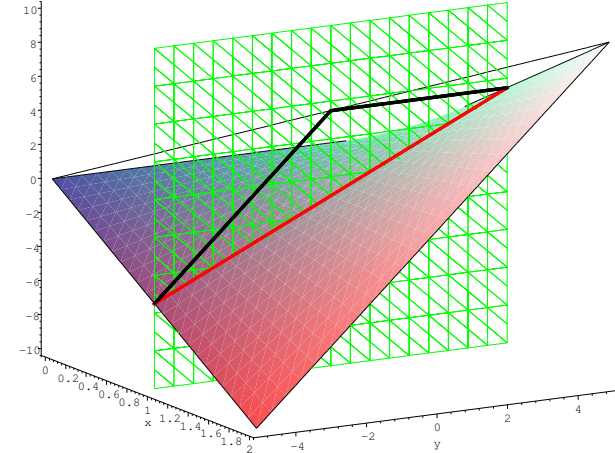
Thank you



$$C = \{(x, y, w) \mid x = 1 \wedge w = xy\}$$



$$R = \{(x, y, w) \mid x = 1 \wedge w = y\}$$



McCormick's rel. of $w = xy$ restricted to $x = 1$

Notice $C = R$ (straight red segment)

Equation $w = y$ can be obtained via RRLT: multiply equation $x = 1$ by y and linearize via w

Introduction

Reformulation-
Linearization
Technique

RLT literature
review

Reduced RLT

▷ RRLT literature
review

Announcement

General theory

Automatic
reformulation

Application to
quantum chemistry

New developments

Why bother?

Thank you

RRLT literature review

Introduction
Reformulation- Linearization Technique
RLT literature review
Reduced RLT
RRLT literature review
▷ Announcement
General theory
Automatic reformulation
Application to quantum chemistry
New developments
Why bother?
Thank you

L. , ITOR 2004:

RRLT constraints are linearly independent

Preliminary results on pooling problems



Inf. Trans. in Op. Res. 11 (2004) 33-41

INTERNATIONAL
TRANSACTIONS
IN OPERATIONAL
RESEARCH

Reduction constraints for the global optimization of NLPs

Leo Liberti

*DEI, Politecnico di Milano, Piazza L. de Vinci 32, I-20133, Milan, Italy
E-mail: liberti@elet.polimi.it*

Received 3 July 2002; received in revised form 6 December 2002; accepted 13 January 2003

Abstract

Convergence of branch-and-bound algorithms for the solution of NLPs is obtained by finding ever-nearer lower and upper bounds to the objective function. The lower bound is calculated by constructing a convex relaxation of the NLP. Reduction constraints are new linear problem constraints which are (a) linearly independent from the existing constraints; (b) redundant with reference to the original NLP formulation; (c) not redundant with reference to its convex relaxation. Thus, they can be successfully employed to reduce the feasible region of the convex relaxation without cutting the feasible region of the original NLP.

Keywords: global optimization; valid cut; NLP; branch-and-bound

1. Introduction

Global nonlinear optimization has witnessed a remarkable theoretical development in the last decade (Adjiman, Schweiger and Floudas, 1998; Floudas, 2001; Pardalos, Romeijn, 2000). A lot of new algorithms have been proposed, either geared towards a specific problem or class of problems (Hirafuji and Hagan, 2000; Hagglof, Lindberg and Svensson, 1995), or more general (Adjiman et al., 1998; Adjiman, Androulakis and Floudas, 1998; Kesavan and Barton, 2000; Ryoo and Sahinidis 1995; Smith and Pantelides, 1999; Vaidyanathan and El-Halwagi, 1996). Software implementations of these algorithms, however, are scarce, not easily available, and more importantly, not really ready for practical use. One could draw a parallel with the development of LP solvers, where the straight implementation of an algorithm is usually not enough to give birth to a good piece of code; all sorts of 'implementation tricks' are necessary to this end. We feel that at the present state, global optimization solvers for NLPs are in their basic form, with the algorithm in place but devoid of other speeding-up devices which are crucial for practical usability.

In this article we describe the theory and implementation of one such speeding-up device, named 'method of reduction constraints', to be used in a branch-and-bound solution framework.

© 2004 International Federation of Operational Research Societies.
Published by Blackwell Publishing Ltd.

Introduction
Reformulation- Linearization Technique
RLT literature review
Reduced RLT
RRLT literature review
Announcement ▷ General theory
Automatic reformulation Application to quantum chemistry
New developments
Why bother?
Thank you

L. , JOGO 2005:

General theory of RRLT con-
straints

Reformulation proofs

Journal of Global Optimization (2005) 33: 157–196
DOI 10.1007/s10898-004-0864-2

© Springer 2005

Linearity Embedded in Nonconvex Programs

LEO LIBERTI

DEI, Politecnico di Milano, Piazza L. da Vinci 32, I-20133 Milano, Italy (e-mail: liberti@elet.polimi.it)

(Received 4 May 2003; accepted in revised form 19 May 2004)

Abstract. Nonconvex programs involving bilinear terms and linear equality constraints often appear more nonlinear than they really are. By using an automatic symbolic reformulation we can substitute some of the bilinear terms with linear constraints. This has a dramatically improving effect on the tightness of any convex relaxation of the problem, which makes deterministic global optimization algorithms like spatial Branch-and-Bound much more efficient when applied to the problem.

Key words: Bilinear, Convex relaxation, Global optimization, MINLP, Reduction constraint, Reformulation, RLT

1. Introduction

This paper is concerned with programming problems of the form:

$$\left. \begin{aligned} \min_x \quad & x^T Q x + c^T x + f(x), \\ & Ax = b, \\ & g(x) = 0, \\ & h(x) \leq 0, \\ & x \in X, \\ & x^L \leq x \leq x^U, \end{aligned} \right\} \quad (1)$$

where $Q = (q_{ij})$ is an $n \times n$ matrix, $x, c, x^L, x^U \in \mathbb{R}^n$, $A = (a_{ij})$ is an $m \times n$ matrix having rank m , $b \in \mathbb{R}^m$, $f: \mathbb{R}^n \rightarrow \mathbb{R}$, $g: \mathbb{R}^n \rightarrow \mathbb{R}^{m_1}$, $h: \mathbb{R}^n \rightarrow \mathbb{R}^{m_2}$ and X is an arbitrary subset of \mathbb{R}^n (which might express integrality constraints on the decision variables, for example). Notice f, g, h are completely arbitrary functions. Notice also that we assume $m \leq n$, otherwise the feasible region may be empty. Such a formulation is very general and encompasses many instances of problems arising from mathematical modelling of real life processes.

