Planning Competition for Logistics Robots In Simulation
Scenario and Challenges

Tim Niemueller
Knowledge-based Systems Group, RWTH Aachen University

Erez Karpas
Technion – Israel Institute of Technology

Tiago Vaquero and Eric Timmons
Massachusetts Institute of Technology

The tutorial has received generous support from FESTO
Outline

14:00–14:30  General Introduction and Competition Description
14:30–15:00  Planning and Executive Challenges
15:00–15:30  Hands-on Part I: Overview and bootup

Coffee Break and Demo

16:00–17:00  Hands-on Part II: Exploring the Simulation
17:00–17:30  Discussion and Feedback
**Planning Perspective**

- Focus on plan generation
- Robotics not as testbed
- Execution gets less attention

**Robotics Perspective**

- Focus often on various topics
- Integration for evaluation
- Planning labor-intensive
Planning Perspective

- Focus on plan generation
- Robotics not as testbed
- Execution gets less attention

Robotics Perspective

- Focus often on various topics
- Integration for evaluation
- Planning labor-intensive

Goals

1. Foster closer cooperation among communities
2. Develop grounded expertise with robotic scenarios, platforms, decision architectures, system integration and evaluation
RoboCup Logistics League 2015
Game Basics

- Task: In-factory production logistics
- Goal: variant production
- Two teams playing on common field
- Each team has 3 robots
- Multi-robot coordination task

Two Game Phases

- *Exploration*: detect and report machines
- *Production*: produce and deliver by using processing stations spread across field
RoboCup Logistics League 2015

Playing Field

- Team colors: cyan and magenta
- Exclusive machines spread across field
- Mirrored at middle axis
RoboCup Logistics League – Machines

Common
- Based on Festo MPS
- Marker to identify machine
- Signal light to indicate state
- Each team has exclusive set
- Similar handling for all types

Machine Types (per team)
1 × Base Station (BS): retrieve bases
2 × Ring Station (RS): mount colored rings
2 × Cap Station (CS): buffer/mount caps
1 × Delivery Station (DS): final delivery
RCLL Robot Platform (Team Carologistics)

- Custom Gripper
- Signal Camera
- Additional Laptop
- Conveyor Camera
- Marker Camera
- Wifi
- Internal Computer
- Laser Scanner
- Common Platform
- Festo Robotino 3
Semi-autonomous Referee Box

**Tasks**
- Determines randomized orders and machine failures
- Posts orders dynamically
- Scoring and evaluation
- Instructs MPS stations

**Planning and Benchmarking**
- Accountable environment agency
- Same controller in simulation
- Records extensive data
- Limited uncertainty

⇒ Repeatability benchmarks

Logs game information and all communication
Game Phases

Exploration (4 min)
- Machines show light code specific for machine type
- Robot must recognize and announced this type

Production (15 min)
- Orders are posted dynamically, e.g.
  "Deliver 1x P[red base, yellow and green ring, gray cap] in time window [123, 206] to gate 3"
- Robots must complete production chain leading to products
- Coordination is required for effective resource usage
- Machines may fail, other robots on the field
Game Phases

**Exploration (4 min)**
- Machines show light code specific for machine type
- Robot must recognize and announced this type

**Production (15 min)**
- Orders are posted dynamically, e.g.
  \[\text{"Deliver 1x } P\{\text{red base, yellow and green ring, gray cap}\} \text{ in time window [123, 206] to gate 3"}\]
- Robots must complete production chain leading to products
- Coordination is required for effective resource usage
- Machines may fail, other robots on the field
RoboCup Logistics League – Production

Product Composition

- Products of four complexities (number of rings)
- Base (3 colors) + 0–3 rings (4 colors) + cap (2 colors)
- Order of ring colors is important
- Some ring colors require additional material
- Actual product variants randomized by referee box
- Orders have lead time of a few minutes

Order Elements (posted dynamically by refbox)

- Product to deliver (and number thereof)
- Time window in which to deliver
$C_0$ Production

- Retrieve base with cap from shelf at CS

![Diagram showing production process]
$C_0$ Production

- Retrieve base with cap from shelf at CS
- Prepare CS to retrieve cap
- Feed base into CS

Already simple product has several fragile points and cooperation potential.
\( C_0 \) Production

- Retrieve base with cap from shelf at CS
- Prepare CS to retrieve cap
- Feed base into CS
- Discard cap-less base
$C_0$ Production

- Retrieve base with cap from shelf at CS
- Prepare CS to retrieve cap
- Feed base into CS
- Discard cap-less base
- Prepare BS to provide black base
- Retrieve base from BS
**C₀ Production**

- Retrieve base with cap from shelf at CS
- Prepare CS to retrieve cap
- Feed base into CS
- Discard cap-less base
- Prepare BS to provide black base
- Retrieve base from BS
- Prepare CS to mount cap
- Feed black base to CS
$C_0$ Production

- Retrieve base with cap from shelf at CS
- Prepare CS to retrieve cap
- Feed base into CS
- Discard cap-less base
- Prepare BS to provide black base
- Retrieve base from BS
- Prepare CS to mount cap
- Feed black base to CS
- Retrieve black base with cap from CS
RoboCup Logistics League – Production Example

