

Task scheduling for dual-arm industrial robots through constraint programming

MiniZinc modeling and solver comparison

Tommy Kvant

Institute of Computer Science
Lund University

February 24, 2015



Outline

- 1 Introduction
 - Project goal
 - MiniZinc
 - YuMi®
- 2 Case Study
- 3 Model
 - Tasks
- 4 Evaluation
 - Defining tasks
 - Grouping
 - Filter
- 5 Conclusions
 - Solvers
 - Results
- 5 Conclusions
 - Conclusions

Introduction - Project goal

- Constraint Programming model for dual-armed robots such as YuMi®
 - Change tools
 - Carry only one component at the time
 - Same duration for tool changes, regardless of direction
 - Use trays, fixtures and outputs
- Implement the model in MiniZinc
- Test the model with 6 solvers and compare the results

Introduction - MiniZinc

- Declarative language
- Medium level
- Translates to FlatZinc
- Aims to be standard
- Many solvers can read FlatZinc

Introduction - YuMi®

- Dual-armed robot
- Flexible - multiple tools
- Fine motor skills

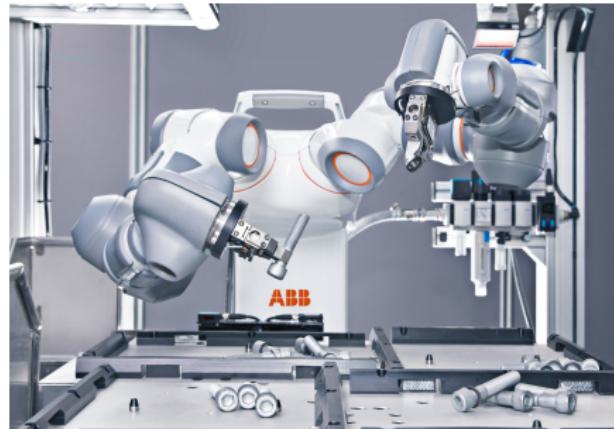


Photo: ABB

Case Study

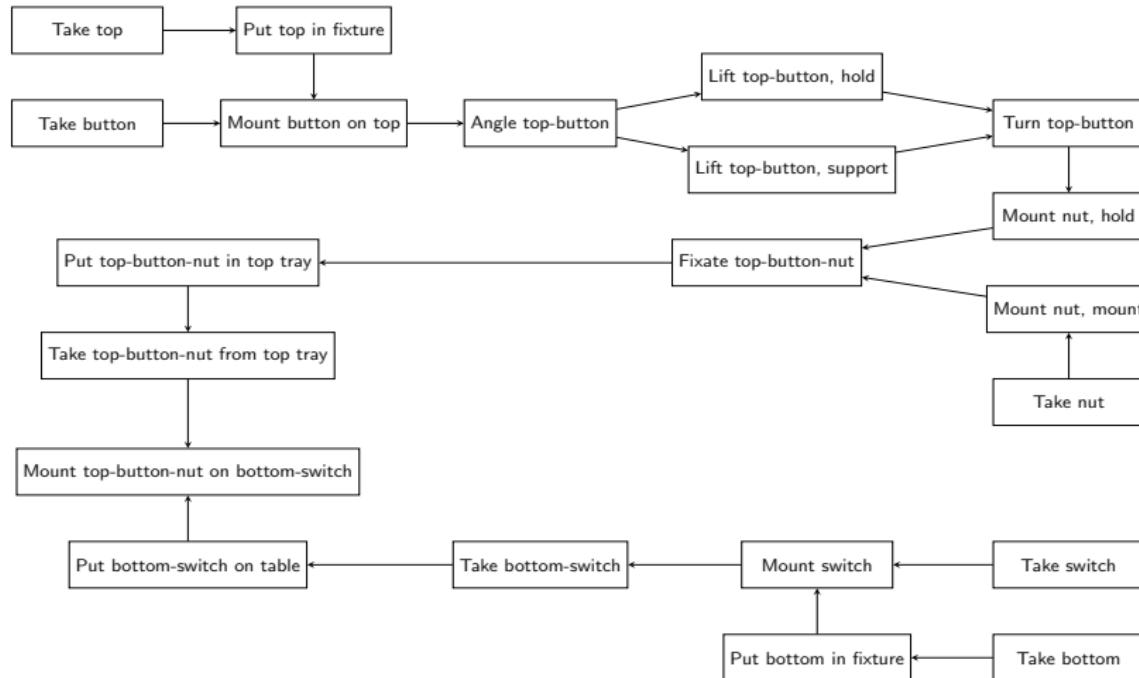


Case Study

Physical Entities

- Machines
- Tools
- Components
- Tray
- Fixture
- Output

Assembly

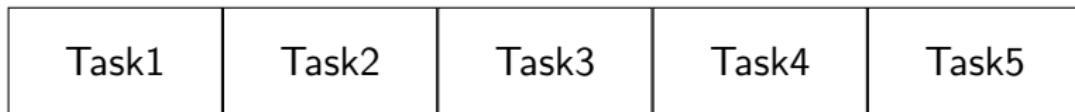


Model

Job Shop Problem

- n jobs, varying size
- m identical machines
- NP-complete for $m \geq 2$ and $n \geq 3$

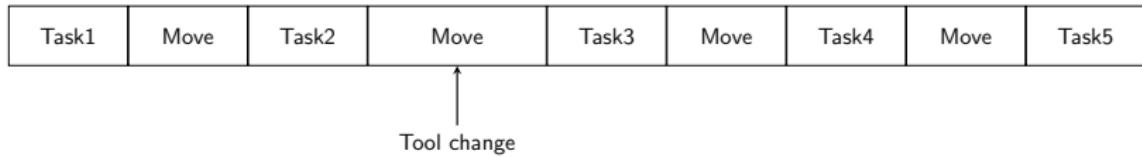
Model



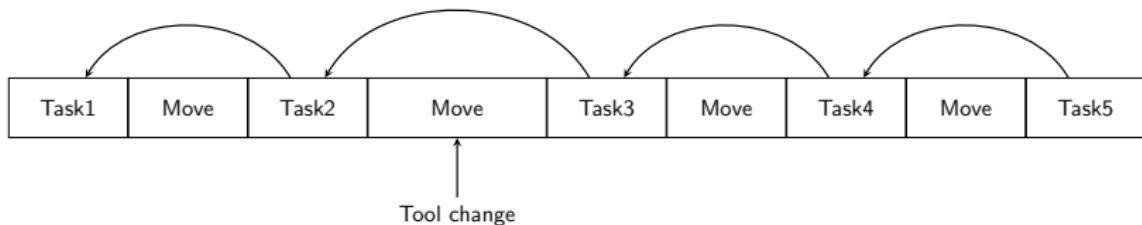
Model

Task1	Move	Task2	Move	Task3	Move	Task4	Move	Task5
-------	------	-------	------	-------	------	-------	------	-------

