Large Neighborhood Search in Column Generation Algorithms

Marco Lübbecke and Christian Puchert

RWTH Aachen University
Chair of Operations Research

International Workshop on Column Generation
Bromont, Québec, June 10-13, 2012

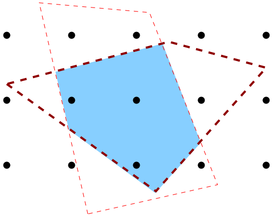
Outline

Outline

Our Setting

We are solving MIPs of the form

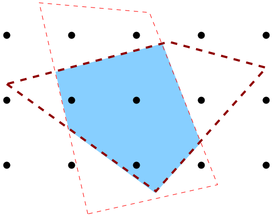
$$\begin{aligned} \min \quad & c^T x \\ \text{s. t.} \quad & Ax \geq b \\ & Dx \geq d \\ & x = \mathbb{Z}^q \times \mathbb{Q}^{n-q}. \end{aligned}$$



Our Setting

We are solving MIPs of the form

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax \geq b \\ & Dx \geq d \\ & x = \mathbb{Z}^q \times \mathbb{Q}^{n-q}. \end{aligned}$$



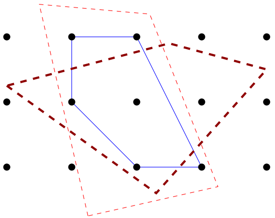
In our applications, D has a block structure, i.e.

$$Dx \geq d \quad \Longleftrightarrow \quad \begin{array}{l} D^k x^k \geq d^k \text{ for all } k \\ x = x^1 \times \cdots \times x^K. \end{array}$$

Our Setting

We are solving MIPs of the form

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax \geq b \\ & Dx \geq d \\ & x = \mathbb{Z}^q \times \mathbb{Q}^{n-q}. \end{aligned}$$



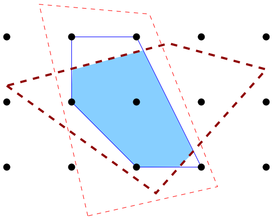
Apply *Dantzig-Wolfe Decomposition*, i.e. convexify the blocks by

$$x^k = \sum_{p \in P_k} \lambda_{kp} x^{kp}, \ \sum_{p \in P_k} \lambda_{kp} = 1, \ \lambda \geq 0,$$

where $x^{kp} \in \mathbb{Z}^{q_k} \times \mathbb{Q}^{n_k - q_k}$.

Our Setting

$\rightsquigarrow$ Master problem:



$$
\begin{aligned}
\min \quad & \sum_k \sum_{p \in P_k} c^{kp} \lambda_{kp} \\
\text{s.t.} \quad & \sum_k \sum_{p \in P_k} a^{kp} \lambda_{kp} \geq b \\
& \sum_{p \in P_k} \lambda_{kp} = 1 \quad \forall k \\
& x^k = \sum_{kp \in P_k} \lambda_{kp} x^{kp} \quad \forall k \\
& \lambda \geq 0 \\
& x \in \mathbb{Z}^q \times \mathbb{Q}^{n-q}
\end{aligned}
$$

$\Longrightarrow$ potentially stronger relaxation!

Primal Heuristics

- Important aspect of MIP solving: finding good feasible solutions to get strong primal bounds

Primal Heuristics

- Important aspect of MIP solving: finding good feasible solutions to get strong primal bounds
- The earlier good solutions are found, the more B&B nodes can be pruned

Primal Heuristics

- Most heuristics work with an LP feasible solution $\tilde{x}$, some others also with already known feasible solutions (*improvement heuristics*)

Primal Heuristics

- Most heuristics work with an LP feasible solution $\tilde{x}$, some others also with already known feasible solutions (*improvement heuristics*)
- Often, $\tilde{x}$ comes from the LP relaxation

Primal Heuristics

- Most heuristics work with an LP feasible solution $\tilde{x}$, some others also with already known feasible solutions (*improvement heuristics*)
- Often, $\tilde{x}$ comes from the LP relaxation
- However, an $\tilde{x}$ can also be obtained by translating the master LP solution into the original space

