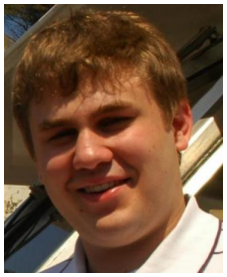
Generic Branch-Price-and-Cut

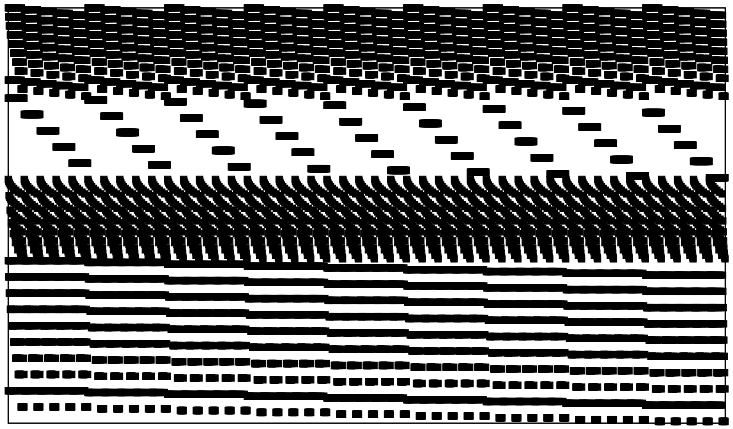
A Status Report

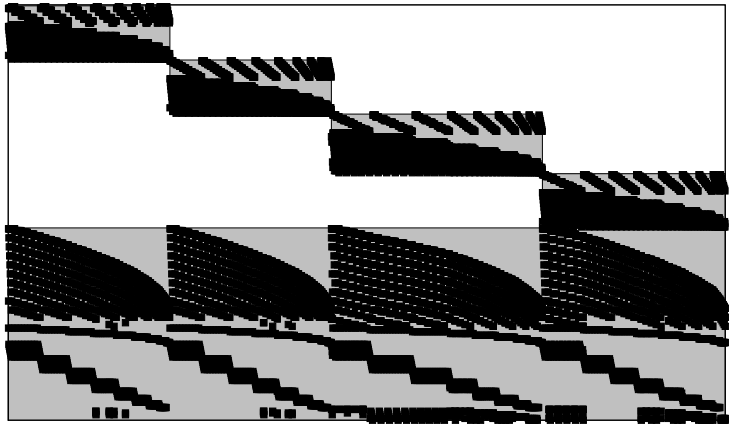
Column Generation · Bromont · June 12, 2012

OPERATIONSRESEARCH RWTHAACHEN

T-Shirts



can you spot the difference?



Status Quo: On the Positive Side

- many exciting developments in branch-and-price
- some problems solvable *only* by B&P
- more and more experts (look around you!)
- more and more "standard" in IP solving
- many successful applications (and growing!)
- good frameworks for implementation available
- branch-and-price much easier to implement than years ago

Status Quo: On the Negative Side

- implementation still non-trivial
- requires a good deal of work
- need expert knowledge, experience, and "tricks"

Status Quo: On the Negative Side

- implementation still non-trivial
- requires a good deal of work
- need 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

Goals of our Project

- we would like to make B&P accessible to a broad audience
- generic solver for *structured mixed integer programs*
- fully automatic branch-price-and-cut
- no user-interaction, usable for non-experts
- several related/similar projects: `BaPCod`, `DIP`, `G12`
- report only on our solver: `GCG`

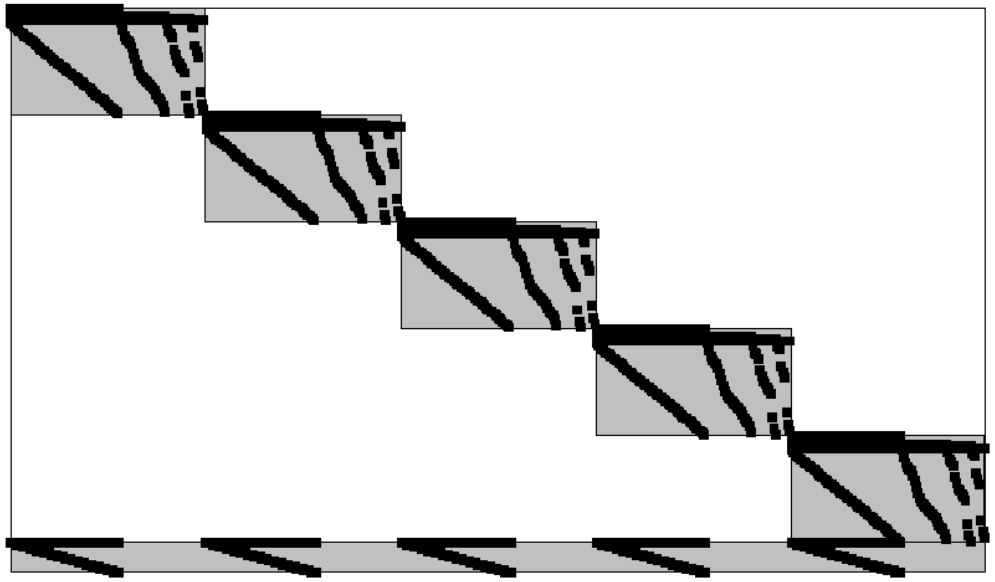
Our Generic BP&C Solver GCG

- based on state-of-the-art MIP solver SCIP
- automatic detection of decomposable structures
- alternatively: user provides a decomposition
- DW reformulation, column generation, branch-and-price
- synchronizes two trees, "original" and "extended" formulation
- convexification, discretization, aggregation
- parallelization of subproblems
- branching on original variables, Ryan/Foster, ...
- integration of generic (combinatorial) cutting planes
- primal heuristics particular to decompositions
- preprocessing, propagation, ... (inherited from SCIP)

⋮

Dantzig-Wolfe: Traditional Realm

- Dantzig-Wolfe works particularly well for bordered block-angular matrix structure

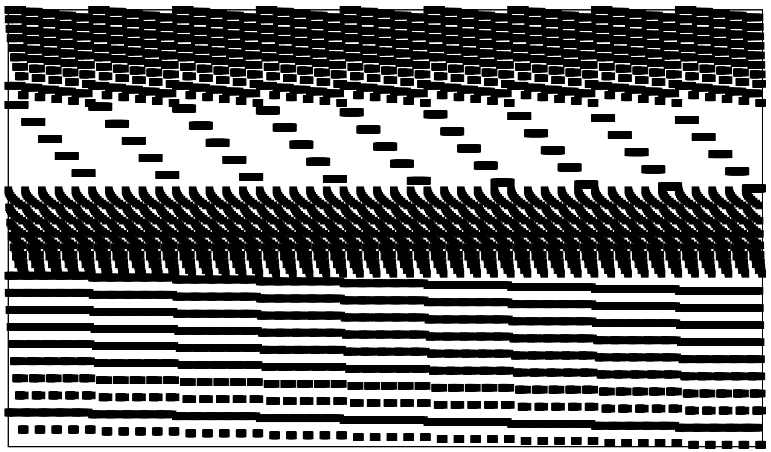


rout

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

Block-Angular Structure?

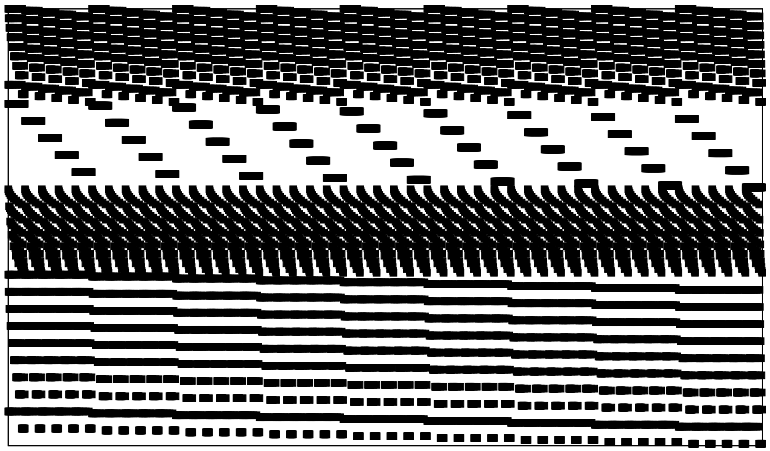
- key to automatic decomposition: detect "structure"



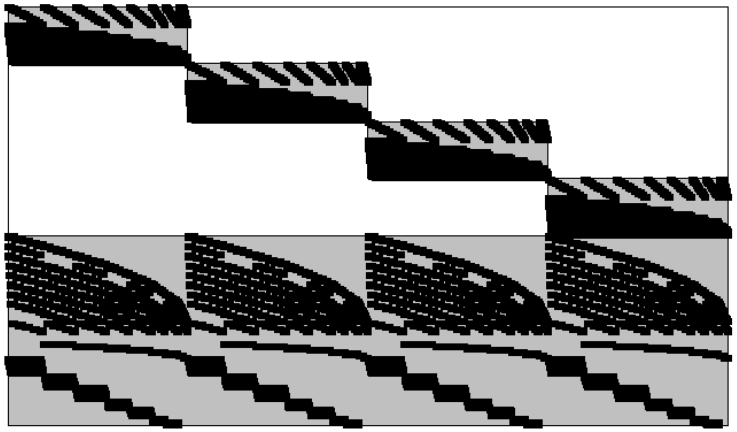
10teams

Block-Angular Structure

- key to automatic decomposition: detect "structure"



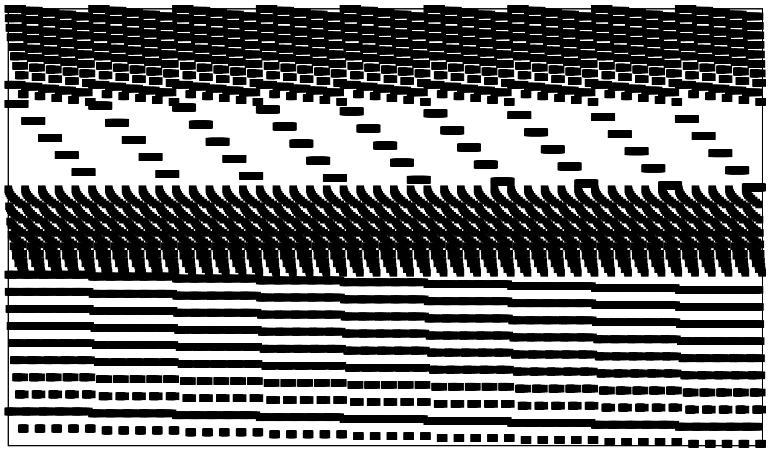
10teams



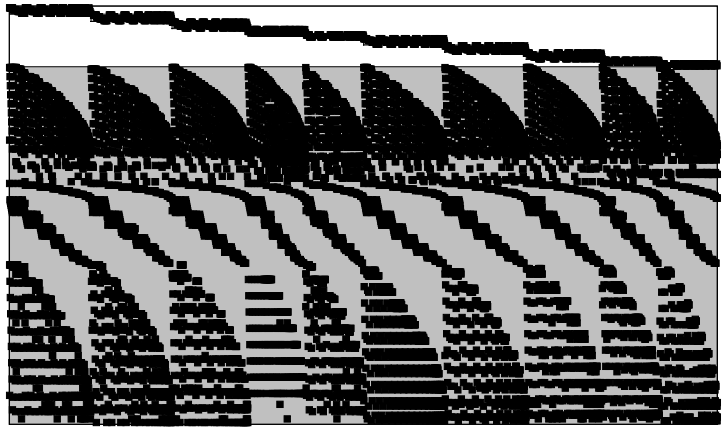
10teams

Block-Angular Structure

- key to automatic decomposition: detect "structure"