Because the theory developed herein will enable us to substitute some of the bilinear terms with linear constraints, we can restrict our attention to a more standard formulation (2) of the bilinear problem. This does not mean that the methods described only applies to problems in formulation (2),

Introduction
Reformulation-Linearization Technique
RLT literature review
Reduced RLT
RRLT literature review
Announcement
General theory
Automatic
▷ reformulation
Application to quantum chemistry
New developments
Why bother?
Thank you

L. & Pantelides, JOGO 2006:

Graph-based automatic reformulation algorithm

Full computational results on pooling and blending problems

Journal of Global Optimization (2006) 36: 161–189
DOI 10.1007/s10898-006-9005-4

© Springer 2006

An Exact Reformulation Algorithm for Large Nonconvex NLPs Involving Bilinear Terms

LEO LIBERTI¹ and CONSTANTINOS C. PANTELIDES²

¹CNRS LIX, École Polytechnique, F-91128 Palaiseau, France

(e-mail: liberti@lix.polytechnique.fr)

²Centre for Process Systems Engineering, Department of Chemical Engineering and Chemical Technology, Imperial College London, SW7 2BY London, UK

(e-mail: c.pantelides@imperial.ac.uk)

(Received 2 February 2004; accepted in revised form 29 January 2006; Published online 22 April 2006)

Abstract. Many nonconvex nonlinear programming (NLP) problems of practical interest involve bilinear terms and linear constraints, as well as, potentially, other convex and non-convex terms and constraints. In such cases, it may be possible to augment the formulation with additional linear constraints (a subset of Reformulation-Linearization Technique constraints) which do not affect the feasible region of the original NLP but tighten that of its convex relaxation to the extent that some bilinear terms may be dropped from the problem formulation. We present an efficient graph-theoretical algorithm for effecting such exact reformulations of large, sparse NLPs. The global solution of the reformulated problem using spatial Branch-and-Bound algorithms is usually significantly faster than that of the original NLP. We illustrate this point by applying our algorithm to a set of pooling and blending global optimization problems.

Key words: Bilinear, Convex relaxation, Global optimization, NLP, Reformulation-linearization technique, RRLT constraints

1. Introduction

We consider the solution of nonlinear programs (NLPs) of the following standard form.

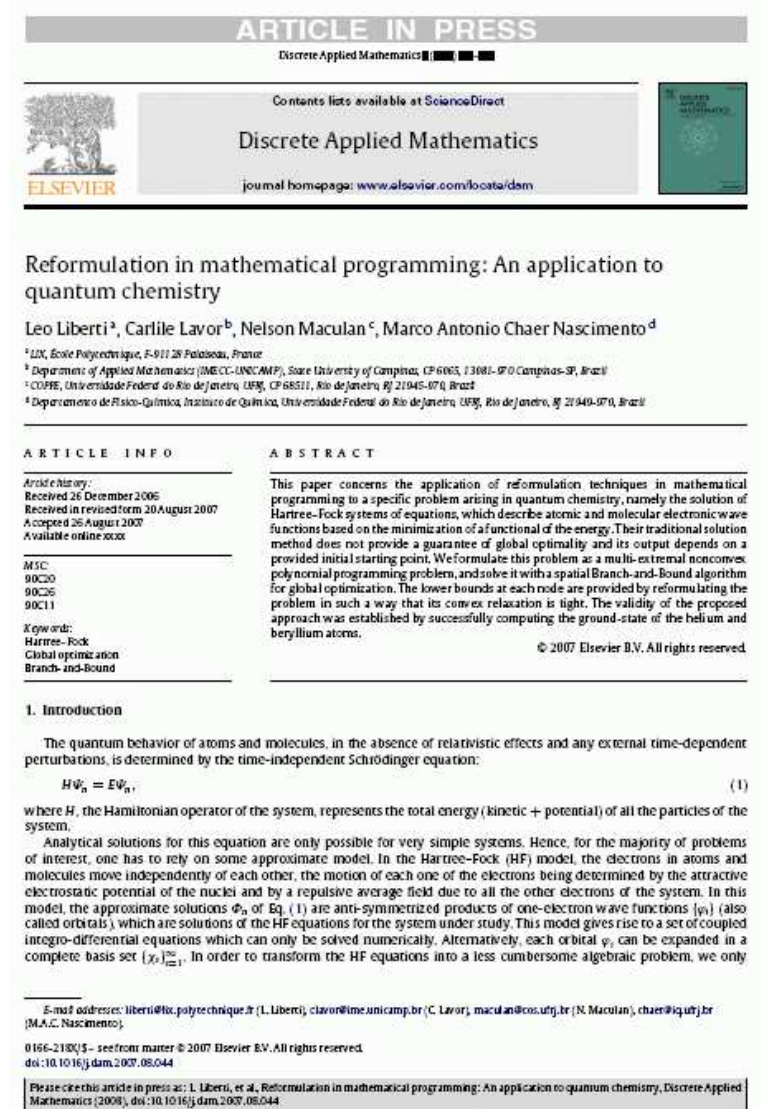
$$\begin{aligned}
 [P]: \quad & \min_{z} \quad z_t & (1) \\
 & Az = b & (2) \\
 & z_t = z_j z_k \quad \forall i, j, k \in B & (3) \\
 & z_t = \frac{z_j z_l}{z_k} \quad \forall i, j, k \in F & (4) \\
 & z_t = f_i(z_j) \quad \forall i, j \in N & (5) \\
 & z^L \leq z \leq z^U & (6)
 \end{aligned}$$

where $z = (z_1, \dots, z_p) \in \mathbb{R}^p$ are the problem variables, l is an index in the set $\{1, \dots, p\}$, $A = (a_{ij})$ is an $M \times p$ matrix of rank M , $b \in \mathbb{R}^M$, B, F are

Introduction
Reformulation- Linearization Technique
RLT literature review
Reduced RLT
RRLT literature review
Announcement
General theory
Automatic reformulation
Application to quantum ▷ chemistry
New developments
Why bother?
Thank you

L. , Lavor, Maculan, Chaer Nascimento, DAM 2009:

Application of an RRLT-2 subset
to solving Hartree-Fock systems



Introduction

Reformulation-
Linearization
Technique

RLT literature
review

Reduced RLT

RRLT literature
review

New
▷ developments

Optimal RRLT

RRLT for polynomial
programming 1/3

RRLT for polynomial
programming 2/3

RRLT for polynomial
programming 3/3

Sparsity 1/4

Sparsity 2/4

Sparsity 3/4

Sparsity 4/4

Why bother?

