Aggregating Dual Variables in Column Generation: Applications for Bounded Vertex Coloring

Daniel Porumbel[*], François Clautiaux[**]

[*]Université d'Artois, France
[**]Université de Lille 1, France

The Optimal Dual Values in Column Generation

Recurent Properties of Optimal Dual Values:

- Same dual values for set-covering primal constraints of
  - items of same weight in `Cutting-Stock`
  - vertices with the same neighbor set in `Graph Coloring`
- Cutting-Stock: the Dual Feasible Functions [Lueker, 1983] provide a good estimation of the dual optimum values.

Set-Covering Column Generation: the Primal

Goal: minimize the number of selected patterns

$$\min \sum_{a \in cols} x_a$$

cols: Column Set

$$\sum_{a \in cols} a_i x_a \geq b_i \ , \ \forall i \in [1..n]$$

$$x \in \mathbb{Z}^{|cols|}$$

Set-Covering Constraints Dualized to dual vector y

Set-Covering Column Generation: the Dual

$$\max b^\top y$$

$$\left.\begin{array}{ll} a^\top y \leq 1, & \forall a \in cols \\ y_i \geq 0, & i \in [1..n] \end{array}\right\} \mathbf{P}$$

Valid $a = [a_1 \; a_2 \ldots a_n]^\top$ for `Pure Cutting-Stock`

a_i: item i selected a_i times

Capacity constraint: $\sum w_i a_i \leq C, \forall a \in cols$

Set-Covering Column Generation: the Dual

$$\max b^\top y$$
$$\left.\begin{array}{ll} a^\top y \leq 1, & \forall a \in cols \\ y_i \geq 0, & i \in [1..n] \end{array}\right\} \mathbf{P}$$

Valid $a = [a_1\ a_2 \ldots a_n]^\top$ for `Pure Cutting-Stock`

a_i: item i selected a_i times
Capacity constraint: $\sum w_i a_i \leq C$, $\forall a \in cols$

Set-Covering Column Generation: the Dual

$$\max b^\top y$$
$$\left.\begin{array}{ll} a^\top y \leq 1, & \forall a \in cols \\ y_i \geq 0, & i \in [1..n] \end{array}\right\} \mathbf{P}$$

Bounded Vertex Coloring or Cutting-Stock with conflicts

a_i: item i selected a_i times

Capacity constraint: $\sum w_i a_i \leq C$

new: $a_i, a_j > 0 \implies \boxed{i}, \boxed{j}$ unlinked vertices in a conflict graph

Set-Covering Column Generation: the Dual

$$
\left.\begin{array}{ll}
\max b^\top y & \\
a^\top y \leq 1, & \forall a \in cols \\
y_i \geq 0, & i \in [1..n]
\end{array}\right\} \mathbf{P}
$$

Applications

Cutting-Stock

- cutting rolls of metal/paper into smaller units
- fitting existing items into bins (bin-packing)

Cutting Stock with Conflicts

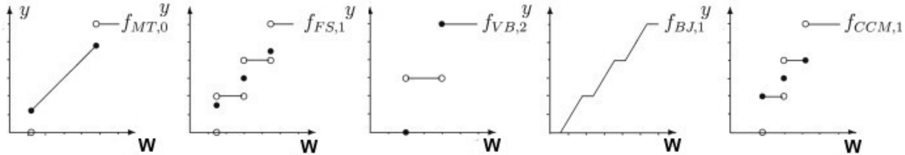
- frequency assignment with bounded capacities
- a pattern should contains items of different types

Rationale: High-Quality Duals in Pure Cut-Stock

$$y_i \leftarrow f(w_i), \quad \forall i \in [1..n]$$

- f =Dual Feasible Function $\implies$ $\mathbf{y}$ feasible in $\mathbf{P}$
    - DFF property: $\sum w_i \leq C \implies \sum f(w_i) \leq 1$

      [Clautiaux et al, A survey of dual-feasible functions for bin packing problems, AOR, 2008], [Rietz et al, Theoretical investigations on maximal dual feasible functions OR Letters]



$\implies$ Even optimal dual vectors can be constructed for Classical Cutting-Stock

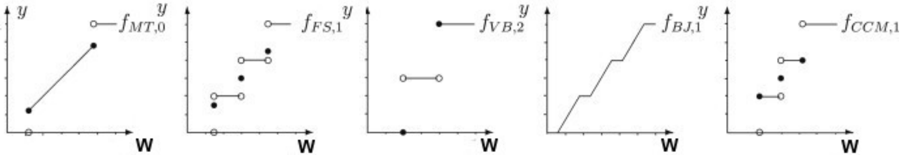
- DFFs are very regular functions and easy to calculate

Rationale: High-Quality Duals in Pure Cut-Stock

$$y_i \leftarrow f(w_i), \quad \forall i \in [1..n]$$

- f =Dual Feasible Function $\implies$ **y** feasible in **P**
    - DFF property: $\sum w_i \leq C \implies \sum f(w_i) \leq 1$

      [Clautiaux et al, A survey of dual-feasible functions for bin packing problems, AOR, 2008], [Rietz et al, Theoretical investigations on maximal dual feasible functions OR Letters]



