Learning how to decompose

Markus Kruber · Marco Lübbecke · Axel Parmentier

September 6, 2017

Operations Research
RWTH AACHEN UNIVERSITY

Scope

$$\begin{array}{ll} \min & cx \\ \text{s.t.} & Ax = b \\ & x \in \mathbb{R}^n_+ \times \mathbb{Z}^p_+ \end{array}$$

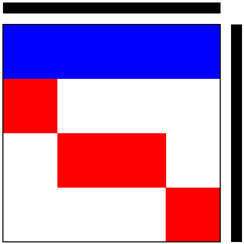


Figure: Decomposition of A

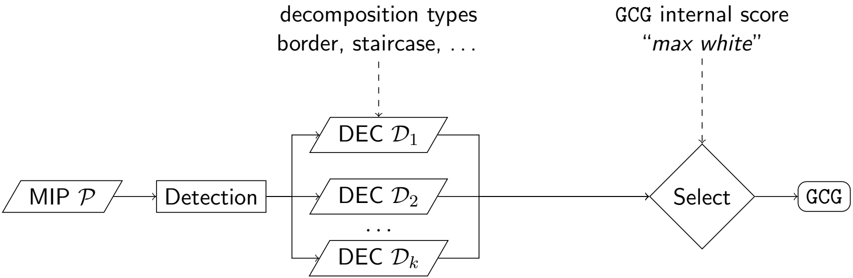
Motivation

GCG performance highly depends on how well the decomposition catches the problem structure.

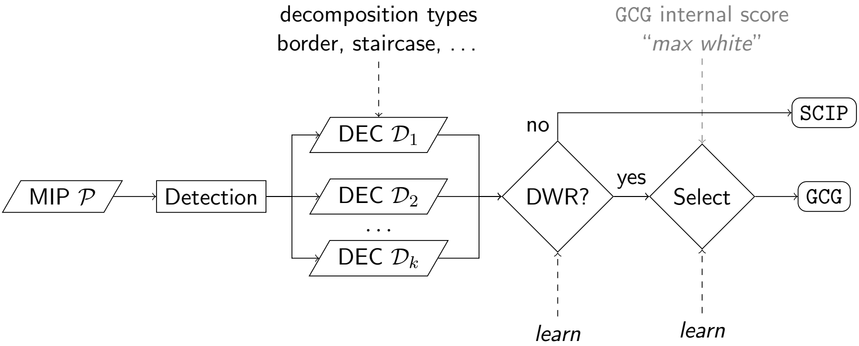
In comparison to the previous talk:

- ~~Finding~~ Evaluating
- ~~Intuition~~ Learning

Automatic Decomposition in GCG



Automatic Decomposition in GCG



Our work: a supervised learning approach to *select the best decomposition* (using no decomposition is often the best answer).

Operations Research

RWTH AACHEN UNIVERSITY

Table of Content

Supervised Learning

Given an *input* X, a *classifier* is a model (function) f that predicts a *variable of interest* $Y \in S$

$$Y = f(X)$$

- f is *binary* classifier if $S = \{0, 1\}$.
- *learn* a classifier: find the f_θ that fits best a *training set* $((x_i, y_i), i = 1, \ldots, n)$ among a family $(f_\theta, \theta \in \Theta)$
- Standard binary classifiers / learning algorithms when $X \in \mathbb{R}^d$

Input variables X:

- MIP $\mathcal{P}$
- Decomposition(s) $\mathcal{D}$

Variables Y of interest:

- Should we use `SCIP` or `GCG`?
- Which decomposition should we use?

Input X for Standard Classifiers

We want to use standard classifiers for f (SVM, KNN, RF). Need an input in $\mathbb{R}^d$.

- Input size not fixed
- Ordering of columns / rows not fixed

Input X for Standard Classifiers

We want to use standard classifiers for f (SVM, KNN, RF). Need an input in $\mathbb{R}^d$.

- Input size not fixed
- Ordering of columns / rows not fixed

$$\boxed{\begin{array}{c}\text{Input}\\(\mathcal{P},\mathcal{D})\end{array}} \xrightarrow[\text{map } \phi]{\text{Feature}} \boxed{\begin{array}{c}\text{Features vector}\\\phi(\mathcal{P},\mathcal{D})\in\mathbb{R}^d\end{array}} \xrightarrow[f_\theta]{\text{Classifier}} \boxed{\begin{array}{c}\text{Output}\\f_\theta\left(\phi\left(\mathcal{P},\mathcal{D}\right)\right)\end{array}}$$

Define ϕ
$>$80 features

Choose *family* f
Learn θ

Feature Map ϕ

Examples of features used:

