Experimenting with Robotic Intra-Logistics Domains

Martin Gebser [1] Philipp Obermeier [1] Thomas Otto [1] Torsten Schaub [1] Orkunt Sabuncu [2] Van Nyugen [3] Tran Cao Son [3]

[1]University of Potsdam, Germany

[2]TED University, Ankara, Turkey

[3]New Mexico State University, Las Cruces, USA

Potassco

Outline I

Outline I

Motivation

Answer Set Programming

Motivation

Answer Set Programming

- Declarative problem solving for combinatorial problems

Motivation

Answer Set Programming

- Declarative problem solving for combinatorial problems
- Large spectrum of applications in academia and industry

Potassco

Motivation

Answer Set Programming

- Declarative problem solving for combinatorial problems
- Large spectrum of applications in academia and industry

Potassco

Motivation

Answer Set Programming

- Declarative problem solving for combinatorial problems
- Large spectrum of applications in academia and industry

Dynamic Real-World Apps

Potassco

Motivation

Answer Set Programming

- Declarative problem solving for combinatorial problems
- Large spectrum of applications in academia and industry

Dynamic Real-World Apps

- Logistics, manufacturing, automation, scheduling, etc.

Motivation

Answer Set Programming

- Declarative problem solving for combinatorial problems
- Large spectrum of applications in academia and industry

Dynamic Real-World Apps

- Logistics, manufacturing, automation, scheduling, etc.
- Large instance sizes

Potassco

Motivation

Answer Set Programming

- Declarative problem solving for combinatorial problems
- Large spectrum of applications in academia and industry

Dynamic Real-World Apps

- Logistics, manufacturing, automation, scheduling, etc.
- Large instance sizes
- Complex processes

Motivation

Answer Set Programming

- Declarative problem solving for combinatorial problems
- Large spectrum of applications in academia and industry

Dynamic Real-World Apps

- Logistics, manufacturing, automation, scheduling, etc.
- Large instance sizes
- Complex processes

Motivation

Answer Set Programming

- Declarative problem solving for combinatorial problems
- Large spectrum of applications in academia and industry

Dynamic Real-World Apps

- Logistics, manufacturing, automation, scheduling, etc.
- Large instance sizes
- Complex processes

Potassco

Motivation

Answer Set Programming

- Declarative problem solving for combinatorial problems
- Large spectrum of applications in academia and industry

Dynamic Real-World Apps

- Logistics, manufacturing, automation, scheduling, etc.
- Large instance sizes
- Complex processes

↯ Lack of Real-Life Test Data ↯

- Existing benchmark suites kept intentionally simplistic
- No industrial scale test data in the public domain

Potassco

Motivation

Answer Set Programming

- Declarative problem solving for combinatorial problems
- Large spectrum of applications in academia and industry

Dynamic Real-World Apps

- Logistics, manufacturing, automation, scheduling, etc.
- Large instance sizes
- Complex processes

Robotic Intra-Logistics as Benchmark Domain

- Hard, dynamic planning problem related to MAS, scheduling, temporal logics, CSP, uncertainty, etc.
- Key concern of industry 4.0

Robotic Intra-Logistics

- Robotics systems for logistics and warehouse automation based on hundreds of
  - mobile robots
  - movable shelves

Potassco

Robotic Intra-Logistics

- Robotics systems for logistics and warehouse automation based on hundreds of
  - mobile robots
  - movable shelves
- Main tasks: order fulfillment, i.e.
  - routing
  - order picking
  - replenishment

Potassco

Robotic Intra-Logistics

- Robotics systems for logistics and warehouse automation based on hundreds of
  - mobile robots
  - movable shelves
- Main tasks: order fulfillment, i.e.
  - routing
  - order picking
  - replenishment
- Many competing industry solutions:
  - Amazon, Dematic, Genzebach, Gray Orange, Swisslog