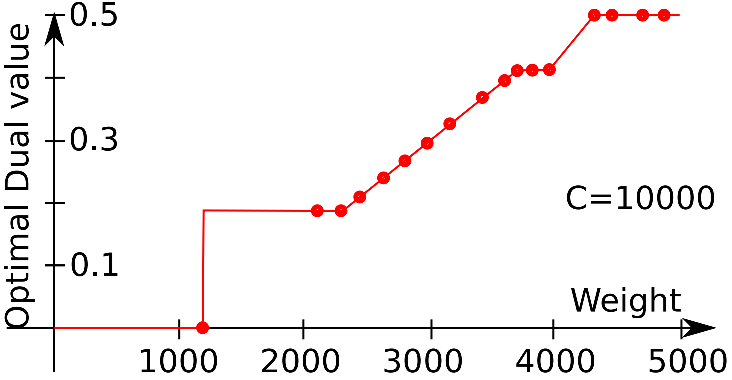
$\implies$ Even optimal dual vectors can be constructed for Classical Cutting-Stock

- DFFs are very regular functions and easy to calculate

Rationale: The Optimal Dual Solution

A more complex solution: `CSTR20b50c2p3.dat` (C=10000) from

[Vanderbeck. Computational Study of a Column Generation algorithm for Bin Packing and Cutting Stock. Math Prog. 1999]



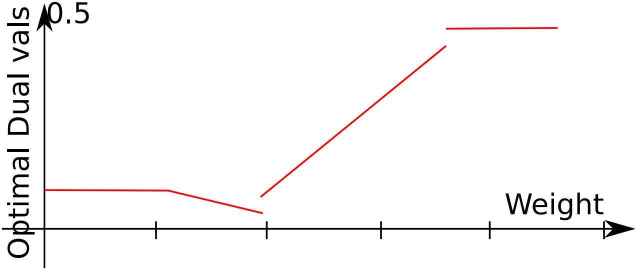
$\boxed{\text{Goal: Algorithmic Construction}}$ of groupwise linear functions

Optimal Duals: CutStock with $\boxed{\text{with Conflicts}}$

The optimal duals do not verify classical Cutting-Stock relations:

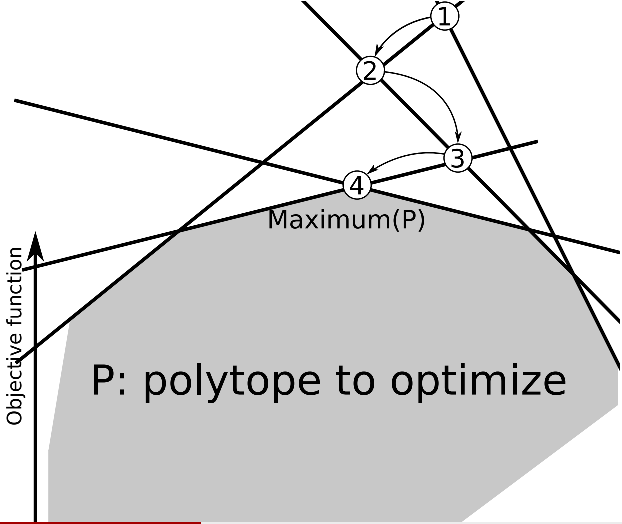
$$y_i \le y_j, \text{ if } w_i \le w_j$$

$$y_i + y_j \le y_k, \text{ if } w_i + w_j \le w_k$$



- Classical DFF lower bounds do not work
  - Optimal Solutions can even be non-monotone
- $\boxed{\text{Group-wise Linear Duals}}$ can provide feasible solutions

Classical Column Generation: convergent upper bounds



Column generation reaches the **P** optimum ④ via: ① → ② → ③ → ④

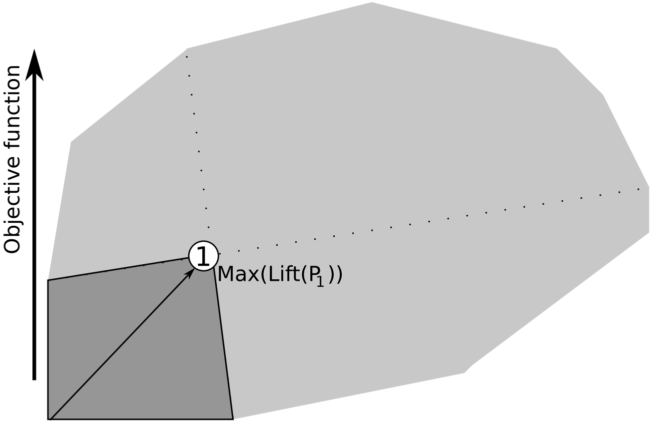
However, ④ is a lower bound for the integer optimum

