

Distributed Planning

Artur Daudrich

Research lab – Summer term 2014

University Koblenz-Landau

Agenda

- Introduction
- Task Sharing
- Result Sharing
- Distributed Planning
- Planning & Execution

Introduction

What is a Distributed Problem Solving and Planning?

- agents work together to solve problems...
 - ...which cannot be accomplished by one agent
 - ...which are accomplished better with others
 - more quickly, completely, precisely, certainly

Motivation

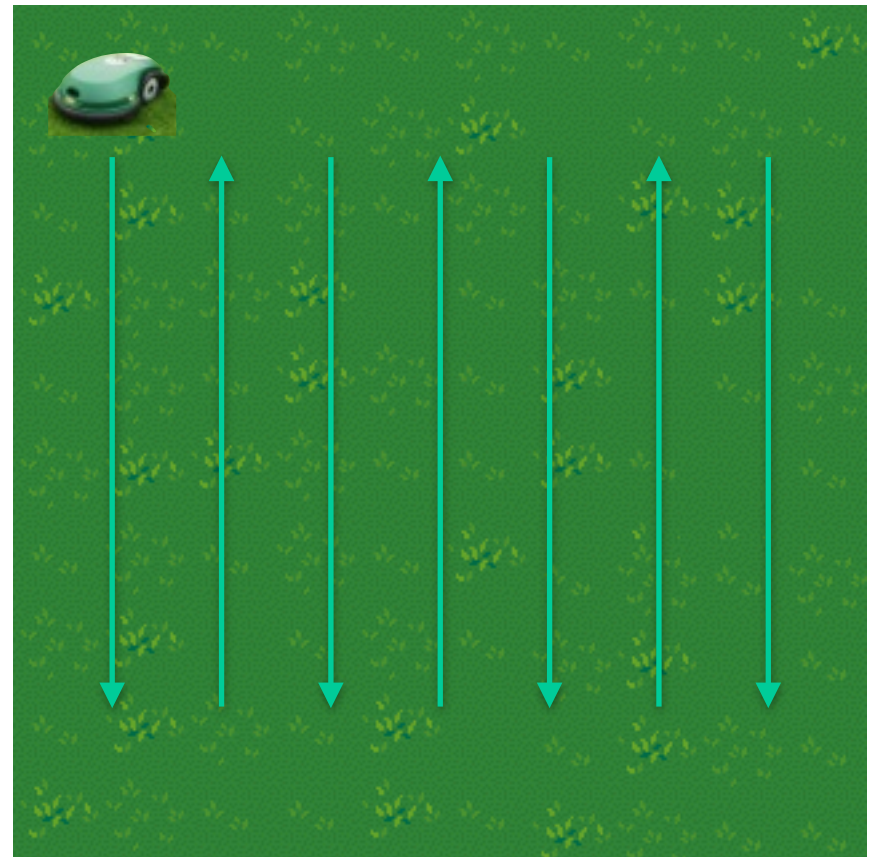
Garden Agent System

Task: Mow the lawn (1 agent)

General Strategy:

1. define requirements for solution
2. find solution for problem
3. check if solution satisfies requirements
4. solve problem or find other solution

Optimize -> Homogeneous System



Task Sharing

Homogeneous Systems

What is a Homogeneous System?

Assumption:

Every agent has the same capabilities and the same expertise (omni-capable agents).

General Idea:

If an agent is assigned with too many or a too large task, enlist help of other agents with few or no tasks.

