Column generation stabilization: dual cuts

Operational Research Complements

Giovanni Righini

Università degli Studi di Milano

Based on: T. Gschwind, S. Irnich, *Dual inequalities for stabilized column generation revisited*, INFORMS Journal on Computing 28, 1 (2016) 175-194

A primal-dual master problem

Consider an extended formulation and its dual:

$$P)\ \ z_P = \min c^T \lambda \quad \text{s.t. } A\lambda = b,\ \ \lambda \geq 0 \qquad\qquad D)\ \ z_D = \max b^T \pi \quad \text{s.t. } A^T \pi \leq c$$

Pricing: find column a^j s.t. $c_j - \pi^T a^j < 0$.

Stabilized formulation

Dual cuts (Valério de Carvalho (2005), Ben Amor et al. (2006), Gschwind and Irnich (2016)): add valid cuts to the dual problem D, to avoid wild oscillations of variables π.

$$\begin{aligned} \tilde{P})\ \ z_P = \min\, & c^T\lambda + e^T y \\ \text{s.t. } & A\lambda + Ey = b \\ & \lambda \geq 0, y \geq 0 \end{aligned} \qquad\qquad \begin{aligned} \tilde{D})\ \ z_D = \max\, & b^T\pi \\ \text{s.t. } & A^T\pi \leq c \\ & E^T\pi \leq e \end{aligned}$$

$\tilde{D}$ is a restriction if D; $\tilde{P}$ is a relaxation of P.

Dual optimal inequalities

Inequalities $E^T \pi \leq e$ are called dual inequalitiees (DI).

Let D^* be the set of optimal dual solutions of D.

A dual inequality is a dual optimal inequality (DOI) if all $\pi \in D^*$ satisfy it.

A set of dual inequalities are deep dual optimal inequalities (DDOIs) if at least one solution $\pi \in D^*$ satisfies them.

Equivalence

The following statements are equivalent:

1. $\{E^T\pi \leq e\}$ are DDOIs.
2. $\exists\pi^* \in D^*$ which is feasible (and therefore optimal) also for $\tilde{D}$.
3. $z_D^* = z_{\tilde{D}}^*$.
4. $z_P^* = z_{\tilde{P}}^*$.
5. $\forall(\tilde{\lambda}, y)$ feasible for $\tilde{P}$, $\exists\lambda$ feasible for P, with $c^T\lambda \leq c^T\tilde{\lambda} + e^t y$.
6. $\exists(\lambda^*, y^*)$ optimal for $\tilde{P}$ with $Ey^* = 0$ and λ^* is optimal for P.
7. Every π^* optimal for $\tilde{D}$ is also optimal for D.

Exchange property

Consider

- a constraint matrix $A \geq 0$ with integer coefficients, row set I and column set J;
- an integer positive rhs vector b;
- a unit cost c for all columns.

Consider two column vectors $r \geq 0$ and $t \geq 0$.

A matrix $A \in \mathcal{Z}_+^{I \times J}$ has the (r, t) exchange property if

$$(a^j \in A) \wedge (a^j \geq r) \Rightarrow a^j - r + t \in A.$$

Every column $a^j \geq r$ of A can be transformed into another column of A, replacing r with t.

Remark. The exchange property is asymmetric: the (r, t) exchange property does not imply the (t, r) exchange property, in general.

Exchange property

If A has the (r, t) exchange property and (λ^*, π^*) is a pair of optimal solutions of P and D, then

$$\sum_{i \in I} r_i \pi_i^* \geq \sum_{i \in I} t_i \pi_i^*$$

or

$$\lambda_k^* = 0 \ \ \forall k \in J : a^k \geq r.$$

Exchange property and DOIs

Consider a column vector u^h such that $u_h = 1$ and $u_i = 0 \; \forall i \neq h$.

If A has the (r, t) exchange property when $r = u^h$, then we indicate this as the (h, t) exchange property: entries on row h can be feasibly replaced by entries in a subset of other rows.

