Aggregation Technique of Mixed Integer Rounding Cut and Cutting Plane Selection

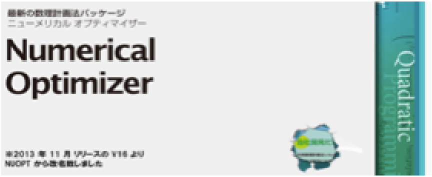
*Koichi Fujii

NTT DATA Mathematical Systems Inc.

2019/03/29

Background : Joint work with SCIP team

- Visited Zuse Institute Berlin from April 2018 to March 2019, joined SCIP team
- SCIP : Non-commercial Constraint Integer Programs Solver
- Idea: push forward optimization together to tackle the practical problems



Summary : Aggregation Technique of Mixed Integer Rounding Cut

Motivation

Cutting plane is an essential feature of branch-and-bound

Goal

Improve **Mixed integer rounding cut**, one of the most important cutting plane

Our Results

Improve heuristics of generating Mixed integer rounding cut

Summary : Cutting Plane Selection

Motivation

How to select cutting planes is an important issue in the implementation

Goal

Improve the strategy of cutting planes selection.

Our Results

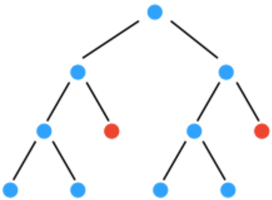
Assess the strategy **cloud points selection** / **one step selection**.

Agenda

The Cutting Plane

MILP

$$\min_{\boldsymbol{x}} \left\{ \boldsymbol{c}^T \boldsymbol{x} \mid \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \boldsymbol{x} \in \mathbb{Z}^n_{\geq 0} \times \mathbb{R}^m_{\geq 0} \right\}$$

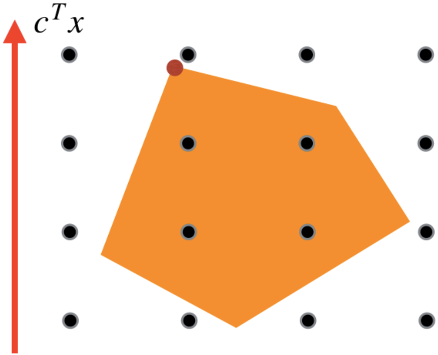


- LP relaxation is solved at each node
- Node can be pruned if (local) lower bound exceeds upper bound (integer feasible solution)

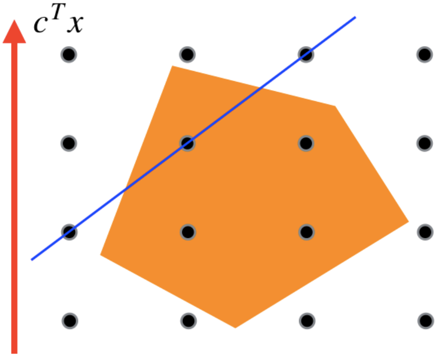
The Cutting Plane

MILP

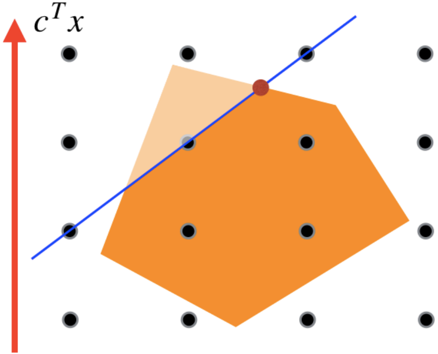
$$\min_{\boldsymbol{x}} \left\{ \boldsymbol{c}^T \boldsymbol{x} \mid \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \boldsymbol{x} \in \mathbb{Z}^n_{\geq 0} \times \mathbb{R}^m_{\geq 0} \right\}$$



The Cutting Plane



The Cutting Plane



Cutting plane improves lower bound (not always).