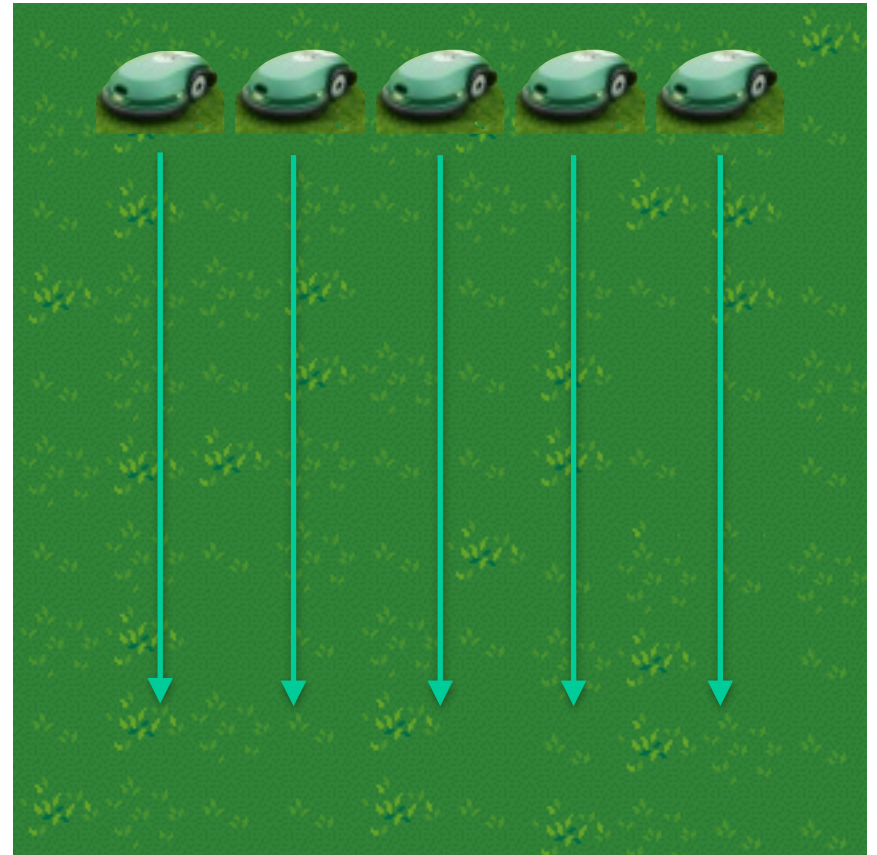
Task Sharing

Homogeneous Systems

Task: Mow the lawn (system of agents)

Planning Strategy:

1. Task decomposition
2. Task allocation
3. Task accomplishment
4. Result synthesis



Task Sharing

Homogeneous Systems

Task: Mow the lawn & clean the pond

- Need for agents with extended capabilities
- What if new task: „Cut the hedge“
- Omni-capable agent ?



Task Sharing

Homogeneous Systems

Problem:

- omni-capable agents often overkill
- most of capabilities wasted

Solution:

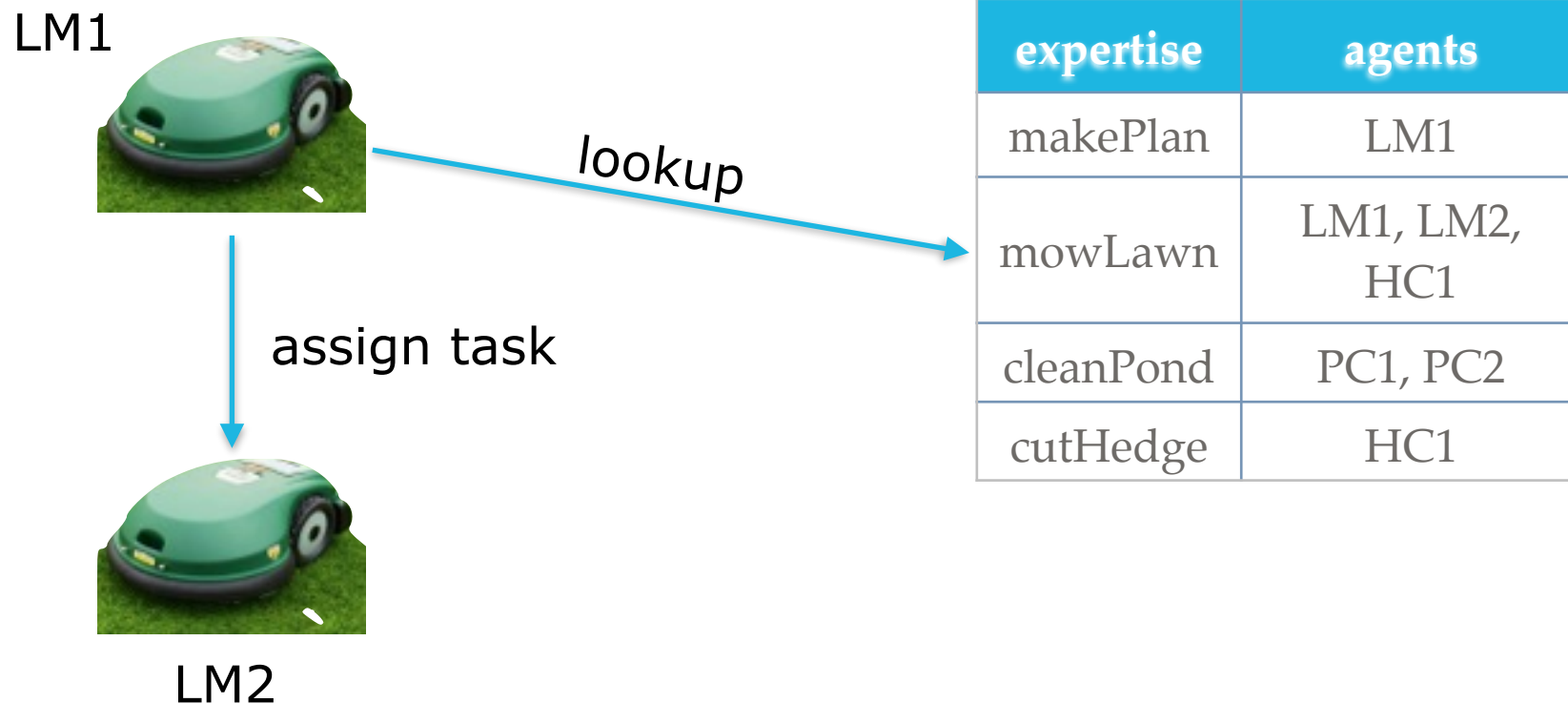
- Heterogeneous Systems: combination of specialists