Robotic Intra-Logistics

Outline I

asprilo Benchmark Suite

Main Components

- Standardized benchmark domains

asprilo Benchmark Suite

Main Components

- Standardized benchmark domains
  - Concise problem specification
  - Domains ranging from MAPF to full order fulfillment

asprilo Benchmark Suite

Main Components

- Standardized benchmark domains
- Versatile instance generator

asprilo Benchmark Suite

Main Components

- Standardized benchmark domains
- Versatile instance generator
  - Rich set of customization options
  - Leverages multi-shot ASP for generation

asprilo Benchmark Suite

Main Components

- Standardized benchmark domains
- Versatile instance generator
- Visualizer for instances and plans

asprilo Benchmark Suite

Main Components

- Standardized benchmark domains
- Versatile instance generator
- Visualizer for instances and plans
  - Animated playback of plans
  - Graphical editor for instances

asprilo Benchmark Suite

Main Components

- Standardized benchmark domains
- Versatile instance generator
- Visualizer for instances and plans
- Solution checker with error feedback

asprilo Benchmark Suite

Main Components

- Standardized benchmark domains
- Versatile instance generator
- Visualizer for instances and plans
- Solution checker with error feedback
  - Specific error descriptions
  - Modular design, easily extensible

asprilo Benchmark Suite

Main Components

- Standardized benchmark domains
- Versatile instance generator
- Visualizer for instances and plans
- Solution checker with error feedback

asprilo Benchmark Suite

Main Components

- Standardized benchmark domains
- Versatile instance generator
- Visualizer for instances and plans
- Solution checker with error feedback

Resources

- Website at `http://potassco.org/asprilo`

asprilo Benchmark Suite

Main Components

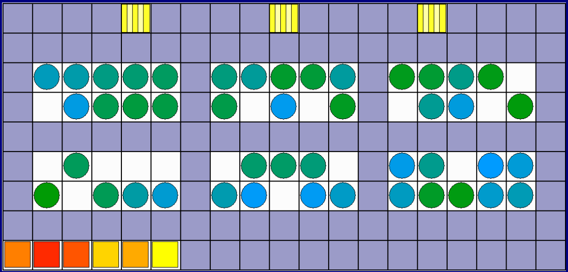
- Standardized benchmark domains
- Versatile instance generator
- Visualizer for instances and plans
- Solution checker with error feedback

Resources

- Website at `http://potassco.org/asprilo`
- ICLP'18 paper, also available at `https://arxiv.org/abs/1804.10247`

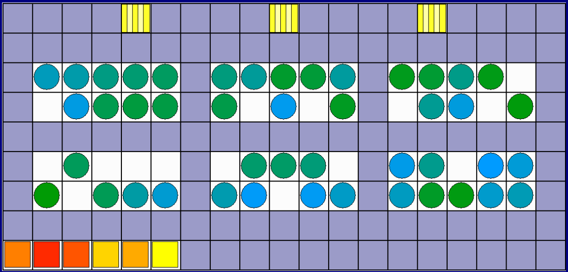
General Domain A

- The warehouse is laid out as a (partial) 2-dimensional grid
- Shelves store products in a certain quantity, each shelf occupies a single grid node
- Mobile robots move and navigate through the warehouse along the grid, can carry shelves and deliver product units to picking stations



General Domain A

- Highway nodes are special grid nodes where robots must never put down a shelf
- A set of orders is initially provided, an order is fulfilled if all its requested product units are delivered to its assigned picking station
- Main Goal: plan robot actions such that all orders will be fulfilled



Domain A Demo

Domains A, B, C, M

Domain A most general domain

Complexity

Domains A, B, C, M

Domain A most general domain

Domain B ignores product quantities

Complexity

Domains A, B, C, M

Domain A most general domain

Domain B ignores product quantities

Domain C ignores product quantities
delivery actions at once

Complexity

Domains A, B, C, M

Complexity

Domain A most general domain

Domain B ignores product quantities

Domain C ignores product quantities
delivery actions at once

Domain M only move actions
singleton orders and shelves
reach shelves with ordered products

Domain M Demo

Outline I

Exemplary Evaluation

- Exemplary benchmark evaluation to showcase *asprilo*

Exemplary Evaluation

- Exemplary benchmark evaluation to showcase *asprilo*
- Key questions of the analysis

Exemplary Evaluation

- Exemplary benchmark evaluation to showcase *asprilo*
- Key questions of the analysis
  1. What is the impact of different representations of grid positions?

Exemplary Evaluation

- Exemplary benchmark evaluation to showcase *asprilo*
- Key questions of the analysis
  1. What is the impact of different representations of grid positions?
  2. What is the impact of increasingly complex domains?

Exemplary Evaluation

- Exemplary benchmark evaluation to showcase *asprilo*
- Key questions of the analysis
  1. What is the impact of different representations of grid positions?
  2. What is the impact of increasingly complex domains?
  3. What is the impact of decoupling sources of combinatorics?