Mixed Integer Rounding Cuts

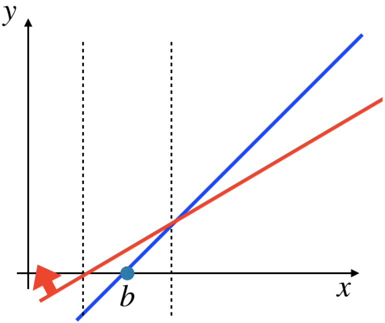
Heuristic generating procedure mixed integer rounding cut [Marchand, Wolsey, 1998]:

- Start with one constraint of the problem
- [**Bound Substitution**] Complement variables if LP solution is closer to upper bound
- [**Rounding**] Apply MIR formula
- [**Aggregation**] If no violated cut is found, add another problem constraint to the current aggregated inequality, go to **Bound Substitution**

Mixed Integer Rounding Cuts: Rounding

Proposition 3.1 (Wolsey Marchand '98)

Let $S := \{x, y \in \mathbb{Z} \times \mathbb{R}_+ \mid x{-}y \leq b\}$. *Then,* $x - \frac{y}{1-f_0} \leq \lfloor b \rfloor$ *is valid for* S *with* $f_0 = b - \lfloor b \rfloor$



Mixed Integer Rounding Cuts: Bound Substitution

Basic strategy : choose the closest bound distance base on LP solution

- Substitute with simple lower bound $\bar{y}_j = y_j - l_j$
- Substitute with simple upper bound $\bar{y}_j = u_j - y_j$
- Substitute with variable lower bound $\bar{y}_j = y_j - l_j x_j$ when $y_j \geq l_j x_j$
- Substitute with variable upper bound $\bar{y}_j = u_j x_j - y_j$ when $y_j \leq u_j x_j$

What is the best choice for degenerate case?

$$l_j \leq y_j \leq u_j \tag{1}$$

$$l_j x_j \leq y_j \leq u_j x_j \tag{2}$$

$$y_j^* = x_j^* = 0 \tag{3}$$

Mixed Integer Rounding Cuts: Experiment of Bound Substitution

$$
\begin{aligned}
\text{minimize} \quad & \sum_i c_i x_i + \sum_i d_i y_i && && (4)\\
\text{subject to} \quad & x_i \le u_i y_i, && \forall i \in I && (5)\\
& x_i \ge l_i y_i, && \forall i \in I && (6)\\
& \sum_i x_i \ge b && && (7)\\
& x_i \in \mathbb{R}, y_i \in \{0,1\} && && (8)
\end{aligned}
$$

Mixed Integer Rounding Cuts: Experiment of Bound Substitution

Regard knapsack constraint as a base inequality.
There are five choices for each variables to aggregate.

- (case1) Substitute with variable lower bound
- (case2) Substitute with variable lower bound + substitute integer varibale
- (case3) Substitute with variable upper bound
- (case4) Substitute with variable upper bound + substitute integer variable
- (case5) No substitution

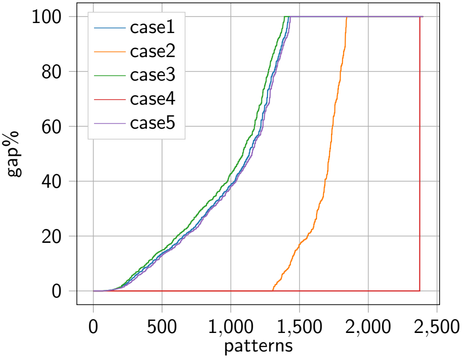
Mixed Integer Rounding Cuts: Experiment of Bound Substitution

Regard knapsack constraint as a base inequality.
There are five choices for each variables to aggregate.