- ①, ② and ③ are upper bounds for a lower bound

Convergent Lower Bounds by $\boxed{\text{Aggregation}}$

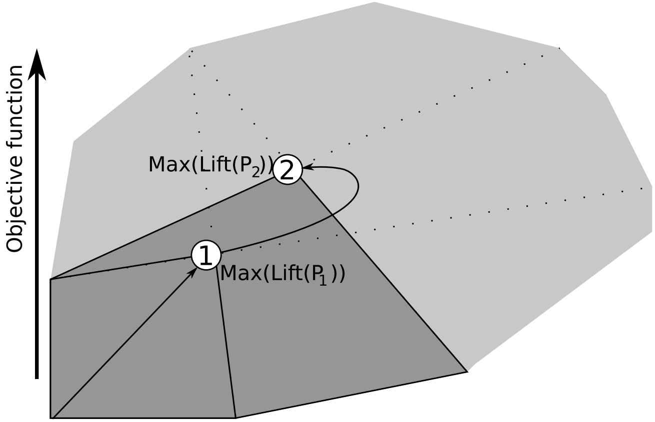
Simplest Dual Aggregation $y_i = \alpha, \forall i \in [1..n] \rightarrow$ Polytope $\mathbf{P}_1$

- opt($\mathbf{P}_1$)$\leq$opt($\mathbf{P}$)



Convergent Lower Bounds by $\boxed{\text{Aggregation}}$

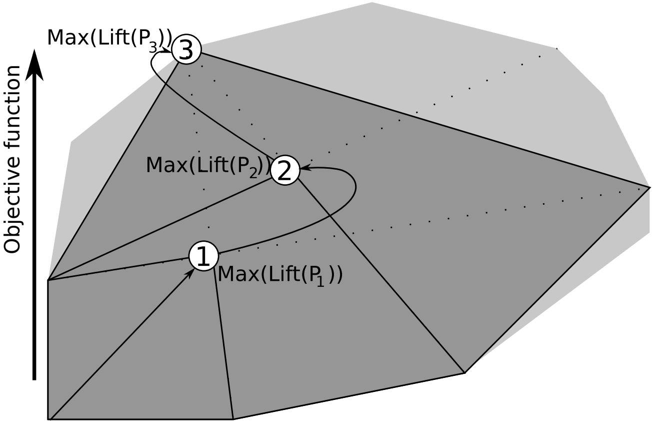
Dual aggregation with two groups: $y_i = \begin{cases} \alpha^1 & \text{if } i \in [1..n_1] \\ \alpha^2 & \text{if } i \in [n_1+1..n] \end{cases}$



Convergent Lower Bounds by $\boxed{\text{Aggregation}}$

A series of increasing $\boxed{\text{lower bounds}}$ is produced:

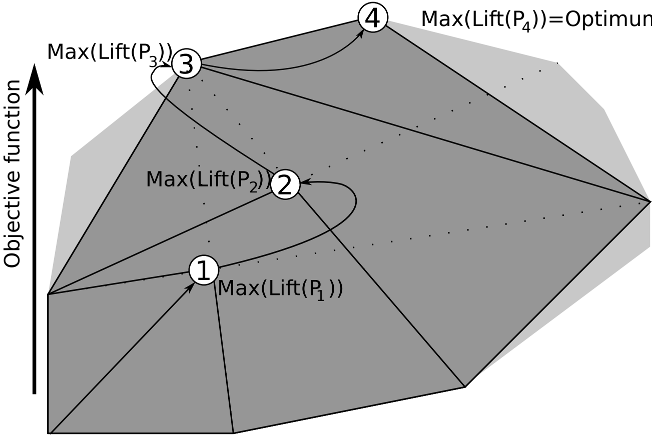
- opt($\mathbf{P}_1$)$\leq$ opt($\mathbf{P}_2$)$\leq$ opt($\mathbf{P}_3$)$\leq \cdots \leq$ opt($\mathbf{P}$)



Convergent Lower Bounds by $\boxed{\text{Aggregation}}$

Progressive refinement of the aggregation $\rightarrow$ **P** optimum

- The optimum is reached when $k = n$ or even bevore



An Illustration: Re-Writing $\mathbf{P}$ using $y_i = \alpha, \forall i$

Transforming $\mathbf{P} \to \mathbf{P}_1$ via aggregation

$$\left.\begin{array}{ll} \multicolumn{2}{c}{\max b^T y} \\ a^T y \leq 1, & \forall a \in cols \\ y_i \geq 0, & i \in [1..n] \end{array}\right\} \mathbf{P} \implies \left.\begin{array}{ll} \multicolumn{2}{c}{\max \left(\sum b_i\right) \cdot \alpha} \\ \left(\sum a_i\right) \cdot \alpha \leq 1, & \forall a \in cols \\ \alpha \geq 0, & i \in [1..n] \end{array}\right\} \mathbf{P}_1$$

One variable (α) is sufficient to describe $\mathbf{P}_1$:

- the objective function becomes $\sum b_i y_i = \sum b_i \alpha = \left(\sum b_i\right) \cdot \alpha$
- $a^T y \leq 1$ is $\sum a_i y_i \leq 1$, or $\left(\sum a_i\right) \cdot \alpha \leq 1$
- $y_i \geq 0$ becomes $\alpha \geq 0$