Task Sharing

Heterogeneous Systems

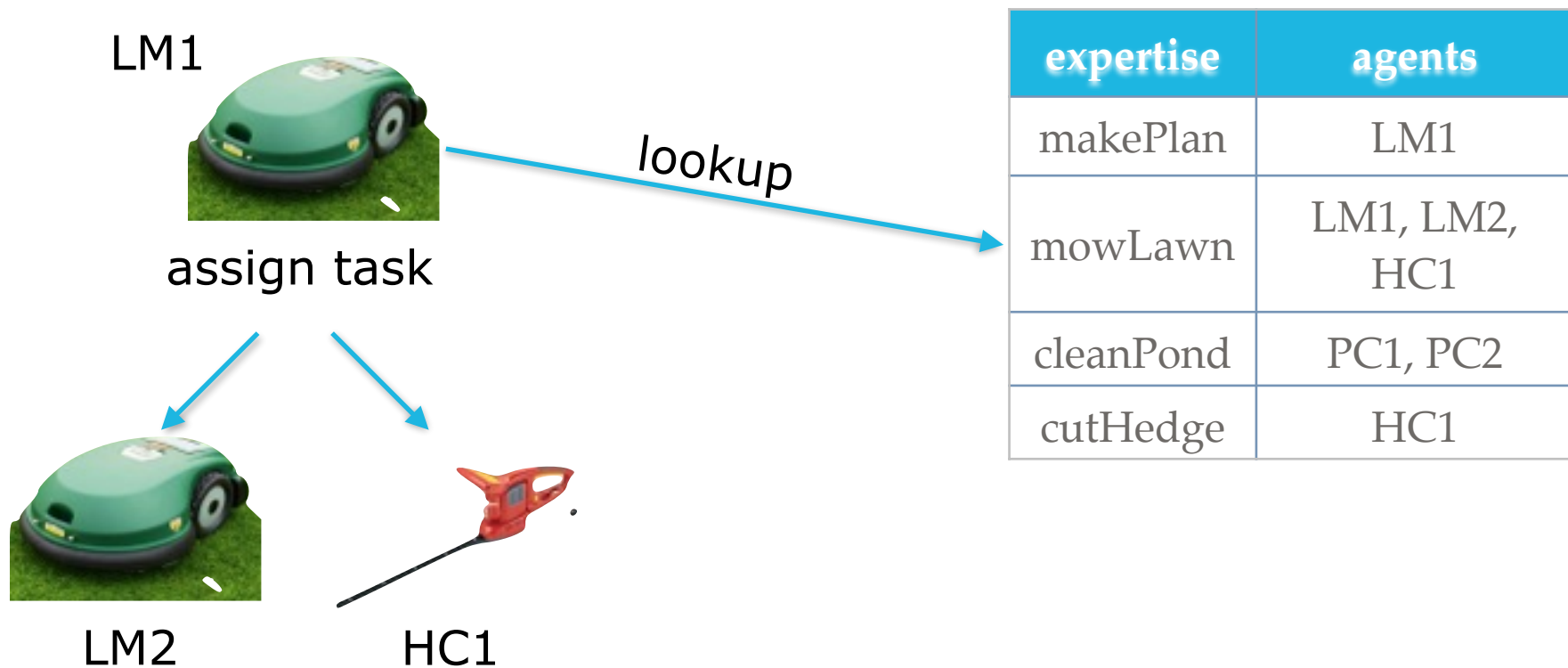
Assign Tasks: Directed Contract



Task Sharing

Heterogeneous Systems