If A has the (h, t) exchange property, then

$$\forall \pi^* \in D^*, \pi_h^* \geq \sum_{i \in I} t_i \pi_i^*.$$

This result can be used to obtain DOIs.

The cutting stock problem

The CSP: find cutting patterns for cutting rolls of given width L into items $i \in I$ of prescribed width $w_i \leq L$, such that the total number of rolls is minimized and given demands b_i $\forall i \in I$ are fulfilled.

Compact formulation.

$$
\begin{aligned}
\text{minimize } z = \sum_{v \in V} y_v & \\
\text{s.t.} \quad \sum_{i \in I} w_i a_{ij} \leq L y_v & \qquad \forall v \in V \\
\sum_{v \in V} a_{ij} = b & \qquad \forall i \in I \\
a \in \mathcal{Z}_+^{I \times V} & \\
y \in \mathcal{B}^V. &
\end{aligned}
$$

The cutting stock problem

In the extended formulation each column $j \in J$ corresponds to a cutting pattern containing a_{ij} items of type $i \in I$. Each column variable $\lambda \in \mathcal{Z}_+^J$ indicates how many patterns of type $j \in J$ must be cut.

Demand constraints $\rightarrow$ master; capacity constraints $\rightarrow$ pricing.

The pricing sub-problem is an integer knapsack problem.

$$\begin{aligned} \min z = & \sum_{j \in J} \lambda_j \\ \text{s.t.} & \sum_{j \in J} a_{ij} \lambda_j \geq b_i \quad \forall i \in I \\ & \lambda \in \mathcal{Z}_+^J \end{aligned}$$

$$\begin{aligned} \min \overline{c} = & 1 - \sum_{i \in I} a^i \pi_i \\ \text{s.t.} & \sum_{i \in I} w_i a^i \leq L \\ & a \in \mathcal{Z}_+^I \end{aligned}$$

The cutting stock problem

Assume that an instance of the CSP contains an item $h \in I$ and a subset $S \subseteq I \backslash \{h\}$ such that $w_h \geq \sum_{s \in S} w_s$. Then

$$\pi_h \geq \sum_{s \in S} \pi_s$$

is a DOI (subset inequality).

In general, given a coefficient vector $t \in \mathcal{Z}^S_{>0}$ such that $w_h \geq \sum_{s \in S} t_s w_s$,

$$\pi_h \geq \sum_{s \in S} t_s \pi_s$$

is a DOI (weighted subset inequality).

The cutting stock problem

The corresponding column in the primal master problem has

$$E_{ik} = \begin{cases} t_i & i \in S \\ -1 & i = h \\ 0 & \text{otherwise} \end{cases}$$

If it is more difficult to insert h than S is a feasible solution, then the reward for inserting h must be not smaller than the total reward for S.

The presence of the additional column accounts for the possible replacement of h with s in any column of the RMP: it implicitly corresponds to inserting in the RMP all columns that could be obtained from the columns already in the RMP by replacing h with S.

Covering property

If a matrix $A \in \mathcal{Z}_+^{I \times J}$ has the $(h, 0)$ exchange property for every row $h \in I$, then

1. $\forall \lambda : A\lambda \geq b, \lambda \geq 0, \; \exists \lambda' : A\lambda' = b, \lambda' \geq 0$ with $\sum_{j \in J} \lambda_j = \sum_{j \in J} \lambda'_j$;
2. $\pi^*_h \geq 0 \; \forall \pi^* \in D^*$.

When A has the covering property, positive entries on every row h can be decreased, still preserving feasibility.

- Cutting stock problem: deleting an item from a feasible pattern yields a feasible pattern;
- Bin packing problem: removing an item from a feasible bin subset yields a feasible bin subset;
- Vertex coloring problem: removing a vertex from an independent set yields an independent set.

This motivates the preference for set covering formulations of the master problem, instead of set partitioning.