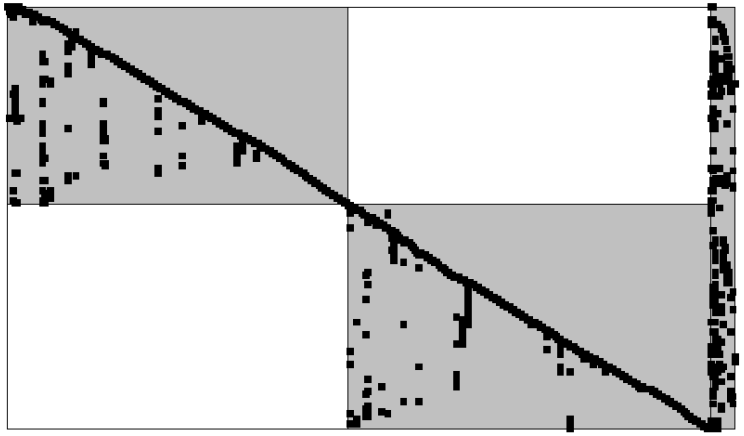


10teams

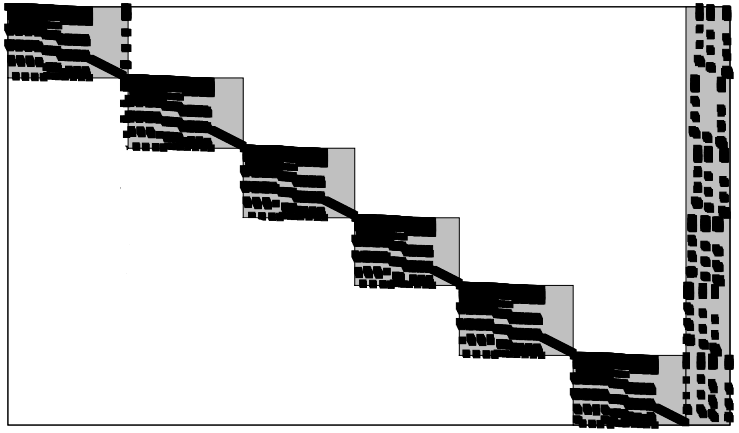


10teams

Often: Linking Variables



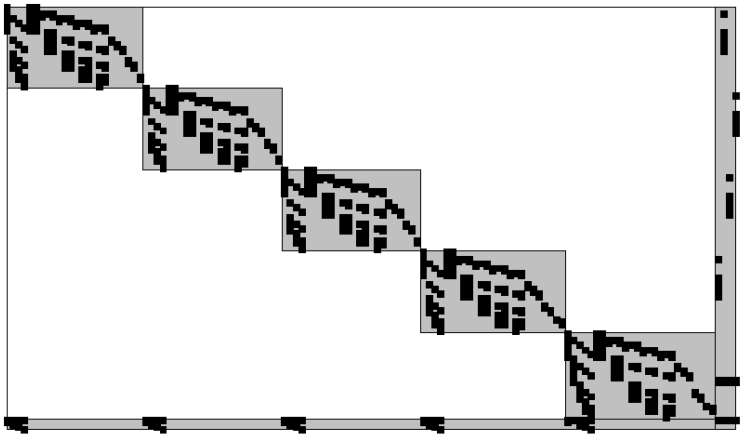
timtab1



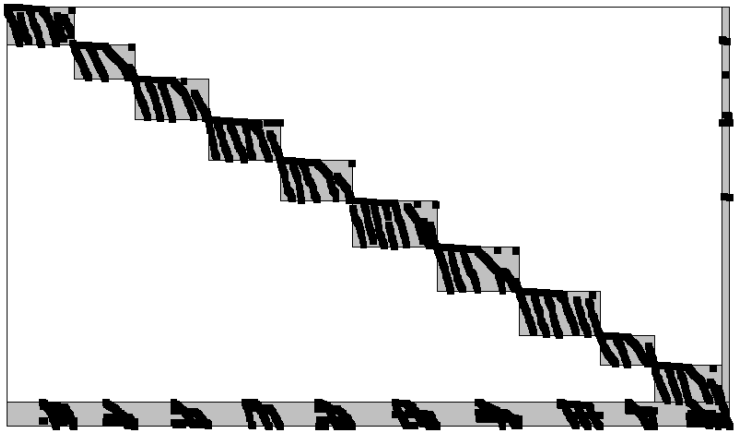
gesa2

- e.g., replace linking variable x by copies x^i, enforce $x = x^i$
- alternatively, use DW reformulation $x^i = \sum_{p \in P^i} \lambda_p p$

More General: Arrowhead Structure



noswot



aflow30a

Where do these Pictures come from?

Where do these Pictures come from?

- "folklore" connection: (hyper-)graphs and matrices
  - several possibilities, e.g., (variant of Ferris/Horn, 1998)
  - vertex for every non-zero entry
  - hyperedge for each row, resp. each column
  - (different weights for different types of vars and conss)

Where do these Pictures come from?

- "folklore" connection: (hyper-)graphs and matrices
  - several possibilities, e.g., (variant of Ferris/Horn, 1998)
  - vertex for every non-zero entry
  - hyperedge for each row, resp. each column
  - (different weights for different types of vars and conss)

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

Where do these Pictures come from?

- "folklore" connection: (hyper-)graphs and matrices
  - several possibilities, e.g., (variant of Ferris/Horn, 1998)
  - vertex for every non-zero entry
  - hyperedge for each row, resp. each column
  - (different weights for different types of vars and conss)

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

Where do these Pictures come from?

- "folklore" connection: (hyper-)graphs and matrices
  - several possibilities, e.g., (variant of Ferris/Horn, 1998)
  - vertex for every non-zero entry
  - hyperedge for each row, resp. each column
  - (different weights for different types of vars and conss)

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

Where do these Pictures come from?

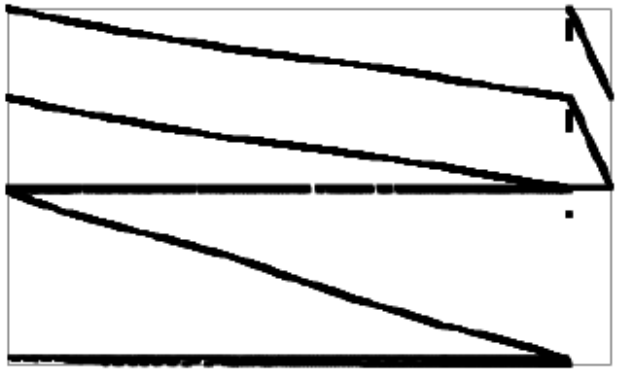
- "folklore" connection: (hyper-)graphs and matrices
  - several possibilities, e.g., (variant of Ferris/Horn, 1998)
  - vertex for every non-zero entry
  - hyperedge for each row, resp. each column
  - (different weights for different types of vars and conss)

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

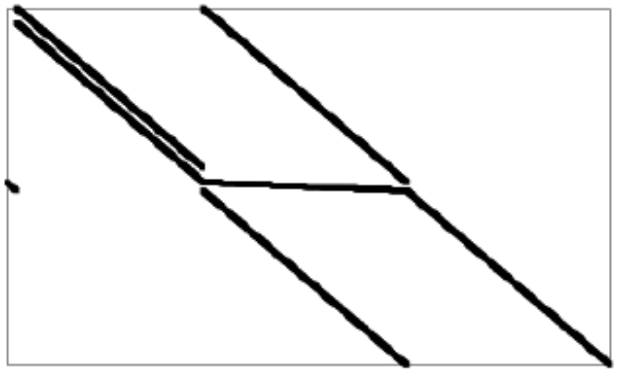
Where do these Pictures come from?

- "folklore" connection: (hyper-)graphs and matrices
  - several possibilities, e.g., (variant of Ferris/Horn, 1998)
  - vertex for every non-zero entry
  - hyperedge for each row, resp. each column
  - (different weights for different types of vars and conss)
- arrowhead detection $\equiv k$-way hypergraph partitioning
  $\equiv k$-way vertex separation
- NP-hard problems, but effective heuristics available
- last year we used `hMETIS` 2.0pre1
- meanwhile we know about better algorithms