- Nb variables/constraints
- Variable types
- Constraint types
- Products of features
- Nb linking variables / constraints
- Nb blocks
- $\min$, $\max$, mean block size
- Detector used (indicator)
- Detection quality metrics

Labeling - Definition of Y

Training set

- MIP $\mathcal{P}$
- Decompositions $\mathcal{D}$
- `SCIP` run on each $\mathcal{P}$
- `GCG` run on each $(\mathcal{P}, \mathcal{D})$

Given an input $(\mathcal{P}, \mathcal{D})$ we learn a classifier

$$Y = f(\phi(\mathcal{P}, \mathcal{D}))$$

where $Y = 1$ if `GCG` on $(\mathcal{P}, \mathcal{D})$ is *better* than `SCIP` on $\mathcal{P}$ after a given timelimit t.

- `GCG` solves $\mathcal{P}$ and `SCIP` doesn't
- Both solve $\mathcal{P}$ and `GCG` is faster
- Neither solve $\mathcal{P}$ but `GCG`'s gap is smaller (approx.)

Question: Should we use `SCIP` or `GCG`?

Classifiers based on Decomposition Quality

$f(\phi(\mathcal{P}, \mathcal{D})) \in [0, 1]$: probability that GCG with $\mathcal{D}$ beats SCIP.

Given the decompositions $\mathcal{D}_1, \ldots, \mathcal{D}_k$ available and the remaining time t, *use GCG if*

$$\max_i f(\phi(\mathcal{P}, \mathcal{D}_i)) \geq \alpha$$

We take $0.5 < \alpha \leq 1$: decomposition is not a default choice.

If we use GCG, *select the decomposition*

$$\arg\max_i f(\phi(\mathcal{P}, \mathcal{D}_i))$$

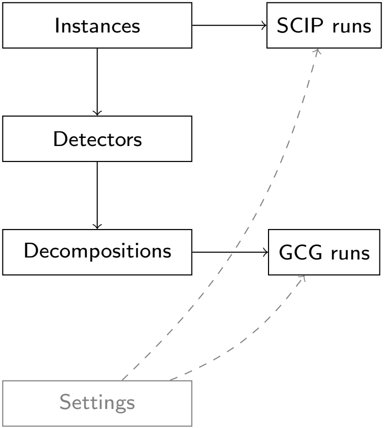
Question: Which decomposition should we use?

Table of Content

Technical Setup

- database
- python interface
- `SCIP` version 3.2.1, `GCG` 2.1.1.
  i7-2600 3.4GHz PC,
  8MB cache, 16GB RAM
- ~ 135 days computing time

Database Schema



Distribution of Instances

SCIP results	all	clr	stcv	cpmp	sdlb	ctst	gap	ntlb	ltsz	bp	rap	stbl	cvrp	miplib
		structured												non-str
instances	400	25	25	25	25	25	25	25	25	25	25	25	25	100
opt. sol.	65.5%	19	3	18	10	25	23	25	25	6	12	22	6	68
feas. sol.	31.5%	6	21	7	11	-	2	-	-	19	12	3	19	26
no sol.	3.0%	-	1	-	4	-	-	-	-	-	1	-	-	6

Structured Instances

coloring (clr)
set covering (stcv)
capacitated p-median (cpmp)
survivable fixed telecom network design (sdlb)
cutting stock (ctst)
generalized assignment (gap)
network design (ntlb)
resource allocation (rap)
capacitated vehicle routing (cvrp)
lot sizing (ltsz)
bin packing (bp)
stable set (stbl)

Splitting Training and Testset

Reminder: datapoints $(\mathcal{P}, \mathcal{D})$

Split training and test set by mip instances, to avoid a biased estimator.

Distribution of decompositions per MIP instance:

- Average: ~ 15.3
- Standard Deviation: ~ 9.0

	Instances	Decompositions
Training	269 ($\sim$ 2/3)	4434
Test	131 ($\sim$ 1/3)	2069

Overall Performance

- Test set of 131 MIP instances, 99 structured and 32 unstructured.
- `GCG` better than `SCIP` on 34 instances.

Nearest neighbor classifier of `scikit-learn` library.

Overall Performance

- Test set of 131 MIP instances, 99 structured and 32 unstructured.
- GCG better than SCIP on 34 instances.

Nearest neighbor classifier of `scikit-learn` library.

Instances	All				Structured				Non-structured			
Solver	SCIP	GCG	SL	OPT	SCIP	GCG	SL	OPT	SCIP	GCG	SL	OPT
No opt. sol.	52	66	44	39	39	37	31	26	13	29	14	13
CPU time (h)	111.3	142.6	93.1	85.7	83.5	82.2	65.9	58.5	27.8	56.8	29.2	27.2
Geo. mean (s)	127.1	370.4	78.6	67.8	73.4	146.9	39.2	32.2	672.9	5145.0	766.0	646.5

- SCIP: apply SCIP to all instances
- GCG: apply GCG with build-in selection tool
- SL: our supervised learning scheme
- OPT: best decomposition selected each time

Solver Selection Accuracy

Avoid using GCG when there is no appropriate structure.