Potassco

Exemplary Evaluation

- Exemplary benchmark evaluation to showcase *asprilo*
- Key questions of the analysis
  1. What is the impact of different representations of grid positions?
  2. What is the impact of increasingly complex domains?
  3. What is the impact of decoupling sources of combinatorics?
- Test instances created with the generator

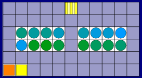
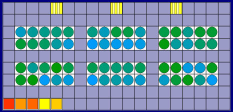
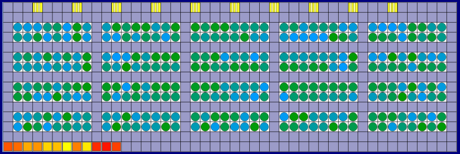
Exemplary Evaluation

- Exemplary benchmark evaluation to showcase *asprilo*
- Key questions of the analysis
  1. What is the impact of different representations of grid positions?
  2. What is the impact of increasingly complex domains?
  3. What is the impact of decoupling sources of combinatorics?
- Test instances created with the generator
- Referential encodings for *asprilo*'s domains

Exemplary Evaluation

- Exemplary benchmark evaluation to showcase *asprilo*
- Key questions of the analysis
  1. What is the impact of different representations of grid positions?
  2. What is the impact of increasingly complex domains?
  3. What is the impact of decoupling sources of combinatorics?
- Test instances created with the generator
- Referential encodings for *asprilo*'s domains
- Detailed setup description, instances, encodings and results available at `http://potassco.org/asprilo/experiments`

Instances

Name	Generator Call and Resulting Layout
small:	`gen -x 11 -y 6 -X 4 -Y 2 -p 1 -s 16 -P 16 -u 16 -H --prs 1 -r 2 -o 2` 
medium:	`gen -x 19 -y 9 -X 5 -Y 2 -p 3 -s 60 -P 60 -u 60 -H --prs 1 -r 5 -o 5` 
large:	`gen -x 46 -y 15 -X 8 -Y 2 -p 10 -s 320 -P 320 -u 320 -H --prs 1 -r 12 -o 12` 

Potassco

clingo Encoding for Domain M

routing

```
time(1..horizon).

direction((X,Y)) :- X=-1..1, Y=-1..1, |X+Y|=1.
nextto((X,Y),(X',Y'),(X+X',Y+Y')) :- position((X,Y)), direction((X',Y')), position((X+X',Y+Y')).

{ move(R,D,T) : direction(D) } 1 :- isRobot(R), time(T).

position(R,C,T) :- move(R,D,T), position(R,C',T-1),     nextto(C',D,C).     % movement effect and precond.
                :- move(R,D,T), position(R,C ,T-1), not nextto(C ,D,_).

position(R,C,T) :- position(R,C,T-1), not move(R,_,T), isRobot(R), time(T). %inertia

moveto(C',C,T) :- nextto(C',D,C), position(R,C',T-1), move(R,D,T).         % edge collision
:- moveto(C',C,T), moveto(C,C',T), C < C'.

:- { position(R,C,T) : isRobot(R) }  > 1, position(C), time(T).             % vertex collision
```

clingo Encoding for Domain M

routing to shelves

```
time(1..horizon).

direction((X,Y)) :- X=-1..1, Y=-1..1, |X+Y|=1.
nextto((X,Y),(X',Y'),(X+X',Y+Y')) :- position((X,Y)), direction((X',Y')), position((X+X',Y+Y')).

{ move(R,D,T) : direction(D) } 1 :- isRobot(R), time(T).

position(R,C,T) :- move(R,D,T), position(R,C',T-1),     nextto(C',D,C).     % movement effect and precond.
                :- move(R,D,T), position(R,C ,T-1), not nextto(C ,D,_).

position(R,C,T) :- position(R,C,T-1), not move(R,_,T), isRobot(R), time(T). %inertia

moveto(C',C,T) :- nextto(C',D,C), position(R,C',T-1), move(R,D,T).         % edge collision
:- moveto(C',C,T), moveto(C,C',T), C < C'.

:- { position(R,C,T) : isRobot(R) }  > 1, position(C), time(T).            % vertex collision


processed(O,A) :- ordered(O,A), shelved(S,A), position(S,C,0), position(R,C,horizon), isRobot(R).

processed(O) :- isOrder(O), processed(O,A) : ordered(O,A).

:- not processed(O), isOrder(O).
```

