Slide 1

Column Basis Reduction, Decomposable Knapsack and Cascade Problems

Gábor Pataki

Dept. of Statistics and Operations Research
UNC, Chapel Hill

joint work with Bala Krishnamoorthy

Dept. of Mathematics, Washington State University

Slide 2

What is basis reduction ?

Given integral matrix A, basis reduction (BR) computes a unimodular $U (\Leftrightarrow \det U = \pm 1)$ st. the columns of AU are "short" and "nearly" orthogonal.

Example

$$A = \begin{pmatrix} 289 & 18 \\ 466 & 29 \\ 273 & 17 \end{pmatrix}, \; U = \begin{pmatrix} 1 & -15 \\ -16 & 241 \end{pmatrix}, \; AU = \begin{pmatrix} 1 & 3 \\ 2 & -1 \\ 1 & 2 \end{pmatrix}.$$

Computing AU $\Leftrightarrow$ doing *elementary column operations* on A:

- adding an integer multiple of a column to another; multiplying a column by -1; swapping columns.

Slide 3

Reformulating equality constrained IP feasibility problems

Aardal, Hurkens, Lenstra (1998); Aardal, Bixby, Hurkens, Lenstra, Smeltink (1999); Aardal, Lenstra (2004); Louvaux, Wolsey (2003).

$$\begin{array}{c} x \in \mathcal{Z}^n \\ Ax = d \\ \ell \le x \le u \\ \downarrow \qquad \textbf{Reformulation} \\ \lambda \in \mathcal{Z}^{n-m} \\ \ell \le B\lambda + x_d \le u \end{array}$$

Slide 4

Here

$$\{\, x \in \mathcal{Z}^n \mid Ax = d \,\} \quad = \quad \{\, x_d + B\lambda \mid \lambda \in \mathcal{Z}^{n-m} \,\}$$

- $[B, x_d]$ is
  - integral, columns are short and nearly orthogonal.
  - found by doing **basis reduction** on an enlarged matrix using two large constants N_1, N_2.
- The reformulated problem of finding

$$\lambda \in \mathcal{Z}^{n-m},\ \ell \le B\lambda + x_d \le b$$

  proved experimentally *much* easier to solve for some problems, e.g. the Cornuejols-Dawande instances.

Slide 5

Questions

1. Why only equality constrained problems?
2. Why does it work?

Slide 6

Rest of talk

1. Column BR: simplified reformulation for arbitrary IPs. 2 variants: in range space and null space.
2. Computational study.
3. Analysis for a general problem class, called *decomposable knapsack problems*.

Slide 7

Rangespace reformulation

$$\begin{aligned} P &= \{\, x \mid \ell \le \ Ax \le b \,\} \\ \tilde{P} &= \{\, y \mid \ell \le (AU)y \le b\} \end{aligned}$$

where U is unimodular.

There is 1-1 correspondence between

$$P \cap \mathcal{Z}^n \text{ and } \tilde{P} \cap \mathcal{Z}^n$$

given by

$$Uy = x$$

We choose U so columns of AU are reduced. We can do the same if some of the "$\le$" are actually "$=$".

Slide 8

Nullspace reformulation

If

$$A_1 x = b_1$$

is a subset of the inequalities in $\ell \le Ax \le b$, then

$$\{\, x \in \mathcal{Z}^n \mid A_1 x = b_1 \,\} = \{\, x_d + B_1 \lambda \mid \lambda \in \mathcal{Z}^{n-m} \,\}$$

$[B_1, x_d]$ is found by a Hermite Normal Form (HNF) computation; columns are *not* in general short and orthogonal.

Substitute $B_1\lambda + x_d$ for x, and do the rangespace reformulation.

If all constraints are equalities, then essentially equivalent to the Aardal et al. reformulation.

Slide 9

- Such a simple reformulation actually works for essentially all hard IPs used to test "nontraditional" IP algorithms!
- We need a problem class on which we can *analyze* its action.