- Test on 1000 random generated instances
- Generate cutting plane on each case
- Implemented by PySCIPOpt (SCIP Python interface)

Mixed Integer Rounding Cuts: Experiment of Bound Substitution

- The experiment shows that case3 is the best choice



Mixed Integer Rounding Cuts: Bound Substitution

$$l_i x_i \leq y_i \leq u_i x_i \tag{9}$$

$$l_i \leq y_i \leq u_i \tag{10}$$

Modifed rules for mixed integer rounding cuts (work with Robert Gottwald):

- Substitute with variable upper bound if $u_i - u_i x_i^* \geq l_i - l_i x_i^*$
- Substitute with variable lower bound if $l_i - l_i x_i^* \geq u_i - u_i x_i^*$

```
if( ((*bestlbtype) >= 0 || (*bestubtype) >= 0)
      && !SCIPisEQ(scip, *bestlb - simplelb, simpleub - *bestub) )
{
   if( *bestlb - simplelb > simpleub - *bestub )
      *selectedbound = SCIP_BOUNDTYPE_LOWER;
   else
      *selectedbound = SCIP_BOUNDTYPE_UPPER;
}
```

Mixed Integer Rounding Cuts: Computational Results on Unit Commitment Problems

- p_t^h : power production of the thermal units
- u_t^h, v_t^h, w_t^h : status of the thermal units

$$\text{minimize} \quad \sum_h \alpha^h p_t^h + \sum_h \beta^h u_t^h + \sum_h \gamma^h v_t^h \tag{11}$$

$$\text{subject to} \quad \underline{p}^h u_t^h \leq p_t^h \leq \overline{p}^h u_t^h \tag{12}$$

$$(\overline{p}^h - \Delta_+^h) u_{t-1}^h + p_t^h \leq \overline{p}^h + p_{t-1}^h \tag{13}$$

$$(\underline{p}^h - \Delta_-^h) u_t^h + p_{t-1}^h \leq \underline{p}^h + p_t^h \tag{14}$$

$$u_t^h - u_{t-1}^h = v_t^h - w_t^h \tag{15}$$

$$\sum_{t \geq k \geq t - \tau_{min}} v_k^h \leq u_t^h, \quad \sum_{t \geq k \geq t - \tau_{max}} w_k^h \leq 1 - u_t^h \tag{16}$$

$$\sum_h p_t^h \geq D_t \tag{17}$$

Mixed Integer Rounding Cuts: Computational Results on Unit Commitment Problems

bracket	model	modified bound substitution			
		faster	slower	time	node
≥ 0	360	276	26	0.44	0.17
≥ 10	147	138	5	0.33	0.04
≥ 100	5	5	0	0.11	0.01

Table: computational results on UC instances

- Implemented in SCIP 5.0.1
- Achieve overall 2.27x speedup on 360 randomly generated UC instances.
- 3.03x speedup on relatively hard problem (both case take $\geq$ 10 sec)
- 2 more intances can be solved (3600 sec time limit).

Mixed Integer Rounding Cuts: Computational Results on MIPLIBs

- Tested on mipdev-solvable testset (425 instances include MIPLIB 2010) with 5 seeds.
- On the 40 affected instances, achieve 3.4% speed-up and the primal-dual integral (performance measure of primal heuristics) >20% improvement
- The code will be release in the next version of SCIP.

Mixed Integer Rounding Cuts : Aggregation

Heuristic generating procedure mixed integer rounding cut by Marchand and Wolsey (1998):

- Start with one constraint of the problem.
- [**Bound Substitution**] Complement variables if LP solution is closer to upper bound.
- [**Rounding**] Apply MIR formula.
- [**Aggregation**] If no violated cut is found, add another problem constraint to the current aggregated inequality, go to Bound Substitution.

Mixed Integer Rounding Cuts: Aggregation

- SCIP scores each constraint for **Aggregation**
- The score is weighted sum of density, slackness and the product of dual and fractionality
- Some adhoc rules: prefer equalities which contain a single continuous variable [Ambros et al., 2017]

Mixed Integer Rounding Cuts: Aggregation

Cutting plane for capacitated facility location problem (join work with Chen Weikun).