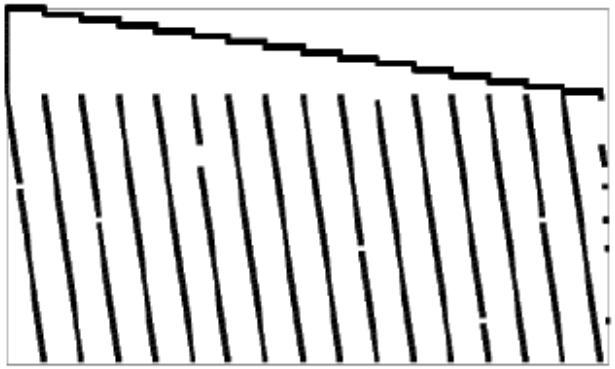
Forcing MIPLIB2003 into Structure



(a) *p2756*



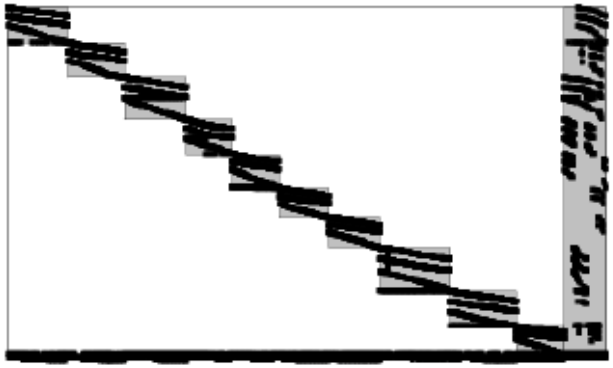
(b) *set1ch*



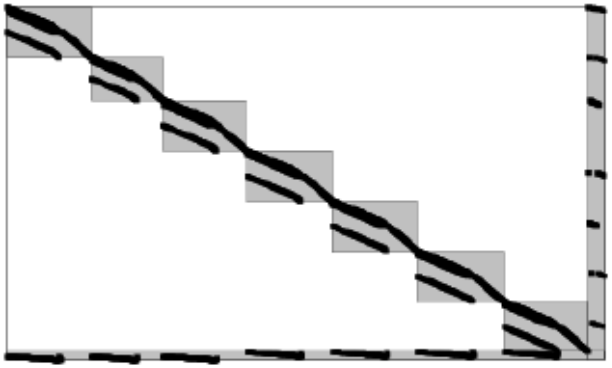
(c) *opt1217*

(d) *manna81*

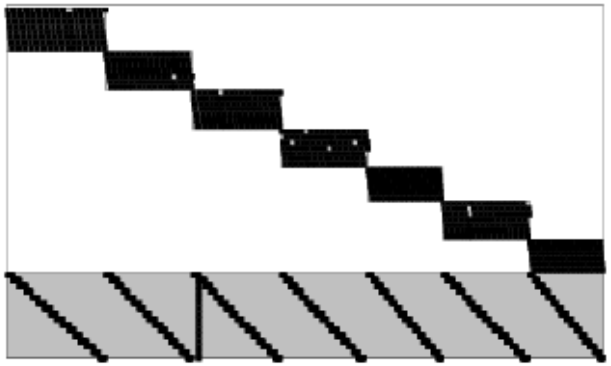
Forcing MIPLIB2003 into Structure



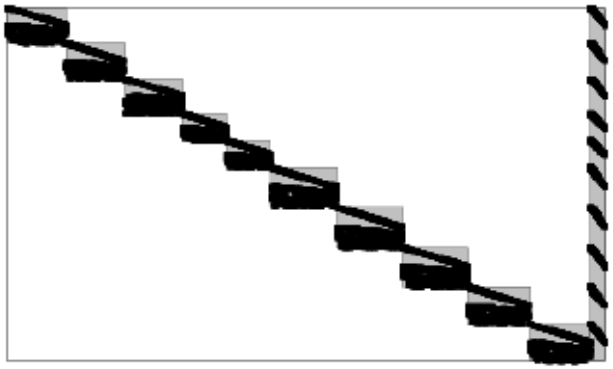
(a) *p2756*



(b) *set1ch*



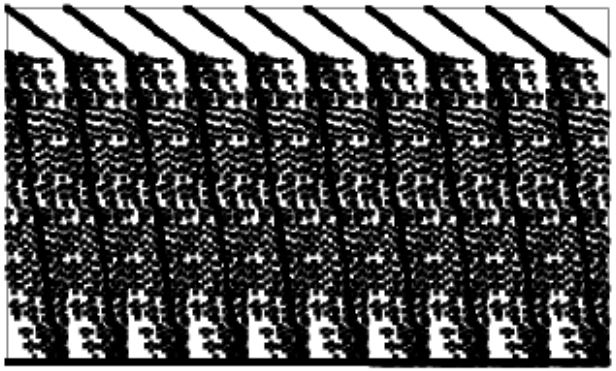
(c) *opt1217*



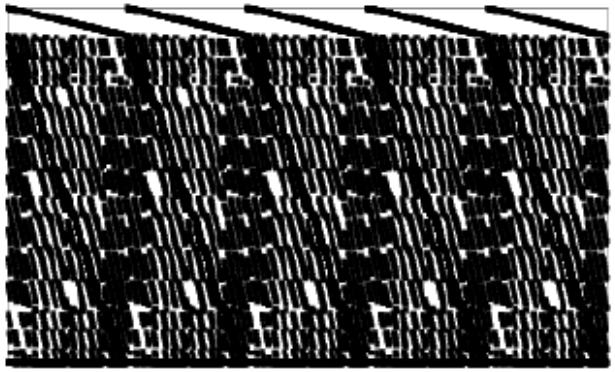
(d) *manna81*

Forcing MIPLIB2010 into Structure

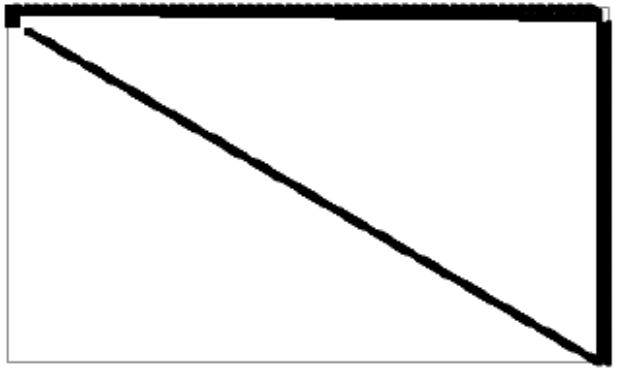
(a) *m100n500k4r1*



(b) *mine-90-10*

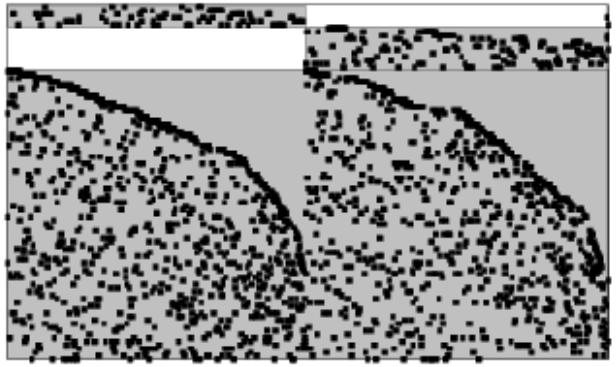


(c) *mine-166-5*

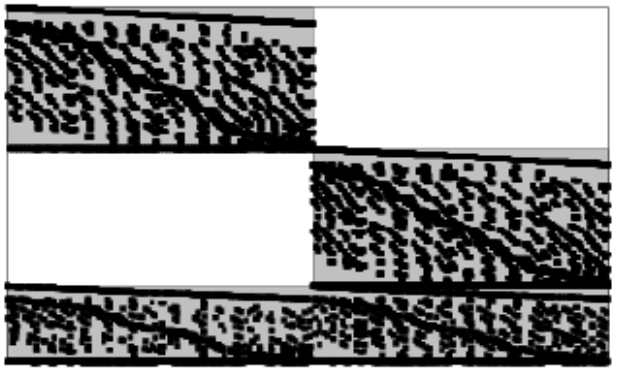


(d) *neos-686190*

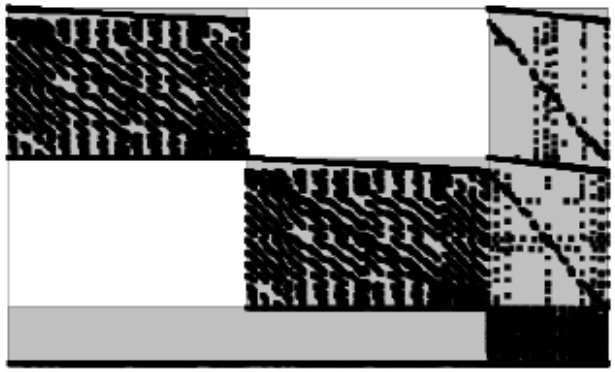
Forcing MIPLIB2010 into Structure



(a) *m100n500k4r1*



(b) *mine-90-10*

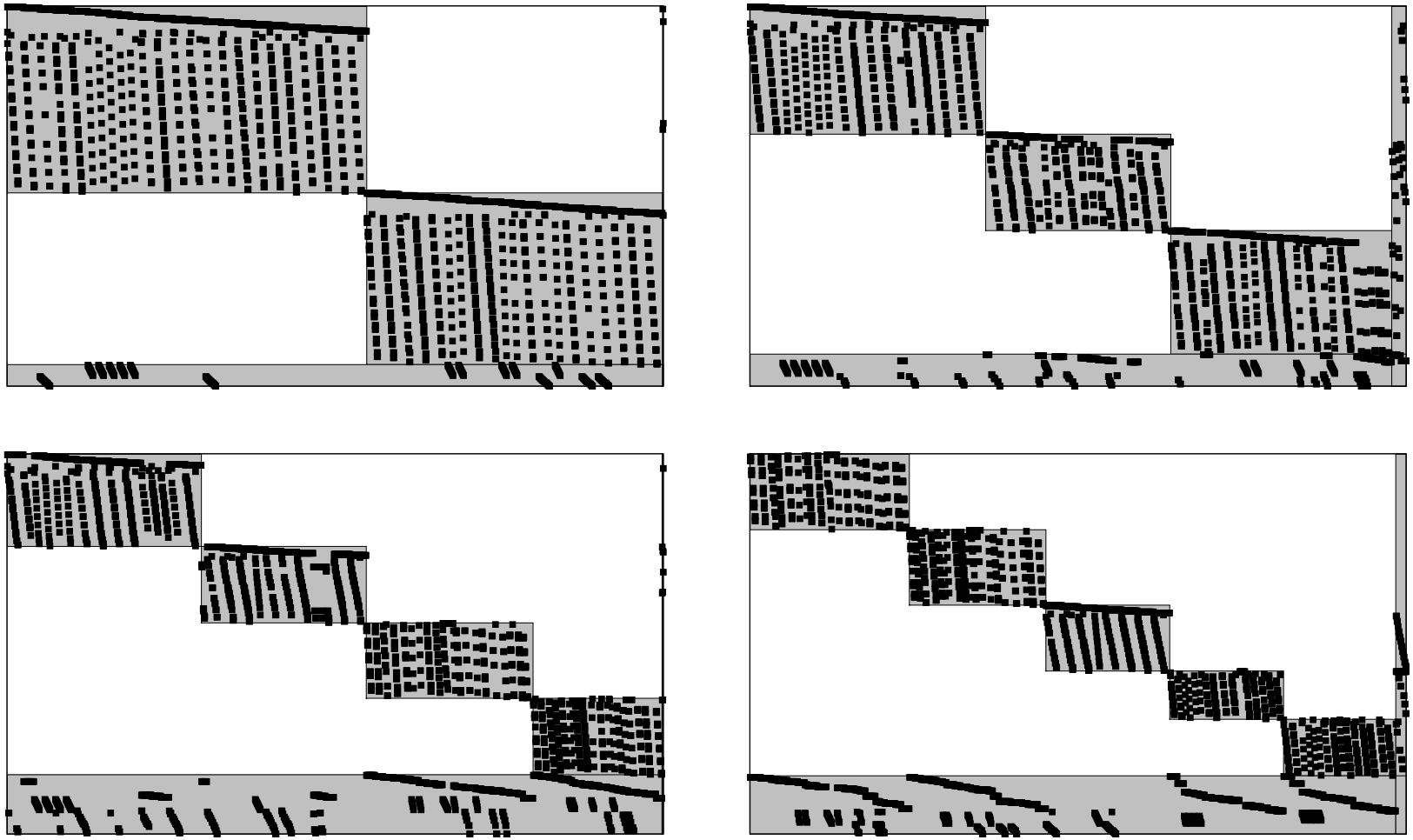


(c) *mine-166-5*

(d) *neos-686190*

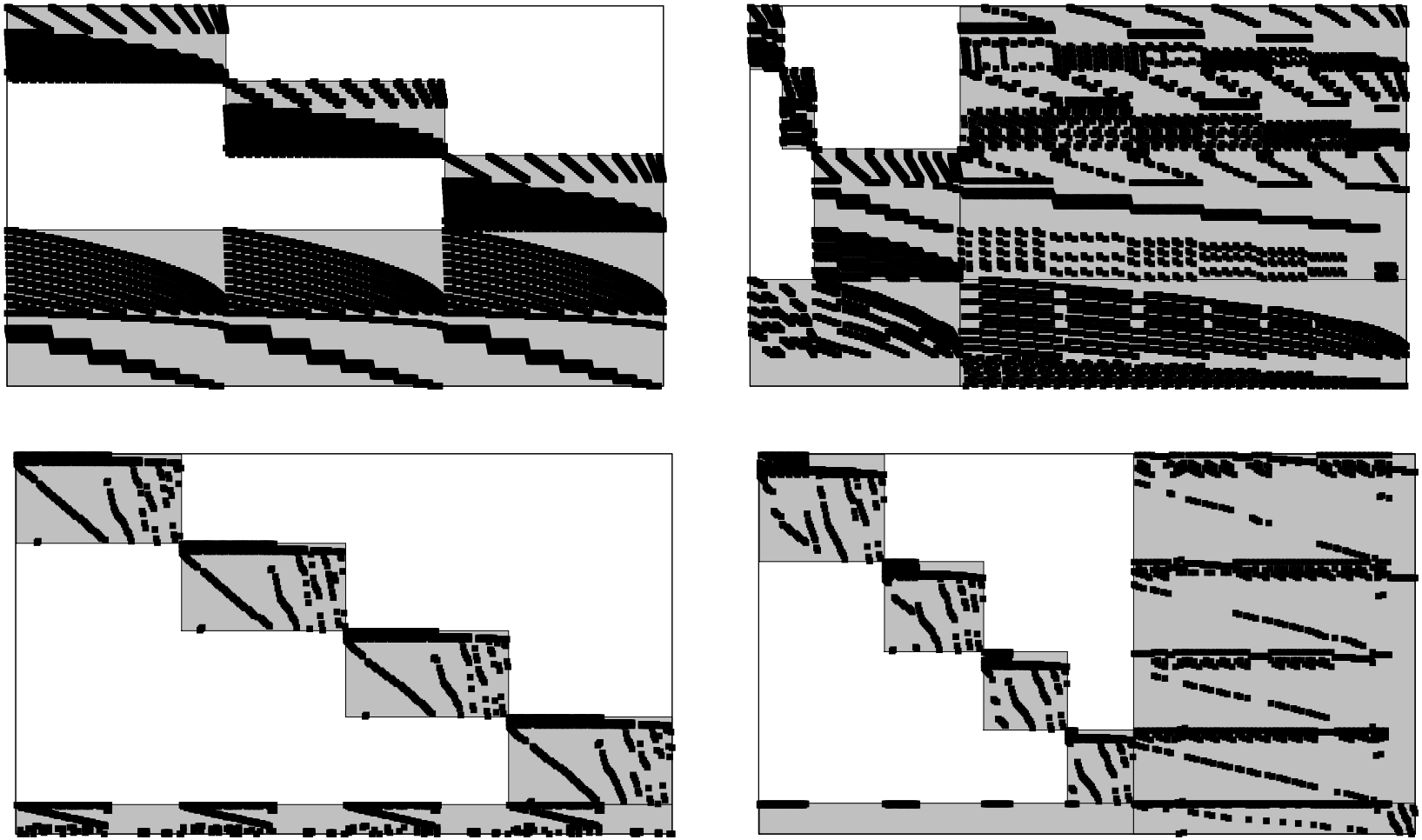
What is a "good" Decomposition: Blocks

- influence of the number k of blocks on the "shape" (*fiber*)



What is a "good" Decomposition: Visual

- visual differences for different decompositions (*10teams*, *rout*)



Quality of a Decomposition

- caveat! don't think there is structure only b/c you see it
- a matrix can be permuted in many ways
- measure for the quality of a decomposition is needed
- this is a theoretically very challenging question
- here we used heuristic quality measures like "border area"

MIPLIBs: Instance Selection

Selected MIPLIB 2003 and 2010 instances for which

- the optimum is known
- the density is between 0.05% and 5%
- the number of non-zeros is not larger than 20,000
- the percentage of discrete variables is at least 20%

MIPLIB2003: Quality in Root Node

						LP	DWR automatic		DWR manual		CPLEX+cuts	
instance	rows	cols	k	ℓ	c	gap	gap	closed	gap	closed	gap	closed
10teams	2025	230	4	0	107	0.76	0.00	**100.00**	0.00	**100.00**	0.00	**100.00**
aflow30a	842	479	2	0	28	15.10	14.71	2.59	10.23	32.27	5.35	**64.57**
aflow40b	2728	1442	5	0	39	13.90	13.91	0.00	13.91	0.00	6.47	**53.44**
fiber	1298	363	2	2	21	61.55	1.07	**98.27**	1.07	**98.27**	1.77	97.13
fixnet6	878	478	4	3	14	69.85	18.89	72.96	18.89	72.96	6.06	**91.33**
gesa2-o	1224	1248	5	65	0	1.18	0.07	**94.17**	0.05	**96.16**	0.10	91.51
gesa2	1224	1392	3	65	0	1.18	0.06	**95.16**	0.03	**97.82**	0.21	82.43
glass4	322	396	3	16	0	33.33	25.12	**24.63**	25.12	**24.63**	33.33	0.00
harp2	2993	112	5	0	39	0.61	0.66	0.00	0.65	0.00	0.37	**39.61**
manna81	3321	6480	2	78	0	1.01	0.08	92.05	0.08	92.05	0.00	**100.00**
mkc	5325	3411	2	0	29	8.51	0.16	**98.13**	0.16	**98.13**	3.77	55.68
modglob	422	291	2	18	3	1.49	0.77	48.11	0.12	**91.74**	0.14	90.94
noswot	128	182	5	21	3	4.88	0.49	**90.00**	0.49	**90.00**	4.88	0.00
opt1217	769	64	4	0	16	25.13	25.13	0.00	25.13	0.00	0.00	**100.00**
p2756	2756	755	4	39	13	13.93	0.34	**97.59**	0.27	**98.07**	5.65	59.42
pp08a	240	136	2	16	0	62.61	2.27	**96.83**	2.27	**96.83**	2.53	95.97
pp08aCUTS	240	246	2	16	0	25.43	2.27	**91.09**	2.27	**91.09**	3.81	85.03
rout	556	291	5	0	16	8.88	0.68	**92.34**	0.68	**92.34**	8.86	0.26
set1ch	712	492	3	20	8	41.31	2.84	93.14	2.13	94.84	0.92	**97.78**
timtab1	397	171	2	13	0	96.25	14.60	**84.84**	14.60	**84.84**	39.05	59.43
timtab2	675	294	4	25	0	92.38	39.83	**56.89**	25.26	**72.66**	46.00	50.20
tr12-30	1080	750	3	24	0	89.12	93.47	0.00	2.83	96.83	0.68	**99.24**
vpm2	378	234	2	7	0	28.08	1.71	**93.92**	1.71	**93.92**	6.44	77.05
arithm. mean						30.28	11.27	66.18	**6.43**	**74.56**	7.67	69.17

MIPLIB2003: Quality in Root Node

