GCG: A Generic Branch-Price-and-Cut Solver

`#SCIP Workshop 2014` · Berlin · 10/01/2014

RWTHAACHEN UNIVERSITY

Dantzig-Wolfe Reformulation

- original problem

$$
\begin{aligned}
\min \quad & c^T x \\
\text{s.t.} \quad & Ax \geq b \\
& Dx \geq d \\
& x \in \mathbb{Z}^n_+
\end{aligned}
$$

Dantzig-Wolfe Reformulation

- original problem

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax \geq b \\ & Dx \geq d \\ & x \in \mathbb{Z}_+^n \end{aligned}$$

- reformulation of $x \in X := \left\{ x \in \mathbb{Z}_+^n : Dx \geq d \right\}$ as convex combination of extreme points $\boldsymbol{x}^p$ of $conv(X)$:

$$\sum_{p \in P} \boldsymbol{x}^p \lambda_p = x$$

$$\sum_{p \in P} \lambda_p = 1$$

$$\lambda_p \geq 0 \quad \forall\, p \in P$$

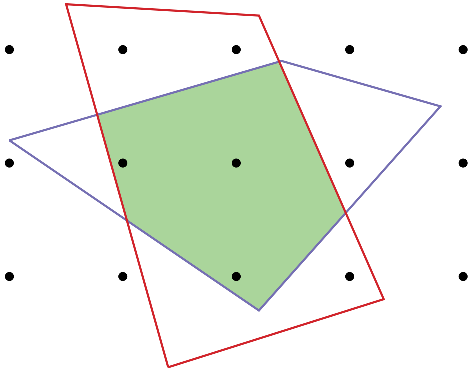
RWTH AACHEN UNIVERSITY

Dantzig-Wolfe Reformulation

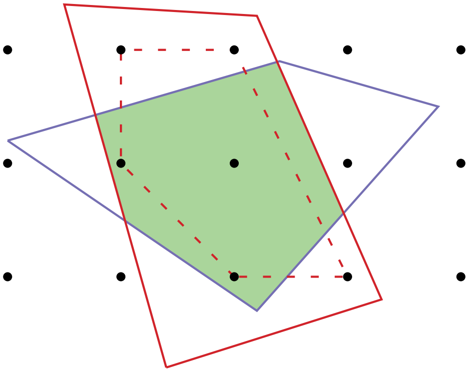
- (integer) master problem

$$
\begin{aligned}
\min \quad & \sum_{p \in P} (c^T \boldsymbol{x}^p) \lambda_p & & \\
\text{s. t.} \quad & \sum_{p \in P} (A \boldsymbol{x}^p) \lambda_p & \geq \quad & b \\
& \sum_{p \in P} \lambda_p & = \quad & 1 \\
& \lambda_p & \geq \quad & 0 \qquad \forall p \in P \\
& \sum_{p \in P} \boldsymbol{x}^p \lambda_p & = \quad & x \\
& x & \in \quad & \mathbb{Z}_+^n
\end{aligned}
$$

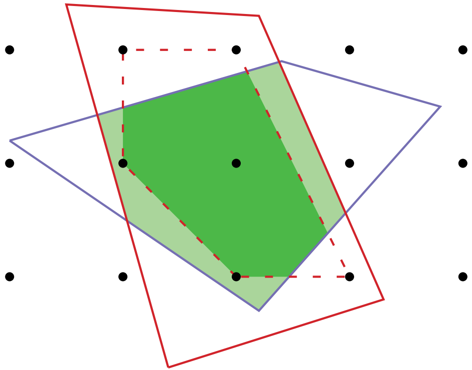
Dantzig-Wolfe Reformulation



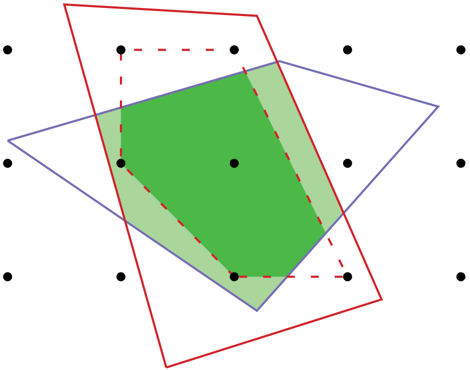
Dantzig-Wolfe Reformulation



Dantzig-Wolfe Reformulation



Dantzig-Wolfe Reformulation



- potentially stronger LP relaxation
- *huge* number of variables (one variable per extreme point)
- solve via *branch-price-and-cut*