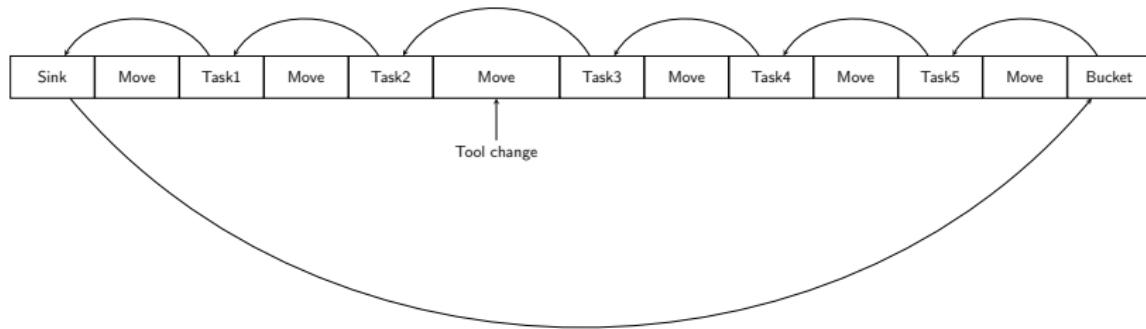
Model



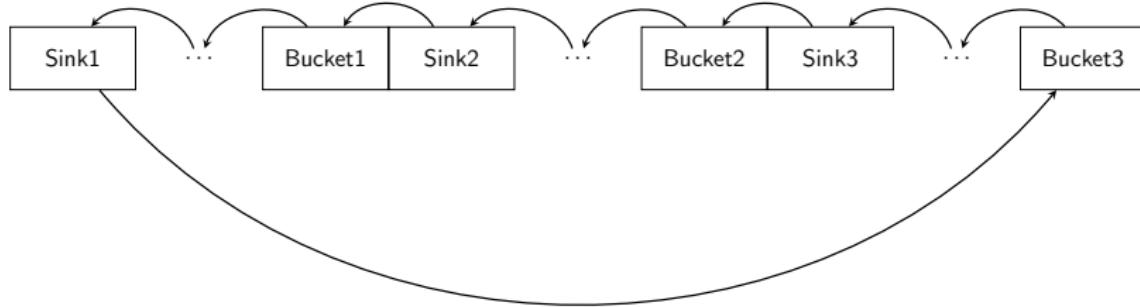
Model



Model



Model



Defining Tasks

- Components
- Tray, Fixture, Output
- Tool
- Action - Taking, Putting, Mounting, Moving

Model

Group tasks

- Ordered group
- Concurrent group

Filter

- Temporal filter
- Predecessor filter

Evaluation

- Test with 6 solvers
 - Assembly times
 - Solver time
 - FlatZinc file
- MiniZinc 1.6 & 2.0.1
- Combination of filters

Criteria

- FlatZinc parser
- Free

Solvers Tested

- G12/FD
- JaCoP
- Gecode
- or-tools
- Opturion CPX
- Choco3

Assembly Times

Manual Time
516 t.u.

Assembly Times

Manual Time
516 t.u.

Solver Time
512 t.u.

Solver Time

	Pred & Temp		Pred		Temp		None	
	1.6	2.0.1	1.6	2.0.1	1.6	2.0.1	1.6	2.0.1
G12/FD	-	-	-	-	-	-	-	-
JaCoP	658	-	1011156	-	-	-	-	-
Gecode	-	60	-	71761	-	99	-	71186
or-tools	271	!	380	!	302	!	457	!
Opturion CPX	-	!	-	!	-	!	-	!
Choco3	-	-	-	-	-	-	-	-

Solver Time

	Pred	-	-	-	-	-	-	-	-
	1.6	-	658	1011156	71761	99	-	71186	-
G12/FD	-	-	-	-	-	-	-	-	-
JaCoP	658	-	1011156	-	-	-	-	-	-
Gecode	-	60	-	71761	-	99	-	71186	-
or-tools	271	!	380	!	302	!	457	!	-
Opturion CPX	-	!	-	!	-	!	-	!	-
Choco3	-	-	-	-	-	-	-	-	-

Solver Time

	Pred	-	-	-	-	-	-	-	-
G12/FD	-								
JaCoP	658	-	1011156	-	-	-	-	-	-
Gecode	-	60	-	71761	-	99	-	71186	
or-tools	271	!	380	!	302	!	457	!	
Opturion CPX	-	!	-	!	-	!	-	-	!
Choco3	-	-	-	-	-	-	-	-	-