Thank you

New developments

Introduction

Reformulation-
Linearization
Technique

RLT literature
review

Reduced RLT

RRLT literature
review

New developments

▷ Optimal RRLT

RRLT for polynomial
programming 1/3

RRLT for polynomial
programming 2/3

RRLT for polynomial
programming 3/3

Sparsity 1/4

Sparsity 2/4

Sparsity 3/4

Sparsity 4/4

Why bother?

Thank you

- Feasible region of QCQP: use R_N instead of C

Introduction

Reformulation-
Linearization
Technique

RLT literature
review

Reduced RLT

RRLT literature
review

New developments

▷ Optimal RRLT

RRLT for polynomial
programming 1/3

RRLT for polynomial
programming 2/3

RRLT for polynomial
programming 3/3

Sparsity 1/4

Sparsity 2/4

Sparsity 3/4

Sparsity 4/4

Why bother?

Thank you

- Feasible region of QCQP: use R_N instead of C
- R_N relies on quadratic constraints $\forall (j, \ell) \in N$ ($w_{j\ell} = x_j x_\ell$)

Introduction

Reformulation-
Linearization
Technique

RRLT literature
review

Reduced RLT

RRLT literature
review

New developments

▷ Optimal RRLT

RRLT for polynomial
programming 1/3

RRLT for polynomial
programming 2/3

RRLT for polynomial
programming 3/3

Sparsity 1/4

Sparsity 2/4

Sparsity 3/4

Sparsity 4/4

Why bother?

Thank you

- Feasible region of QCQP: use R_N instead of C
- R_N relies on quadratic constraints $\forall (j, \ell) \in N$ ($w_{j\ell} = x_j x_\ell$)
- **Degree of freedom:** choice of basic/nonbasic partition B, N of $[n] \times [n]$

Introduction

Reformulation- Linearization Technique

RRLT literature review

Reduced RLT

RRLT literature review

New developments

▷ Optimal RRLT

RRLT for polynomial programming 1/3

RRLT for polynomial programming 2/3

RRLT for polynomial programming 3/3

Sparsity 1/4

Sparsity 2/4

Sparsity 3/4

Sparsity 4/4

Why bother?

Thank you

- Feasible region of QCQP: use R_N instead of C
- R_N relies on quadratic constraints $\forall (j, \ell) \in N$ ($w_{j\ell} = x_j x_\ell$)
- **Degree of freedom:** choice of basic/nonbasic partition B, N of $[n] \times [n]$
- $(j, \ell) \leftrightarrow$ volume $V_{j\ell}$ of conv. env. of $x_j x_\ell$

Introduction

Reformulation-
Linearization
Technique

RRLT literature
review

Reduced RLT

RRLT literature
review

New developments

▷ Optimal RRLT

RRLT for polynomial
programming 1/3

RRLT for polynomial
programming 2/3

RRLT for polynomial
programming 3/3

Sparsity 1/4

Sparsity 2/4

Sparsity 3/4

Sparsity 4/4

Why bother?

Thank you

- Feasible region of QCQP: use R_N instead of C
- R_N relies on quadratic constraints $\forall (j, \ell) \in N$ ($w_{j\ell} = x_j x_\ell$)
- **Degree of freedom:** choice of basic/nonbasic partition B, N of $[n] \times [n]$
- $(j, \ell) \leftrightarrow$ volume $V_{j\ell}$ of conv. env. of $x_j x_\ell$
- *Convexity gap:* $\mathcal{V}(N) = \sum_{(j, \ell) \in N} V_{j\ell}$

Introduction

Reformulation- Linearization Technique

RRLT literature review

Reduced RLT

RRLT literature review

New developments

▷ Optimal RRLT

RRLT for polynomial programming 1/3

RRLT for polynomial programming 2/3

RRLT for polynomial programming 3/3

Sparsity 1/4

Sparsity 2/4

Sparsity 3/4

Sparsity 4/4

Why bother?

Thank you

- Feasible region of QCQP: use R_N instead of C
- R_N relies on quadratic constraints $\forall (j, \ell) \in N$ ($w_{j\ell} = x_j x_\ell$)
- **Degree of freedom:** choice of basic/nonbasic partition B, N of $[n] \times [n]$
- $(j, \ell) \leftrightarrow$ volume $V_{j\ell}$ of conv. env. of $x_j x_\ell$
- *Convexity gap:* $\mathcal{V}(N) = \sum_{(j, \ell) \in N} V_{j\ell}$
- Let $N^* = \arg \min_N \mathcal{V}(N)$

Introduction

Reformulation-
Linearization
Technique

RRLT literature
review

Reduced RLT

RRLT literature
review

New developments

▷ Optimal RRLT

RRLT for polynomial
programming 1/3

RRLT for polynomial
programming 2/3

RRLT for polynomial
programming 3/3

Sparsity 1/4

Sparsity 2/4

Sparsity 3/4

Sparsity 4/4

Why bother?

Thank you

- Feasible region of QCQP: use R_N instead of C
- R_N relies on quadratic constraints $\forall (j, \ell) \in N$ ($w_{j\ell} = x_j x_\ell$)
- **Degree of freedom:** choice of basic/nonbasic partition B, N of $[n] \times [n]$
- $(j, \ell) \leftrightarrow$ volume $V_{j\ell}$ of conv. env. of $x_j x_\ell$
- *Convexity gap:* $\mathcal{V}(N) = \sum_{(j, \ell) \in N} V_{j\ell}$
- Let $N^* = \arg \min_N \mathcal{V}(N)$

Smaller gap \Rightarrow tight bound more likely

Introduction

Reformulation- Linearization Technique

RRLT literature review

Reduced RLT

RRLT literature review

New developments

▷ Optimal RRLT

RRLT for polynomial programming 1/3

RRLT for polynomial programming 2/3

RRLT for polynomial programming 3/3

Sparsity 1/4

Sparsity 2/4

Sparsity 3/4

Sparsity 4/4

Why bother?