instance	rows	cols	k	ℓ	c	LP gap	DWR automatic gap	DWR automatic closed	DWR manual gap	DWR manual closed	CPLEX+cuts gap	CPLEX+cuts closed
10teams	2025	230	4	0	107	0.76	0.00	**100.00**	0.00	**100.00**	0.00	**100.00**
aflow30a	842	479	2	0	28	15.10	14.71	2.59	10.23	32.27	5.35	**64.57**
aflow40b	2728	1442	5	0	39	13.90	13.91	0.00	13.91	0.00	6.47	**53.44**
fiber	1298	363	2	2	21	61.55	1.07	**98.27**	1.07	**98.27**	1.77	97.13
fixnet6	878	478	4	3	14	69.85	18.89	72.96	18.89	72.96	6.06	**91.33**
gesa2-o	1224	1248	5	65	0	1.18	0.07	**94.17**	0.05	**96.16**	0.10	91.51
gesa2	1224	1392	3	65	0	1.18	0.06	**95.16**	0.03	**97.82**	0.21	82.43
glass4	322	396	3	16	0	33.33	25.12	**24.63**	25.12	**24.63**	33.33	0.00
harp2	2993	112	5	0	39	0.61	0.66	0.00	0.65	0.00	0.37	**39.61**
manna81	3321	6480	2	78	0	1.01	0.08	92.05	0.08	92.05	0.00	**100.00**
mkc	5325	3411	2	0	29	8.51	0.16	**98.13**	0.16	**98.13**	3.77	55.68
modglob	422	291	2	18	3	1.49	0.77	48.11	0.12	**91.74**	0.14	90.94
noswot	128	182	5	21	3	4.88	0.49	**90.00**	0.49	**90.00**	4.88	0.00
opt1217	769	64	4	0	16	25.13	25.13	0.00	25.13	0.00	0.00	**100.00**
p2756	2756	755	4	39	13	13.93	0.34	**97.59**	0.27	**98.07**	5.65	59.42
pp08a	240	136	2	16	0	62.61	2.27	**96.83**	2.27	**96.83**	2.53	95.97
pp08aCUTS	240	246	2	16	0	25.43	2.27	**91.09**	2.27	**91.09**	3.81	85.03
rout	556	291	5	0	16	8.88	0.68	**92.34**	0.68	**92.34**	8.86	0.26
set1ch	712	492	3	20	8	41.31	2.84	93.14	2.13	94.84	0.92	**97.78**
timtab1	397	171	2	13	0	96.25	14.60	**84.84**	14.60	**84.84**	39.05	59.43
timtab2	675	294	4	25	0	92.38	39.83	**56.89**	25.26	**72.66**	46.00	50.20
tr12-30	1080	750	3	24	0	89.12	93.47	0.00	2.83	96.83	0.68	**99.24**
vpm2	378	234	2	7	0	28.08	1.71	**93.92**	1.71	**93.92**	6.44	77.05
arithm. mean						30.28	11.27	66.18	**6.43**	**74.56**	7.67	69.17

MIPLIB2003: Quality in Root Node

instance	rows	cols	k	ℓ	c	LP gap	DWR automatic gap	DWR automatic closed	DWR manual gap	DWR manual closed	CPLEX+cuts gap	CPLEX+cuts closed
10teams	2025	230	4	0	107	0.76	0.00	**100.00**	0.00	**100.00**	0.00	**100.00**
aflow30a	842	479	2	0	28	15.10	14.71	2.59	10.23	32.27	5.35	**64.57**
aflow40b	2728	1442	5	0	39	13.90	13.91	0.00	13.91	0.00	6.47	**53.44**
fiber	1298	363	2	2	21	61.55	1.07	**98.27**	1.07	**98.27**	1.77	97.13
fixnet6	878	478	4	3	14	69.85	18.89	72.96	18.89	72.96	6.06	**91.33**
gesa2-o	1224	1248	5	65	0	1.18	0.07	**94.17**	0.05	**96.16**	0.10	91.51
gesa2	1224	1392	3	65	0	1.18	0.06	**95.16**	0.03	**97.82**	0.21	82.43
glass4	322	396	3	16	0	33.33	25.12	**24.63**	25.12	**24.63**	33.33	0.00
harp2	2993	112	5	0	39	0.61	0.66	0.00	0.65	0.00	0.37	**39.61**
manna81	3321	6480	2	78	0	1.01	0.08	92.05	0.08	92.05	0.00	**100.00**
mkc	5325	3411	2	0	29	8.51	0.16	**98.13**	0.16	**98.13**	3.77	55.68
modglob	422	291	2	18	3	1.49	0.77	48.11	0.12	**91.74**	0.14	90.94
noswot	128	182	5	21	3	4.88	0.49	**90.00**	0.49	**90.00**	4.88	0.00
opt1217	769	64	4	0	16	25.13	25.13	0.00	25.13	0.00	0.00	**100.00**
p2756	2756	755	4	39	13	13.93	0.34	**97.59**	0.27	**98.07**	5.65	59.42
pp08a	240	136	2	16	0	62.61	2.27	**96.83**	2.27	**96.83**	2.53	95.97
pp08aCUTS	240	246	2	16	0	25.43	2.27	**91.09**	2.27	**91.09**	3.81	85.03
rout	556	291	5	0	16	8.88	0.68	**92.34**	0.68	**92.34**	8.86	0.26
set1ch	712	492	3	20	8	41.31	2.84	93.14	2.13	94.84	0.92	**97.78**
timtab1	397	171	2	13	0	96.25	14.60	**84.84**	14.60	**84.84**	39.05	59.43
timtab2	675	294	4	25	0	92.38	39.83	**56.89**	25.26	**72.66**	46.00	50.20
tr12-30	1080	750	3	24	0	89.12	93.47	0.00	2.83	96.83	0.68	**99.24**
vpm2	378	234	2	7	0	28.08	1.71	**93.92**	1.71	**93.92**	6.44	77.05
arithm. mean						30.28	11.27	66.18	**6.43**	**74.56**	7.67	69.17