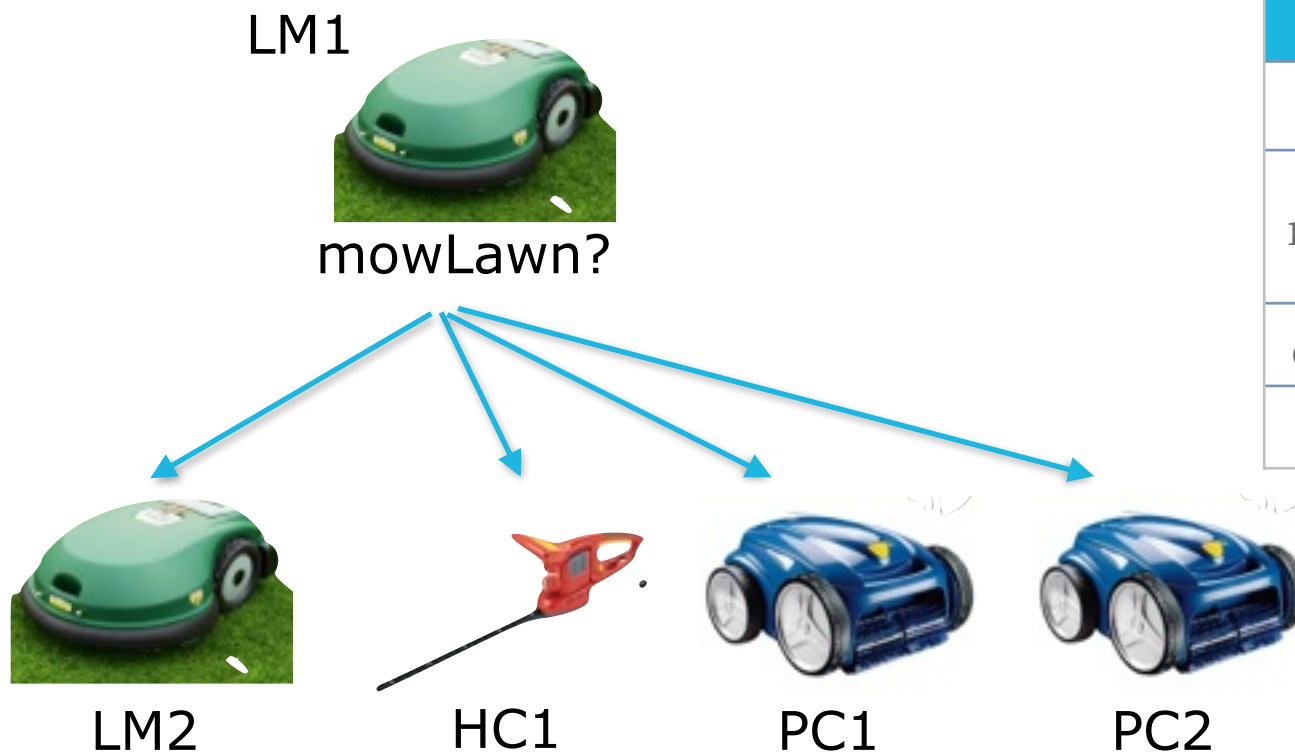
Assign Tasks: Focussed Addressing



Task Sharing

Heterogeneous Systems

Broadcast Contracting

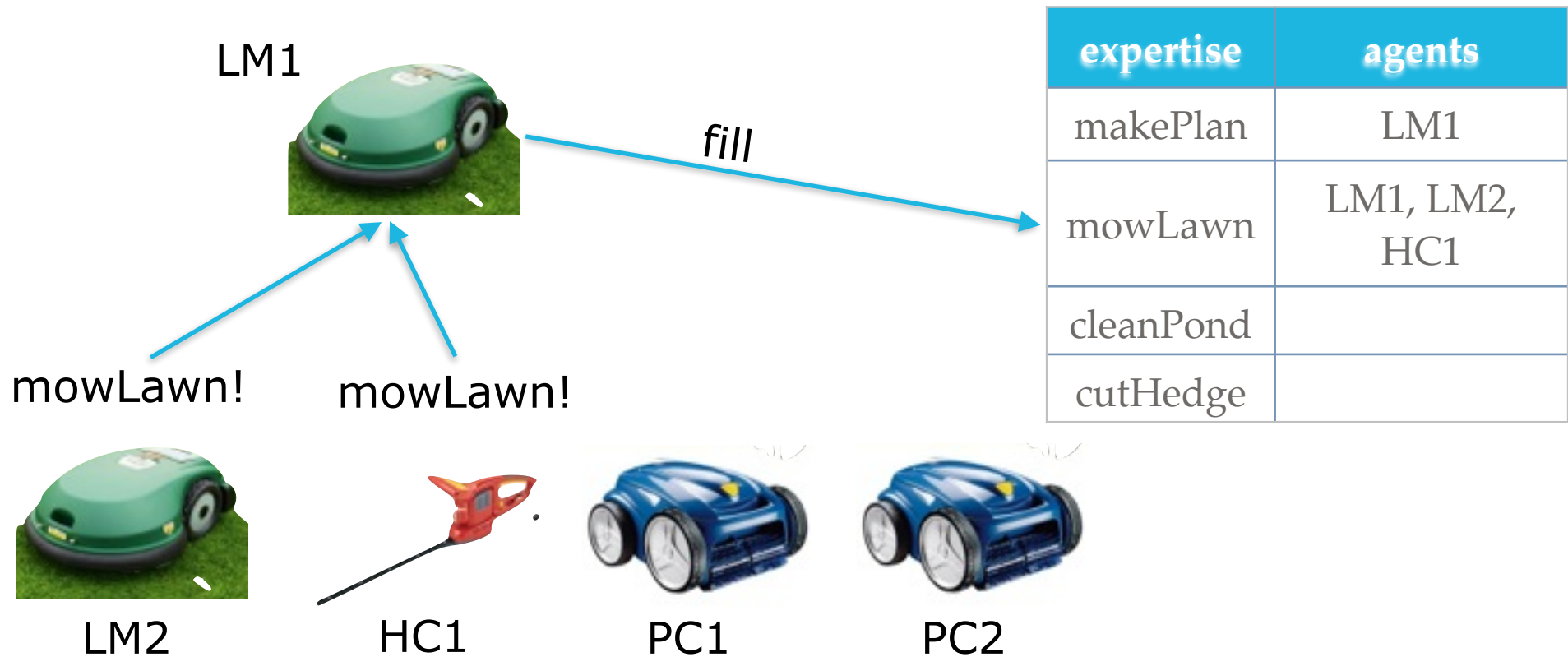