Solver Time

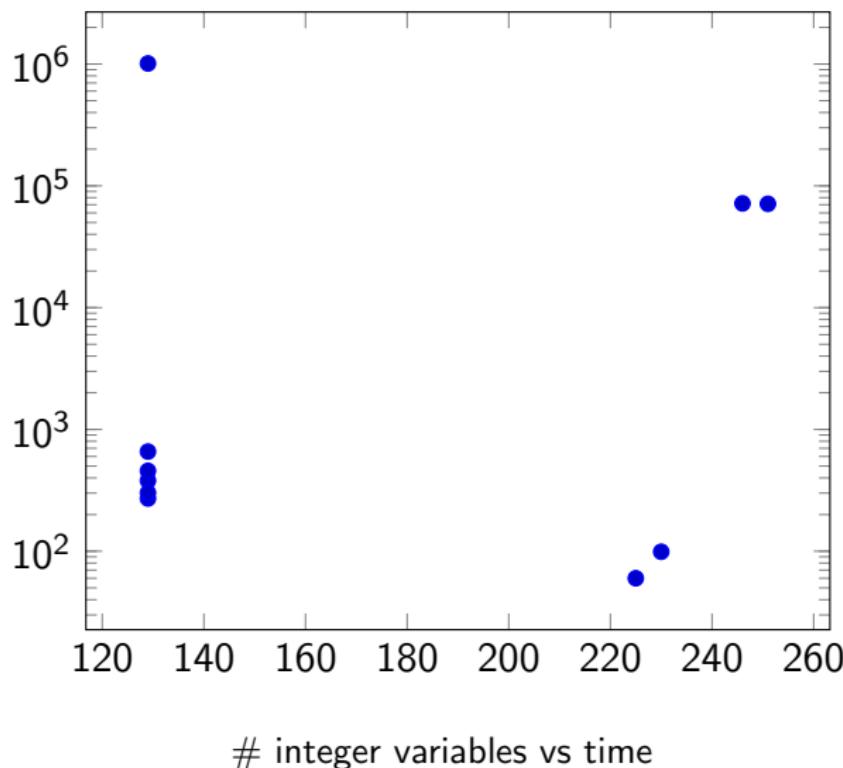
	Pred & Temp		Pr					
	1.6	2.0.1	1.6					
G12/FD	-	-	-					
JaCoP	658	-	1011156					
Gecode	-	60	-	71761	-	99	-	71186
or-tools	271	!	380	!	302	!	457	!
Opturion CPX	-	!	-	!	-	!	-	!
Choco3	-	-	-	-	-	-	-	-

Solver Time

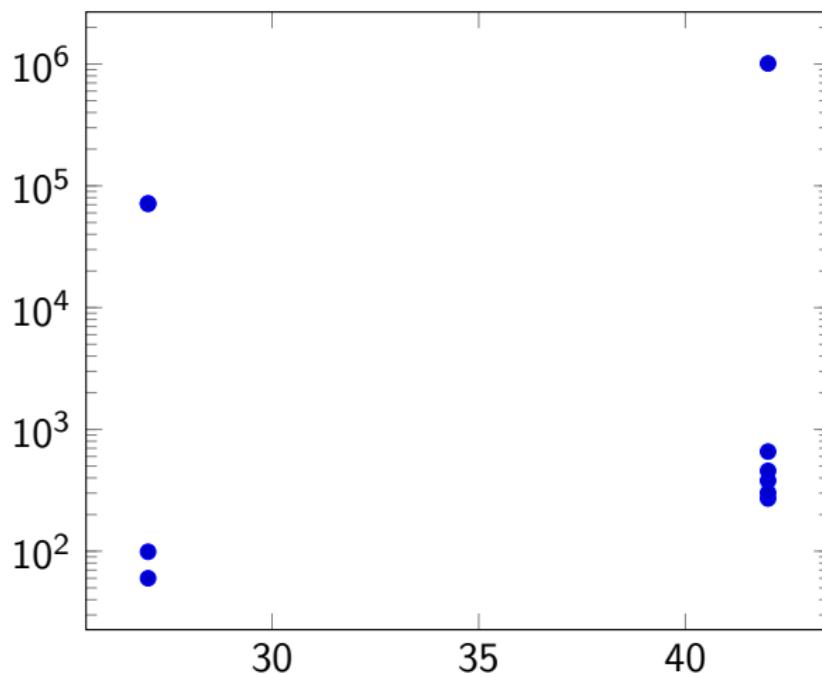
	Pred & Temp		Pr						
	1.6	2.0.1		1.6	71761	-	99	-	71186
G12/FD	-	-	-						
JaCoP	658	-	1011156						
Gecode	-	60	-	!	302	!	457	!	
or-tools	271	!	380	!	-	!	-	-	
Opturion CPX	-	!	-	-	-	-	-	-	
Choco3	-	-	-	-	-	-	-	-	