Slide 10

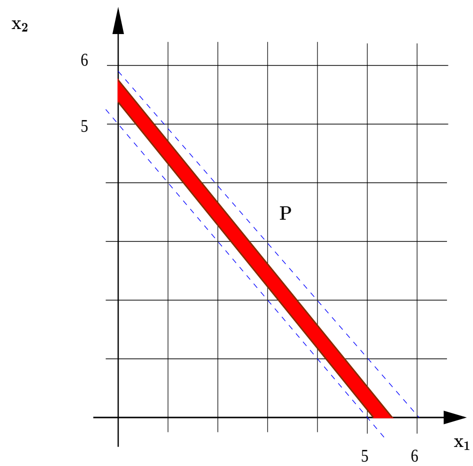
Branching on a constraint

Given polyhedron P, integral vector c,

- $\mathrm{width}(c, P) = \max\{\, cx \mid x \in P \,\} - \min\{\, cx \mid x \in P \,\}$.
- **branching on** cx means creating the branches $cx = \lceil \min \rceil$, $cx = \lceil \min \rceil + 1, \ldots, cx = \lfloor \max \rfloor$.
- If the interval $[\min, \max]$ contains no integer, then P contains no integral point.

Slide 11

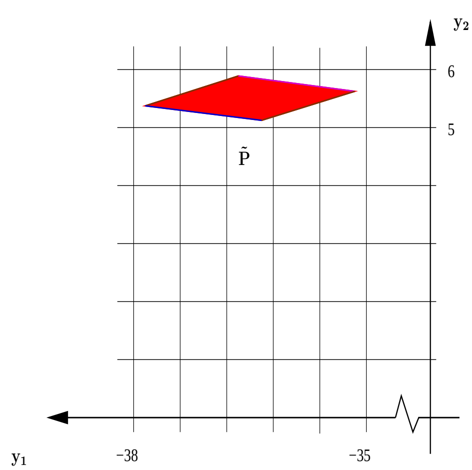
Example :
$$\begin{aligned} 106 \leq \quad 21x_1 + 19x_2 \quad &\leq 113 \\ x_1, x_2 \in \quad &\in [0,6] \cap \mathcal{Z} \end{aligned}$$



Hard for branching on x_is.

Easy for branching on $x_1 + x_2$: max = 5.94, min = 5.04.

Slide 12



After reformulation: branching on y_2 proves infeasibility.

Slide 13

2-level decomposable knapsack problems

The example is an instance of

$$(KP_2) \quad \beta' \leq a\,x\; \leq \beta, \qquad 0 \leq x \leq u, \quad x \in \mathcal{Z}^n,$$

where

- $a = pM + r$, with $p \in \mathcal{Z}^n_+$, $r \in \mathcal{Z}^n$; M large;
- β, β' chosen, so KP_2 is LP-feasible, IP-infeasibility proven by branching on px.
- In the example, $(21, 19) = (1, 1) * 20 + (1, -1)$.

Slide 14

What does the reformulation do on these?

Recall general reformulation:

$$P = \{x \,|\, \ell \leq Ax \leq b\,\} \Leftrightarrow \tilde{P} = \{y \,|\, \ell \leq\; (AU)y \leq b\}$$

Slide 15

Basis reduction in range space

We choose U unimodular, s.t.

$$\begin{pmatrix} pM + r \\ I \end{pmatrix} U \quad \text{is reduced.}$$

Theorem: M suff. large $\Rightarrow$

$$pU = (\,\overbrace{0\ \ldots\ 0}^{n-1}\ \alpha\,) \quad \text{for some } \alpha \in \mathcal{Z} \setminus \{\,0\,\}.$$

Corollary:

$$U\,y = x \;\Rightarrow\; pUy \;=\; p\,x \;\;\Rightarrow\; \alpha y_n = \; p\,x$$

$\Rightarrow$ branching on y_n proves infeasibility.

Slide 16

"Sufficiently large" means:

- If LLL (Lenstra, Lenstra, Lovasz) reduction is used, $M > 2^{n+1}\,\|\,p\,\|\|\,r\,\|^2$.
- If KZ (Korkhine-Zolotarev) reduction is used, $M > \sqrt{n}\,\|\,p\,\|\|\,r\,\|^2$.

Slide 17

Basis reduction in null space

Can be used if $\beta = \beta'$ $\rightarrow$ reformulation has $n-1$ variables.