An Illustration: Re-Writing $\mathbf{P}$ using $y_i = \alpha, \forall i$

Transforming $\mathbf{P} \to \mathbf{P}_1$ via aggregation

$$\left.\begin{array}{ll} & \max b^T y \\ a^T y \leq 1, & \forall a \in cols \\ y_i \geq 0, & i \in [1..n] \end{array}\right\} \mathbf{P} \quad \Longrightarrow \quad \left.\begin{array}{ll} & \max \left(\sum b_i\right) \cdot \alpha \\ \left(\sum a_i\right) \cdot \alpha \leq 1, & \forall a \in cols \\ \alpha \geq 0, & i \in [1..n] \end{array}\right\} \mathbf{P}_1$$

One variable (α) is sufficient to describe $\mathbf{P}_1$:

- the objective function becomes $\sum b_i y_i = \sum b_i \alpha = \left(\sum b_i\right) \cdot \alpha$
- $a^T y \leq 1$ is $\sum a_i y_i \leq 1$, or $\left(\sum a_i\right) \cdot \alpha \leq 1$
- $y_i \geq 0$ becomes $\alpha \geq 0$

Optimizing $\mathbf{P}_1$ by Column Generation

Transforming $\mathbf{P} \to \mathbf{P}_1$ via aggregation

$$\left.\begin{array}{ll} & \max b^T y \\ a^T y \leq 1, & \forall a \in cols \\ y_i \geq 0, & i \in [1..n] \end{array}\right\} \mathbf{P} \quad \Longrightarrow \quad \left.\begin{array}{ll} & \max \left(\sum b_i\right) \cdot \alpha \\ \left(\sum a_i\right) \cdot \alpha \leq 1, & \forall a \in cols \\ \alpha \geq 0, & i \in [1..n] \end{array}\right\} \mathbf{P}_1$$

1. One variable α
2. Fewer constraints: aggreagte all $a \in cols$ with equal $\sum a_i$
3. Easier sub-problem: min red cost $= 1 - \max_{a \in cols} \sum a_i y_i$
   - Classical **Cutting-Stock**:
     $$max_{a \in cols}\left(\sum a_i\right) = \lfloor \frac{C}{w_{\min}} \rfloor$$
   - **Cutting-Stock with Conflicts**:
     `Maximum Stable instead of Maximum` $y-$`Weighted Stable`
     - Stable respecting capacity constraint
     - If graph=disjoint components $\implies$ multiple-choice knapsack

Optimizing $\mathbf{P}_1$ by Column Generation

Transforming $\mathbf{P} \rightarrow \mathbf{P}_1$ via aggregation

$$\left.\begin{array}{ll} \multicolumn{2}{c}{\max b^T y} \\ a^T y \leq 1, & \forall a \in cols \\ y_i \geq 0, & i \in [1..n] \end{array}\right\} \mathbf{P} \Longrightarrow \left.\begin{array}{ll} \multicolumn{2}{c}{\max \left(\sum b_i\right) \cdot \alpha} \\ \left(\sum a_i\right) \cdot \alpha \leq 1, & \forall a \in cols \\ \alpha \geq 0, & i \in [1..n] \end{array}\right\} \mathbf{P}_1$$

1. One variable α
2. Fewer constraints: aggreagte all $a \in cols$ with equal $\sum a_i$
3. Easier sub-problem: min red cost $= 1 - \max_{a \in cols} \sum a_i y_i$
   - Classical **Cutting-Stock**:
     $$max_{a \in cols} \left(\sum a_i\right) = \lfloor \frac{C}{w_{\min}} \rfloor$$
   - **Cutting-Stock with Conflicts**:
     `Maximum Stable` instead of `Maximum` $y-$`Weighted Stable`
     - Stable respecting capacity constraint
     - If graph=disjoint components $\Longrightarrow$ multiple-choice knapsack

The Group-Wise Linear Aggregation

Consider a partition of y into groups $y^1, y^2 \ldots y^k$. Given group j:

- y^j is vector $[y^j_1 \; y^j_2 \; \ldots y^j_{n_j}]^T$ of size n_j
- α^j, β^j: slope and y-intercept, new decision variables
- $w^j, b^j, a^j, \ldots$ coefficients associated to group j

The Aggregation Formula

- $y^j_i = w^j_i \cdot \alpha^j + \beta^j$

OR

- $y^j = w^j \alpha^j + \mathbf{1}_{n_j} \beta^j$

The Group-Wise Linear Aggregation

Consider a partition of y into groups $y^1, y^2 \dots y^k$. Given group j:

- y^j is vector $[y^j_1 \ y^j_2 \ \dots y^j_{n_j}]^T$ of size n_j
- α^j, β^j: slope and y-intercept, new decision variables
- $w^j, b^j, a^j, \dots$ coefficients associated to group j

Re-writing the objective function: replace y^j by $\left(w^j\alpha^j + \mathbf{1}_{n_j}\beta^j\right)$

$$b^T y = \sum_{j=1}^{k} \left(b^j\right)^T y^j = \sum_{j=1}^{k} \left(b^j\right)^T \left(w^j\alpha^j + \mathbf{1}_{n_j}\beta^j\right)$$

$$= \sum_{i=1}^{k} \left(\left(b^j\right)^T w^j\right) \cdot \alpha^j + \left(\left(b^j\right)^T \mathbf{1}_{n_j}\right) \cdot \beta^j$$