Thank you

- Feasible region of QCQP: use R_N instead of C
- R_N relies on quadratic constraints $\forall (j, \ell) \in N$ ($w_{j\ell} = x_j x_\ell$)
- **Degree of freedom:** choice of basic/nonbasic partition B, N of $[n] \times [n]$
- $(j, \ell) \leftrightarrow$ volume $V_{j\ell}$ of conv. env. of $x_j x_\ell$
- *Convexity gap:* $\mathcal{V}(N) = \sum_{(j, \ell) \in N} V_{j\ell}$
- Let $N^* = \arg \min_N \mathcal{V}(N)$

Smaller gap \Rightarrow tight bound more likely

- B, N partition $[n] \times [n] \Rightarrow N^* = [n] \times [n] \setminus \arg \max_B \mathcal{V}(B)$

- Introduction
- Reformulation-Linearization Technique
- RRLT literature review
- Reduced RRLT
- RRLT literature review
- New developments
 - ▷ Optimal RRLT
 - RRLT for polynomial programming 1/3
 - RRLT for polynomial programming 2/3
 - RRLT for polynomial programming 3/3
 - Sparsity 1/4
 - Sparsity 2/4
 - Sparsity 3/4
 - Sparsity 4/4
- Why bother?
- Thank you

- Feasible region of QCQP: use R_N instead of C
- R_N relies on quadratic constraints $\forall (j, \ell) \in N$ ($w_{j\ell} = x_j x_\ell$)
- **Degree of freedom:** choice of basic/nonbasic partition B, N of $[n] \times [n]$
- $(j, \ell) \leftrightarrow$ volume $V_{j\ell}$ of conv. env. of $x_j x_\ell$
- *Convexity gap:* $\mathcal{V}(N) = \sum_{(j,\ell) \in N} V_{j\ell}$
- Let $N^* = \arg \min_N \mathcal{V}(N)$

Smaller gap \Rightarrow tight bound more likely

- B, N partition $[n] \times [n] \Rightarrow N^* = [n] \times [n] \setminus \arg \max_B \mathcal{V}(B)$
- Reduces to finding a max-weight basis of a linear system

Introduction

Reformulation- Linearization Technique

RRLT literature review

Reduced RLT

RRLT literature review

New developments

▷ Optimal RRLT

RRLT for polynomial programming 1/3

RRLT for polynomial programming 2/3

RRLT for polynomial programming 3/3

Sparsity 1/4

Sparsity 2/4

Sparsity 3/4

Sparsity 4/4

Why bother?

Thank you

- Feasible region of QCQP: use R_N instead of C
- R_N relies on quadratic constraints $\forall (j, \ell) \in N$ ($w_{j\ell} = x_j x_\ell$)
- **Degree of freedom:** choice of basic/nonbasic partition B, N of $[n] \times [n]$
- $(j, \ell) \leftrightarrow$ volume $V_{j\ell}$ of conv. env. of $x_j x_\ell$
- *Convexity gap:* $\mathcal{V}(N) = \sum_{(j, \ell) \in N} V_{j\ell}$
- Let $N^* = \arg \min_N \mathcal{V}(N)$

Smaller gap \Rightarrow tight bound more likely

- B, N partition $[n] \times [n] \Rightarrow N^* = [n] \times [n] \setminus \arg \max_B \mathcal{V}(B)$
- Reduces to finding a max-weight basis of a linear system
- Greedy algorithm solves problem optimally

Introduction

Reformulation-
Linearization
Technique

RLT literature
review

Reduced RLT

RRLT literature
review

New developments

Optimal RRLT

RRLT for
polynomial
▷ programming 1/3

RRLT for polynomial
programming 2/3

RRLT for polynomial
programming 3/3

Sparsity 1/4

Sparsity 2/4

Sparsity 3/4

Sparsity 4/4

Why bother?

Thank you

Consider general polynomial programming MINLP

$$\left. \begin{array}{ll} \min_{x \in \mathbb{R}^{n_1} \times \mathbb{Z}^{n_2}} & g_0(x) \\ \forall i \leq q & g_i(x) \leq 0 \\ & A\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \in \mathcal{X} \cap [\mathbf{x}^L, \mathbf{x}^U] \end{array} \right\}$$

where $g_i \in \mathbb{Q}[\mathbf{x}]$ for all $i \in \{0, \dots, q\}$

- **Reformulation:** for all $J \subseteq [n - 1]$ multiply $A\mathbf{x} = \mathbf{b}$ by

$$\prod_{j \in J} x_j$$

- Introduction
- Reformulation-
Linearization
Technique
- RLT literature
review
- Reduced RLT
- RRLT literature
review
- New developments
- Optimal RRLT
- RRLT for polynomial
programming 1/3
 - RRLT for
polynomial
▷ programming 2/3
- RRLT for polynomial
programming 3/3
- Sparsity 1/4
- Sparsity 2/4
- Sparsity 3/4
- Sparsity 4/4
- Why bother?
- Thank you

Introduction

Reformulation-
Linearization
Technique

RRLT literature
review

Reduced RRLT

RRLT literature
review

New developments

Optimal RRLT

RRLT for polynomial
programming 1/3

RRLT for
polynomial
▷ programming 2/3

RRLT for polynomial
programming 3/3

Sparsity 1/4

Sparsity 2/4

Sparsity 3/4

Sparsity 4/4

Why bother?

