Generating Dual Solutions in MILP

MIP Workshop, Stamford, May 2026

Menal Güzelsoy Laszlo Ladanyi
SAS Institute Inc.

Agenda

1. LP Duality/Bound Tightening
2. Improved Bound Tightening
3. Generating LP Dual Solutions
4. Generating IP Dual Functions

SAS MILP Solver

- Node presolver - one of the important components
- SAS MILP solver node presolver has around 20 techniques
- Most of them are well-known, generic techniques
  - Reduced cost fixing/degenerate reduced cost fixing
  - Dual solutions/rays pools
  - Probing
  - Orbital probing
  - Constraint propagation
  - Disjoint solver
  - Solution enumeration ...
- About half of them use LP duality/dual information to tighten bounds

LP Duality

- Consider the following LP problem

$$
\begin{aligned}
z = \min\ & cx \\
\text{s.t. } & L \le Ax \le U \\
& l \le x \le u
\end{aligned}
$$

- For a given *dual feasible solution* y, define

$$z_y(L,U,l,u) = \sum_{i:y_i>0} L_i y_i + \sum_{i:y_i<0} U_i y_i + \sum_{j:r_j>0} l_j r_j + \sum_{j:r_j<0} u_j r_j$$

where $r_j = c_j - A_j^T y$

- Then, we have $z \ge z_y(L,U,l,u)$

Column Bound Tightening

- We are given a cutoff $\bar{z}$
- If $r_j = c_j - A_j^T y > 0$, then increasing l_j also increases z_y
- If there is $l_j < l'_j < u_j$ such that

$$z' = z_y(L, U, l, u) + (l'_j - l_j) r_j \geq \bar{z},$$

  then we know that there are no better feasible solutions where $x_j > l'_j$

- Therefore, we can tighten u_j to l'_j where

$$l'_j = l_j + \frac{\bar{z} - z_y(L, U, l, u)}{r_j}$$

- Same applies to upper bound side

Improvements

- All apply to dual rays also – no need for cutoff
- All apply to row bounds also – we can tighten row bounds
- Integrality improves the tightening
- Primal formulation is tightened:
  - Implications
  - Row min/max

Implications

- Assume $l_k \to l'_k$ implies a set of bound changes:

$$l'_s \leq x_s \leq u'_s, s \in S$$

  - Conflict graph/structured rows
  - Constraint/domain propagation
  - Dual conflicts (if setting $x_1 = 1, x_2 = 1 \to z_y \geq \bar{z}$, then, we have $x_1 = 1 \to x_2 = 0$)
- In addition to x_k, primal side can be modified to include the changes in S
- y is still a dual feasible solution for the problem where we set l_k, l_S, u_S to l'_k, l'_S, u'_S and we have

$$z_y(L, U, l', u') \geq z_y(L, U, l, u)$$

- So if we can tighten the upper bound of x_k for the modified problem, then it can be tightened for the original problem also.

Implications

- We now have $z' = z_y(L,U,l,u) + (l'_k - l_k)r_k + \Delta_S$, where

$$\Delta_S = \sum_{s:r_s>0}(l'_s - l_s)r_s + \sum_{s:r_s<0}(u'_s - u_s)r_s.$$

- If $z' \geq \bar{z}$, then we can tighten u_k to l'_k where

$$l'_k = l_k + \frac{\bar{z} - z_y(L,U,l,u) - \Delta_S}{r_k}$$

Row Min/Max

- The min/max of a row can be bounded using the column bounds:

$$\underline{R}_i = \sum_{j:a_{ij}>0} a_{ij} l_j + \sum_{j:a_{ij}<0} a_{ij} u_j$$

$$\overline{R}_i = \sum_{j:a_{ij}<0} a_{ij} l_j + \sum_{j:a_{ij}>0} a_{ij} u_j$$

- Moving $l_k \rightarrow l'_k$ moves $\underline{R}_i$ or $\overline{R}_i$:
  - if $a_{ik} > 0 \rightarrow \underline{R'}_i = \underline{R}_i + a_{ik}(l'_k - l_k)$
  - if $a_{ik} < 0 \rightarrow \overline{R'}_i = \overline{R}_i + a_{ik}(l'_k - l_k)$
- $\underline{R'}_i$ and $\overline{R'}_i$ can also be improved by implications:

$$\underline{R'}_i = \underline{R}_i + \max\{0, a_{ik}(l'_k - l_k)\} + \sum_{s:a_{is}>0} a_{is}(l'_s - l_s) + \sum_{s:a_{is}<0} a_{is}(u'_s - u_s)$$

$$\overline{R'}_i = \overline{R}_i + \min\{0, a_{ik}(l'_k - l_k)\} + \sum_{s:a_{is}<0} a_{is}(l'_s - l_s) + \sum_{s:a_{is}>0} a_{is}(u'_s - u_s)$$

