Basis Reduction, and the Complexity of Branch-and-Bound

Gábor Pataki

Department of Statistics and Operations Research

UNC Chapel Hill

joint work with **Mustafa Tural, IMA**

and

Erick B. Wong, University of British Columbia

Talk outline

Talk outline

- B&B for Integer Programming, and why it is bad.

Talk outline

- **B&B for Integer Programming, and why it is bad.**
- **Basis reduction.**

Talk outline

- **B&B for Integer Programming, and why it is bad.**
- **Basis reduction.**
- **Reformulating IPs via basis reduction.**

Talk outline

- B&B for Integer Programming, and why it is bad.
- Basis reduction.
- Reformulating IPs via basis reduction.
- B&B is a good algorithm after reformulation – solving almost all IPs at the root node.

Talk outline

- **B&B for Integer Programming, and why it is bad.**
- **Basis reduction.**
- **Reformulating IPs via basis reduction.**
- **B&B is a good algorithm after reformulation – solving almost all IPs at the root node.**
- **Meanings: low density knapsacks; random IPs become easier, as the coefficients become larger.**

Bounded Integer Programming (IP) Feasibility Problem

$$\begin{pmatrix} \ell_1 \\ \ell_2 \end{pmatrix} \leq \begin{pmatrix} A \\ I \end{pmatrix} x \leq \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} \qquad (IP)$$

$$x \in \mathbb{Z}^n$$

Here A **is** $m \times n$**.**

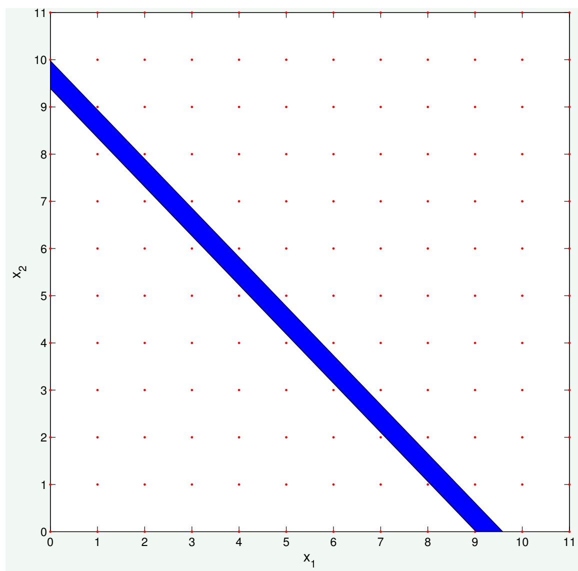
Branch-and-Bound (B&B) to Solve IP Problems

- **First proposed by Land and Doig in the 60s.**
- **Solve the LP relaxation, and get x^*.**
- **If x^* has a fractional component, say x_i^*, divide the problem into subproblems by fixing x_i to its possible integer values (branching).**
- **Continue in this way until an integral solution is found in a subproblem, or all the subproblems are LP infeasible.**
- **Implemented in most software.**

But: Worst case can be quite bad

In the infeasible problem below, branching either on x_1 or x_2 generates 10 nodes.

$$460 \leq 51x_1 + 49x_2 \leq 489$$
$$0 \leq x_1, x_2 \leq 10$$
$$x_1, x_2 \in \mathbb{Z}$$



Theoretically efficient methods for IP

In a sense, the best we can ask for is: polynomial running time, when n is fixed. This is achieved by:

- Lenstra's algorithm (1983);
- Kannan's algorithm (1987);
- Generalized basis reduction of Lovász and Scarf (1992).

- These are more involved than B&B.

Theoretically efficient methods for IP

Integer optimization w.o. binary search: Eisenbrand 2003; Eisenbrand, Laue, 2005.

We can even count the number of solutions in polynomial time, when n is fixed: Barvinok (1994); Dyer-Kannan (1997); De Loera et al (2005); Koeppe (2006).

An augmentation framework for IP: the Integral Basis Method of Haus, Weismantel, and Koeppe (2003), and by Haus (2004).

What is Basis Reduction?

For rational matrix B, basis reduction (BR) finds unimodular U ($\Leftrightarrow$ U integral & $\det U = \pm 1$) such that the columns of BU are short and nearly orthogonal.

Example $$B = \begin{pmatrix} 289 & 18 \\ 466 & 29 \\ 273 & 17 \end{pmatrix}, \quad U = \begin{pmatrix} 1 & -15 \\ -16 & 241 \end{pmatrix}, \quad BU = \begin{pmatrix} 1 & 3 \\ 2 & -1 \\ 1 & 2 \end{pmatrix}.$$

Variants: LLL (Lenstra, Lenstra, Lovász) Reduction, KZ (Korkin-Zolotarev) Reduction, Reciprocal KZ (RKZ) Reduction.