Thank you

- **Reformulation:** for all $J \subseteq [n - 1]$ multiply $A\mathbf{x} = \mathbf{b}$ by $\prod_{j \in J} x_j$
- **Linearization:** replace each term $\prod_{j \in J} x_j$ by the added variable w_J (for all $J \subseteq [n]$)

- Introduction
- Reformulation-Linearization Technique
- RRLT literature review
- Reduced RRLT
- RRLT literature review
- New developments
- Optimal RRLT
- RRLT for polynomial programming 1/3
 - RRLT for polynomial programming 2/3
- RRLT for polynomial programming 3/3
- Sparsity 1/4
- Sparsity 2/4
- Sparsity 3/4
- Sparsity 4/4
- Why bother?
- Thank you

- **Reformulation:** for all $J \subseteq [n - 1]$ multiply $A\mathbf{x} = \mathbf{b}$ by $\prod_{j \in J} x_j$
- **Linearization:** replace each term $\prod_{j \in J} x_j$ by the added variable w_J (for all $J \subseteq [n]$)
- Adjoin defining constraints $w_J = \prod_{j \in J} x_j$

- Introduction
- Reformulation-Linearization Technique
- RRLT literature review
- Reduced RRLT
- RRLT literature review
- New developments
- Optimal RRLT
- RRLT for polynomial programming 1/3
- RRLT for polynomial programming 2/3
- RRLT for polynomial programming 3/3
- Sparsity 1/4
- Sparsity 2/4
- Sparsity 3/4
- Sparsity 4/4
- Why bother?
- Thank you

- **Reformulation:** for all $J \subseteq [n - 1]$ multiply $A\mathbf{x} = \mathbf{b}$ by $\prod_{j \in J} x_j$
- **Linearization:** replace each term $\prod_{j \in J} x_j$ by the added variable w_J (for all $J \subseteq [n]$)
- Adjoin defining constraints $w_J = \prod_{j \in J} x_j$
- Define natural extensions of C, R_N :

$$C = \{(\mathbf{x}, \mathbf{w}) \mid A\mathbf{x} = \mathbf{b} \wedge \forall J \subseteq [n - 1] (w_J = \prod_{j \in J} x_j)\}$$

$$R_N = \{(\mathbf{x}, \mathbf{w}) \mid A\mathbf{x} = \mathbf{b} \wedge \forall J \subseteq [n - 1] (A\mathbf{w}_J = \mathbf{b}w_J) \wedge \forall J \in N (w_J = \prod_{j \in J} x_j)\}$$

where $\mathbf{w}_J = (w_{(J,1)}, \dots, w_{(J,n)})$

- Introduction
- Reformulation-Linearization Technique
- RRLT literature review
- Reduced RRLT
- RRLT literature review
- New developments
- Optimal RRLT
- RRLT for polynomial programming 1/3
- RRLT for polynomial programming 2/3
- RRLT for polynomial programming 3/3
- Sparsity 1/4
- Sparsity 2/4
- Sparsity 3/4
- Sparsity 4/4
- Why bother?
- Thank you

- **Reformulation:** for all $J \subseteq [n-1]$ multiply $A\mathbf{x} = \mathbf{b}$ by $\prod_{j \in J} x_j$
- **Linearization:** replace each term $\prod_{j \in J} x_j$ by the added variable w_J (for all $J \subseteq [n]$)
- Adjoin defining constraints $w_J = \prod_{j \in J} x_j$
- Define natural extensions of C, R_N :

$$C = \{(\mathbf{x}, \mathbf{w}) \mid A\mathbf{x} = \mathbf{b} \wedge \forall J \subseteq [n-1] (w_J = \prod_{j \in J} x_j)\}$$

$$R_N = \{(\mathbf{x}, \mathbf{w}) \mid A\mathbf{x} = \mathbf{b} \wedge \forall J \subseteq [n-1] (A\mathbf{w}_J = \mathbf{b}w_J) \wedge \forall J \in N (w_J = \prod_{j \in J} x_j)\}$$

where $\mathbf{w}_J = (w_{(J,1)}, \dots, w_{(J,n)})$

Main result $C = R_N$ still holds

- Introduction
- Reformulation-
Linearization
Technique
- RLT literature
review
- Reduced RLT
- RRLT literature
review
- New developments
- Optimal RRLT
- RRLT for polynomial
programming 1/3
- RRLT for polynomial
programming 2/3
 - RRLT for
polynomial
programming 3/3
- Sparsity 1/4
- Sparsity 2/4
- Sparsity 3/4
- Sparsity 4/4
- Why bother?
- Thank you

- Choice of optimal N extends from quadratic case, but:

- Introduction
- Reformulation-
Linearization
Technique
- RLT literature
review
- Reduced RLT
- RRLT literature
review
- New developments
- Optimal RRLT
- RRLT for polynomial
programming 1/3
- RRLT for polynomial
programming 2/3
 - RRLT for
polynomial
▷ programming 3/3
- Sparsity 1/4
- Sparsity 2/4
- Sparsity 3/4
- Sparsity 4/4
- Why bother?
- Thank you

- ☐ Choice of optimal N extends from quadratic case, but:
- ☐ **Added complication:**

- Introduction
- Reformulation-
Linearization
Technique
- RLT literature
review
- Reduced RLT
- RRLT literature
review
- New developments
- Optimal RRLT
- RRLT for polynomial
programming 1/3
- RRLT for polynomial
programming 2/3
 - RRLT for
polynomial
▷ programming 3/3
- Sparsity 1/4
- Sparsity 2/4
- Sparsity 3/4
- Sparsity 4/4
- Why bother?
- Thank you

- Choice of optimal N extends from quadratic case, but:
- **Added complication:**
 - V_{ij} and V_{ijk} are expressed in different units of measure

Introduction

Reformulation-
Linearization
Technique

RRLT literature
review

Reduced RRLT

RRLT literature
review

New developments

Optimal RRLT

RRLT for polynomial
programming 1/3

RRLT for polynomial
programming 2/3

RRLT for
polynomial
▷ programming 3/3

Sparsity 1/4

Sparsity 2/4

Sparsity 3/4

Sparsity 4/4

Why bother?