Row interchange property

Consider a master problem where $A \in \mathcal{B}^{I \times J}$, $b_i = 1 \ \forall i \in I$, $c_j = 1 \ \forall j \in J$.

Given two rows $h \in I$ and $i \in I$, the matrix A has the (h, i) row interchange property if

$$(a^j \in A) \wedge (a_{hj} \geq 1) \wedge (a_{ij} = 0) \Rightarrow a^j - u^h + u^i \in A.$$

We say that (h, i) is a valid replacement for A.

The set of inequalities $\pi_i - \pi_h \leq 0$ for all valid replacements is a set of DDOIs.

- Bin packing problem: pairs of items with $w_h \geq w_i$;
- Vertex coloring problem: pairs of vertices with $N(h) \cup \{h\} \supseteq N(i)$ (where $N(i)$ indicates the neighborhood of any vertex i).

Constraint aggregation

In a BPP with identical items (i.e. such that $w_i = w_h$ for two items $i \neq h \in I$), the corresponding rows can be aggregated, summing up their entries for each column and in the rhs. The pricing problem becomes a Bounded Integer Knapsack Problem.

The same can happen in the Cutting Stock Problem and in the Vertex Coloring Problem.

The set of equalities $\pi_i = \pi_h \; \forall i, h \in I : w_i = w_h$ is a set of DDOIs for the Bin Packing Problem.

Constraint aggregation

Let $\alpha \in \Re$ and let $h \in I$ and $i \in I$ two row indices. Let a_i and a_h the two corresponding row vectors defining the equality constraints $a_i\lambda = b_i$ and $a_h\lambda = b_h$. Then

$$(a_h + \alpha a_i)\tilde{\lambda} = b_h + \alpha b_i \Leftrightarrow \begin{bmatrix} a_h \\ a_i \end{bmatrix} \tilde{\lambda} + \begin{bmatrix} \alpha \\ -1 \end{bmatrix} y = \begin{bmatrix} b_h \\ b_i \end{bmatrix}, y \in \Re$$

Aggregating the two constraints is the same as inserting a suitable dual cut $\alpha\pi_h = \pi_i$ in $\tilde{D}$ and the corresponding primal variable y in $\tilde{P}$.

Furthermore, consider an optimal dual solution $\pi^* \in D^*$ s.t. $\alpha\pi_h^* = \pi_i^*$. Then P and $\tilde{P}$ are equivalent to an aggregated formulation $\tilde{P}'$ where rows i and h are replaced by the aggregated constraint $(a_h + \alpha a_i)\tilde{\lambda}$. The dual solution $\pi^{*'}$ defined by

$$\pi_k^{*'} = \begin{cases} \pi_k^* & k \neq h, i \\ \pi_h^* & \text{for the aggregated row} \end{cases}$$

is an optimal solution of the dual of $\tilde{P}'$.

Constraint elimination

The special case $\alpha = 0$ leads to constraint elimination.

- Set Covering Problem: if $a_{ij} \geq a_{hj}\ \forall j \in J$, then row i is redundant;
- Vertex Coloring Problem: if $N(h) \supseteq N(i)$, then row i is redundant.

The equalities $\pi_i = 0$ for all redundant rows are DDOIs.

Separation

(Weighted) subset inequalities can be exponentially many. Instead of generating only some of them a priori, it may be profitable to dynamically generate them as needed, as in cutting planes algorithms.

For each problem a tailored separation algorithm must be devised.

In general, dynamic programming works fine for problems resembling variations of the Knapsack Problem.

In some cases it may be easier to generate violated dual inequalities that are not DOIs or DDOIs.

This overstabilization in general leads to primal infeasibility, triggering the need for repairing the solution.

Overstabilization: an example

Bin Packing Problem. $I = \{1, \dots, 4\}$, $w = [5, 2, 2, 2]$, $L = 10$.

The MP contains the four columns $\begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}$.