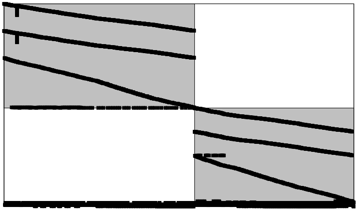
RWTH AACHEN UNIVERSITY

Branch-Price-and-Cut

- solve LP relaxation by applying *column generation*
  - start with a subset of all variables
  - iteratively add negative reduced cost variables/columns by solving a subproblem/pricing problem over X
- specialized branching
- specialized cutting planes
- ...

Dantzig-Wolfe: Traditional Realm

- bordered block-angular matrix structure
- subproblem corresponding to each block



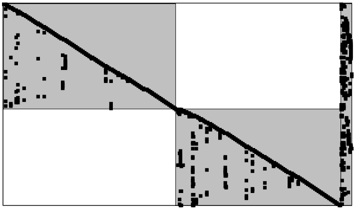
p2756

2 "blocks"

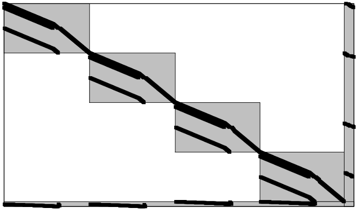
- vehicle routing, generalized assignment, cutting stock, p-median, bin packing, graph coloring, many more, ...

Dantzig-Wolfe: Extending the Applicability

- linking variables and double-bordered block-angular matrices
- e.g., replace linking variable x by copies x^k for each bock k, enforce $x = x^k$ Guinard (2004)



timtab1



set1ch

- *every* matrix is double-bordered block-angular!

Implementation of Branch-Price-and-Cut

- good frameworks available (e.g., SCIP)
- implementation still non-trivial and time-consuming
- needs expert knowledge, experience, and "tricks"

Implementation of Branch-Price-and-Cut

- good frameworks available (e.g., SCIP)
- implementation still non-trivial and time-consuming
- needs expert knowledge, experience, and "tricks"
- many implementations from scratch, little re-use of code
- implementations not done, because of lack of knowledge
- implementations not dared, because of unclear outcome
- many users have no access to a state-of-the-art methodology

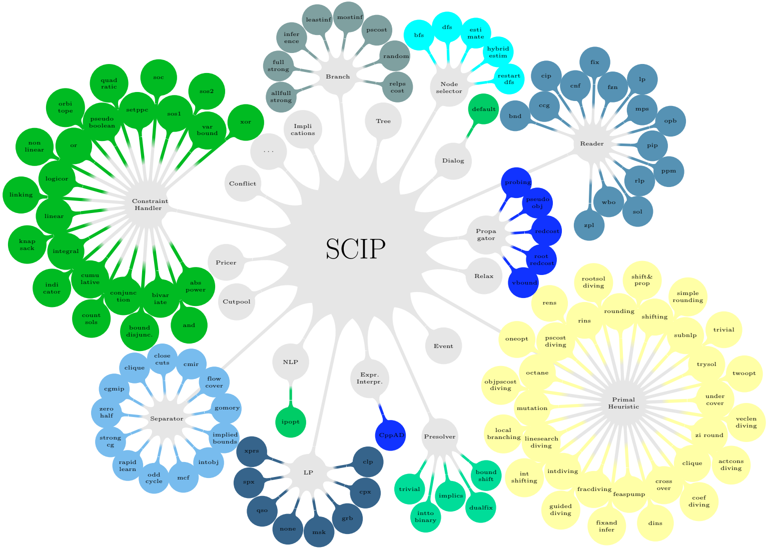
The GCG Project: Generic Column Generation

- generic solver for *structured mixed integer programs*
- fully automatic branch-price-and-cut
- no user-interaction, usable for non-experts