Inspecting constraint $\sum_{j=1}^{k}\left(a^j\right)^T y^j \leq 1$

Re-writing the constraints: replace y^j by $\left(w^j\alpha^j + \mathbf{1}_{n_j}\beta^j\right)$

$$\left(a^j\right)^T y^j = \left(a^j\right)^T \left(w^j\alpha^j + \mathbf{1}_{n_j}\beta^j\right) = \left(\left(a^j\right)^T w^j\right)\cdot\alpha^j + \left(\left(a^j\right)^T \mathbf{1}_{n_j}\right)\cdot\beta^j$$

$$= c^j\cdot\alpha^j + \boxed{\left(\left(a^j\right)^T \mathbf{1}_{n_j}\right)}\cdot\beta^j$$

Constraint aggregation at fixed total weight $c^j = \left(a^j\right)^T w^j$ in group j

$\boxed{\left(\left(a^j\right)^T \mathbf{1}_{n_j}\right)}$: calculate min $m(j, c^j)$, max $M(j, c^j)$

$\Rightarrow$ Given $c^1, c^2, \ldots c^k$, term j in $\sum_{j=1}^{k}\left(a^j\right)^T y^j \leq 1$ is between

$$c^j\cdot\alpha^j + \boxed{M(j, c^j)}\cdot\beta^j \text{ and } c^j\cdot\alpha^j + \boxed{m(j, c^j)}\cdot\beta^j$$

$\Rightarrow$ Number of constraints not depending on the number of items!

Inspecting constraint $\sum_{j=1}^{k} \left(a^j\right)^T y^j \leq 1$

Re-writing the constraints: replace y^j by $\left(w^j \alpha^j + \mathbf{1}_{n_j} \beta^j\right)$

$$\left(a^j\right)^T y^j = \left(a^j\right)^T \left(w^j \alpha^j + \mathbf{1}_{n_j} \beta^j\right) = \left(\left(a^j\right)^T w^j\right) \cdot \alpha^j + \left(\left(a^j\right)^T \mathbf{1}_{n_j}\right) \cdot \beta^j$$

$$= c^j \cdot \alpha^j + \boxed{\left(\left(a^j\right)^T \mathbf{1}_{n_j}\right)} \cdot \beta^j$$

Constraint aggregation at fixed total weight $c^j = \left(a^j\right)^T w^j$ in group j

$\boxed{\left(\left(a^j\right)^T \mathbf{1}_{n_j}\right)}$: calculate min $m(j, c^j)$, max $M(j, c^j)$

$\Rightarrow$ Given $c^1, c^2, \ldots c^k$, term j in $\sum_{j=1}^{k} \left(a^j\right)^T y^j \leq 1$ is between

$$c^j \cdot \alpha^j + \boxed{M(j, c^j)} \cdot \beta^j \text{ and } c^j \cdot \alpha^j + \boxed{m(j, c^j)} \cdot \beta^j$$

$\Rightarrow$ Number of constraints not depending on the number of items!

Inspecting constraint $\sum_{j=1}^{k} \left(a^j\right)^T y^j \leq 1$

Re-writing the constraints: replace y^j by $\left(w^j\alpha^j + \mathbf{1}_{n_j}\beta^j\right)$

$$\left(a^j\right)^T y^j = \left(a^j\right)^T \left(w^j\alpha^j + \mathbf{1}_{n_j}\beta^j\right) = \left(\left(a^j\right)^T w^j\right)\cdot\alpha^j + \left(\left(a^j\right)^T \mathbf{1}_{n_j}\right)\cdot\beta^j$$

$$= c^j \cdot \alpha^j + \boxed{\left(\left(a^j\right)^T \mathbf{1}_{n_j}\right)} \cdot \beta^j$$

Constraint aggregation at fixed total weight $c^j = \left(a^j\right)^T w^j$ in group j

$\boxed{\left(\left(a^j\right)^T \mathbf{1}_{n_j}\right)}$: calculate min $m(j, c^j)$, max $M(j, c^j)$

$\Rightarrow$ Given $c^1, c^2, \ldots c^k$, term j in $\sum_{j=1}^{k} \left(a^j\right)^T y^j \leq 1$ is between

$$c^j \cdot \alpha^j + \boxed{M(j, c^j)} \cdot \beta^j \text{ and } c^j \cdot \alpha^j + \boxed{m(j, c^j)} \cdot \beta^j$$

$\Rightarrow$ Number of constraints not depending on the number of items!