expertise	agents
makePlan	LM1
mowLawn	LM1
cleanPond	
cutHedge	

Task Sharing

Heterogeneous Systems

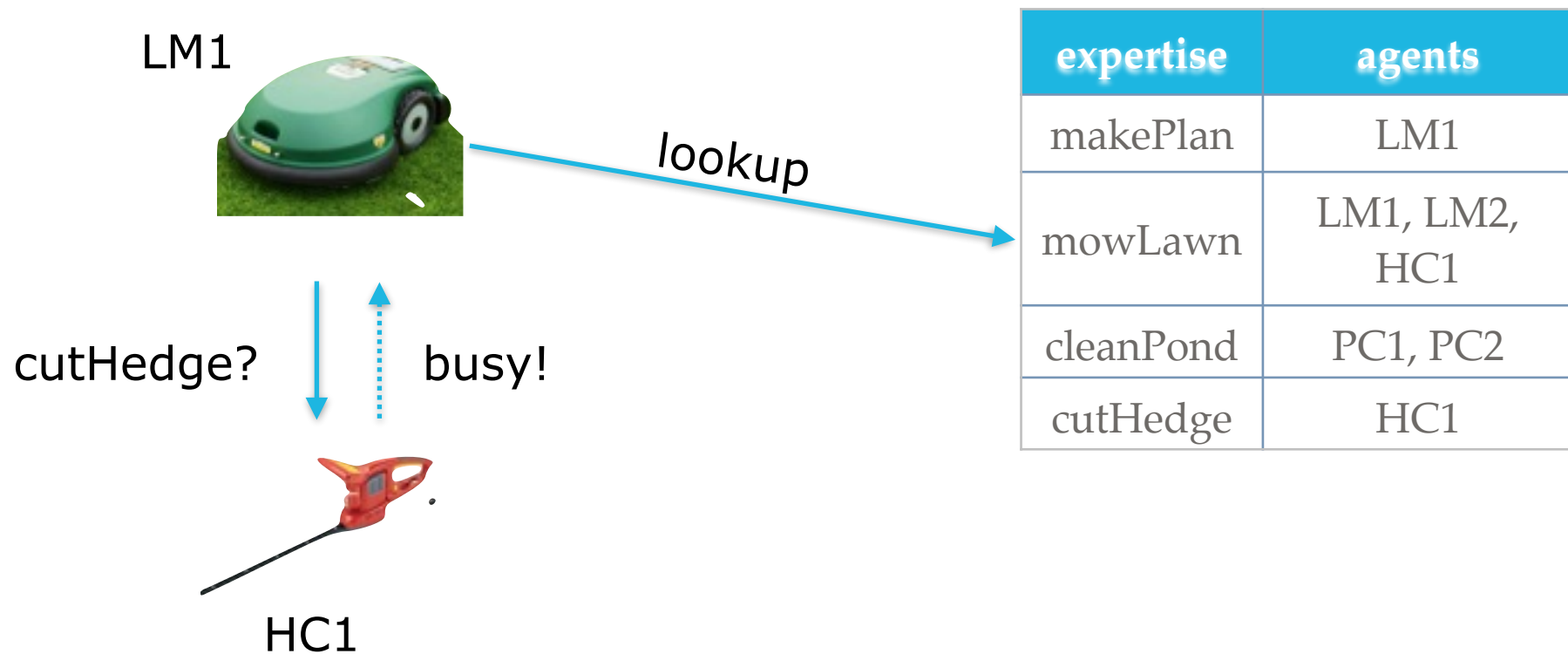