$C_0$ Production
- Retrieve base with cap from shelf at CS
- Prepare CS to retrieve cap
- Feed base into CS
- Discard cap-less base
- Prepare BS to provide black base
- Retrieve base from BS
- Prepare CS to mount cap
- Feed black base to CS
- Retrieve black base with cap from CS
- Prepare DS for slide specified in order
- Deliver to DS
RoboCup Logistics League – Production Example

$C_0$ Production

- Retrieve base with cap from shelf at CS
- Prepare CS to retrieve cap
- Feed base into CS
- Discard cap-less base
- Prepare BS to provide black base
- Retrieve base from BS
- Prepare CS to mount cap
- Feed black base to CS
- Retrieve black base with cap from CS
- Prepare DS for slide specified in order
- Deliver to DS

Already simple product has several fragile points and cooperation potential.
<table>
<thead>
<tr>
<th>Sub-task</th>
<th>Description</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Additional base</td>
<td>Feed an additional base into an RS</td>
<td>+2</td>
</tr>
<tr>
<td>Finish $CC_0$ step</td>
<td>Finish the work order for a color requiring no additional base</td>
<td>+5</td>
</tr>
<tr>
<td>Finish $CC_1$ step</td>
<td>Finish the work order for a color requiring one additional base</td>
<td>+10</td>
</tr>
<tr>
<td>Finish $CC_2$ step</td>
<td>Finish the work order for a color requiring two additional bases</td>
<td>+20</td>
</tr>
<tr>
<td>Finish $C_1$ pre-cap</td>
<td>Mount the last ring of a $C_1$ product</td>
<td>+10</td>
</tr>
<tr>
<td>Finish $C_2$ pre-cap</td>
<td>Mount the last ring of a $C_2$ product</td>
<td>+30</td>
</tr>
<tr>
<td>Finish $C_3$ pre-cap</td>
<td>Mount the last ring of a $C_3$ product</td>
<td>+80</td>
</tr>
<tr>
<td>Mount cap</td>
<td>Mount the cap on a product</td>
<td>+10</td>
</tr>
<tr>
<td>Delivery</td>
<td>Deliver one of the final product variants to the designated loading zone at the time specified in the order</td>
<td>+20</td>
</tr>
</tbody>
</table>
Towards a Robotics Planning Competition

RoboCup Logistics League (RCLL)
- In-factory manufacturing logistics in Smart Factory
- Maintain and optimize material flow in production
- Competition under the RoboCup umbrella

RCLL as a Planning Competition and Benchmark
- Cooperative and competitive aspects, partially observable, non-deterministic, dynamic
- Typical: local, distributed, incremental strategy
- Desired: planning for global optimization
- Challenges: coordination, execution, robustness
Towards a Robotics Planning Competition

RoboCup Logistics League (RCLL)
- In-factory manufacturing logistics in Smart Factory
- Maintain and optimize material flow in production
- Competition under the RoboCup umbrella

RCLL as a Planning Competition and Benchmark
- Cooperative and competitive aspects, partially observable, non-deterministic, dynamic
- Typical: local, distributed, incremental strategy
- Desired: planning for global optimization
- Challenges: coordination, execution, robustness

Medium complex benchmark domain focusing on efficient planning/scheduling and execution integration
CLIPS-based Incremental Task-Level Reasoning

- Only commit to single step at a time
- Strategic behavior with coarse tasks
- Reason about current knowledge

- CLIPS rule-based system
- Efficient reasoning with many updates
- Distributed, local-scope, incremental

(defrule s1-t23-s0
  (state IDLE) (holding S1)
  (machine (mtype ?mt&M2\_3) (name ?n)
    (loaded-with $?lw&:(contains$ S0 ?lw)) )
=>
  (assert (task-candidate goto ?n)) )
RoboCup Logistics League – Simulation

- Readily integrated 3D simulation with environment agency
- Based on software stack by RoboCup Team Carologistics
Simulation Architecture

Gazebo

RCLL Environment

Models

Gazebo Robot 1
Motor, Laser, Cam, ...

Gazebo Robot 2
Motor, Laser, Cam, ...

Gazebo Robot 3
Motor, Laser, Cam, ...

Gazebo API

Referee Box

Visualization

Robot 1
Fawkes, ROS, ...

Robot 2
Fawkes, ROS, ...

Robot 3
Fawkes, ROS, ...
Fawkes Robot Software Framework
- Functional software components
- Lua-based Behavior Engine for skill execution
- Path planning and locomotion

ROS
- Full integration with simulation
- Encapsulates communication with referee box
- Visualization tool
Planning System Architecture

Task-Level Planning Executive
  - Task Planner
  - Execution Monitoring System
    - Domain Model
    - State Estimator
  - Robot Software Stack
    - Behaviors/Skills
  - Simulation Environment
    - Agency

Referee Box
  - Goals
  - Execution Status
  - Execution Request
  - Behaviors/Skills Execution Request
  - Feedback
Tracks are determined by what can be replaced or extended.