We can similarly prove: M suff. large $\Rightarrow$ branching on y_{n-1} in reformulation $\equiv$branching on px in original problem.

Slide 18

A classic example of a decomposable knapsack problem: Jeroslow's problem

$$\begin{aligned} 2(x_1 + \ldots + x_n) &= n \\ x_i \in \{\, 0, 1 \,\}^n & \end{aligned}$$

where n is odd. In B&B branching on the x_i no node is pruned above level $n/2$. If we branch on $x_1 + \ldots x_n$, we solve it at the root.

Here $p = e$, $r = 0$, $M = 2$.

Slide 19

Other examples:

1. $p = e$, $r = (2^0, \ldots, 2^{n-1})$, $u = e$, $M = 2^{n+\ell+1}$: Todd's problem from Chvátal "Hard knapsack problems" (1983).
2. $p = e$, $r = (1, \ldots, n)$, $u = e$, $M = n(n+1)$: Avis' problem from same paper.
3. A modification of (1): Gu, Nemhauser (2001).
4. $p \geq 0$, r arbitrary, $u = +\infty$, $\beta = \beta'$: Aardal-Lenstra Frobenius problems.

Out of these: (1) and (2) take $2^{n/2}$ nodes for ordinary B&B ; in (4) has a $\beta = \text{const}^* M^2$ for which problem is infeasible.

Slide 20

Algorithms that find thin directions to branch on

- H. W. Lenstra (1983); Kannan (1987); Eisenbrand (2004): polytime algorithms for IP in fixed dimensions. Implementation: Gao, Zhang (2002); Modification and implementation: Mehrotra, Li (2004).
- Generalized BR: Lovasz, Scarf (1990); Implementation: Cook, Rutherford, Scarf, Shallcross (1993); Modification and implementation: Mehrotra, Li (2004).

Slide 21

When thinner $\neq$ better

$$5660 \leq 520x_1 + 725x_2 + 1156x_3 + 1574x_4 + 1794x_5 + 1829x_6$$
$$+2023x_7 + 2221x_8 + 2267x_9 + 2465x_{10} + 2496x_{11} \leq 5661$$
$$x_i \in \{0,1\} \ (i = 1, \ldots, 11). \tag{1}$$

- IP-infeasible, and 'reasonably" hard for B&B .
- If Q = LP relaxation, then $\min_{c \text{ integral}} \text{width}(c, Q) = 1 - 0$, attained at e_i.
- $\exists\, p_1$ integral: $\text{width}(p_1, Q) = 25.34 - 24.30 \Rightarrow$ constraint $p_1 x = 25$ can be added to LP.
- If Q' = new LP relaxation, then $\exists\, p_2$ integral: $\text{width}(p_2, Q') = 14.93 - 14.02 \Rightarrow$ proves IP-infeasibility.

Slide 22

- So, a direction with width $= 1.04$ beats all directions with width 1!
- Such problems are called *cascade* problems: branching on a good direction has a "cascade" effect.
- There are more extreme examples, with width in good direction ≈ 1.5.

Slide 23

$t+1$-level decomposable knapsack problems

- For $a = p_1 M_1 + p_2 M_2 + \ldots + p_t M_t + r$, with $M_1 > M_2 > \ldots > M_t$ and suitable β, β'

$$(KP_{t+1}) \quad \beta' \leq a\,x \leq \beta, \quad 0 \leq x \leq u, \quad x \in \mathcal{Z}^n$$

Problem is

- easy, if branching on $p_1 x$, $p_2 x$, ..., $p_t x$.
- hard, if branching on x_j variables, if parameters suitably chosen.
- cascade problems can be constructed this way.

Slide 24

When using the rangespace reformulation: compute U so that

$$\begin{pmatrix} \sum_{i=1}^t p_i M_i + r \\ I \end{pmatrix} U \quad \text{is reduced.}$$

Theorem: If separation between $M_1 > M_2 > \ldots > M_t$ is suitably large, then

$$\begin{pmatrix} p_1 \\ p_2 \\ \vdots \\ p_t \end{pmatrix} U = \begin{pmatrix} 0 & 0\ldots & 0 & 0 & 0 & * \\ 0 & 0\ldots & 0 & 0 & * & * \\ \vdots & & & & & \\ 0 & 0\ldots & * & \ldots & * & * \end{pmatrix}$$

Remark: When computing U, we do not know the decomposition!!