Primal Heuristics

- Most heuristics work with an LP feasible solution $\tilde{x}$, some others also with already known feasible solutions (*improvement heuristics*)
- Often, $\tilde{x}$ comes from the LP relaxation
- However, an $\tilde{x}$ can also be obtained by translating the master LP solution into the original space
- In our setting, there are *two spaces* to look for feasible solutions:
  - the master variables (where the relaxation is solved)
  - the original variables

Primal Heuristics

Common paradigms:

- Round an LP feasible solution while trying to avoid violation of the linear constraints

Primal Heuristics

Common paradigms:

- Round an LP feasible solution while trying to avoid violation of the linear constraints
- Quickly go down the Branch-and-Bound tree (*diving*)

Primal Heuristics

Common paradigms:

- Round an LP feasible solution while trying to avoid violation of the linear constraints
- Quickly go down the Branch-and-Bound tree (*diving*)
- Search a neighborhood of an LP feasible and/or some feasible solutions (*Large Neighborhood Search*)

Outline

Large Neighborhood Search

- For general MIPs, a number of LNS heuristics have already been developed

Large Neighborhood Search

- For general MIPs, a number of LNS heuristics have already been developed
- A neighborhood is defined by adding constraints and/or changing variable bounds and thus restricting the feasible space

Large Neighborhood Search

- For general MIPs, a number of LNS heuristics have already been developed
- A neighborhood is defined by adding constraints and/or changing variable bounds and thus restricting the feasible space
- The neighborhood is searched by solving the resulting sub-MIP

Large Neighborhood Search

- For general MIPs, a number of LNS heuristics have already been developed
- A neighborhood is defined by adding constraints and/or changing variable bounds and thus restricting the feasible space
- The neighborhood is searched by solving the resulting sub-MIP
- The sub-MIP is smaller and therefore hopefully easier to solve

Classical LNS heuristics

Let $\tilde{x}$ be LP feasible, $\bar{x}$ be LP and integer feasible.

- RENS (Relaxation Enforced Neighborhood Search):
  For each variable, set its bounds to

$$\lfloor \tilde{x}_i \rfloor \leq \tilde{x}_i \leq \lceil \tilde{x}_i \rceil;$$

Classical LNS heuristics

Let $\tilde{x}$ be LP feasible, $\bar{x}$ be LP and integer feasible.

- RENS (Relaxation Enforced Neighborhood Search):
  For each variable, set its bounds to

$$\lfloor \tilde{x}_i \rfloor \leq \tilde{x}_i \leq \lceil \tilde{x}_i \rceil;$$

- RINS (Relaxation Induced Neighborhood Search):
  Fix each variable which satisfies $\tilde{x}_i = \bar{x}_i$.

Classical LNS heuristics

Let $\bar{x}, \bar{x}_1, \ldots, \bar{x}_\kappa$ be LP and integer feasible.

- Crossover:
  Fix each variable for which $\bar{x}_1 = \cdots = \bar{x}_\kappa$;

Classical LNS heuristics

Let $\bar{x}, \bar{x}_1, \ldots, \bar{x}_\kappa$ be LP and integer feasible.

- Crossover:
  Fix each variable for which $\bar{x}_1 = \cdots = \bar{x}_\kappa$;
- OneOpt:
  For each integer variable, try to shift $\bar{x}_i$ in a direction that improves the objective and preserves LP feasibility.

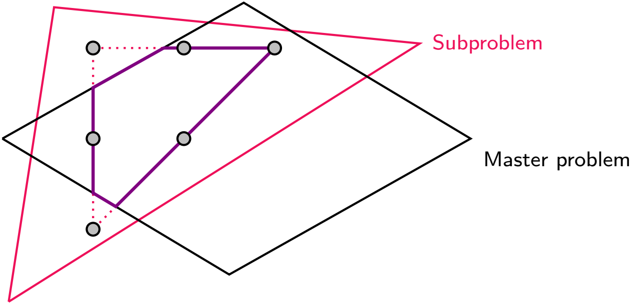
Outline

Extreme Point Heuristics

- In B&P, each $\tilde{x}$ is a convex combination of points x^p
- The points are integer feasible and may only violate the $Ax \geq b$ constraints