Is GCG on $(\mathcal{P}, \mathcal{D})$ better than SCIP on $\mathcal{P}$?

		All instances		Structured		Non-structured	
Classifier	Pred.	SCIP 74.0%	GCG 26.0%	SCIP 68.7%	GCG 31.3%	SCIP 90.6%	GCG 9.4%
RBF Unbal.	SCIP	TN	FN				
	GCG	FP	TP				
KNN distance.	SCIP						
	GCG						
RF Unbal.	SCIP						
	GCG						
RF Bal.	SCIP						
	GCG						

Solver Selection Accuracy

Avoid using GCG when there is no appropriate structure.

Is GCG on $(\mathcal{P}, \mathcal{D})$ better than SCIP on $\mathcal{P}$?

		All instances		Structured		Non-structured	
		SCIP	GCG	SCIP	GCG	SCIP	GCG
Classifier	Pred.	74.0%	26.0%	68.7%	31.3%	90.6%	9.4%
RBF Unbal.	SCIP	70.3%	22.1%	63.7%	23.2%	90.6%	9.4%
	GCG	3.8%	3.8%	5.1%	8.1%	0.0%	0.0%
KNN distance.	SCIP	69.5%	12.3%	64.6%	11.1%	84.4%	6.3%
	GCG	4.5%	13.7%	4.1%	20.2%	6.3%	3.1%
RF Unbal.	SCIP	63.4%	11.5%	55.6%	13.1%	87.5%	6.3%
	GCG	10.7%	14.5%	13.1%	18.2%	3.1%	3.1%
RF Bal.	SCIP	60.3%	10.7%	50.5%	11.1%	90.6%	9.4%
	GCG	13.7%	15.3%	18.2%	20.2%	0.0%	0.0%

Best Decomposition Selection Accuracy

Classifier	All instances	`GCG` selected by class.	`GCG` the best
RBF	42.7%	80.0%	76.7%
KNN	58.8%	88.9%	77.4%
RF unbalanced	51.1%	72.7%	76.5%
RF balanced	64.9%	71.1%	79.4%

Best Decomposition Selection Accuracy

Classifier	All instances	GCG selected by class.	GCG the best
RBF	42.7%	80.0%	76.7%
KNN	58.8%	88.9%	77.4%
RF unbalanced	51.1%	72.7%	76.5%
RF balanced	64.9%	71.1%	79.4%

How to improve this performance?

Table of Content

Outlook: a Bunch of Decompositions

New designed detection loop (previous talk)

- Beat SCIP on more instances
- Harder Machine Learning task

Outlook: a Bunch of Decompositions

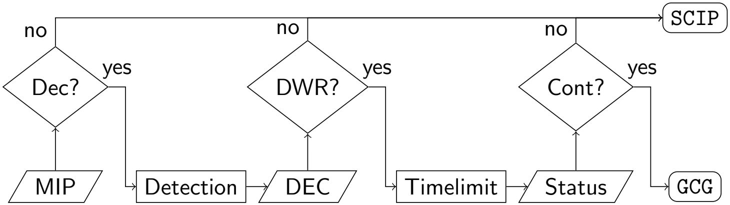
New designed detection loop (previous talk)

- Beat SCIP on more instances
- Harder Machine Learning task

Enrich the dataset

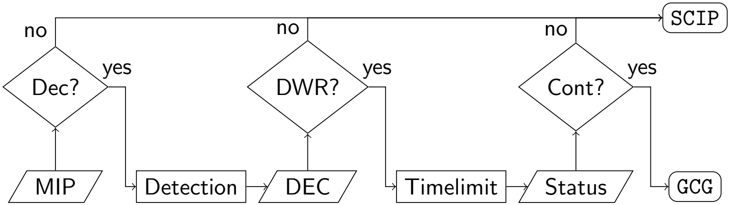
- More decompositions
- More instances
- More meaningful features
- Consider performance variability
- . . . a lot more computation time

Additional Decisions in GCG



1. Prediction before detection
2. DWR? and selection
3. Enrich feature quality by exploiting run time features

Additional Decisions in GCG



1. Prediction before detection
2. DWR? and selection
3. Enrich feature quality by exploiting run time features

Thank you for your attention!

Backup

- geom. mean: $\bar{x}_{\text{geom}} = \sqrt[n]{\prod_{i=1}^{n} x_i}$
- un/balanced: weights associated with classes.