Solver Time

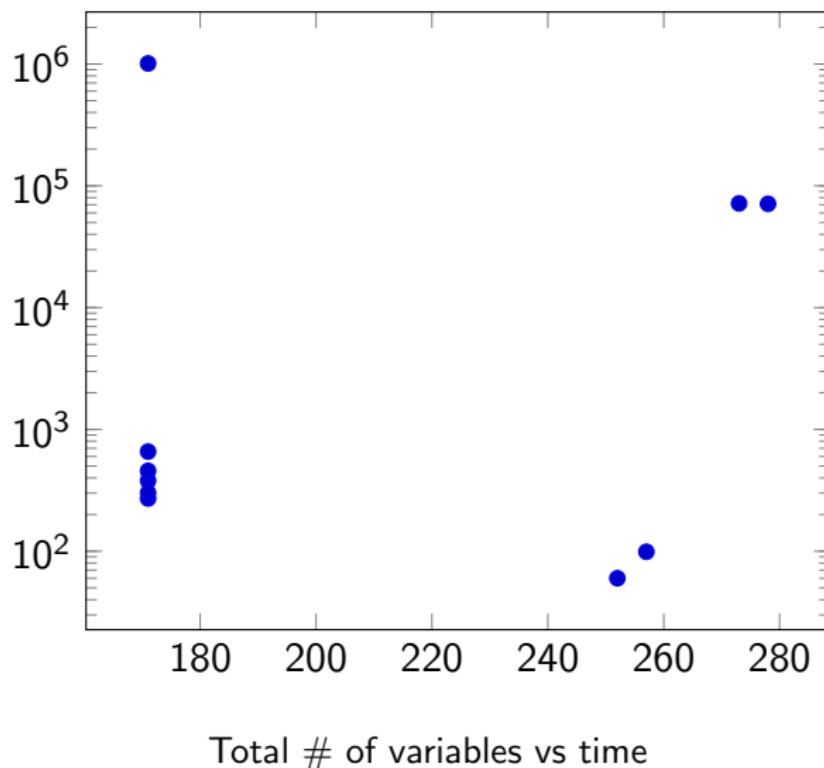
	Pred & Temp	Pred	Temp	None
G12/FD				2.0.1
JaCoP				-
Gecode				-
or-tools				!
Opturion CPX				!
Choco3	-	-	-	-