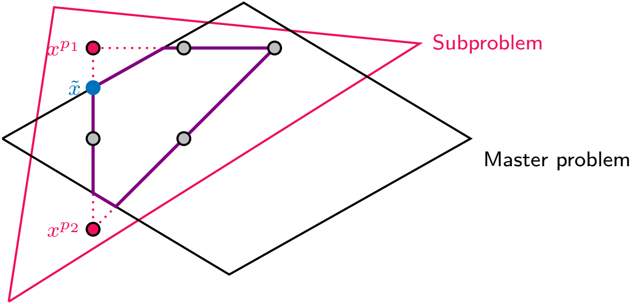
Extreme Point Heuristics

- In B&P, each $\tilde{x}$ is a convex combination of points x^p
- The points are integer feasible and may only violate the $Ax \geq b$ constraints



Extreme Point Heuristics

- In B&P, each $\tilde{x}$ is a convex combination of points x^p
- The points are integer feasible and may only violate the $Ax \geq b$ constraints



Extreme Point Heuristics

In Column Generation: Define neighborhoods in terms of extreme points x^p:

- Extreme Point Crossover:
  Choose κ extreme points $x^{p_1}, \ldots, x^{p_\kappa}$; fix each variable x_i that satisfies $x_i^{p_1} = \cdots = x_i^{p_\kappa}$;

Extreme Point Heuristics

In Column Generation: Define neighborhoods in terms of extreme points x^p:

- Extreme Point Crossover:
  Choose κ extreme points $x^{p_1}, \ldots, x^{p_\kappa}$; fix each variable x_i that satisfies $x_i^{p_1} = \cdots = x_i^{p_\kappa}$;
- Extreme Point RINS:
  Choose again κ extreme points and let them "vote": Fix a variable x_i if for at least $\alpha \cdot 100$ percent of the extreme points x^p, we have $x_i^p = \tilde{x}_i$.

Which extreme points are chosen?

There are a number of possible choices for x^p:

- all x^p generated so far;

Which extreme points are chosen?

There are a number of possible choices for x^p:

- all x^p generated so far;
- all x^p that contribute to the current $\tilde{x}$, i.e. for which $\lambda_p > 0$;

Which extreme points are chosen?

There are a number of possible choices for x^p:

- all x^p generated so far;
- all x^p that contribute to the current $\tilde{x}$, i.e. for which $\lambda_p > 0$;
- a random choice.

Remarks

- The extreme point heuristics are inspired by Crossover and RINS; in contrast to them, however, they do not need to know any feasible solution

Remarks

- The extreme point heuristics are inspired by Crossover and RINS; in contrast to them, however, they do not need to know any feasible solution
- They work on the original variables; however, they use information yielded by the master problem

Remarks

- The extreme point heuristics are inspired by Crossover and RINS; in contrast to them, however, they do not need to know any feasible solution
- They work on the original variables; however, they use information yielded by the master problem
- For Binary Programs and Mixed Binary Programs, their neighborhoods are related to the neighborhood of the RENS heuristic

Extreme Points vs. Feasible Solutions

Is it justified to use extreme points instead of feasible solutions? (They may be, after all, infeasible.)

$\leadsto$ See next slide for an answer...

Extreme Points vs. Feasible Solutions

Experiment: Compare extreme points with known feasible solutions.