Inspecting constraint $\sum_{j=1}^{k} \left(a^j\right)^T y^j \leq 1$

Re-writing the constraints: replace y^j by $\left(w^j \alpha^j + \mathbf{1}_{n_j} \beta^j\right)$

$$\left(a^j\right)^T y^j = \left(a^j\right)^T \left(w^j \alpha^j + \mathbf{1}_{n_j} \beta^j\right) = \left(\left(a^j\right)^T w^j\right) \cdot \alpha^j + \left(\left(a^j\right)^T \mathbf{1}_{n_j}\right) \cdot \beta^j$$

$$= c^j \cdot \alpha^j + \boxed{\left(\left(a^j\right)^T \mathbf{1}_{n_j}\right)} \cdot \beta^j$$

Constraint aggregation at fixed total weight $c^j = \left(a^j\right)^T w^j$ in group j

$\boxed{\left(\left(a^j\right)^T \mathbf{1}_{n_j}\right)}$: calculate min $m(j, c^j)$, max $M(j, c^j)$

$\Rightarrow$ Given $c^1, c^2, \ldots c^k$, term j in $\sum_{j=1}^{k} \left(a^j\right)^T y^j \leq 1$ is between

$$c^j \cdot \alpha^j + \boxed{M(j, c^j)} \cdot \beta^j \text{ and } c^j \cdot \alpha^j + \boxed{m(j, c^j)} \cdot \beta^j$$

$\Rightarrow$ Number of constraints not depending on the number of items!

Ex. $C = 100$, $w^1 = [10\ 20\ 30\ 41]$, $b^1 = [4\ 1\ 1\ 1]$

$\boxed{y_i = \alpha^1 w_i + \beta^1}$ applied on pattern [3 1 0 1]

$$3y_1 + y_2 + y_4 \leq 1 \rightarrow 3(10\alpha^1 + \beta^1) + (20\alpha^1 + \beta^1) + (41\alpha^1 + \beta^1) \leq 1$$
$$\rightarrow 91\alpha^1 + 5\beta^1 \leq 1$$

Other patterns of total weight 91:

pattern	[3 1 0 1]	[2 0 1 1]	[0 1 1 1]
weight	$3 \cdot 10 + 20 + 41$	$2 \cdot 10 + 30 + 41$	$20 + 30 + 31$
constraint	$91\alpha^1 + 5\beta^1 \leq 1$	$91\alpha^1 + 4\beta^1 \leq 1$	$91\alpha^1 + 3\beta^1 \leq 1$

- $M(1, 91) = 5$, $m(1, 91) = 3 \Rightarrow$
  - $91\alpha^1 + 5\beta^1 \leq 1$ and $91\alpha^1 + 3\beta^1 \leq 1$ are enough

Add a second group $w^2 = [88\ 89\ 90]$ and $b^2 = [1\ 1\ 1]$:

- $M(2, 88) = M(2, 89) = M(2, 90) = 1$
- The strongest constraint involving both groups is: $10\alpha^1 + \beta^1 + 90\alpha^2 + \beta^2 \leq 1$

Ex. $C = 100$, $w^1 = [10\ 20\ 30\ 41]$, $b^1 = [4\ 1\ 1\ 1]$

$\boxed{y_i = \alpha^1 w_i + \beta^1}$ applied on pattern [3 1 0 1]

$$3y_1 + y_2 + y_4 \le 1 \rightarrow 3(10\alpha^1 + \beta^1) + (20\alpha^1 + \beta^1) + (41\alpha^1 + \beta^1) \le 1$$
$$\rightarrow 91\alpha^1 + 5\beta^1 \le 1$$

Other patterns of total weight 91:

pattern	[3 1 0 1]	[2 0 1 1]	[0 1 1 1]
weight	$3 \cdot 10 + 20 + 41$	$2 \cdot 10 + 30 + 41$	$20 + 30 + 31$
constraint	$91\alpha^1 + 5\beta^1 \le 1$	$91\alpha^1 + 4\beta^1 \le 1$	$91\alpha^1 + 3\beta^1 \le 1$

- $M(1, 91) = 5$, $m(1, 91) = 3 \Rightarrow$
  - $91\alpha^1 + 5\beta^1 \le 1$ and $91\alpha^1 + 3\beta^1 \le 1$ are enough

Add a second group $w^2 = [88\ 89\ 90]$ and $b^2 = [1\ 1\ 1]$:

- $M(2, 88) = M(2, 89) = M(2, 90) = 1$
- The strongest constraint involving both groups is:
  $10\alpha^1 + \beta^1 + 90\alpha^2 + \beta^2 \le 1$

Ex. $C = 100$, $w^1 = [10\ 20\ 30\ 41]$, $b^1 = [4\ 1\ 1\ 1]$

$\boxed{y_i = \alpha^1 w_i + \beta^1}$ applied on pattern [3 1 0 1]

$$3y_1 + y_2 + y_4 \leq 1 \rightarrow 3(10\alpha^1 + \beta^1) + (20\alpha^1 + \beta^1) + (41\alpha^1 + \beta^1) \leq 1$$
$$\rightarrow 91\alpha^1 + 5\beta^1 \leq 1$$

Other patterns of total weight 91:

pattern	[3 1 0 1]	[2 0 1 1]	[0 1 1 1]
weight	$3 \cdot 10 + 20 + 41$	$2 \cdot 10 + 30 + 41$	$20 + 30 + 31$
constraint	$91\alpha^1 + 5\beta^1 \leq 1$	$91\alpha^1 + 4\beta^1 \leq 1$	$91\alpha^1 + 3\beta^1 \leq 1$