- $x_{i,j}$: assigned to a facility
- y_j : facility

$$\text{minimize} \quad \sum_{i,j} c_{i,j} x_{i,j} + \sum_j d_j y_j \tag{18}$$

$$\text{subject to} \quad \sum_i d_j x_{i,j} \leq y_j \tag{19}$$

$$\sum_j x_{i,j} = 1 \tag{20}$$

If we sum up all the constraints, we get a knapsack constraint $\sum y_j \leq \sum_j d_j$. $\Rightarrow$ this leads to significant speedup.

Mixed Integer Rounding Cuts: Aggregation

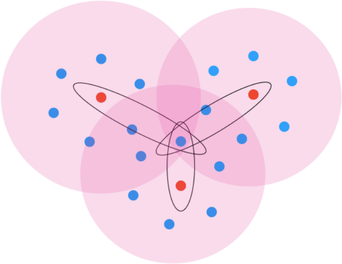
Capacitated facility location cut

Step 1: Find partition constraints $\sum_j x_{i,j} = 1$

Step 2: Check the constraints which will be covered by partition constrains

Step 3: Aggregate partition constraints and covered constraints

Step 4: Generate MIR cut



Mixed Integer Rounding Cuts: Computational Results on Single Facility Location Problem

		with capacitated facility location cut			
$\geq$ *time*(*sec*)	model	faster	slower	time	node
≥ 0	100	59	39	0.37	0.22
≥ 10	96	56	38	0.44	0.29
≥ 100	67	38	27	0.68	0.62

Table: Computational results on capacitated facility location instances

- 100 instances from http://www.math.nsc.ru/AP/benchmarks/CFLP/cflp-eng.html
- implemented in SCIP 6.0.0
- overall 2.70x speedup on 100 capacitated facility location instances.
- 13 more intances can be solved (7200 sec time limit).

Mixed Integer Rounding Cuts: Computational Results on MIPLIBs

- Only 4(/112) instances are affected on mipdev-solvable testset(425), no improvement.
- Future work: can we generalize more?

Summary : Aggregation Technique of Mixed Integer Rounding Cut

- Modified bound substitution rules is effective for practical problems as unit commitment problem
- Adhoc aggregation rules (capacitated facility location cuts) is effective for a certain problem class
    - Still need more generalization

Summary : Cutting Plane Selection

Motivation

How to select cutting planes is an important issue in the implementation

Goal

Improve the strategy of cutting planes selection.

Our Results

Assess the strategy **cloud points selection** / **one step selection**.

Cutting Plane Selection

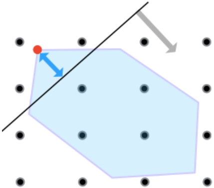
Greedy strategy for cutting plane selection: [Achterberg, 2009, Wesselmann and Suhl, 2012]

- Sort the cuts by their score
- Select the cuts which have less similarity
- similarity := parallelism $p(\alpha_1, \alpha_2) = \frac{\alpha_1^t \alpha_2}{||\alpha_1||||\alpha_2||}$

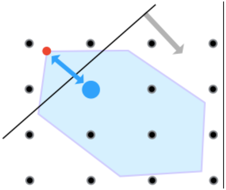
Cutting Plane Selection: Previous work

How to score the cutting planes $\boldsymbol{\alpha}^t \boldsymbol{x} \leq \boldsymbol{\beta}$

efficacy $\frac{\boldsymbol{\alpha} \boldsymbol{x_{lp}} - \boldsymbol{\beta}}{||\boldsymbol{\alpha}||}$



directed cutoff distance $\frac{\boldsymbol{\alpha} \boldsymbol{x_{lp}} - \boldsymbol{\beta}}{\boldsymbol{\alpha}^t (\boldsymbol{x}^{int} - \boldsymbol{x}^{lp})}$