Thank you

- Choice of optimal N extends from quadratic case, but:
- **Added complication:**
 - V_{ij} and V_{ijk} are expressed in different units of measure
 - Summing up V_J for J 's of different sizes may not make much sense

- Introduction
- Reformulation-
Linearization
Technique
- RRLT literature
review
- Reduced RRLT
- RRLT literature
review
- New developments
- Optimal RRLT
- RRLT for polynomial
programming 1/3
- RRLT for polynomial
programming 2/3
- RRLT for
 polynomial
- ▷ programming 3/3
- Sparsity 1/4
- Sparsity 2/4
- Sparsity 3/4
- Sparsity 4/4
- Why bother?
- Thank you

- Choice of optimal N extends from quadratic case, but:
- **Added complication:**
 - V_{ij} and V_{ijk} are expressed in different units of measure
 - Summing up V_J for J 's of different sizes may not make much sense
- Multi-objective problem: $\forall p \in [n] \quad \max \sum_{|J|=p} V_J$

- Introduction
- Reformulation-Linearization Technique
- RRLT literature review
- Reduced RRLT
- RRLT literature review
- New developments
- Optimal RRLT
- RRLT for polynomial programming 1/3
- RRLT for polynomial programming 2/3
 - RRLT for polynomial programming 3/3
- Sparsity 1/4
- Sparsity 2/4
- Sparsity 3/4
- Sparsity 4/4
- Why bother?
- Thank you

- Choice of optimal N extends from quadratic case, but:
- **Added complication:**
 - V_{ij} and V_{ijk} are expressed in different units of measure
 - Summing up V_J for J 's of different sizes may not make much sense
- Multi-objective problem: $\forall p \in [n] \quad \max_{|J|=p} \sum V_J$

Thm.

Efficient solution is an optimum of $\max_{J \subseteq [n]} \sum V_J$

- Introduction
- Reformulation-Linearization Technique
- RRLT literature review
- Reduced RRLT
- RRLT literature review
- New developments
 - Optimal RRLT
 - RRLT for polynomial programming 1/3
 - RRLT for polynomial programming 2/3
 - RRLT for polynomial programming 3/3
 - Sparsity 1/4
 - Sparsity 2/4
 - Sparsity 3/4
 - Sparsity 4/4
- Why bother?
- Thank you

- Choice of optimal N extends from quadratic case, but:
- **Added complication:**
 - V_{ij} and V_{ijk} are expressed in different units of measure
 - Summing up V_J for J 's of different sizes may not make much sense
- Multi-objective problem: $\forall p \in [n] \quad \max_{|J|=p} \sum V_J$

Thm.

Efficient solution is an optimum of $\max_{J \subseteq [n]} \sum V_J$

- Greedy is still OK

Introduction

Reformulation- Linearization Technique

RLT literature review

Reduced RLT

RRLT literature review

New developments

Optimal RRLT

RRLT for polynomial programming 1/3

RRLT for polynomial programming 2/3

RRLT for polynomial programming 3/3

▷ Sparsity 1/4

Sparsity 2/4

Sparsity 3/4

Sparsity 4/4

Why bother?

Thank you

- ☐ Polynomial programs are never dense in practice
- ☐ RRLT needs $B \cup N = \mathcal{P}([n])$
- ☐ Need to introduce exponentially many new monomials

?

- β = set of multi-indices for monomials already in problem

Introduction

Reformulation-
Linearization
Technique

RLT literature
review

Reduced RLT

RRLT literature
review

New developments

Optimal RRLT

RRLT for polynomial
programming 1/3

RRLT for polynomial
programming 2/3

RRLT for polynomial
programming 3/3

Sparsity 1/4

▷ Sparsity 2/4

Sparsity 3/4

Sparsity 4/4

Why bother?

Thank you

Introduction

Reformulation-
Linearization
Technique

RLT literature
review

Reduced RLT

RRLT literature
review

New developments

Optimal RRLT

RRLT for polynomial
programming 1/3

RRLT for polynomial
programming 2/3

RRLT for polynomial
programming 3/3

Sparsity 1/4

▷ Sparsity 2/4

Sparsity 3/4

Sparsity 4/4

Why bother?

Thank you

- ☐ β = set of multi-indices for monomials already in problem
- ☐ Every new monomial $J \notin \beta$ yields a new variable w_J

Introduction

Reformulation-
Linearization
Technique

RLT literature
review

Reduced RLT

RRLT literature
review

New developments

Optimal RRLT

RRLT for polynomial
programming 1/3

RRLT for polynomial
programming 2/3

RRLT for polynomial
programming 3/3

Sparsity 1/4

▷ Sparsity 2/4

Sparsity 3/4

Sparsity 4/4

Why bother?

Thank you

- ☐ β = set of multi-indices for monomials already in problem
- ☐ Every new monomial $J \notin \beta$ yields a new variable w_J
- ☐ Sometimes $\exists J \notin \beta$ s.t. w_J yields > 1 new RRLT constr.

Introduction

Reformulation-
Linearization
Technique

RLT literature
review

Reduced RLT

RRLT literature
review

New developments

Optimal RRLT

RRLT for polynomial
programming 1/3

RRLT for polynomial
programming 2/3

RRLT for polynomial
programming 3/3

Sparsity 1/4

▷ Sparsity 2/4

Sparsity 3/4

Sparsity 4/4

Why bother?

Thank you

- β = set of multi-indices for monomials already in problem
- Every new monomial $J \notin \beta$ yields a new variable w_J
- Sometimes $\exists J \notin \beta$ s.t. w_J yields > 1 new RRLT constr.
- E.g.:

$$x_1 + x_2 = 1 \wedge 2x_1 - x_2 = 3 \wedge \beta = \{(1, 3)\}$$

one new monomial $(x_2x_3) \Rightarrow$ **two** new RRLT constraints

Introduction

Reformulation-
Linearization
Technique

RLT literature
review

Reduced RLT

RRLT literature
review

New developments

Optimal RRLT

RRLT for polynomial
programming 1/3

RRLT for polynomial
programming 2/3

RRLT for polynomial
programming 3/3

Sparsity 1/4

▷ Sparsity 2/4

Sparsity 3/4

Sparsity 4/4

Why bother?

Thank you

- β = set of multi-indices for monomials already in problem
- Every new monomial $J \notin \beta$ yields a new variable w_J
- Sometimes $\exists J \notin \beta$ s.t. w_J yields > 1 new RRLT constr.
- E.g.:

$$x_1 + x_2 = 1 \wedge 2x_1 - x_2 = 3 \wedge \beta = \{(1, 3)\}$$

one new monomial $(x_2x_3) \Rightarrow$ **two** new RRLT constraints

$$\begin{aligned} x_1 + x_2 = 1 & \quad (\times x_3 =) & w_{13} + w_{23} = x_3 \\ 2x_1 - x_2 = 3 & \quad (\times x_3 =) & 2w_{13} - w_{23} = 3x_3 \end{aligned}$$

- **Principle:** one new equation, one fewer degrees of freedom

Introduction

Reformulation-
Linearization
Technique

RLT literature
review

Reduced RLT

RRLT literature
review

New developments

Optimal RRLT

RRLT for polynomial
programming 1/3

RRLT for polynomial
programming 2/3

RRLT for polynomial
programming 3/3

Sparsity 1/4

▷ Sparsity 2/4

Sparsity 3/4

Sparsity 4/4

Why bother?