- started in 2008 with Gerald Gamrath's Diploma thesis
- project initiator and head: M. Lübbecke
- current maintainers and main developers:
  M. Bergner, G. Gamrath, Chr. Puchert, J. Witt
- part of the SCIP Optimization Suite
- download at `www.or.rwth-aachen.de/gcg` or `scip.zib.de`

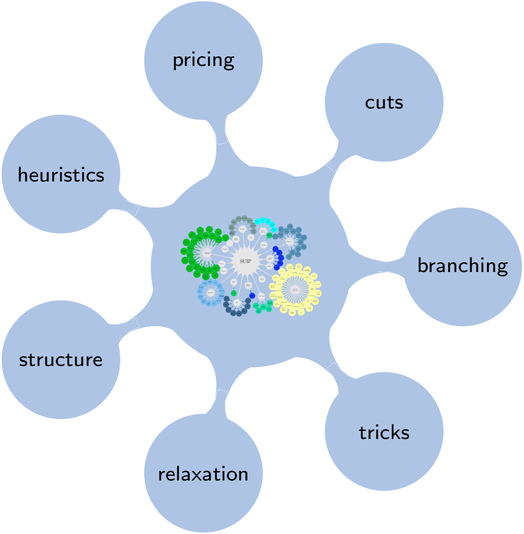
- related projects: `BaPCod`, `DIP`, `SAS/OR DECOMP`

The Branch-Price-and-Cut Framework SCIP



RWTH AACHEN UNIVERSITY

Extending the Framework to a Solver: GCG



Structure: Provided by the User

```
PRESOLVED
0
NBLOCKS
5
BLOCK 1
c2
c7
c8
...
BLOCK 2
c3
c9
c10
c19
...
MASTERCONSS
c1
c57
...
```

- user provides Dantzig-Wolfe reformulation
  - label master constraints $Ax \geq b$
  - label constraints of subproblem(s) $Dx \geq d$

  in accordance to `.lp`/`.mps` file
- `.dec`, `.blk`, `.ref` file formats
- very easy to provide in e.g., ZIMPL

- 2.0 ▸ wishlist

Structure: Detected Automatically

- heuristic evaluation of decompositions Bergner et al. (2014)
- set {packing, covering, partitioning} + connected components
- staircase (several variants), classification of constraints , . . .
- graph partitioning Ferris, Horn (1998)
- graph clustering
- theoretical notion of "goodness" of decomposition quality (ongoing research, very hard)

▶ 2.0 ▶ devel ▶ wishlist

Sketch: Structure Detection Based on Graphs

- representations of matrices $A = (a_{ij})$ by (hyper-)graphs

- vertex j per variable x_j, edge jk iff $a_{ij}, a_{ik} \neq 0$ for some i
- vertex i per row, vertex j per variable x_j, edge ij iff $a_{ij} \neq 0$
- vertex per $a_{ij} \neq 0$, hyperedge per variable x_j/row i

- key: a connected component induces a block in A

Sketch: Structure Detection Based on Graphs

- representations of matrices $A = (a_{ij})$ by (hyper-)graphs

- vertex j per variable x_j, edge jk iff $a_{ij}, a_{ik} \neq 0$ for some i
- vertex i per row, vertex j per variable x_j, edge ij iff $a_{ij} \neq 0$
- vertex per $a_{ij} \neq 0$, hyperedge per variable x_j/row i

- key: a connected component induces a block in A
- example: hypergraph representation Aykanat et al. (2004)

$$\begin{pmatrix} 1 & 1 & & & & 1 \\ & 1 & & & 1 & \\ 1 & 1 & & & & \\ & & 1 & & & 1 \\ & & 1 & 1 & 1 & \\ 1 & & 1 & & 1 & 1 \end{pmatrix}$$

Sketch: Structure Detection Based on Graphs

- representations of matrices $A = (a_{ij})$ by (hyper-)graphs

- vertex j per variable x_j, edge jk iff $a_{ij}, a_{ik} \neq 0$ for some i
- vertex i per row, vertex j per variable x_j, edge ij iff $a_{ij} \neq 0$
- vertex per $a_{ij} \neq 0$, hyperedge per variable x_j/row i