Slide 25

Corollary: Branching on y_n, $y_{n-1}, \ldots, y_{n-t}$ in reformulation

$\Leftrightarrow$ branching on $p_1 x$, $p_2 x$, ..., $p_t x$ in original problem.

Analogous result for nullspace reformulation.

- That is, column BR
  - takes the *unknown* "dominant" branching combinations;
  - transforms them into individual variables;
  - lines them up in reverse order of significance!

Slide 26

Computational results

- BR: by NTL library of Victor Shoup.
- IP solver: CPLEX 9.0.
- Machine: 3.2 GHz Linux PC.
- We adapted column BR to deal with optimization problems.
- We report: time and B&B nodes taken by CPLEX 9.0 *after* reformulation.
- We do not report: time taken *without* reformulation (even in the simplest case, it is a few hundred thousand B&B nodes; usually it is $+\infty$).

Slide 27

To solve

$$\begin{array}{rrcl} \max & cx & & \\ st. & Ax & \leq & b \\ & x & \in & \mathcal{Z}^n \end{array}$$

we replace A with AU, c with cU, where U makes

$$\begin{pmatrix} c \\ A \end{pmatrix} U$$

reduced.

Slide 28

Maximization versions of integer subset sum

$$\begin{array}{rrcl} \max & ax & & \\ st. & ax & \leq & \beta \\ & x & \in & \mathcal{Z}^n_+. \end{array} \tag{2}$$

First four instances from Cornuéjols, Urbaniak, Weismantel, Wolsey (1998). Last (shown below) from Wolsey: Integer Programming (1999).

$$(12228, 36679, 36682, 48908, 61139, 73365); 89716837$$

Number of B&B nodes after column BR: 5, 0, 9, 0, 10.

Slide 29

Feasibility versions of same instances

For (a, β), $\beta_a :=$ optimal value. Then check the feasibility of

$$\begin{aligned} ax &= \beta_a \\ x &\in \mathcal{Z}^n_+, \end{aligned} \tag{3}$$

using 1) rangespace reformulation, 2) nullspace reformulation. Number of B&B nodes is between 0 and 10 for all 5 instances, for both choices.

Same happens, if rhs is chosen as $\beta_a + \gcd(a)$.

Slide 30

Marketshare problems (Cornuéjols, Dawande)

We need to find

$$x \in \{0, 1\}^n, \quad Ax = d,$$

where $m = 6$ or $m = 7$, $n = 10(m-1)$. A, d are generated to make the problem difficult.

	range space		null space	
	# BB	CPU	# BB	CPU
ms1	288597	175.30	51887	32.80
ms2	220803	165.40	52920	43.70

Slide 31

Relaxed marketshare problems

Same data, but we want to find

$$x \in \{0, 1\}^n, \quad d - 1 \leq Ax \leq d.$$

After column BR

- *markshare1:* $85,466$ nodes, 53 seconds; *markshare2:* $250,368$ nodes, 211 seconds.

Slide 32

$\boxed{\textbf{Cascade2}}$

The "big brother" of the 11-variable instance.

- $n = 100$ variables, $a_j \leq 14,000$, β, $\beta' \leq 100,000$.
- Original problem does not solve by CPLEX after enumerating 2 billion B&B nodes.
- Easy, if we branch on $p_1 x$, then $p_2 x$.
- Reformulation solves at rootnode.

Slide 33

Caveats

- There are hard IPs for which the reformulation does *not* work :-(
- The reformulation uncovers the hidden “dominant” directions in the polyhedron - but in some hard problems, these may not exist, if the problem is symmetric.

Slide 34

Conclusions and further work

- A general, and very simple reformulation technique for arbitrary IPs.
- A fairly general class of IPs that are provably hard for ordinary B&B .
- Analysis: the provably hard problems turn into provably easy ones: the reformulation “uncovers” the hidden, dominant directions.
- The *cascade* problems: thinner $\neq$ better!
- Works well in on most small, hard IPs from the literature.