Thank you

- β = set of multi-indices for monomials already in problem
- Every new monomial $J \notin \beta$ yields a new variable w_J
- Sometimes $\exists J \notin \beta$ s.t. w_J yields > 1 new RRLT constr.
- E.g.:

$$x_1 + x_2 = 1 \wedge 2x_1 - x_2 = 3 \wedge \beta = \{(1, 3)\}$$

one new monomial $(x_2x_3) \Rightarrow$ **two** new RRLT constraints

$$\begin{aligned} x_1 + x_2 = 1 \quad (\times x_3 =) \quad w_{13} + w_{23} &= x_3 \\ 2x_1 - x_2 = 3 \quad (\times x_3 =) \quad 2w_{13} - w_{23} &= 3x_3 \end{aligned}$$

- **Principle:** one new equation, one fewer degrees of freedom

Create fewer J 's than new RRLT constraints

□ Problem:

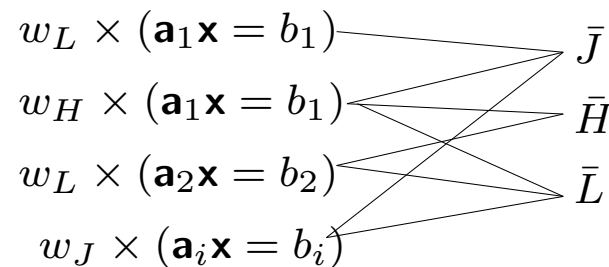
Look for subset ρ of rows of $A\mathbf{x} = \mathbf{b}$ to be multiplied by a subset σ of $\mathcal{P}([n-1])$ such that the number of new vars w_J is $<$ number of new RRLT constraints

- Introduction
- Reformulation-Linearization Technique
- RRLT literature review
- Reduced RLT
- RRLT literature review
- New developments
- Optimal RRLT
- RRLT for polynomial programming 1/3
- RRLT for polynomial programming 2/3
- RRLT for polynomial programming 3/3
- Sparsity 1/4
- Sparsity 2/4
- ▷ Sparsity 3/4
- Sparsity 4/4
- Why bother?
- Thank you

□ Problem:

Look for subset ρ of rows of $A\mathbf{x} = \mathbf{b}$ to be multiplied by a subset σ of $\mathcal{P}([n-1])$ such that the number of new vars w_J is $<$ number of new RRLT constraints

□ Formalization: consider bipartite graph (U, V, E)

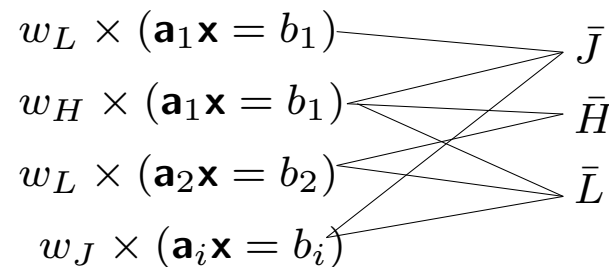


- Introduction
- Reformulation-Linearization Technique
- RLT literature review
- Reduced RLT
- RRLT literature review
- New developments
- Optimal RRLT
- RRLT for polynomial programming 1/3
- RRLT for polynomial programming 2/3
- RRLT for polynomial programming 3/3
- Sparsity 1/4
- Sparsity 2/4
- ▷ Sparsity 3/4
- Sparsity 4/4
- Why bother?
- Thank you

□ Problem:

Look for subset ρ of rows of $A\mathbf{x} = \mathbf{b}$ to be multiplied by a subset σ of $\mathcal{P}([n-1])$ such that the number of new vars w_J is $<$ number of new RRLT constraints

□ Formalization: consider bipartite graph (U, V, E)



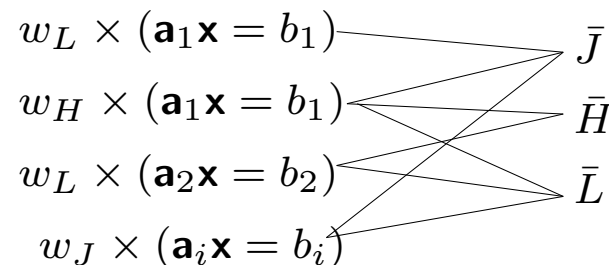
□ $U =$ row i by var. w_J (indexed by (i, J))

- Introduction
- Reformulation-Linearization Technique
- RLT literature review
- Reduced RLT
- RRLT literature review
- New developments
- Optimal RRLT
- RRLT for polynomial programming 1/3
- RRLT for polynomial programming 2/3
- RRLT for polynomial programming 3/3
- Sparsity 1/4
- Sparsity 2/4
- ▷ Sparsity 3/4
- Sparsity 4/4
- Why bother?
- Thank you

□ Problem:

Look for subset ρ of rows of $A\mathbf{x} = \mathbf{b}$ to be multiplied by a subset σ of $\mathcal{P}([n-1])$ such that the number of new vars w_J is $<$ number of new RRLT constraints

□ Formalization: consider bipartite graph (U, V, E)



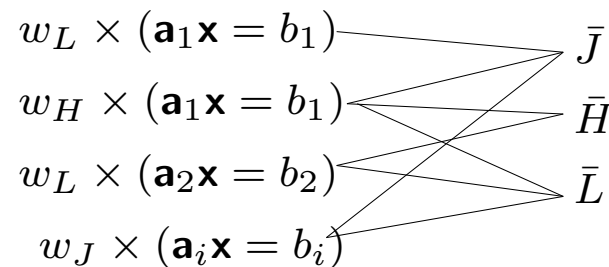
- $U = \text{row } i \text{ by var. } w_J \text{ (indexed by } (i, J))$
- $V = \text{var. } w_{\bar{J}} \text{ with } \bar{J} \notin \beta \text{ (indexed by } \bar{J})$