- key: a connected component induces a block in A
- example: hypergraph representation Aykanat et al. (2004)

$$\begin{pmatrix} 1 & 1 & & & & 1 \\ & 1 & & & 1 & \\ 1 & 1 & & & & \\ & & 1 & & & 1 \\ & & 1 & 1 & 1 & \\ 1 & & 1 & & 1 & 1 \end{pmatrix}$$

Sketch: Structure Detection Based on Graphs

- representations of matrices $A = (a_{ij})$ by (hyper-)graphs

- vertex j per variable x_j, edge jk iff $a_{ij}, a_{ik} \neq 0$ for some i
- vertex i per row, vertex j per variable x_j, edge ij iff $a_{ij} \neq 0$
- vertex per $a_{ij} \neq 0$, hyperedge per variable x_j/row i

- key: a connected component induces a block in A
- example: hypergraph representation Aykanat et al. (2004)

$$
\begin{pmatrix}
1 & 1 & & & & 1 \\
 & 1 & & & 1 & \\
1 & 1 & & & & \\
 & & 1 & & & 1 \\
 & & 1 & 1 & 1 & \\
1 & & 1 & & 1 & 1
\end{pmatrix}
$$

Sketch: Structure Detection Based on Graphs

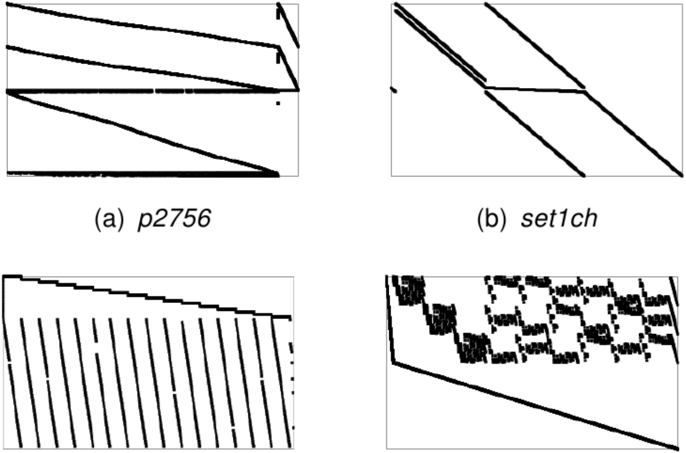
- representations of matrices $A = (a_{ij})$ by (hyper-)graphs

- vertex j per variable x_j, edge jk iff $a_{ij}, a_{ik} \neq 0$ for some i
- vertex i per row, vertex j per variable x_j, edge ij iff $a_{ij} \neq 0$
- vertex per $a_{ij} \neq 0$, hyperedge per variable x_j/row i

- key: a connected component induces a block in A
- example: hypergraph representation Aykanat et al. (2004)

$$\begin{pmatrix} 1 & 1 & & & & 1 \\ & 1 & & & 1 & \\ 1 & 1 & & & & \\ & & 1 & & & 1 \\ & & 1 & 1 & 1 & \\ 1 & & 1 & & 1 & 1 \end{pmatrix}$$

Forcing MIPLIB2003 into Structure

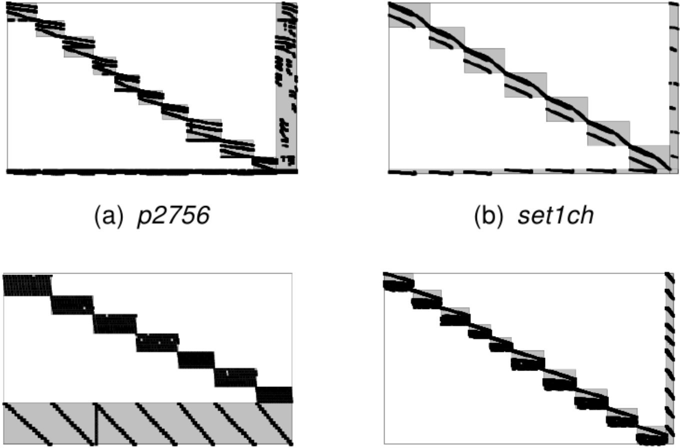


(a) *p2756* (b) *set1ch*

(c) *opt1217* (d) *manna81*

Bergner et al. (2014)