Cutting Plane Selection: Cloud Points

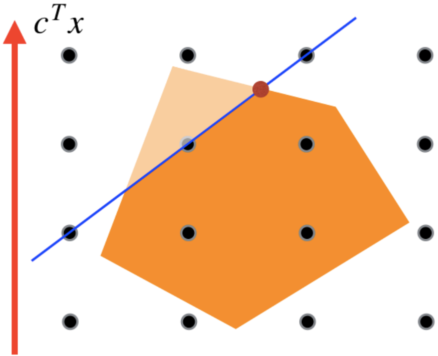
Evaluate efficacy based on multiple points on the **optimal face** [Achterberg, 2013]:

Step 1: Pumping : generate points $\boldsymbol{x^k}$ randomly on optimal face by changing objective function.

Step 2: Return minimum efficacy : $\min_k \frac{\boldsymbol{\alpha x_{lp}^k} - \boldsymbol{\beta}}{||\boldsymbol{\alpha}||}$

Cutting Plane Selection: One Step in Dual Simplex

Evaluate how much the objective function will be improved in one step of dual simplex after adding the cutting plane.



Cutting Plane Selection: One Step in Dual Simplex

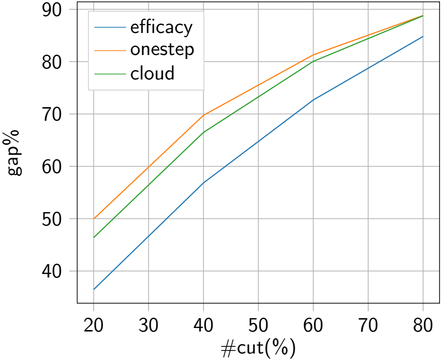
Evaluate how much the objective function will be improved in one step of dual simplex after adding the cutting plane.

Proposition 4.1

If the cutting plane $\boldsymbol{\alpha}^T \mathbf{x_{lp}} \leq \boldsymbol{\beta}$ have only positive coefficients and all variables are in nonbasic at the optimal basis (e.g. **classical** *Gomory cuts), then the objective function will be improved at least* $\min_j \frac{d_j}{\alpha_j} \cdot (\boldsymbol{\alpha} \mathbf{x_{lp}} - \boldsymbol{\beta})$ *where d_j is a reduced cost of variable x_j.*

Cutting Plane Selection: Computational Results of Root Gap

- Select cutting plane based on its score
- Root gap with limited number of adding cutting plane
- Ignore parallelism
- Take average on MIPLIB 2010 instances



Cutting Plane Selection: Computational Results of Number of Nodes

$\geq$ #*node*	model	cloud #node	onestep #node
≥ 0	1050	0.96	0.96
≥ 16	565	0.92	0.91
≥ 128	348	0.94	0.85

- Randomly generated 30 cities TSP problems
- 4% node reduction overall with cloud points selection / one step selection
- Onestep achieve 15% node reduction on hard instances (≥ 100 nodes in every case).

Cutting Plane Selection: Computational Results of # Node

		cloud		onestep	
$\geq$ # node	model	# node	time	# node	time
$\geq$ 0	1365	0.99	0.99	1.01	1.46
$\geq$ 16	837	0.99	0.99	1.00	1.50
$\geq$ 128	686	1.00	1.00	0.99	1.49
$\geq$ 1024	452	0.97	0.99	0.98	1.52
$\geq$ 16384	178	0.99	1.00	0.97	1.39

- mipdev-solvable instances (425) with 5 seeds
- Could not observe overall improvement
- Onestep achieve 3% node reduction on hard instances ($\geq$ 16384 nodes in every case)
- Onestep get 32% overall slowdown
- Caution: default includes pumping (generating clouding points)

Summary: Cutting Plane Selection

- Evaluate one step in dual simplex has interesting property, though currently too time consuming
- Future work : hybrid method is possible? (as strong branching and reliability branching)
- There are two heuristics in cutting plane
  - Heuristics of generating cutting plane (Bound Substitution / Aggregation)
  - Heuristics of cutting plane selection