**Track 1: Planner and Execution**
- Domain model, task planner, execution monitoring, state estimator

**Track 2: Behaviors and Motion Planning**
- Skills/behaviors, execution engine, motion planner (local/global)

**Track 3: Free style**
- Any component but the simulation (parts) and its interface.
Planning and Robotics Competition

**Challenge**
Integrated planning and execution in a medium complex simulated robotics industry-inspired scenario

**Tracks**
Accommodate diversity by creating several tracks that have their specific extension points (discussion)

**Timeline**
2015: Presentation of RoboCup Logistics League scenario
2016: Tutorial, discussion of scenario and tracks
2017: Competition at ICAPS 2017
Planning Challenges
Erez Karpas
Planning

Initial state and Goal

Planner

Plan
Planning and Execution

- Initial state and Goal
- Executive
- Planner
- Plan
Planning and Execution with Plant

Initial state and Goal

Plan

Executive

State

State estimator

Plant

Action
Challenges: Planning

- Finding sequence of actions for each order — easy
- Assigning robots to orders/subtasks
- Large number of objects
  - Our domain formulation keeps track of each workpiece by its identity
  - Including workpieces that are symmetric
Challenges: Planning and Execution

- Orders arrive on the fly
  - Replanning
  - Anticipating orders
- Competition
  - What is the objective?
Challenges: Planning and Execution with Plant

- Actions can:
  - fail (in predictable and unpredictable ways)
  - take too long (or too little)
- Adjust plan based on opponent/ranking?
Running the Simulation
Starting the Simulation

1. Boot from USB stick
2. Login: robosim/simcomp2017
3. Get laptop on wifi (e.g., eduroam)
4. Run Terminal
5. Update and rebuild simstick

```
# simstick-update
# simstick-rebuild
```

6. Start Simulation

```
# cd robotics/fawkes-robotino/bin
# ./gazsim.bash -x start -r -n 1 -t -a
```

7. In Gazebo, hit F11 twice to get to window mode
Running the Game

### Attention Message
**Robot 2 R-2/Magnum at 172.26.124.22 Lost**

#### RefBox Log
- **21:50:11.463 C**: Machine C-RS1 finished processing, moving to output
- **21:50:11.463 C**: Simulated output at C-RS1
- **21:50:11.463 C**: Machine C-RS1 MPS state DELIVERED (bases added: 0)
- **21:50:11.464 C**: Machine C-RS1 finished processing, ready at output
- **21:50:11.622 C**: Client 23947429 (:ffff:127.0.0.1) disconnected
- **21:50:28.183 C**: Machine M-CS1 recovered
- **21:50:28.183 C**: Machine M-CS1 switching to IDLE state
- **21:51:46.036 C**: Client 23947430 connected from :ffff:127.0.0.1:58034
- **21:51:46.069 C**: Received state AVAILABLE for machine C-CS2
- **21:51:46.069 C**: Machine C-CS2 MPS state AVAILABLE (bases added: 0)
- **21:51:46.070 C**: Machine C-CS2 broken: Input to C-CS2 while not prepared IDLE
- **21:51:46.070 A**: Input to C-CS2 while not prepared IDLE
- **21:51:46.236 C**: Client 23947430 (:ffff:127.0.0.1) disconnected

### Machines
- **C-BS**
- **C-DS**
- **C-RS1**
- **C-RS2**
- **C-CS1**
- **C-CS2**
- **M-BS**
- **M-DS**
- **M-RS1**
- **M-RS2**
- **M-CS1**
- **M-CS2**

### Robots
1. **R-1 (Carologistics)**
   - 172.26.108.81
   - Status: ACT
2. **R-2 (Carologistics)**
   - 172.26.108.82
   - Status: ACT
3. **R-3 (Carologistics)**
   - 172.26.108.83
   - Status: ACT

### Orders
- **1. 4/6/1**
- **3. 0/0/2**
- **5. 0/0/1**
- **7. 6/0/1**

### Game
- **State**: RUNNING
- **Phase**: PRODUCTION
- **Time**: 10:42:199
- **Points**: 83 / 42
- **Cyan**: Carologistics
- **Magenta**: Magnum

### RefBox 0.9.0
- F2 STATE
- F3 PHASE
- F4 TEAM
- F9 ROBOT
- F12 DELIVER

---

26 / 28
Explore ROS Integration

Start Simulation with ROS Integration

```
# cd robotics/fawkes-robotino/bin
# ./gazsim.bash -x start -r -n 1 -t \
   --ros-launch-main \ 
   rcll_fawkes_sim:rcll_fawkes_sim_all_1robot.launch
```

Note: **no** `-a` flag!

Explore Topics

```
# rostopic list
```

- Per robot namespace, `/robot1` etc.
- `/robotN/rcll`: communication with referee box
- `/robotN/rcll_sim`: simulation integration
- `/robot1/skiller`: skill execution action
Run Skill via ROS Action

```bash
# rosrun actionlib axclient.py \\
/robot1/skiller fawkes_msgs/ExecSkillAction
```

Send the following goal:
```
skillstring: 'ppgoto{place=“C-CS1-I”}'
```
(requires to be in the production phase in refbox)