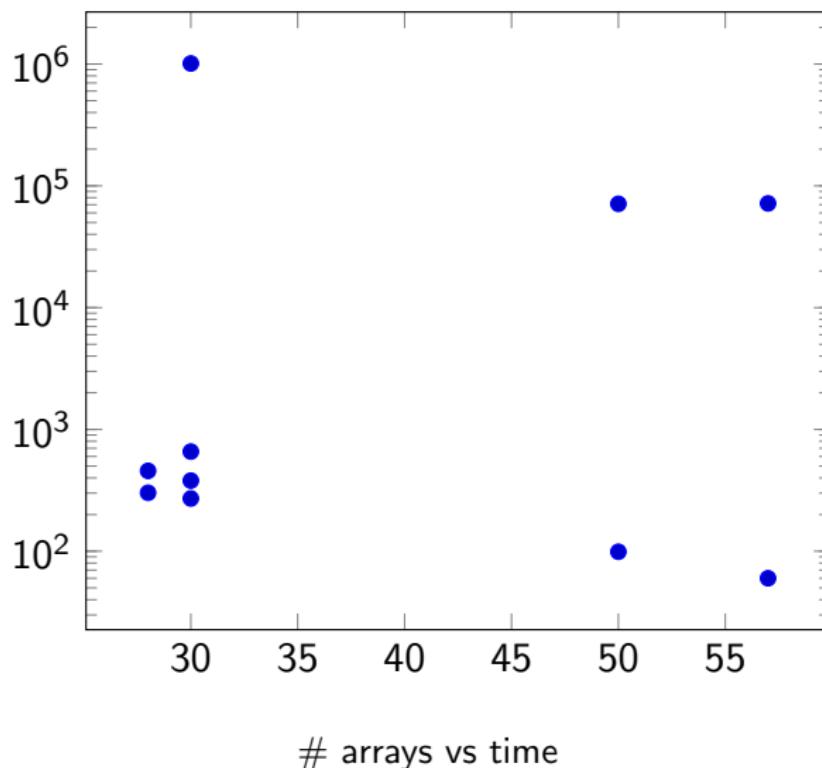


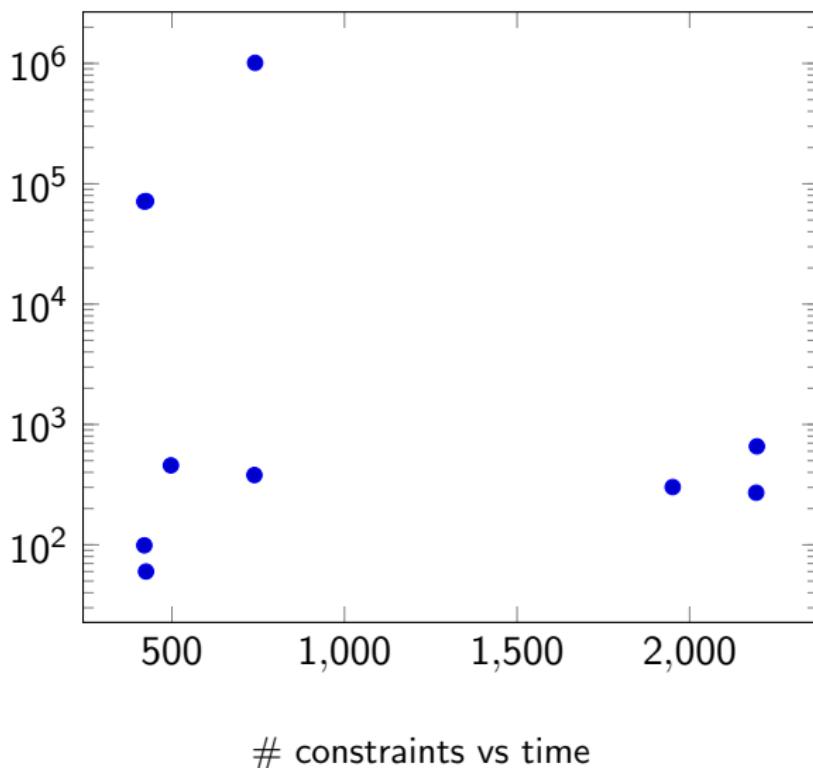
Results

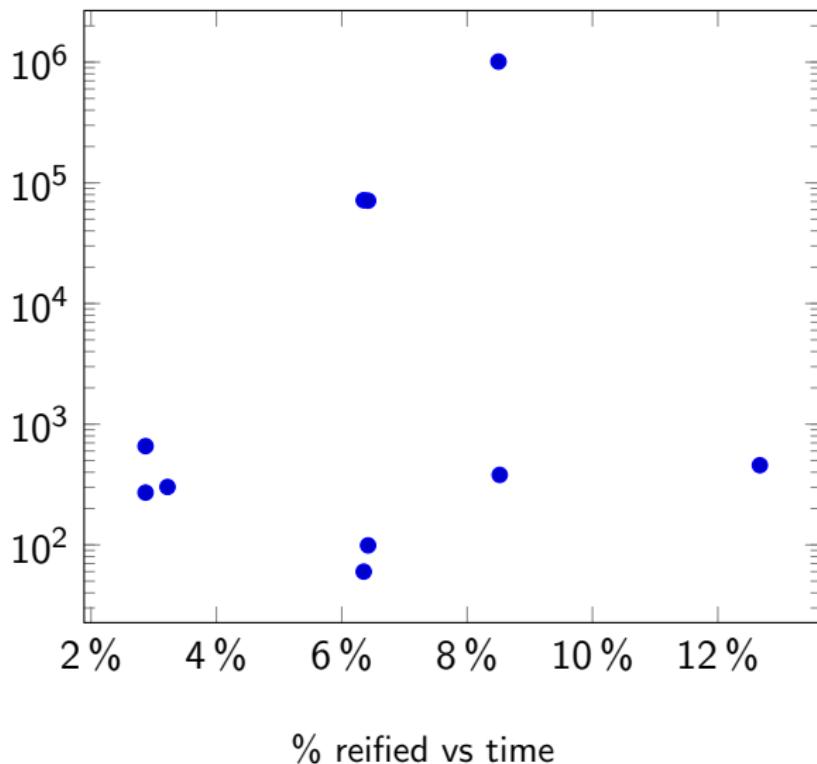


boolean variables vs time

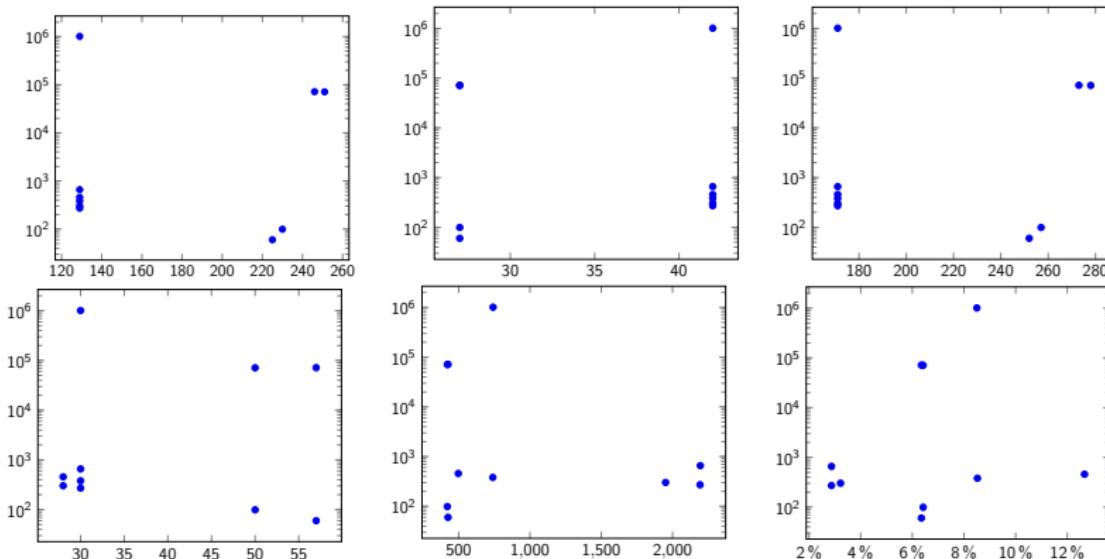








Results



Introduction

ooo

Conclusions

Case Study

Model

oooooooooooo

Evaluation

oooooooooooo

Conclusions

●○

Conclusions

Conclusions

- Model produces solution just as good as handmade solution

Conclusions

- Model produces solution just as good as handmade solution
- Solver performance varies a lot

Conclusions

- Model produces solution just as good as handmade solution
- Solver performance varies a lot
- Best performance: Gecode, all filters, MiniZinc 2.0.1

Conclusions

- Model produces solution just as good as handmade solution
- Solver performance varies a lot
- Best performance: Gecode, all filters, MiniZinc 2.0.1
- Filters presented have a positive impact on runtime

Conclusions

- Model produces solution just as good as handmade solution
- Solver performance varies a lot
- Best performance: Gecode, all filters, MiniZinc 2.0.1
- Filters presented have a positive impact on runtime
- No relation between FlatZinc output and solver runtime

Further work

- Test the result on a real robot
- Further testing of the filters
- More realistic representation of tools available
- Test solvers with more assemblies