Forcing MIPLIB2003 into Structure



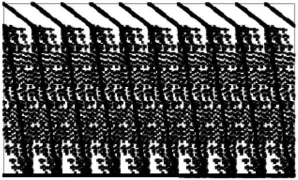
(a) *p2756* (b) *set1ch*

(c) *opt1217* (d) *manna81*

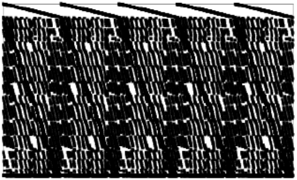
Bergner et al. (2014)

Forcing MIPLIB2010 into Structure

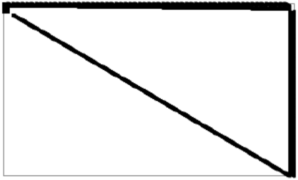
(a) *m100n500k4r1*



(b) *mine-90-10*



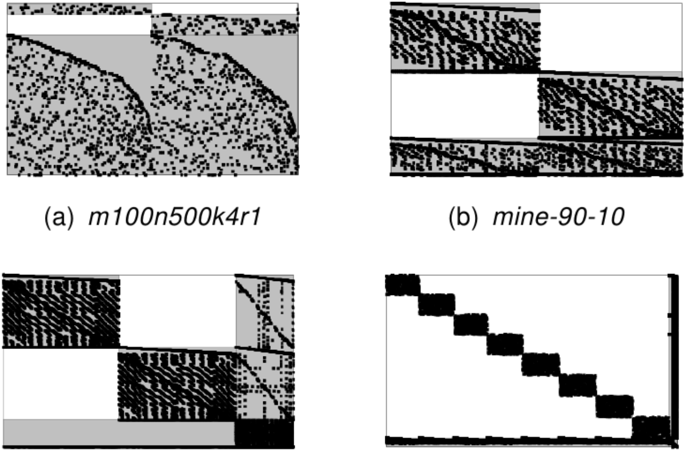
(c) *mine-166-5*



(d) *neos-686190*

Bergner et al. (2014)

Forcing MIPLIB2010 into Structure



(a) *m100n500k4r1*

(b) *mine-90-10*

(c) *mine-166-5*

(d) *neos-686190*

Bergner et al. (2014)

Reformulation

- design: two synchronized instances/trees
- convexification – discretization
- discretization for MIPs
- aggregation of identical subproblems
- symmetry/isomorphism detection of subproblems
- elimination/handling of linking variables
- aggregation of *non*-identical subprobl. Bergner, Dahms (2013)
- handle non-linear constraints in the subproblems

▶ 2.0 ▶ devel ▶ wishlist

Pricing

- (specialized) pricing solver
  - MIP
  - heuristic pricing (but much more is needed)
  - knapsack solver
  - ESPPRC solver
- Farkas pricing for primal LP feasibility
- domain propagation from original to pricing problem
- column pool
- generation of “structured” columns Ghoniem, Sherali (2009)
- more elaborate column management
- dual variable smoothing Wentges (1997), Vanderbeck et al. (2012)
- interior point stabilization
- dual cutting planes Ben Amor et al. (2006)

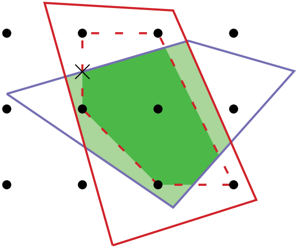
Branching

- on original variables Desrosiers, Soumis, Desrochers (1984)
- pseudo-cost branching
- branching decisions convexified in pricing
- on constraints, specialized Ryan, Foster (1981)
- on constraints, generic Vanderbeck (2010)
- strong branching Røpke (2012)

- 2.0 ▸ devel ▸ wishlist

Cutting Planes

- on original variables
- "combinatorial" cuts
- cuts that need a basis

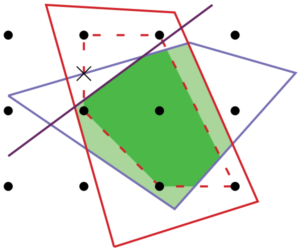


- on master variables Petersen et al. (2008), Jepsen et al. (2008)
- structured separation of extreme points Ralphs, Galati (2006)