- Introduction
- Reformulation-Linearization Technique
- RLT literature review
- Reduced RLT
- RRLT literature review
- New developments
- Optimal RRLT
- RRLT for polynomial programming 1/3
- RRLT for polynomial programming 2/3
- RRLT for polynomial programming 3/3
- Sparsity 1/4
- Sparsity 2/4
- ▷ Sparsity 3/4
- Sparsity 4/4
- Why bother?
- Thank you

□ Problem:

Look for subset ρ of rows of $A\mathbf{x} = \mathbf{b}$ to be multiplied by a subset σ of $\mathcal{P}([n-1])$ such that the number of new vars w_J is $<$ number of new RRLT constraints

□ Formalization: consider bipartite graph (U, V, E)



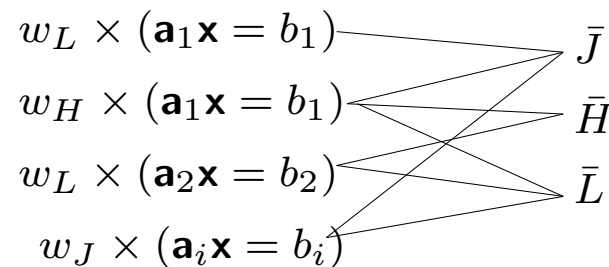
- $U = \text{row } i \text{ by var. } w_J \text{ (indexed by } (i, J))$
- $V = \text{var. } w_{\bar{J}} \text{ with } \bar{J} \notin \beta \text{ (indexed by } \bar{J})$
- **Edges:** $E = \text{incidence of added vars in RRLT constrs}$

- Introduction
- Reformulation-Linearization Technique
- RRLT literature review
- Reduced RRLT
- RRLT literature review
- New developments
- Optimal RRLT
- RRLT for polynomial programming 1/3
- RRLT for polynomial programming 2/3
- RRLT for polynomial programming 3/3
- Sparsity 1/4
- Sparsity 2/4
- ▷ Sparsity 3/4
- Sparsity 4/4
- Why bother?
- Thank you

□ Problem:

Look for subset ρ of rows of $A\mathbf{x} = \mathbf{b}$ to be multiplied by a subset σ of $\mathcal{P}([n-1])$ such that the number of new vars w_J is $<$ number of new RRLT constraints

□ Formalization: consider bipartite graph (U, V, E)



- $U = \text{row } i \text{ by var. } w_J \text{ (indexed by } (i, J))$
- $V = \text{var. } w_{\bar{J}} \text{ with } \bar{J} \notin \beta \text{ (indexed by } \bar{J})$
- **Edges:** $E = \text{incidence of added vars in RRLT constrs}$

Aim: find induced subgraph (U', V', E') such that $|U'|$ is maximum, $|U'| > |V'|$, and $V' = \text{neighb}(U')$

- Introduction
- Reformulation-Linearization Technique
- RLT literature review
- Reduced RLT
- RRLT literature review
- New developments
- Optimal RRLT
- RRLT for polynomial programming 1/3
- RRLT for polynomial programming 2/3
- RRLT for polynomial programming 3/3
- Sparsity 1/4
- Sparsity 2/4
- Sparsity 3/4
- ▷ Sparsity 4/4
- Why bother?
- Thank you

Mathematical Programming formulation:

$$\left. \begin{array}{ll}
 \max & \sum_{(i,J') \in U} u_{i,J'} \\
 & \sum_{(i,J') \in U} u_{i,J'} \geq \sum_{J \notin \beta} v_J + 1 \\
 \forall \{(i, J'), J\} \in E & v_J \geq u_{i,J'} \\
 & u \in \{0, 1\}^{|U|} \\
 & v \in \{0, 1\}^{|\mathcal{P}([n]) \setminus \beta|}.
 \end{array} \right\}$$

- Introduction
- Reformulation-Linearization Technique
- RLT literature review
- Reduced RLT
- RRLT literature review
- New developments
- Optimal RRLT
- RRLT for polynomial programming 1/3
- RRLT for polynomial programming 2/3
- RRLT for polynomial programming 3/3
- Sparsity 1/4
- Sparsity 2/4
- Sparsity 3/4
- ▷ Sparsity 4/4
- Why bother?
- Thank you

Mathematical Programming formulation:

$$\left. \begin{array}{ll}
 \max & \sum_{(i,J') \in U} u_{i,J'} \\
 & \sum_{(i,J') \in U} u_{i,J'} \geq \sum_{J \notin \beta} v_J + 1 \\
 \forall \{(i,J'), J\} \in E & v_J \geq u_{i,J'} \\
 & u \in \{0,1\}^{|U|} \\
 & v \in \{0,1\}^{|\mathcal{P}([n]) \setminus \beta|}.
 \end{array} \right\}$$

Thm.

This problem is in **P**

Proof

Use matching-based algorithm

Introduction

Reformulation-
Linearization
Technique

RLT literature
review

Reduced RLT

RRLT literature
review

New developments

▷ Why bother?

Use within sBB

Thank you

Why bother?

Introduction

Reformulation-
Linearization
Technique

RLT literature
review

Reduced RLT

RRLT literature
review

New developments

Why bother?

▷ Use within sBB

Thank you

- ☐ CPU time in sBB: number of nodes, time to solve each node
- ☐ Need few, small convex relaxation LPs
- ☐ Usually concentrate on *few* (tight bound) but *large* (valid cuts)
- ☐ Different approach: slacken bound, aim to solve each LP faster

Introduction

Reformulation-
Linearization
Technique

RLT literature
review

Reduced RLT

RRLT literature
review

New developments

Why bother?

▷ Use within sBB

Thank you

- ☐ CPU time in sBB: number of nodes, time to solve each node
- ☐ Need few, small convex relaxation LPs
- ☐ Usually concentrate on *few* (tight bound) but *large* (valid cuts)
- ☐ Different approach: slacken bound, aim to solve each LP faster

Outcome:

- bound quality: 0.07% worse;
- CPU improvement: 40%

- ☐ *Future work*: embed in sBB

Introduction

Reformulation-
Linearization
Technique

RLT literature
review

Reduced RLT

RRLT literature
review

New developments

Why bother?

▷ Thank you

Thank you