A simple way to use basis reduction to solve (IP)

Preprocess the problem once to make the columns of the constraint matrix reduced;

Just feed the preprocessed problem to a regular IP solver (which will use branch-and-bound).

A simple way to use basis reduction to solve (IP)

Preprocess the problem once to make the columns of the constraint matrix reduced;

Just feed the preprocessed problem to a regular IP solver (which will use branch-and-bound).

Nullspace reformulation: Aardal-Hurkens-Lenstra '98; Aardal-Bixby-Hurkens-Lenstra-Smeltink '00; Louveaux-Wolsey '02; For equality constrained problems, i.e. when $\ell_1 = w_1$.

Rangespace reformulation: Krishnamoorthy-P. 2005. For general IPs.

Rangespace reformulation of (IP)

$$\begin{pmatrix} \ell_1 \\ \ell_2 \end{pmatrix} \leq \begin{pmatrix} A \\ I \end{pmatrix} x \leq \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} \qquad \text{(IP)} \longrightarrow$$

$$x \in \mathbb{Z}^n$$

$$\begin{pmatrix} \ell_1 \\ \ell_2 \end{pmatrix} \leq \begin{pmatrix} A \\ I \end{pmatrix} Uy \leq \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} \qquad \text{(IP-R)}$$

$$y \in \mathbb{Z}^n$$

where U makes the columns of the constraint matrix reduced.

We use either RKZ- or LLL-reduction.

Nullspace reformulation of (IP)

$$\begin{pmatrix} \ell_1 \\ \ell_2 \end{pmatrix} \leq \begin{pmatrix} A \\ I \end{pmatrix} x \leq \begin{pmatrix} \ell_1 \\ w_2 \end{pmatrix} \qquad \text{(IP)} \longrightarrow$$

$$x \in \mathbb{Z}^n$$

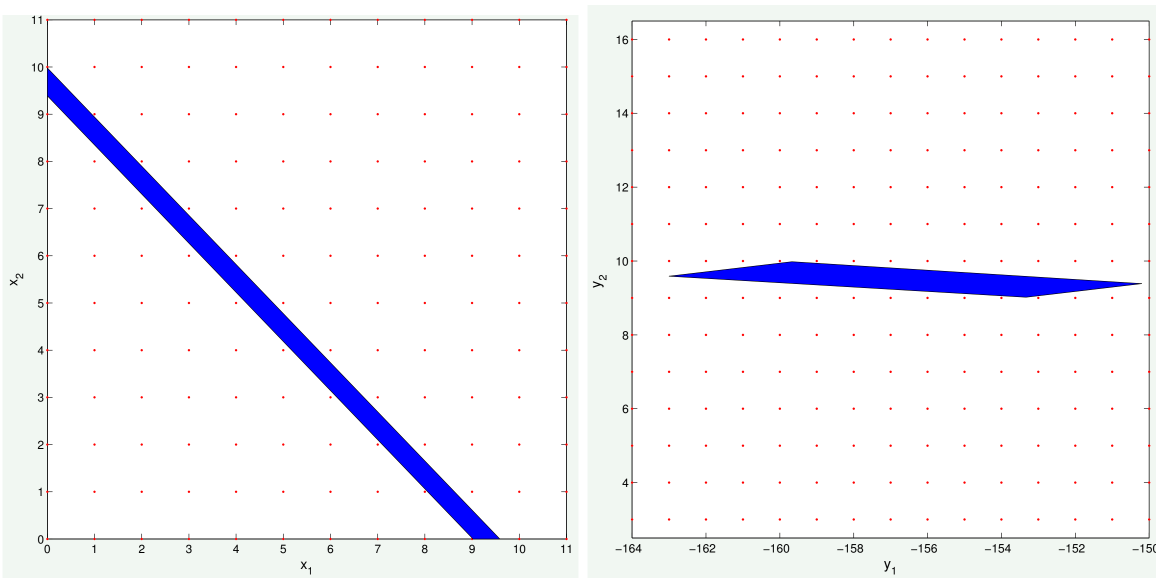
$$\ell_2 \leq B\lambda + x_0 \leq w_2 \qquad \text{(IP-N)}$$

$$\lambda \in \mathbb{Z}^{n-m},$$

where B is a reduced basis of $\{x \in \mathbb{Z}^n | Ax = 0\}$ and $x_0 \in \mathbb{Z}^n$ satisfies $Ax_0 = \ell_1$.

Again, we use either RKZ- or LLL-reduction.