MIPLIB2003: Quality in Root Node

instance	rows	cols	k	ℓ	c	LP gap	DWR automatic gap	DWR automatic closed	DWR manual gap	DWR manual closed	CPLEX+cuts gap	CPLEX+cuts closed
10teams	2025	230	4	0	107	0.76	0.00	**100.00**	0.00	**100.00**	0.00	**100.00**
aflow30a	842	479	2	0	28	15.10	14.71	2.59	10.23	32.27	5.35	**64.57**
aflow40b	2728	1442	5	0	39	13.90	13.91	0.00	13.91	0.00	6.47	**53.44**
fiber	1298	363	2	2	21	61.55	1.07	**98.27**	1.07	**98.27**	1.77	97.13
fixnet6	878	478	4	3	14	69.85	18.89	72.96	18.89	72.96	6.06	**91.33**
gesa2-o	1224	1248	5	65	0	1.18	0.07	**94.17**	0.05	**96.16**	0.10	91.51
gesa2	1224	1392	3	65	0	1.18	0.06	**95.16**	0.03	**97.82**	0.21	82.43
glass4	322	396	3	16	0	33.33	25.12	**24.63**	25.12	**24.63**	33.33	0.00
harp2	2993	112	5	0	39	0.61	0.66	0.00	0.65	0.00	0.37	**39.61**
manna81	3321	6480	2	78	0	1.01	0.08	92.05	0.08	92.05	0.00	**100.00**
mkc	5325	3411	2	0	29	8.51	0.16	**98.13**	0.16	**98.13**	3.77	55.68
modglob	422	291	2	18	3	1.49	0.77	48.11	0.12	**91.74**	0.14	90.94
noswot	128	182	5	21	3	4.88	0.49	**90.00**	0.49	**90.00**	4.88	0.00
opt1217	769	64	4	0	16	25.13	25.13	0.00	25.13	0.00	0.00	**100.00**
p2756	2756	755	4	39	13	13.93	0.34	**97.59**	0.27	**98.07**	5.65	59.42
pp08a	240	136	2	16	0	62.61	2.27	**96.83**	2.27	**96.83**	2.53	95.97
pp08aCUTS	240	246	2	16	0	25.43	2.27	**91.09**	2.27	**91.09**	3.81	85.03
rout	556	291	5	0	16	8.88	0.68	**92.34**	0.68	**92.34**	8.86	0.26
set1ch	712	492	3	20	8	41.31	2.84	93.14	2.13	94.84	0.92	**97.78**
timtab1	397	171	2	13	0	96.25	14.60	**84.84**	14.60	**84.84**	39.05	59.43
timtab2	675	294	4	25	0	92.38	39.83	**56.89**	25.26	**72.66**	46.00	50.20
tr12-30	1080	750	3	24	0	89.12	93.47	0.00	2.83	96.83	0.68	**99.24**
vpm2	378	234	2	7	0	28.08	1.71	**93.92**	1.71	**93.92**	6.44	77.05
arithm. mean						30.28	11.27	66.18	**6.43**	**74.56**	7.67	69.17

MIPLIB2003: Quality in Root Node

instance	rows	cols	k	ℓ	c	LP gap	DWR automatic gap	DWR automatic closed	DWR manual gap	DWR manual closed	CPLEX+cuts gap	CPLEX+cuts closed
10teams	2025	230	4	0	107	0.76	0.00	**100.00**	0.00	**100.00**	0.00	**100.00**
aflow30a	842	479	2	0	28	15.10	14.71	2.59	10.23	32.27	5.35	**64.57**
aflow40b	2728	1442	5	0	39	13.90	13.91	0.00	13.91	0.00	6.47	**53.44**
fiber	1298	363	2	2	21	61.55	1.07	**98.27**	1.07	**98.27**	1.77	97.13
fixnet6	878	478	4	3	14	69.85	18.89	72.96	18.89	72.96	6.06	**91.33**
gesa2-o	1224	1248	5	65	0	1.18	0.07	**94.17**	0.05	**96.16**	0.10	91.51
gesa2	1224	1392	3	65	0	1.18	0.06	**95.16**	0.03	**97.82**	0.21	82.43
glass4	322	396	3	16	0	33.33	25.12	**24.63**	25.12	**24.63**	33.33	0.00
harp2	2993	112	5	0	39	0.61	0.66	0.00	0.65	0.00	0.37	**39.61**
manna81	3321	6480	2	78	0	1.01	0.08	92.05	0.08	92.05	0.00	**100.00**
mkc	5325	3411	2	0	29	8.51	0.16	**98.13**	0.16	**98.13**	3.77	55.68
modglob	422	291	2	18	3	1.49	0.77	48.11	0.12	**91.74**	0.14	90.94
noswot	128	182	5	21	3	4.88	0.49	**90.00**	0.49	**90.00**	4.88	0.00
opt1217	769	64	4	0	16	25.13	25.13	0.00	25.13	0.00	0.00	**100.00**
p2756	2756	755	4	39	13	13.93	0.34	**97.59**	0.27	**98.07**	5.65	59.42
pp08a	240	136	2	16	0	62.61	2.27	**96.83**	2.27	**96.83**	2.53	95.97
pp08aCUTS	240	246	2	16	0	25.43	2.27	**91.09**	2.27	**91.09**	3.81	85.03
rout	556	291	5	0	16	8.88	0.68	**92.34**	0.68	**92.34**	8.86	0.26
set1ch	712	492	3	20	8	41.31	2.84	93.14	2.13	94.84	0.92	**97.78**
timtab1	397	171	2	13	0	96.25	14.60	**84.84**	14.60	**84.84**	39.05	59.43
timtab2	675	294	4	25	0	92.38	39.83	**56.89**	25.26	**72.66**	46.00	50.20
tr12-30	1080	750	3	24	0	89.12	93.47	0.00	2.83	96.83	0.68	**99.24**
vpm2	378	234	2	7	0	28.08	1.71	**93.92**	1.71	**93.92**	6.44	77.05
arithm. mean						30.28	11.27	66.18	**6.43**	**74.56**	7.67	69.17

MIPLIB2010: Quality in Root Node

						LP	DWR automatic		DWR manual		CPLEX+cuts	
instance	rows	cols	k	ℓ	c	gap	gap	closed	gap	closed	gap	closed
beasleyC3	1750	2500	3	52	5	94.64	20.95	77.86	20.95	77.86	0.08	**91.29**
csched010	351	1758	2	1	55	18.52	7.91	**57.28**	7.91	**57.28**	12.62	31.86
enlight13	169	338	2	25	0	100.00	57.55	**42.45**	57.55	**42.45**	80.66	19.34
gmu-35-40	424	1205	2	0	9	0.01	0.03	0.00	0.03	0.00	0.01	**2.54**
m100n500k4r1	100	500	2	11	86	4.17	4.17	**0.00**	4.17	**0.00**	4.17	**0.00**
macrophage	3164	2260	2	7	0	100.00	0.37	**99.63**	0.05	**99.95**	11.48	88.52
mcsched	2107	1747	2	69	2	8.56	1.03	**87.91**	1.03	**87.91**	8.54	0.23
mine-166-5	8429	830	2	166	1320	45.09	9.78	**78.30**	0.00	**100.00**	22.79	49.44
mine-90-10	6270	900	2	0	1264	13.12	6.90	**47.39**	6.71	**48.81**	8.73	33.41
neos-686190	3664	3660	3	59	179	23.70	100.00	0.00	100.00	0.00	21.54	**9.11**
pigeon-10	931	490	2	96	40	11.11	11.11	**0.00**	11.11	**0.00**	11.11	**0.00**
pw-myciel4	8164	1059	2	24	49	100.00	60.00	40.00	0.00	**100.00**	53.54	46.46
ran16x16	288	512	2	0	16	18.48	10.86	41.27	10.82	41.44	6.93	**62.51**
reblock67	2523	670	2	67	112	13.61	1.87	**86.23**	1.87	**86.23**	8.87	34.86
rmine6	7078	1096	2	186	926	1.12	3.22	0.00	1.89	0.00	0.98	**12.26**
rococoC10-001000	1293	3117	2	0	82	34.42	34.43	0.00	34.42	0.00	11.62	**66.25**
arithm. mean						36.66	20.64	41.14	16.16	46.37	16.48	34.26