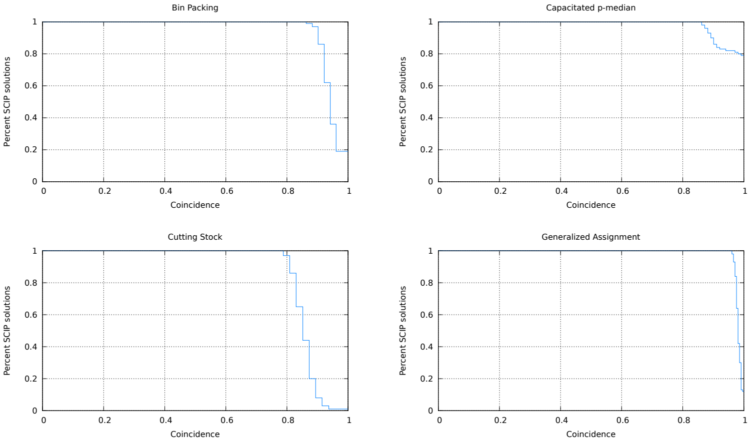


Figure: Coincidence of extreme points with feasible solutions

Outline

Results

- The heuristics have been implemented in `GCG` (see talk by M. Lübbecke tomorrow)
- We tested on several structured problem classes from the literature:
  - Bin Packing (BP)
  - Capacitated p-median (CPMP)
  - Cutting Stock (CS)
  - Generalized Assignment (GAP)

Performance: RENS, RINS, OneOpt

	RENS		RINS		OneOpt	
Test set	found	time	found	time	found	time
BP-bison1	164/714	1.1	0/714	1.0	74/714	1.0
CPMP-optlab	1/188	1.0	1/188	1.0	0/188	1.0
CPMP-orlib	0/5	1.0	0/5	1.0	0/5	1.0
CS-schwerin	0/188	1.0	0/188	1.0	0/188	1.0
CS-waescher	0/21	1.0	0/21	1.0	0/21	1.0
GAP-orlib-min	48/84	1.0	0/84	1.0	0/84	1.0
GAP-yagiura-min	12/57	1.0	0/57	1.0	0/57	1.0

Table: Found solutions and execution times

Performance: Extreme Point Heuristics

	XP Crossover		XP RINS	
Test set	found	time	found	time
BP-bison1	123/714	1.1	134/714	1.1
CPMP-optlab	1/88	1.0	6/88	1.0
CPMP-orlib	0/5	1.0	1/5	1.0
CS-schwerin	139/188	1.0	126/188	1.0
CS-waescher	21/21	1.0	14/21	1.0
GAP-orlib-min	9/36	1.0	27/36	1.0
GAP-yagiura-min	11/57	1.0	19/57	1.0