- $M(1, 91) = 5$, $m(1, 91) = 3 \Rightarrow$
  - $91\alpha^1 + 5\beta^1 \leq 1$ and $91\alpha^1 + 3\beta^1 \leq 1$ are enough

Add a second group $w^2 = [88\ 89\ 90]$ and $b^2 = [1\ 1\ 1]$:

- $M(2, 88) = M(2, 89) = M(2, 90) = 1$
- The strongest constraint involving both groups is: $10\alpha^1 + \beta^1 + 90\alpha^2 + \beta^2 \leq 1$

Solving $w = [10\ 20\ 30\ 41\ 88\ 89\ 90], b = [4\ 1 \ldots 1]$

Take solution below corresponding to $y = [\frac{10}{91}\ \frac{20}{91}\ \frac{30}{91}\ \frac{41}{91}\ \frac{81}{91}\ \frac{81}{91}\ \frac{81}{91}]$;
objective value: $\frac{4\cdot 10+20+30+41+3\cdot 81}{91} = \frac{131+243}{91} = 4 + \frac{10}{91}$

- $\alpha^1 = \frac{1}{91}$, $\beta^1 = 0$
- $\alpha^2 = 0$, $\beta^2 = \frac{81}{91}$

Does this solution violates any constraint of $\mathbf{P}_2$? Non, let us find tightest $\boxed{c^1\alpha^1 + M(1,c^1)\beta^1 + c^2\alpha^2 + M(2,c^2)\beta^2}$ constraints:

- $c^2 = 0 \Rightarrow$ maximum $c^1 = 91 \Rightarrow \boxed{91\alpha^1 + 5\beta^1 \le 1}$
  - no use tyring lower c^1 for $c^2 = 0$
- $c^2 > 0 \Rightarrow c^2 \in \{88, 89, 90\} \Rightarrow$ maximum $c^1 = 10 \Rightarrow$ tightest constraint is $\boxed{10\alpha^1 + \beta^1 + 90\alpha^2 + \beta^2 \le 1}$

Conclusion: the tightest constraints can be found by solving a multiple-choice knapsack problem with k levels. For each level j, there are as many choices as realisable values c^j.

Solving $w = [10\ 20\ 30\ 41\ 88\ 89\ 90], b = [4\ 1 \ldots 1]$

Take solution below corresponding to $y = [\frac{10}{91}\ \frac{20}{91}\ \frac{30}{91}\ \frac{41}{91}\ \frac{81}{91}\ \frac{81}{91}\ \frac{81}{91}]$; objective value: $\frac{4\cdot 10+20+30+41+3\cdot 81}{91} = \frac{131+243}{91} = 4 + \frac{10}{91}$

- $\alpha^1 = \frac{1}{91}$, $\beta^1 = 0$
- $\alpha^2 = 0$, $\beta^2 = \frac{81}{91}$

Does this solution violates any constraint of $\mathbf{P}_2$? Non, let us find tightest $\boxed{c^1\alpha^1 + M(1,c^1)\beta^1 + c^2\alpha^2 + M(2,c^2)\beta^2}$ constraints:

- $c^2 = 0 \Rightarrow$ maximum $c^1 = 91 \Rightarrow \boxed{91\alpha^1 + 5\beta^1 \leq 1}$
  - no use tyring lower c^1 for $c^2 = 0$
- $c^2 > 0 \Rightarrow c^2 \in \{88, 89, 90\} \Rightarrow$ maximum $c^1 = 10 \Rightarrow$ tightest constraint is $\boxed{10\alpha^1 + \beta^1 + 90\alpha^2 + \beta^2 \leq 1}$

Conclusion: the tightest constraints can be found by solving a multiple-choice knapsack problem with k levels. For each level j, there are as many choices as realisable values c^j.

Solving $w = [10\ 20\ 30\ 41\ 88\ 89\ 90], b = [4\ 1 \dots 1]$

Take solution below corresponding to $y = [\frac{10}{91}\ \frac{20}{91}\ \frac{30}{91}\ \frac{41}{91}\ \frac{81}{91}\ \frac{81}{91}\ \frac{81}{91}]$; objective value: $\frac{4 \cdot 10 + 20 + 30 + 41 + 3 \cdot 81}{91} = \frac{131 + 243}{91} = 4 + \frac{10}{91}$

- $\alpha^1 = \frac{1}{91}$, $\beta^1 = 0$
- $\alpha^2 = 0$, $\beta^2 = \frac{81}{91}$

Does this solution violates any constraint of $\mathbf{P}_2$? Non, let us find tightest $\boxed{c^1\alpha^1 + M(1, c^1)\beta^1 + c^2\alpha^2 + M(2, c^2)\beta^2}$ constraints:

- $c^2 = 0 \Rightarrow$ maximum $c^1 = 91 \Rightarrow \boxed{91\alpha^1 + 5\beta^1 \leq 1}$
  - no use tyring lower c^1 for $c^2 = 0$
- $c^2 > 0 \Rightarrow c^2 \in \{88, 89, 90\} \Rightarrow$ maximum $c^1 = 10 \Rightarrow$ tightest constraint is $\boxed{10\alpha^1 + \beta^1 + 90\alpha^2 + \beta^2 \leq 1}$

Conclusion: the tightest constraints can be found by solving a multiple-choice knapsack problem with k levels. For each level j, there are as many choices as realisable values c^j.