clingo Encoding for Domain A

routing + transport + delivery

```
time(1..horizon).

direction((X,Y)) :- X=-1..1, Y=-1..1, |X+Y|=1.
nextto((X,Y),(X',Y'),(X+X',Y+Y')) :- position((X,Y)), direction((X',Y')), position((X+X',Y+Y')).

{    move(R,D,T) : direction(D) ;
   pickup(R,S,T) : isShelf(S)   ;
  putdown(R,S,T) : isShelf(S)   } 1 :- isRobot(R), time(T).

waits(R,T) :- not pickup(R,_,T), not putdown(R,_,T), not move(R,_,T), isRobot(R), time(T).

position(R,C,T) :- move(R,D,T),    position(R,C',T-1),     nextto(C',D,C).
                :- move(R,D,T),    position(R,C ,T-1), not nextto(C, D,_).

 carries(R,S,T) :- pickup(R,S,T),  position(R,C,T-1), position(S,C ,T-1).
                :- pickup(R,S,T),   carries(R,_,T-1).
                :- pickup(R,S,T),   carries(_,S,T-1).
                :- pickup(R,S,T),  position(R,C,T-1), position(S,C',T-1), C != C'.

                :- putdown(R,S,T), not carries(R,S,T-1).

serves(R,S,P,T) :- position(R,C,T), carries(R,S,T),  position(P,C), isStation(P).

position(R,C,T) :- position(R,C,T-1), not    move(R,_,T), isRobot(R), time(T).
 carries(R,S,T) :-  carries(R,S,T-1), not putdown(R,_,T),             time(T).

position(S,C,T) :- position(R,C,T  ),     carries(R,S,T).
position(S,C,T) :- position(S,C,T-1), not carries(_,S,T), isShelf(S), time(T).

moveto(C',C,T) :- nextto(C',D,C), position(R,C',T-1), move(R,D,T).
:- moveto(C',C,T), moveto(C,C',T), C < C'.

 :- { position(R,C,T) : isRobot(R) }  > 1, position(C), time(T).
 :- { position(S,C,T) : isShelf(S) }  > 1, position(C), time(T).
```

Encoding Variants

Variants

clingo	boolean encoding
clingo$_{xy}$	boolean encoding + split positional coordinates
clingcon	linear constraints for positions and product quantities
clingo[DL]	difference constraints for positions and product quantities

Task Assignment

- Robots assigned a subset of shelves and picking stations
- All variants where tested with and without task assignments.

Results

domain	*makespan*	*encoding*	*small*	*medium*	*large*
M	6/10/25	*clingo*	0(0)	0(0)	73(4)
		clingo$_{xy}$	0(0)	16(1)	591(14)
		clingcon	0(0)	37(0)	1168(52)
		clingo[DL]	0(0)	193(1)	1648(96)
M$_a$	6/10/25	*clingo*	0(0)	0(0)	41(2)
		clingo$_{xy}$	0(0)	0(0)	763(27)
		clingcon	0(0)	36(0)	1163(49)
		clingo[DL]	0(0)	86(1)	1679(102)
C^M	20/-/-	*clingo*	805(40)	1800(120)	1800(120)
		clingcon	695(30)	1800(120)	1800(120)
C^M_a	21/35/-	*clingo*	23(1)	370(5)	1800(120)
		clingcon	38(2)	459(15)	1800(120)
B^M	26/-/-	*clingo*	970(53)	1800(120)	1800(120)
		clingcon	807(37)	1800(120)	1800(120)
B^M_a	26/39/-	*clingo*	12(0)	566(19)	1800(120)
		clingcon	29(0)	623(25)	1800(120)
A^M	26/-/-	*clingo*	984(55)	1800(120)	1800(120)
		clingcon	856(41)	1800(120)	1800(120)
A^M_a	26/39/-	*clingo*	12(0)	577(18)	1800(120)
		clingcon	49(1)	625(22)	1800(120)

Experimental results in average run time & number of timeouts

Outline I

Outlook

- Further extending *asprilo* based on user feedback

`http://potassco.org/asprilo`

Outlook

- Further extending *asprilo* based on user feedback
- Explore ASP design patterns and techniques
  - scalability
  - temporal logic
  - preference handling
  - uncertainty
  - online processing

`http://potassco.org/asprilo`

Potassco