Table: Found solutions and execution times

Solution quality: Bin Packing

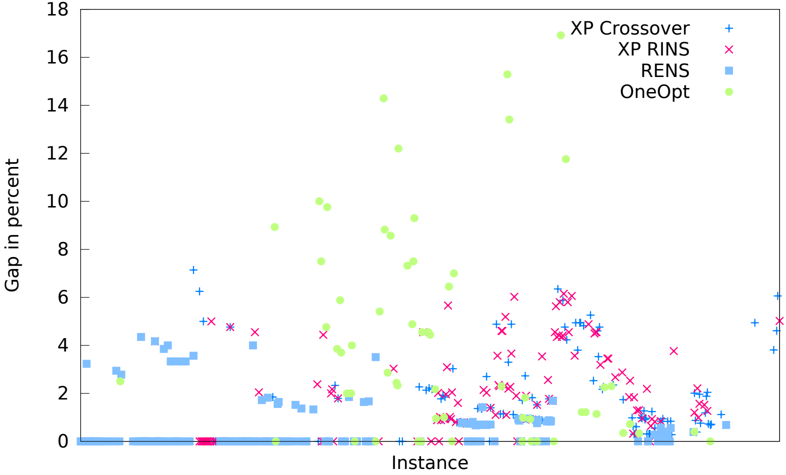


Figure: Gaps for Bin Packing problems

Solution quality: Capacitated p-median

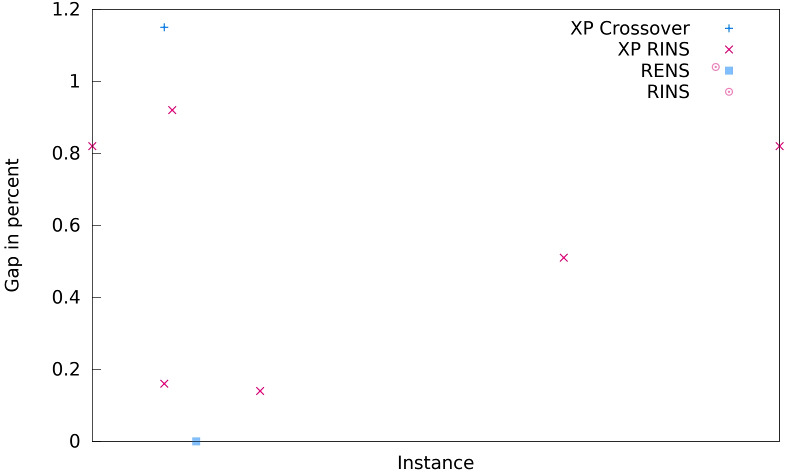


Figure: Gaps for Capacitated p-median problems

Solution quality: Cutting Stock

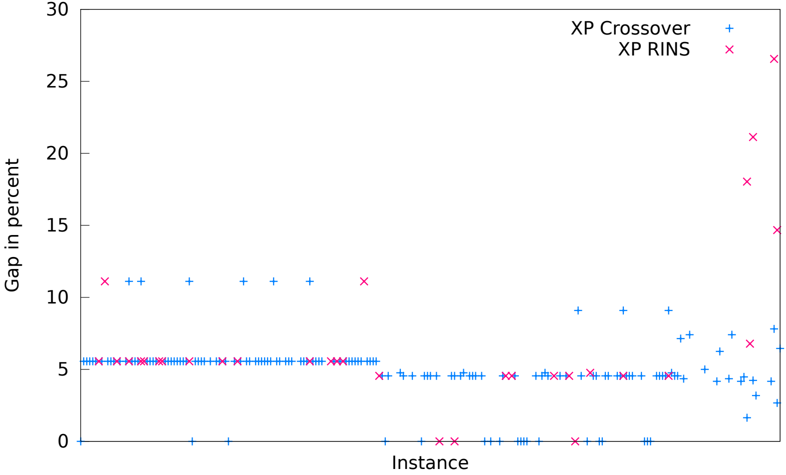


Figure: Gaps for Cutting Stock problems

Solution quality: Generalized Assignment

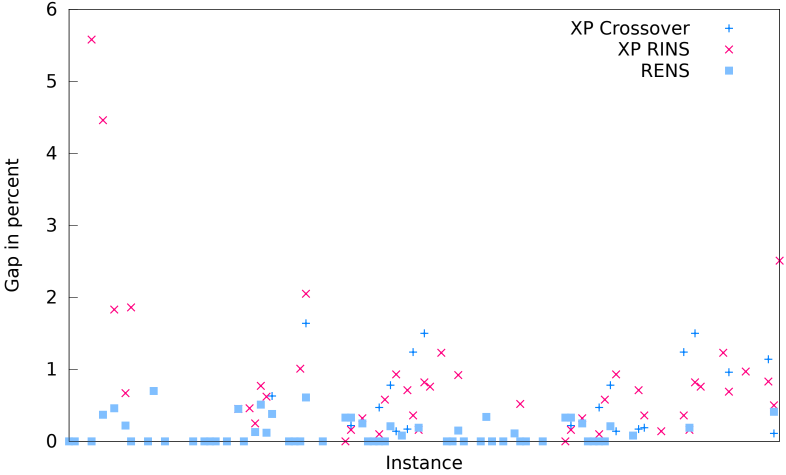


Figure: Gaps for Generalized Assignment problems

Results

- XP Crossover and XP RINS particularly successful on CS, but also on BP and GAP; only a few solutions on CPMP
- Solutions on GAP and CPMP of good quality ($< 6\%$, $< 1.2\%$)
- RENS fails on CS (neighborhood too large), but its solutions are better than those found by XP heuristics (to be expected)
- XP RINS outperforms XP Crossover on GAP
- OneOpt only suitable for BP
- The similarity of extreme points to integer feasible points does not tell us how good the heuristics work

Conclusion

- We tested LNS heuristics in the context of Column Generation
- In particular, we introduced two new heuristics (XP Crossover and XP RINS)
- The new heuristics produce good solutions on a number of different structured problems
- In the future: Tests on more structured problems