MIPLIB2010: Quality in Root Node

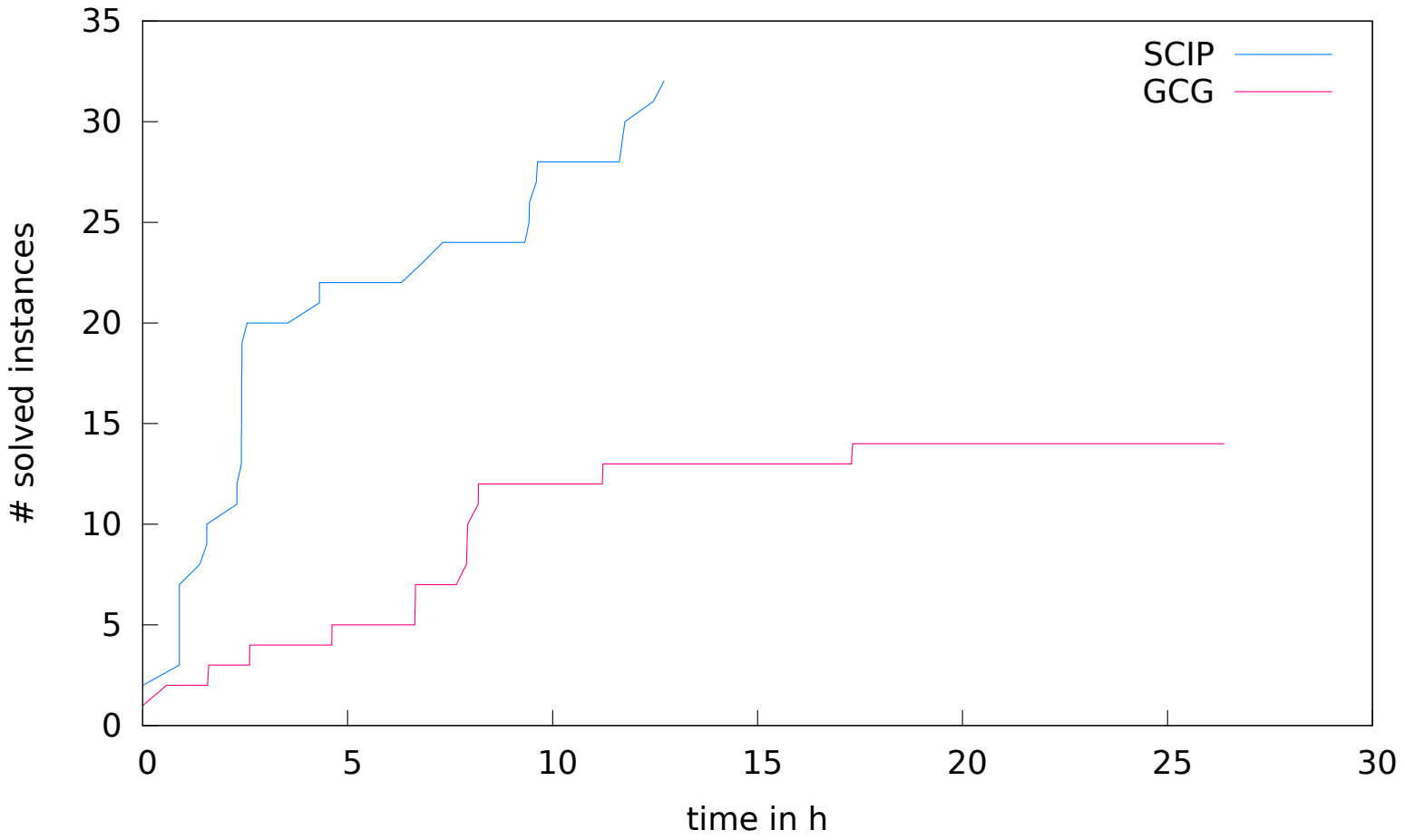
						LP	DWR automatic		DWR manual		CPLEX+cuts	
instance	rows	cols	k	ℓ	c	gap	gap	closed	gap	closed	gap	closed
beasleyC3	1750	2500	3	52	5	94.64	20.95	77.86	20.95	77.86	0.08	**91.29**
csched010	351	1758	2	1	55	18.52	7.91	**57.28**	7.91	**57.28**	12.62	31.86
enlight13	169	338	2	25	0	100.00	57.55	**42.45**	57.55	**42.45**	80.66	19.34
gmu-35-40	424	1205	2	0	9	0.01	0.03	0.00	0.03	0.00	0.01	**2.54**
m100n500k4r1	100	500	2	11	86	4.17	4.17	**0.00**	4.17	**0.00**	4.17	**0.00**
macrophage	3164	2260	2	7	0	100.00	0.37	**99.63**	0.05	**99.95**	11.48	88.52
mcsched	2107	1747	2	69	2	8.56	1.03	**87.91**	1.03	**87.91**	8.54	0.23
mine-166-5	8429	830	2	166	1320	45.09	9.78	**78.30**	0.00	**100.00**	22.79	49.44
mine-90-10	6270	900	2	0	1264	13.12	6.90	**47.39**	6.71	**48.81**	8.73	33.41
neos-686190	3664	3660	3	59	179	23.70	100.00	0.00	100.00	0.00	21.54	**9.11**
pigeon-10	931	490	2	96	40	11.11	11.11	**0.00**	11.11	**0.00**	11.11	**0.00**
pw-myciel4	8164	1059	2	24	49	100.00	60.00	40.00	0.00	**100.00**	53.54	46.46
ran16x16	288	512	2	0	16	18.48	10.86	41.27	10.82	41.44	6.93	**62.51**
reblock67	2523	670	2	67	112	13.61	1.87	**86.23**	1.87	**86.23**	8.87	34.86
rmine6	7078	1096	2	186	926	1.12	3.22	0.00	1.89	0.00	0.98	**12.26**
rococoC10-001000	1293	3117	2	0	82	34.42	34.43	0.00	34.42	0.00	11.62	**66.25**
arithm. mean						36.66	20.64	41.14	16.16	46.37	16.48	34.26

MIPLIB2010: Quality in Root Node

						LP	DWR automatic		DWR manual		CPLEX+cuts	
instance	rows	cols	k	ℓ	c	gap	gap	closed	gap	closed	gap	closed
beasleyC3	1750	2500	3	52	5	94.64	20.95	77.86	20.95	77.86	0.08	**91.29**
csched010	351	1758	2	1	55	18.52	7.91	**57.28**	7.91	**57.28**	12.62	31.86
enlight13	169	338	2	25	0	100.00	57.55	**42.45**	57.55	**42.45**	80.66	19.34
gmu-35-40	424	1205	2	0	9	0.01	0.03	0.00	0.03	0.00	0.01	**2.54**
m100n500k4r1	100	500	2	11	86	4.17	4.17	**0.00**	4.17	**0.00**	4.17	**0.00**
macrophage	3164	2260	2	7	0	100.00	0.37	**99.63**	0.05	**99.95**	11.48	88.52
mcsched	2107	1747	2	69	2	8.56	1.03	**87.91**	1.03	**87.91**	8.54	0.23
mine-166-5	8429	830	2	166	1320	45.09	9.78	**78.30**	0.00	**100.00**	22.79	49.44
mine-90-10	6270	900	2	0	1264	13.12	6.90	**47.39**	6.71	**48.81**	8.73	33.41
neos-686190	3664	3660	3	59	179	23.70	100.00	0.00	100.00	0.00	21.54	**9.11**
pigeon-10	931	490	2	96	40	11.11	11.11	**0.00**	11.11	**0.00**	11.11	**0.00**
pw-myciel4	8164	1059	2	24	49	100.00	60.00	40.00	0.00	**100.00**	53.54	46.46
ran16x16	288	512	2	0	16	18.48	10.86	41.27	10.82	41.44	6.93	**62.51**
reblock67	2523	670	2	67	112	13.61	1.87	**86.23**	1.87	**86.23**	8.87	34.86
rmine6	7078	1096	2	186	926	1.12	3.22	0.00	1.89	0.00	0.98	**12.26**
rococoC10-001000	1293	3117	2	0	82	34.42	34.43	0.00	34.42	0.00	11.62	**66.25**
arithm. mean						36.66	20.64	41.14	16.16	46.37	16.48	34.26

MIPLIBs: Computation Time

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



General IPs: These are the wrong Instances!

- we wish to exploit *structure* in IPs,
  so we need to consider structured IPs!

General IPs: These are the wrong Instances!

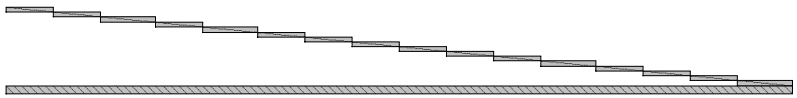
- we wish to exploit *structure* in IPs,
  so we need to consider structured IPs!
- to this end, we are currently building the `strIPlib`
- planned as a web portal as source of information and sandbox
- bin packing, cutting stock, vertex coloring, p-median,
  generalized assignment, vehicle routing, machine scheduling,
  your instances here, ...

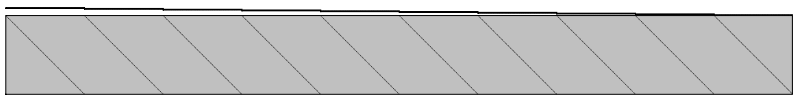
General IPs: These are the wrong Instances!

- we wish to exploit *structure* in IPs,
  so we need to consider structured IPs!
- to this end, we are currently building the `strIPlib`
- planned as a web portal as source of information and sandbox
- bin packing, cutting stock, vertex coloring, p-median, generalized assignment, vehicle routing, machine scheduling, your instances here, ...
- rather small subset for this talk:
  - all N1*, N2* instances from bison 1 binpacking set (360 instances)
  - selected instances from
    - GAP: orlibrary and Yagiura testset (16 instances)
    - cut. stock: Schwerin, Waescher and hard28 (12 instances)
    - cap. p-median: optlab and orlibrary (6 instances)
    - vertex coloring: r-set (4 instances)
  - criterion: solved in less than 600s, varying in nodes and size

Detecting Structure in Structured IPs

- application of hypergraph approach fails surprisingly often!





- # blocks for *gapa_3*: `hmetis`: 16, "natural/best:" 10

Detecting Structure in Structured IPs

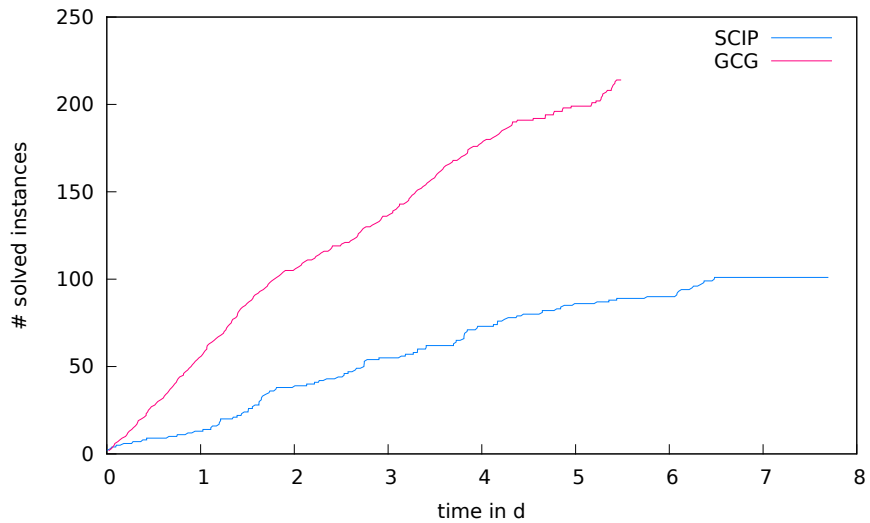
- better, easier, faster detection based on constraint types
  e.g., SPP constraints → master, block detection on remainder

	hmetis			best		
	blocks	nodes	time	blocks	nodes	time
gapa_3	16	1	0.8	10	1	6.0
gapc_5	16	1217	57.8	20	27	6.8
N1C3W4_E	16	≈ 26k	5.41%	50	1	9.2
N1C3W4_L	16	≈ 25k	6.20%	50	22	14.3
out15-1	—	—	—	100	271	212.8
out18-1	—	—	—	100	503	264.9
R50_1g	9	1	47.5	9	6580	7.7
R75_1gb	13	1	411.6	13	10426	228.9
TEST0005	—	—	—	57	59	553.0
TEST0084	—	—	—	46	454	137.1

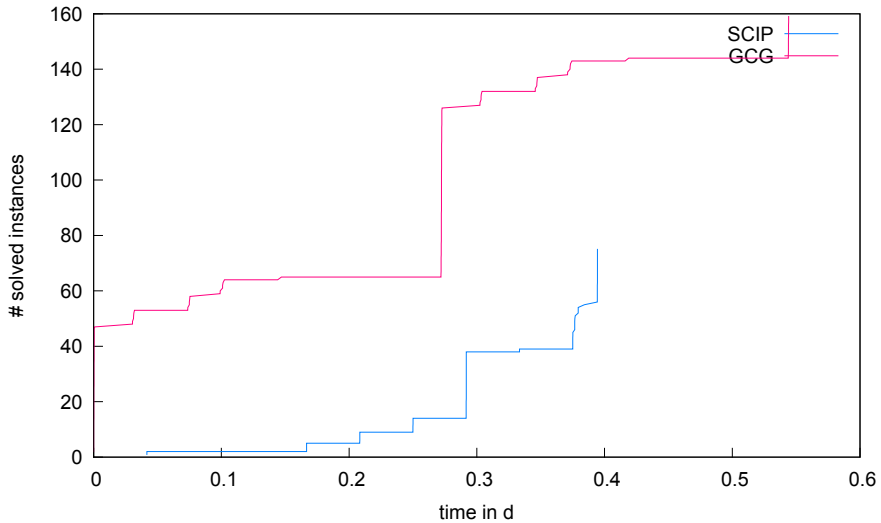
- ample room for (heuristic) improvements

Computation Times for Structured Problems

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



cutting stock



generalized assignment

Benefits from this Approach

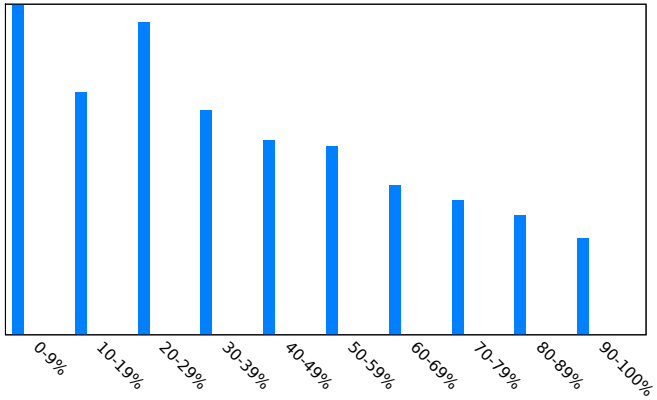
- use a *single* generic implementation on a large number of classical and non-classical applications
- collect all kinds of data and compare in unified way

- caveat!