Row Min/Max

- We can substitute L_i with $\underline{R'}_i$ if $\underline{R'}_i > L_i$.
- We can substitute U_i with $\overline{R'}_i$ if $\overline{R'}_i < U_i$.
- We have $z' = z_y(L, U, l, u) + (l'_k - l_k)r_k + \Delta_S + \Gamma_T$, where

$$\Gamma_T = \sum_{\substack{t: y_t > 0, \\ \underline{R'}_t > L_t}} (\underline{R'}_t - L_t) y_t + \sum_{\substack{t: y_t < 0, \\ \overline{R'}_t < U_t}} (\overline{R'}_t - U_t) y_t$$

- If $z' \geq \bar{z}$, then we can tighten u_k to l'_k where

$$l'_k = l_k + \frac{\bar{z} - z_y(L, U, l, u) - \Delta_S - \Gamma_T}{r_k}$$

Dual information in Branch and Bound

- Bound tightening can be done using any dual feasible solution and/or dual ray.
- Dual information
  - Dual solutions from feasible node LPs
  - Dual rays from infeasible node LPs
  - Degenerate reduced cost fixing
  - Strong branching
  - Objective rotation (Schürmann)
- Usually we need to (re)solve an LP or at the very least do some simplex iterations to get to a new dual solution

Generating Dual Solutions

- We are looking for a heuristic method to generate new dual feasible solutions for bound tightening
- $y + v$ is dual feasible as long as
  - $y_i + v_i > 0$ and $L_i > -\infty$
  - $y_i + v_i < 0$ and $U_i < \infty$
  - $r_j - A_j^T v > 0$ and $l_j > -\infty$
  - $r_j - A_j^T v < 0$ and $u_j < \infty$
- If all columns are bounded in row i, then the direction in that index should only satisfy the row bound feasibility:
  - If only $L_i > -\infty$, then we can pick $v_i \in [-y_i, \infty)$
  - If only $U_i < \infty$, then we can pick $v_i \in (-\infty, -y_i]$
  - If row is bounded on both sides, then $v_i \in (-\infty, \infty)$

Choosing v

- Most obvious choice: $v_i = -y_i$
- Break points:
  - $y_i + v_i \in \{p : p = \frac{r_j + y_i a_{ij}}{a_{ij}}, j \in N\}$
  - Note that these would include the degenerate reduced cost fixing
  - No need to go through all break points
- Other dual solutions:
  - Set $v_i = \pi_i - y_i$ for some i, where π_i is a part of a dual solution seen before
- Any combination

Experimental Results

- $\sim$ 1200 test instances (internal + public), 1h time limit
- Using implications and row min/max in node presolve probing:
  - 46% affected instances - 3% speed up
  - All instances:
    $>$ 10s: 2%
    $>$ 100s: 3%
    $>$ 1000s: 9%
- Generating dual solutions after all other dual based bound tightenings:
  - 33% affected instances - 5% speed up
  - All instances:
    $>$ 10s: 3%
    $>$ 100s: 5%
    $>$ 1000s: 7%

LP Dual Feasible Function

- Consider the LP problem: $z(b) = \min\{cx \mid Ax \geq b,\ x \geq 0\}$
- We want to find a function f that satisfies $f(d) \leq z(d)\ \forall d$
- Let us restrict f:
  - linear and nondecreasing: $f(d) = y^T d$ and $y \geq 0$,
  - satisfies $f(A_j) \leq c_j\ \forall j$
- We can show that such a function f satisfies $f(d) \leq z(d)\ \forall d$
  - For a right hand side b and any feasible solution x:

$$f(b) \leq f(Ax) = \sum_j f(A_j)x_j \leq \sum_j c_j x_j \Rightarrow f(b) \leq z(b)$$

- It is also possible to show that $\exists\, f^*$ with $f^*(b) = z(b)$
- In fact, we can derive the LP dual problem:

$$\max\{f(b) \mid f(A_j) \leq c_j\ \forall j\} = \max\{y^T b \mid y^T A_j \leq c_j\ \forall j,\ y \geq 0\}$$

IP Dual Feasible Function

- IP problem: $z_{\mathsf{IP}}(b) = \min\{cx \mid Ax \geq b,\ x \geq 0,\ x \in \mathbb{Z}^n\}$
- Similarly, we want to find F that satisfies $F(d) \leq z_{\mathsf{IP}}(d)\ \forall d$
- Let us restrict F:
  - subadditive[1], nondecreasing
  - satisfies $F(A_j) \leq c_j\ \forall j$
- We can show that F satisfies $F(d) \leq z_{\mathsf{IP}}(d)\ \forall d$
  - For a right hand side b and a feasible solution x:

$$F(b) \leq F(Ax) = F(\sum_j A_j x_j) \leq \sum_j F(A_j x_j) \leq \sum_j F(A_j) x_j \leq \sum_j c_j x_j \Rightarrow$$
$$F(b) \leq z_{\mathsf{IP}}(b)$$