Broadcast Contracting



Task Sharing

Heterogeneous Systems

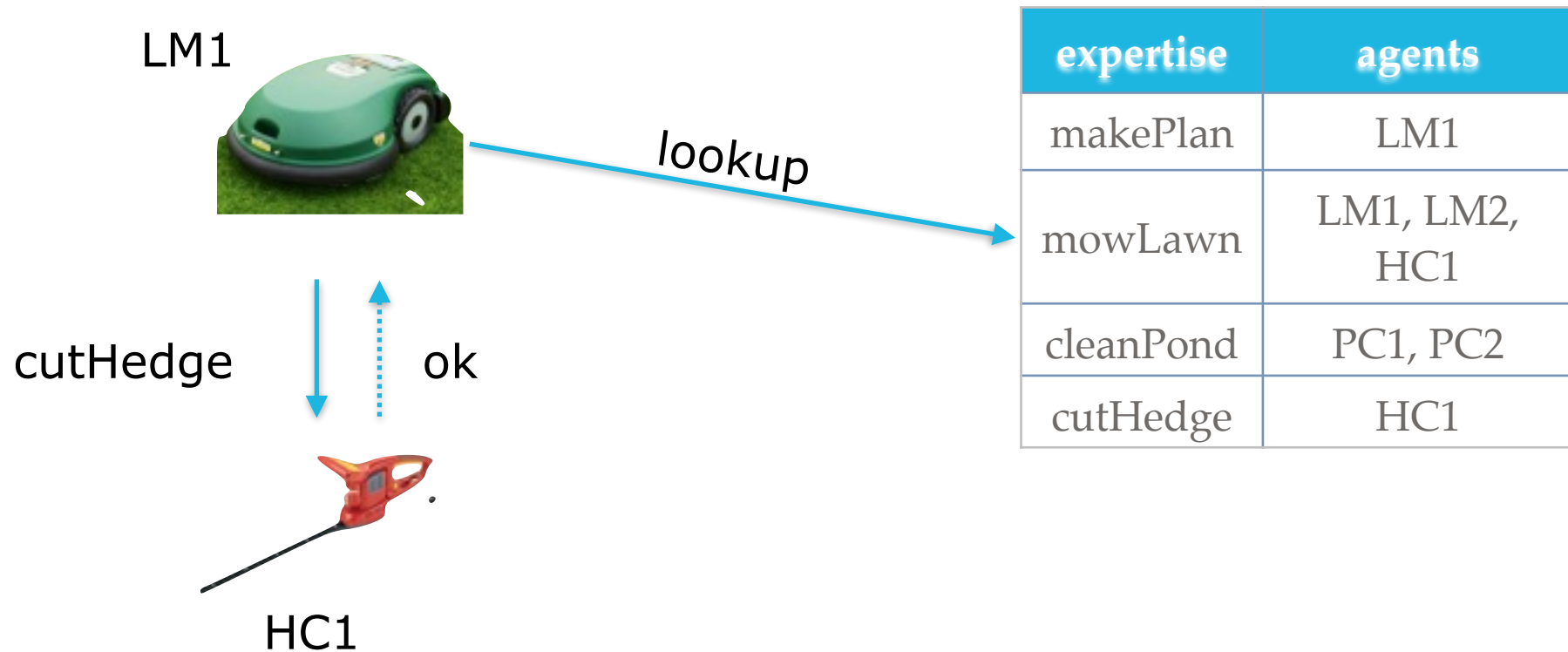
Retry Strategy



Task Sharing

Heterogeneous Systems