► 2.0 ► devel ► wishlist

Cutting Planes

- on original variables
- "combinatorial" cuts
- cuts that need a basis



- on master variables Petersen et al. (2008), Jepsen et al. (2008)
- structured separation of extreme points Ralphs, Galati (2006)

▶ 2.0 ▶ devel ▶ wishlist

Primal Heuristics

- diving on original variables
- diving on master variables Joncour et al. (2010)
- RENS, RINS, and other sub-MIP heuristics
- extreme point crossover
- feasibility pump
- master initialization with solutions from original problem
- structure-exploiting heuristics (like staircase diving)
- set covering heuristics Caprara, Fischetti, Toth (1999)
- set partitioning heuristics

▶ 2.0 ▶ devel ▶ wishlist

“Standard” CG/BP&C Features

- many parameters (number of columns per iteration etc.)
- Lagrangian dual bound + early branching
- parallelization of pricing problems
- (heuristic) reduced cost fixing of original variables
- handling of degeneracy
- dynamic constraint aggregation Desrosiers, Gaulthier, L. (2012)

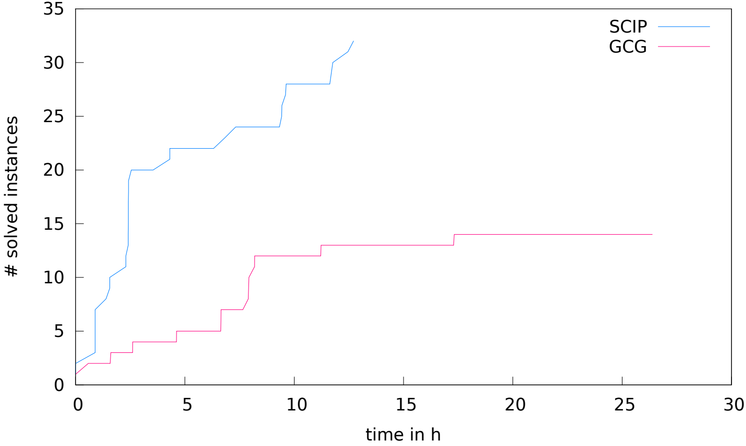
- 2.0 ▸ devel ▸ wishlist ▸ utopia

Statistics

- time spent in preprocessing, pricing, master problem, etc.
- time of generation of LP/IP optimal columns
- information about the decomposition, including visual
- degree of degeneracy
- development of bounds in the tree
- primal integrals Berthold (2012)

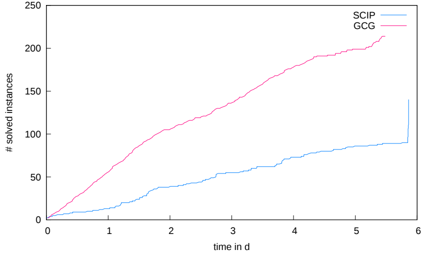
Performance of 1.0 on MIPLIB Selection

- # instances solved to integer optimality, SCIP vs. GCG

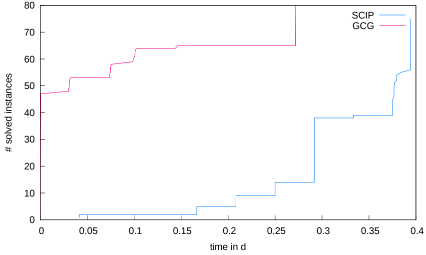


Performance of 1.0 for Structured Models

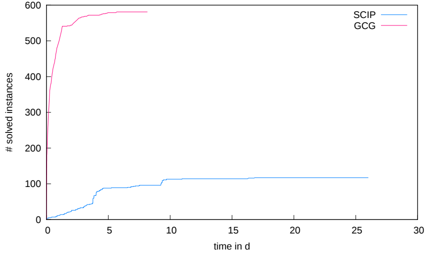
cutting stock:



generalized assignment:



bin packing:



- # instances solved to integer optimality
- GCG vs. SCIP

Take-Home

- automatic decomposition and branch-price-and-cut doable
- lucky for "unstructured" MIPs, successful on "structured"
- challenge: tell promising from unpromising instances

- ideally, this will complement existing state-of-the-art solvers

- download from `scip.zib.de` or `or.rwth-aachen.de/gcg`