- It is also possible to show that $\exists\, F^*$ with $F^*(b) = z_{\mathsf{IP}}(b)$

[1] $F(d_1) + F(d_2) \geq F(d_1 + d_2)$

LP Duality vs IP Duality

LP duality f	IP duality F
linear	subadditive
nondecreasing	nondecreasing
$f(A_j) \leq c_j \ \forall j$	$F(A_j) \leq c_j \ \forall j$
$\Rightarrow$	$\Rightarrow$
$f(d) \leq z(d) \ \forall d$	$F(d) \leq z_{\mathsf{IP}}(d) \ \forall d$
$f^*(b) = z(b)$	$F^*(b) = z_{\mathsf{IP}}(b)$

Subadditive Dual Functions

- IP dual feasible function F can be used for bound tightening
- Assume we move $l_j \to l'_j$ ($b \to b'$)
- If $F(b') \geq \bar{z}$, then we can tighten u_j to l'_j since $z_{\mathsf{IP}}(b') \geq F(b')$
- All the analysis we had for the LP case is valid for the IP case.
- We can apply the implications, row min/max etc.
- How do we find IP dual feasible functions?

Subadditive Dual Functions

- There are ways of generating IP dual feasible functions:
  - Cutting planes / Branch and bound tree
  - Corrected linear dual functions
  - Lagrangian relaxation etc.
- Just as in LP case, we are after practical ways of generating IP dual feasible functions
- We are not looking for an optimal IP dual function
- For LP, by generating LP dual solutions, we were searching for other linear functions close to f
- LP dual function $f(y) = y^T d$ is also subadditive and feasible to the IP dual
- Can we also search the *subadditive neighborhood* of this function?

Subadditive Operations

- Given a linear function f: $f(d) = y^T d,\ y >= 0$
- We can generate subadditive functions with some operations
- Ceiling operation

$$H(d) = \sum_{i \in K} \lceil y_i d_i \rceil + \sum_{i \in M \setminus K} y_i d_i$$

- Max operation using $y, \pi > 0$

$$H(d) = \sum_{i \in K} \max\{y_i d_i, \pi_i d_i\} + \sum_{i \in M \setminus K} y_i d_i$$

- Absolute value operation (not nondecreasing)

$$H(d) = \sum_{i \in K} |y_i d_i| + \sum_{i \in M \setminus K} y_i d_i$$

- Others... ($h(d) = d^p$ where $d >= 0$ and $0 \leq p \leq 1$ etc.)
- Any combination will do
- How can we use these operations to get an IP dual feasible function from an LP dual feasible function?

LP Dual Feasibility

- Let us expand the IP formulation:

$$\begin{aligned} z_{\mathsf{IP}}(b, l, -u) = \min \;\; & cx \\ s.t \;\; Ax &\geq b \\ x &\geq l, \\ -x &\geq -u, \\ x &\geq 0, \;\; x \in \mathbb{Z}^n \end{aligned}$$

- We are given a dual feasible function for the LP relaxation:

$$f(d) = f(d^A, d^l, d^u) = y^T d^A + w^T d^l + v^T d^u$$

- Let $r_j = c_j - A_j^T y \; \forall j$
- We have $w_j = r_j$ if $r_j > 0$ or $v_j = -r_j$ if $r_j < 0 \;\; \forall j$
- LP dual feasibility: $f(A_j, e_j, -e_j) = y^T A_j + c_j - A_j^T y \leq c_j$

IP Dual Feasibility

- Given: $f(d) = f(d^A, d^l, d^u) = y^T d^A + w^T d^l + v^T d^u$
- Let
$$F(d) = F(d^A, d^l, d^u) = H(d^A) + \alpha^T d^l + \beta^T d^u$$

  where
$$H(d^A) = \sum_{i \in K} \lceil y_i (d^A)_i \rceil + \sum_{i \in M \setminus K} y_i (d^A)_i$$
- Let $r_j = c_j - H(A_j)\ \forall j$
- Set $\alpha_j = r_j$ if $r_j > 0$ and $\beta_j = -r_j$ if $r_j < 0\ \ \forall j$
- IP dual feasibility:
$$F(A_j, e_j, -e_j) = H(A_j) + r_j = H(A_j) + c_j - H(A_j) \leq c_j$$

Current Work

- Basic experiments show that we are able to tighten additional bounds
  - Ceiling operation for each row
  - 18% of IP instances are affected
- A non-traditional way of bound tightening that can be used in branch and bound tree
- We are looking for guiding the search and making these basic/fundamental subadditive functions stronger
- Extending to rays and row bounds
- Extending to mixed case

Thank you for your attention.

http://support.sas.com/or