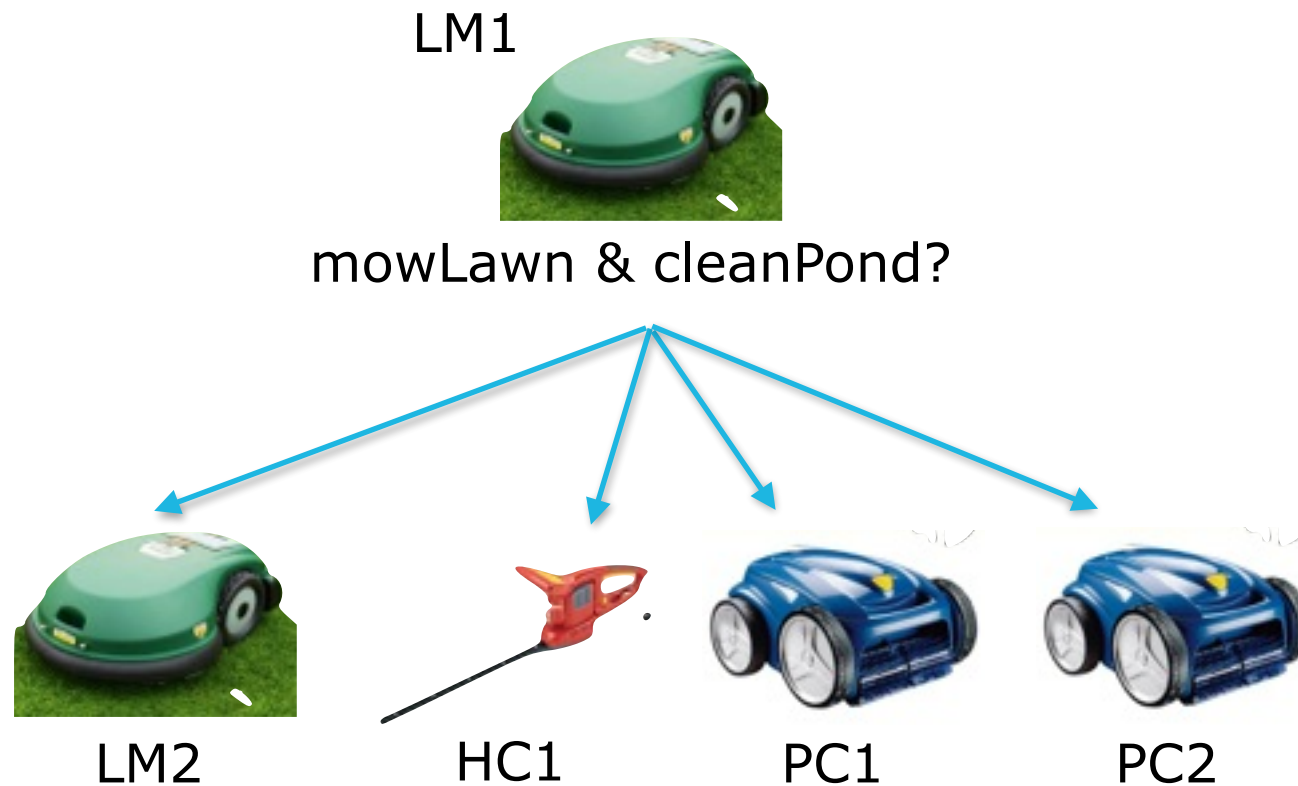
Retry Strategy



Task Sharing

Heterogeneous Systems

Announcement Revision



Task Sharing

Heterogeneous Systems