Solving $w = [10\ 20\ 30\ 41\ 88\ 89\ 90], b = [4\ 1 \dots 1]$

Take solution below corresponding to $y = [\frac{10}{91}\ \frac{20}{91}\ \frac{30}{91}\ \frac{41}{91}\ \frac{81}{91}\ \frac{81}{91}\ \frac{81}{91}]$; objective value: $\frac{4\cdot 10+20+30+41+3\cdot 81}{91} = \frac{131+243}{91} = 4 + \frac{10}{91}$

- $\alpha^1 = \frac{1}{91}$, $\beta^1 = 0$
- $\alpha^2 = 0$, $\beta^2 = \frac{81}{91}$

Does this solution violates any constraint of $\mathbf{P}_2$? Non, let us find tightest $\boxed{c^1\alpha^1 + M(1,c^1)\beta^1 + c^2\alpha^2 + M(2,c^2)\beta^2}$ constraints:

- $c^2 = 0 \Rightarrow$ maximum $c^1 = 91 \Rightarrow \boxed{91\alpha^1 + 5\beta^1 \leq 1}$
  - no use tyring lower c^1 for $c^2 = 0$
- $c^2 > 0 \Rightarrow c^2 \in \{88, 89, 90\} \Rightarrow$ maximum $c^1 = 10 \Rightarrow$ tightest constraint is $\boxed{10\alpha^1 + \beta^1 + 90\alpha^2 + \beta^2 \leq 1}$

Conclusion: the tightest constraints can be found by solving a multiple-choice knapsack problem with k levels. For each level j, there are as many choices as realisable values c^j.

Converging towards the optimum

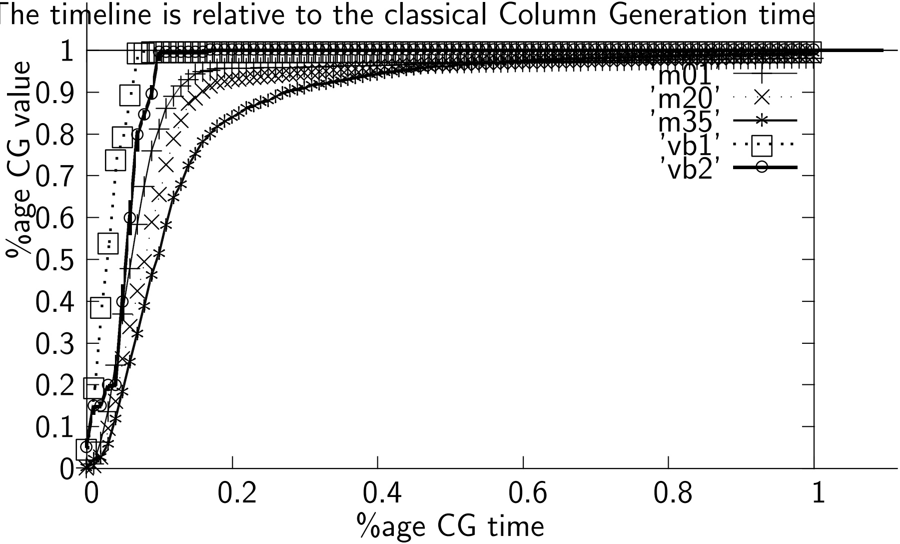
Suppose we have the optimal solution of $\mathbf{P}_k$. Steps to go to $\mathbf{P}_{k+1}$

- The optimum of $\mathbf{P}_k$ can be expressed as a feasible $\mathbf{P}_{k+1}$ solution
- The column generation based on multiple-choice knapsack is applied on $\mathbf{P}_{k+1}$ starting from above solution

The optimum is reached when $k = \lceil \frac{n}{2} \rceil$, or when $\lceil ub \rceil = \lceil lb \rceil$, where lb is calculated from the lower bound lb.

Results: the evolution of the lower bound

The timeline is relative to the classical Column Generation time



Conclusions, Related Work, Perspectives

A. Advantages of the Aggregated Models $\mathbf{P}_1, \mathbf{P}_2, \ldots \mathbf{P}_k$: at each step

- The number of variables is reduced
- The number of columns is reduced
- The sub-problem can become easier

$\Rightarrow$ A reasonably fast algorithm that provides lower bounds before finishing the computation (unlike column generation that converges through exterior solutions)

B. Connexions with other aggregation work in column generation

- [Elhallaoui, Villeneuve, Soumis, Desaulniers, Dynamic Aggregation of Set-Partitioning Constraints in Column Generation, *Operations Research*, 2005] aggregates constraints so as to speed up the convergence through exterior solutions

C. Similar methods can be applied to other problems