Dual constraints:

$$\begin{cases} \pi_1 + \pi_2 + \pi_3 \leq 1 \\ \pi_1 + \pi_2 + \pi_4 \leq 1 \\ \pi_1 + \pi_3 + \pi_4 \leq 1 \\ \pi_2 + \pi_3 + \pi_4 \leq 1 \end{cases}$$

Optimal solutions: $\lambda^* = \left[\frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right]$ and $\pi^* = \left[\frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right]$, $z^* = \frac{4}{3}$.

Overstabilization: an example

Stabilization. Dual inequality $\pi_1 \geq \pi_3 + \pi_4$ (not DDOI).

New optimal dual solution: $\pi^* = \left[\frac{1}{2}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\right]$ and $z^* = \frac{5}{4}$.

Corresponding column in the MP: $\begin{bmatrix} -1 \\ 0 \\ 1 \\ 1 \end{bmatrix}$.

New optimal primal solution: $(\tilde{\lambda}^*, y^*) = \left(\left[\frac{1}{2}, \frac{1}{2}, \frac{1}{4}, 0\right], \frac{1}{4}\right)$, which is infeasible.

To make it feasible, one should replace item 1 with items 3 and 4 in one of the basic columns, which is not possible because at least one of items 3 and 4 would occur twice.

Notation

$(h \leftarrow t, S)$ indicates the WSI $\pi_h \geq \sum_{s \in S} t_s \pi_s$.

K: set of column indices of E (coefficients of y variables in $\tilde{P}$).

The rows of E^T in $\tilde{D}$ are the coefficients in the WSIs. The correspondence between a WSI $(h \leftarrow t, S)$ and its index $k \in K$ is indicated by $k(h \leftarrow t, S)$.

Two columns $j \in J$ and $k \in K$ are WSI-compatible if $a^j - u_h + t \in A$ for $h \in I$ defined by $k = k(h \leftarrow t, S)$.

Recovery algorithm

repeat
  $(\lambda^*, y^*) \leftarrow$ Solve *LRMP*
  while $\exists (j,k) \in J \times K : \lambda^*_j > 0, y^*_k > 0$ and j and k are a compatible WSI pair **do**
    $(j,k) \leftarrow$ *SelectWSIpair*
    $\delta \leftarrow \min\{\lambda^*_j, y^*_k\}$
    Find $j' \in J : a^{j'} = a^j - u_h + t$
    $\lambda^*_j \leftarrow \lambda^*_j - \delta$; $y^*_k \leftarrow y^*_k - \delta$; $\lambda^*_{j'} \leftarrow \lambda^*_{j'} + \delta$
  end while
  if $y^* > 0$ **then**
    Select $h \in I : y^*_k > 0$ for h defined by $k = k(h \leftarrow t, S)$
    Eliminate all WSIs $(h \leftarrow t', S')$ for $S' \subseteq I$, $t' \in \mathcal{Z}^I_{>0}$ from the RMP and forbid their re-generation.
  end if
until $y^* = 0$
return λ^*

Recovery algorithm

By definition of WSI compatible pair, δ is strictly positive.

The assigments $\lambda_j^* \leftarrow \lambda_j^* - \delta$, $y_k^* \leftarrow y_k^* - \delta$ and $\lambda_{j'}^* \leftarrow \lambda_{j'}^* + \delta$ do not affect primal feasibility of $A\tilde{\lambda} + Ey = b$, since $a^{j'} = a^j - u_h + t$, $e_{hk} = -1$ and $e_{sk} = 1\ \forall s \in S$.

They do not affect the value of $z_{\tilde{P}}$ because $c_j = c_{j'} = 1$.

The inner loop succeeds when a feasible primal solution with $y^* = 0$ is found. Otherwise, there must exist an active WSI $(h \leftarrow t, S)$ for which no compatible $j \in J$ exists.

When this happens, all WSIs involving row h are deleted and the master problem is reoptimized.

After at most $|I|$ iterations of the outer loop, the recovery algorithm terminates with an optimal solution λ^*.