Something nice happens when we do this (on previous example)



Analysis, for knapsack problems, assuming $a = \lambda p + r$, with λ large: Krishnamoorthy-P, 2005 (paper: Discrete Optimization, 2009).

Main point:

$$\text{width(last variable, reformulation)} = \text{width}(p, \text{original})$$

$$\text{iwidth(last variable, reformulation)} = \text{iwidth}(p, \text{original})$$

Goal of the analysis

Given $0 < \epsilon < 1$.

If $M >$ function of $\epsilon, n, m, (\ell_1, \ell_2), (w_1, w_2),$

then for all but an ϵ fraction of A matrices with coefficients from $\{1, \ldots, M\}$

(IP-R) solves with at most one node.

Terminology

Def: reverse B&B $\Leftrightarrow$ B&B branching on $y_n, y_{n-1}, \ldots$.

Def: B&B solves an IP at the rootnode $\Leftrightarrow$
at every level of the tree there is at most one node.

1st ingredient

Consider

$$\begin{pmatrix} \ell_1 \\ \ell_2 \end{pmatrix} \leq \begin{pmatrix} A \\ I \end{pmatrix} U y \leq \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} \qquad \text{(IP-R)}$$

$$y \in \mathbb{Z}^n$$

Lemma 1: When we branch on $y_n, y_{n-1}, \ldots, y_1$ in this order, the total number of B&B nodes on the level of y_i is at most

$$\left(\left\lfloor \frac{\|(w_1, w_2) - (\ell_1, \ell_2)\|}{\|b_i^*\|} \right\rfloor + 1 \right) \ldots \left(\left\lfloor \frac{(w_1, w_2) - (\ell_1, \ell_2)\|}{\|b_n^*\|} \right\rfloor + 1 \right),$$

where the b_i^* are the Gram-Schmidt orthogonalization of the constraint matrix.

2nd ingredient: when the $\|b_i^*\|$ are large

Lemma 2 $L :=$ lattice generated by columns of $\begin{pmatrix} A \\ I \end{pmatrix}$.

Lagarias, Lenstra, Schnorr, 1990: If $\begin{pmatrix} A \\ I \end{pmatrix} U$ is RKZ reduced, with Gram-Schmidt vectors $b_1^*, \ldots, b_n^*$, then

$$\|b_i^*\| \geq \frac{1}{\gamma_i}(\text{length of shortest vector in } L),$$

where $\gamma_i \leq 0.2i$ for $i \geq 10$.

3rd ingredient: when the shortest vector in L is long

Lemma 3 $L :=$ lattice generated by columns of $\begin{pmatrix} A \\ I \end{pmatrix}$, $k \in \mathbb{Z}$.

For all, but at most a fraction of

$$\frac{(2k+1)^{n+m}}{M^m}$$

of $A \in \{1, \ldots, M\}^{m \times n}$ the length of the shortest vector in L is at least k.

Main Theorem

Let $0 < \epsilon < 1$. Assume that RKZ reduction is used, and

$$M > \frac{(2\gamma_n \parallel (w_1, w_2) - (\ell_1, \ell_2) \parallel + 1)^{1+n/m}}{\epsilon^{1/m}}.$$

Then for all, but at most a fraction of ϵ of $A \in \{1, \ldots, M\}^{m \times n}$ reverse B&B solves (IP-R) at the rootnode.

Variants

- **Analysis for nullspace reformulation. It only applies when $\ell_1 = w_1$, but better bound on M.**

- **Analysis with LLL reduction. Weaker bound on M, but the reformulations are polynomial time computable.**

Meaning of Main Theorem 1

Subset sum problem:

$$ax = \beta, \quad x \in \{0, 1\}^n. \quad (*)$$

Meaning of Main Theorem 1

Subset sum problem:

$$ax = \beta, \quad x \in \{0,1\}^n. \quad (*)$$

Suppose $a \in \{1, \ldots, M\}^n$, $M \geq 2^{cn^2}$ **(density** $\leq 1/(cn)$**).**

Meaning of Main Theorem 1

Subset sum problem:

$$ax = \beta, \quad x \in \{0,1\}^n. \quad (*)$$

Suppose $a \in \{1, \ldots, M\}^n$, $M \geq 2^{cn^2}$ **(density** $\leq 1/(cn)$**).**

Lagarias, Odlyzko (1985) For almost all a **the solution of (*), if exists, can be found in polynomial time, using LLL reduction.**

Meaning of Main Theorem 1

Subset sum problem:

$$ax = \beta, \ \ x \in \{0,1\}^n. \quad (*)$$

Suppose $a \in \{1,\ldots,M\}^n$, $M \geq 2^{cn^2}$ **(density** $\leq 1/(cn)$**).**

Lagarias, Odlyzko (1985) For almost all a **the solution of (*), if exists, can be found in polynomial time, using LLL reduction.**

Frieze (1986) Simplified algorithm to prove the same.