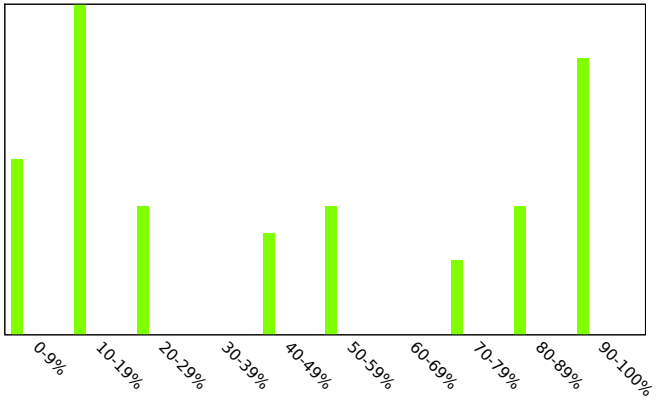
  the following figures only serve the purpose of illustration!

When do we Generate the "Important Columns"?

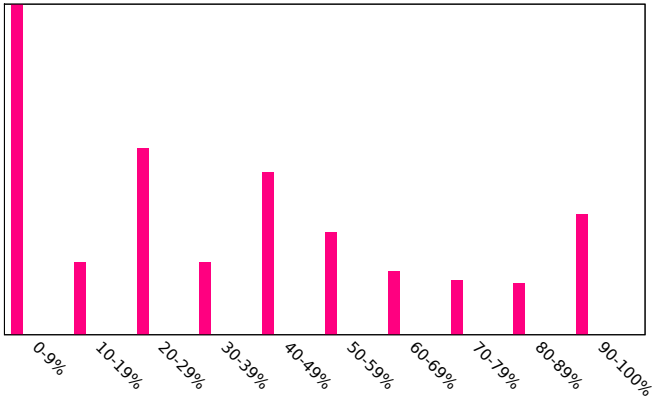
- columns that contribute to the best IP solution



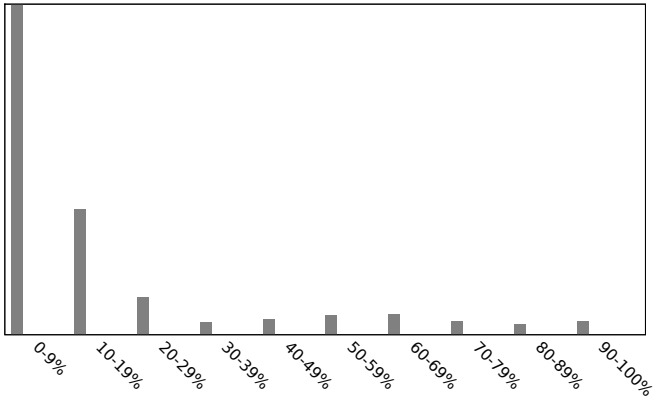
bin packing



vertex coloring



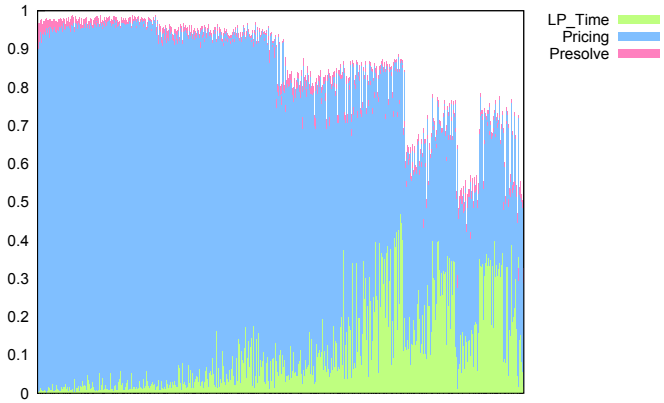
cutting stock



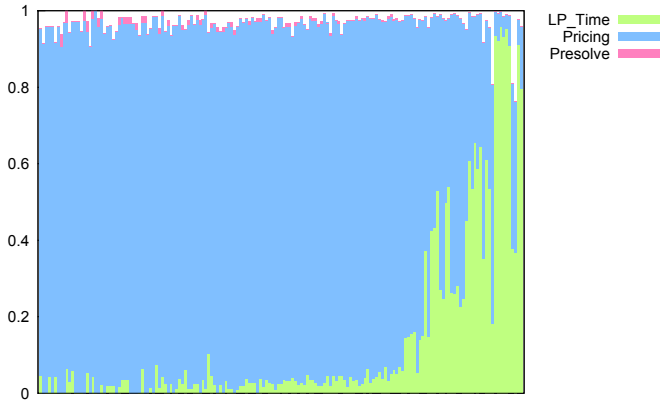
capacitated p-median

Where do we Spend the Computation Time?

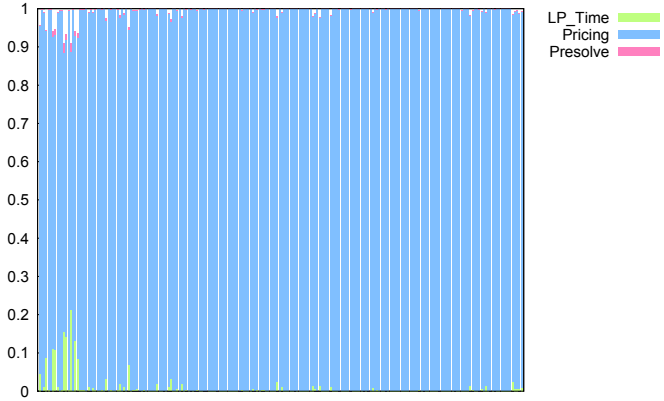
- instances sorted according to computation time



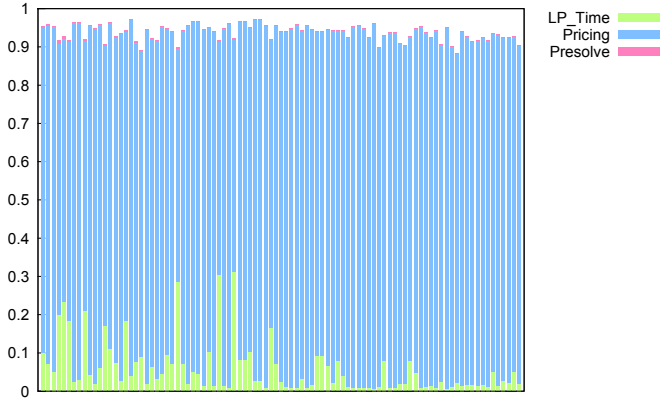
bin packing



generalized assignment



cutting stock



capacitated p-median

Much more is Collected

- degree of degeneracy
- behavior of dual variables
- how do the primal/dual bounds develop in the tree
- which heuristics work well
- effectiveness of features like pseudo-costs, propagation, ...
- pictures/statistics about different decompositions

  ⋮
- this information available about *your instance* via `strIPlib`

Benefits from Building on a State-of-the-Art Solver

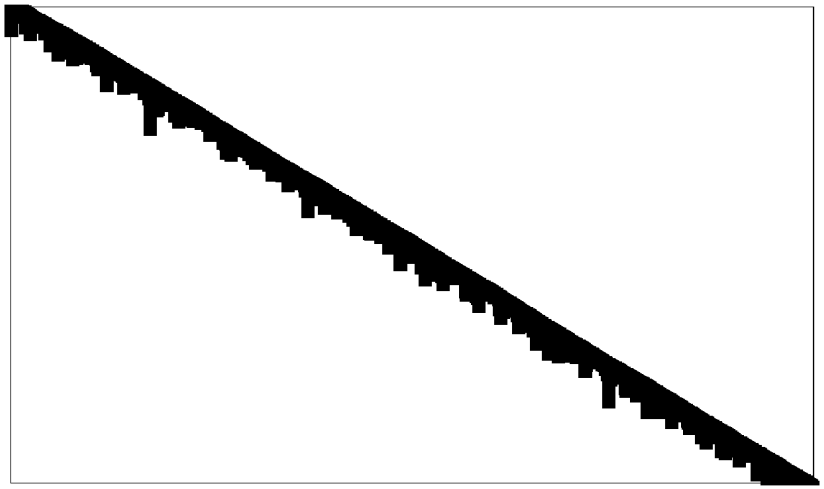
- e.g., presolve (-19% time) and cutting planes (-3% time)

	GCG default settings		no presolve		no cuts	
	Nodes	Time	Nodes	Time	Nodes	Time
10teams	180	32.1	537	170.5	180	37.4
aflow30a	33 246	2064.2	>490	<12.0%	>65 105	<7.3%
aflow40b	>2 521	<41.9%	>4 400	<22.1%	>9 554	>3600.0
fiber	19	95.6	44	151.4	19	104.5
fixnet6	>543	<4.1%	>751	<5.5%	2 111	1006.6
gesa2	1	4.5	1	7.3	1	4.4
gesa2-o	>578	<1.2%	>475	<0.8%	>238	<2.9%
glass4	>67	>3615.7	—	—	>67	>3951.3
harp2	>7	>3672.0	>11 645	<0.5%	>7	>3672.0
manna81	>1	<445.6%	>8	<68.0%	>1	<445.6%
mkc	>100	<5.7%	>100	<40.2%	>485	<7.8%
modglob	7	48.0	—	—	23	60.0
noswot	1	4.1	1	1.2	1	2.1
opt1217	>207 599	<25.0%	>71 917	<25.1%	>253 938	<25.0%
p2756	4 942	892.5	>9 855	<0.1%	>42 006	<0.1%
pp08a	549	50.5	425	42.6	398	36.8
pp08aCUTS	409	41.5	609	58.8	465	50.7
rout	2 111	932.6	2 493	1565.5	2 449	944.2
set1ch	69	17.0	84	31.9	69	26.6
timtab1	>207	>3600.9	>271	>3700.2	>199	>3600.4
timtab2	>21	>3604.0	>15	>3600.7	>32	>3600.2
tr12-30	—	—	>1	>3673.7	—	—
vpm2	131	55.4	371	380.1	131	88.2

— did not finish Farkas pricing, had numerical problems, or a bug

More to Detect!