Meaning of Main Theorem 1

Subset sum problem:

$$ax = \beta, \quad x \in \{0,1\}^n. \quad (*)$$

Suppose $a \in \{1, \ldots, M\}^n$, $M \geq 2^{cn^2}$ **(density** $\leq 1/(cn)$**).**

Lagarias, Odlyzko (1985) For almost all a **the solution of (*), if exists, can be found in polynomial time, using LLL reduction.**

Frieze (1986) Simplified algorithm to prove the same.

Furst, Kannan (1987): For almost all a **(*) can be solved in polynomial time.**

Meaning of Main Theorem 1

Subset sum problem:

$$ax = \beta, \quad x \in \{0,1\}^n. \quad (*)$$

Suppose $a \in \{1, \ldots, M\}^n$, $M \geq 2^{cn^2}$ **(density** $\leq 1/(cn)$**).**

Lagarias, Odlyzko (1985) **For almost all** a **the solution of (*), if exists, can be found in polynomial time, using LLL reduction.**

Frieze (1986) **Simplified algorithm to prove the same.**

Furst, Kannan (1987): **For almost all** a **(*) can be solved in polynomial time.**

Corollary of this work: **For almost all** a **(*) can be solved in polynomial time using branch-and-bound on the LLL-reformulations.**

Meaning of Main Theorem 1

Similar result, generalizing Furst, Kannan (1987) using the RKZ reformulations,

when $M \geq 2^{cn \log n}$ (density $\leq 1/(\log n)$).

Meaning of Main Theorem 2

Practical view: an RKZ basis is easy to compute in practice if $n \leq 100$.

We can get tighter bounds for small sizes. For equality constrained, binary problems, with the RKZ-nullspace reformulation we get:

n	m	bad fraction	M
30	20	10%	31
50	20	10%	1846
50	30	10%	93

In other words

Consider the family of IPs

$$\begin{aligned} Ax &= b \\ x &\in \{0,1\}^n \end{aligned}$$

with 50 variables, 20 constraints, coefficients of A from $\{1,\ldots,1846\}$.

The RKZ-nullspace-reformulation solves at the rootnode for 90% of the instances.

If we have 30 constraints, then same result holds, if coefficients are from $\{1,\ldots,93\}$.

Meaning of Main Theorem 3

Consider marketshare problems (Cornuéjols and Dawande)

$$\begin{aligned} Ax &= b \\ x &\in \{0,1\}^n \end{aligned}$$

and the relaxed versions

$$\begin{aligned} b - e \leq Ax &\leq b \\ x &\in \{0,1\}^n \end{aligned}$$

where $A \in \{1, \ldots, M\}^{5\times 40}$, $b = \lfloor Ae/2 \rfloor$.

Aardal, Bixby, Hurkens, Lenstra, Smeltink (1998): the nullspace reformulations are much easier to solve by commercial solvers, than the original ones.

In the above references, $M = 100$ is used.

Meaning of Main Theorem 3

According to the theory, the reformulations should get easier, as M grows.

Results of a computational experiment: avg number (for 12 instances) of nodes to solve rangespace reformulation of inequality, and nullspace reformulation of equality constrained problems. MIP solver: CPLEX 9.1.

M	EQUALITY	INEQUALITY
100	17531.92	38884.92
1000	1254.42	22899.67
10000	200.83	1975.67

Recap

- **B&B is a classical, and "bad" algorithm from the theoretical point of view.**

Recap

- **B&B is a classical, and "bad" algorithm from the theoretical point of view.**

- **B&B is efficient from a computational complexity viewpoint, after (IP) has been reformulated:**

Recap

- **B&B is a classical, and "bad" algorithm from the theoretical point of view.**

- **B&B is efficient from a computational complexity viewpoint, after (IP) has been reformulated:**

- **For most of the instances, if coefficients are drawn from $\{1, \ldots, M\}$ for a large enough M, then reformulated problem solves at the root.**

Meanings

- **low density subset sum problems can be solved in polynomial time by branch-and-bound;**

Meanings

- **low density subset sum problems can be solved in polynomial time by branch-and-bound;**
- **for small n and m even small values for M suffice;**

Meanings

- **low density subset sum problems can be solved in polynomial time by branch-and-bound;**
- **for small n and m even small values for M suffice;**
- **reformulations of random integer programs become easier as the coefficients grow.**

Thank you!