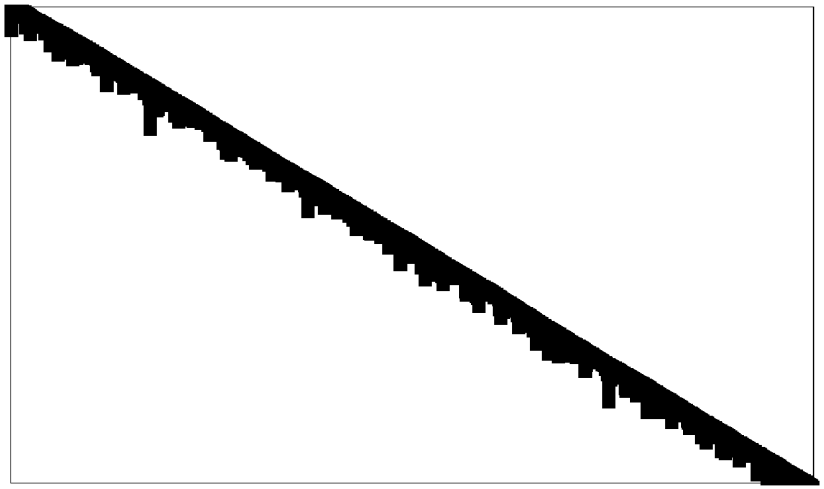
- e.g., problems "indexed over time"



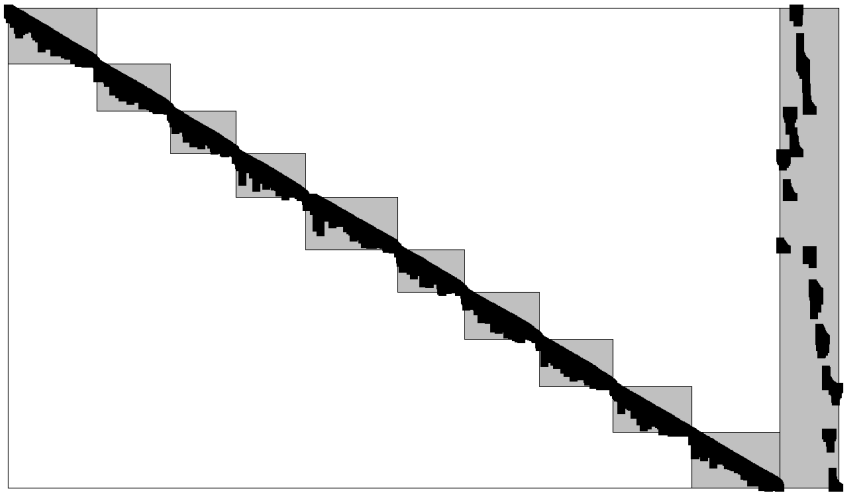
canonical

More to Detect!

- e.g., problems "indexed over time"



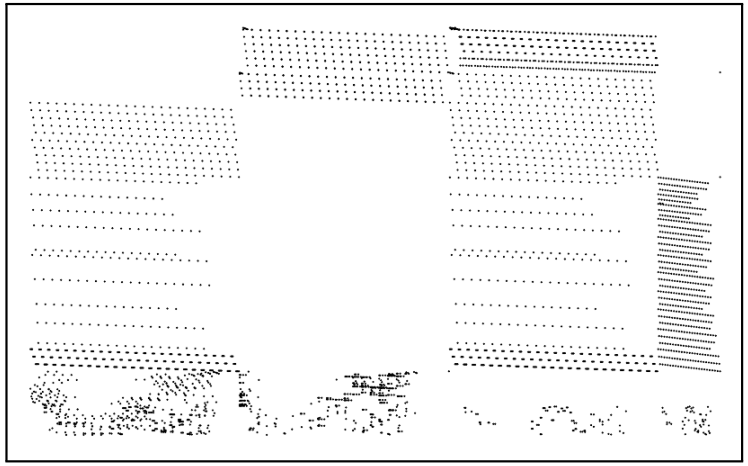
canonical



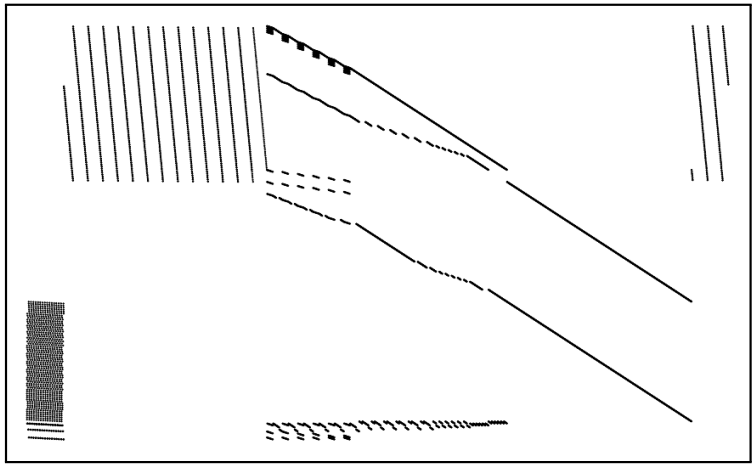
"detected"

- detection $\sim$ "structure enforcement"
- reformulation essentially convexifies all constraints
- gives *very* tight relaxations

More to Detect!

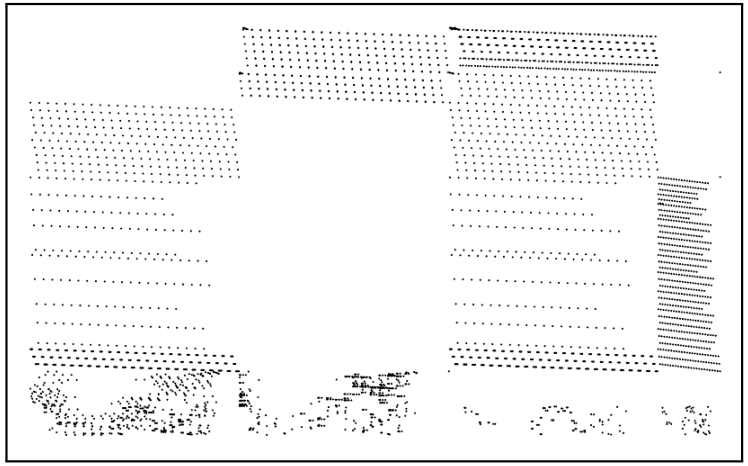


bg512142 original

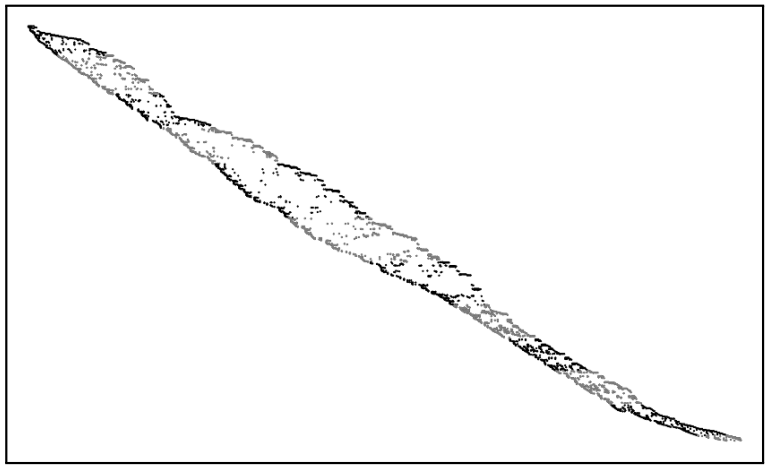


a1c1s1 original

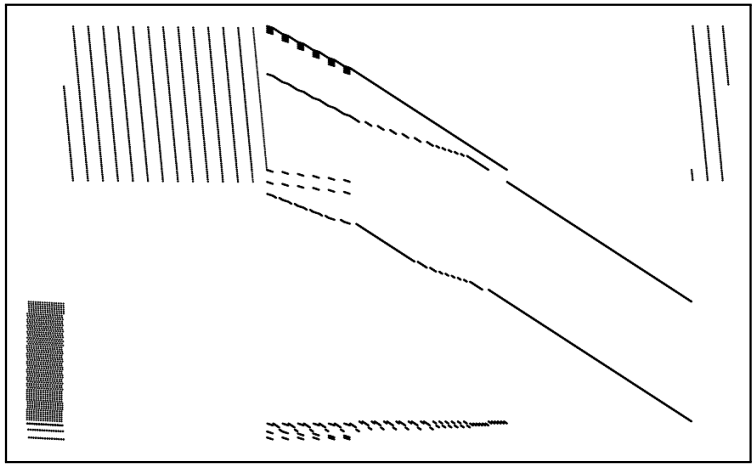
More to Detect!



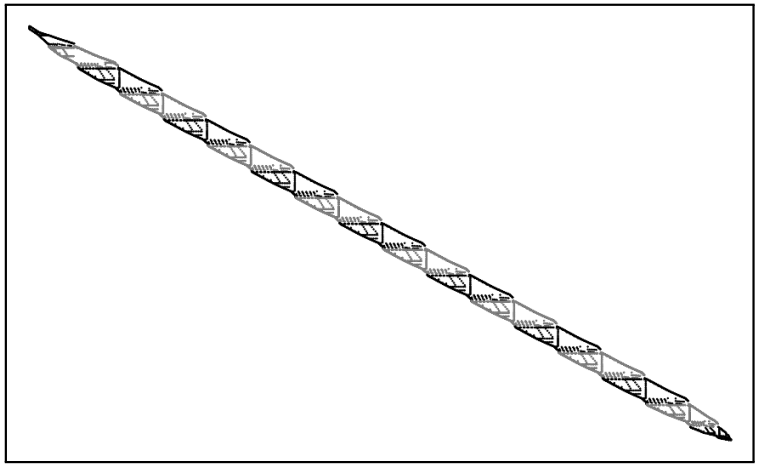
bg512142 original



staircase detected



a1c1s1 original



staircase detected

Feature Request and Call for Instances

- if you were to use a generic BP&C solver:
  what features would you like to see?
- if you like the solver to work well on your problems:
  send us your instances!

Take-Home

- automatic decomposition and branch-price-and-cut doable
- lucky for “unstructured” IPs, successful on “structured”
- challenge: tell promising from unpromising instances
- ideally, this will complement existing state-of-the-art solvers

Take-Home

- automatic decomposition and branch-price-and-cut doable
- lucky for "unstructured" IPs, successful on "structured"
- challenge: tell promising from unpromising instances
- ideally, this will complement existing state-of-the-art solvers

- `GCG` 1.0 to be released within "SCIPoptsuite" at ISMP2012
- `strIPlib` will be made public for you to play (and learn)

Take-Home

- automatic decomposition and branch-price-and-cut doable
- lucky for "unstructured" IPs, successful on "structured"
- challenge: tell promising from unpromising instances
- ideally, this will complement existing state-of-the-art solvers

- `GCG` 1.0 to be released within "SCIPoptsuite" at ISMP2012
- `strIPlib` will be made public for you to play (and learn)

- here are hiring!
  research grant (postdoc) sponsored by Google

August 19-24

www.ismp2012.org