Announcement Revision

LM1



mowLawn & cleanPond?



LM2



HC1



PC1

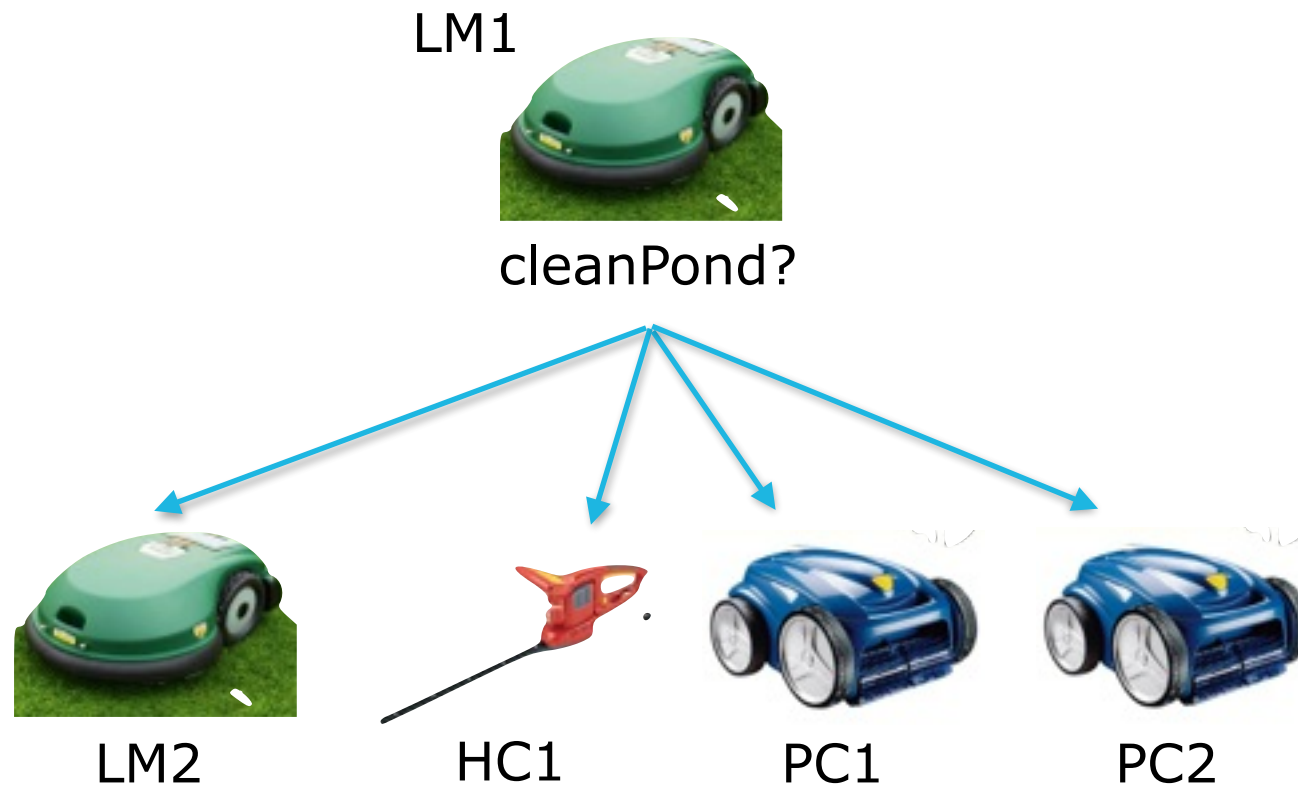


PC2

Task Sharing

Heterogeneous Systems

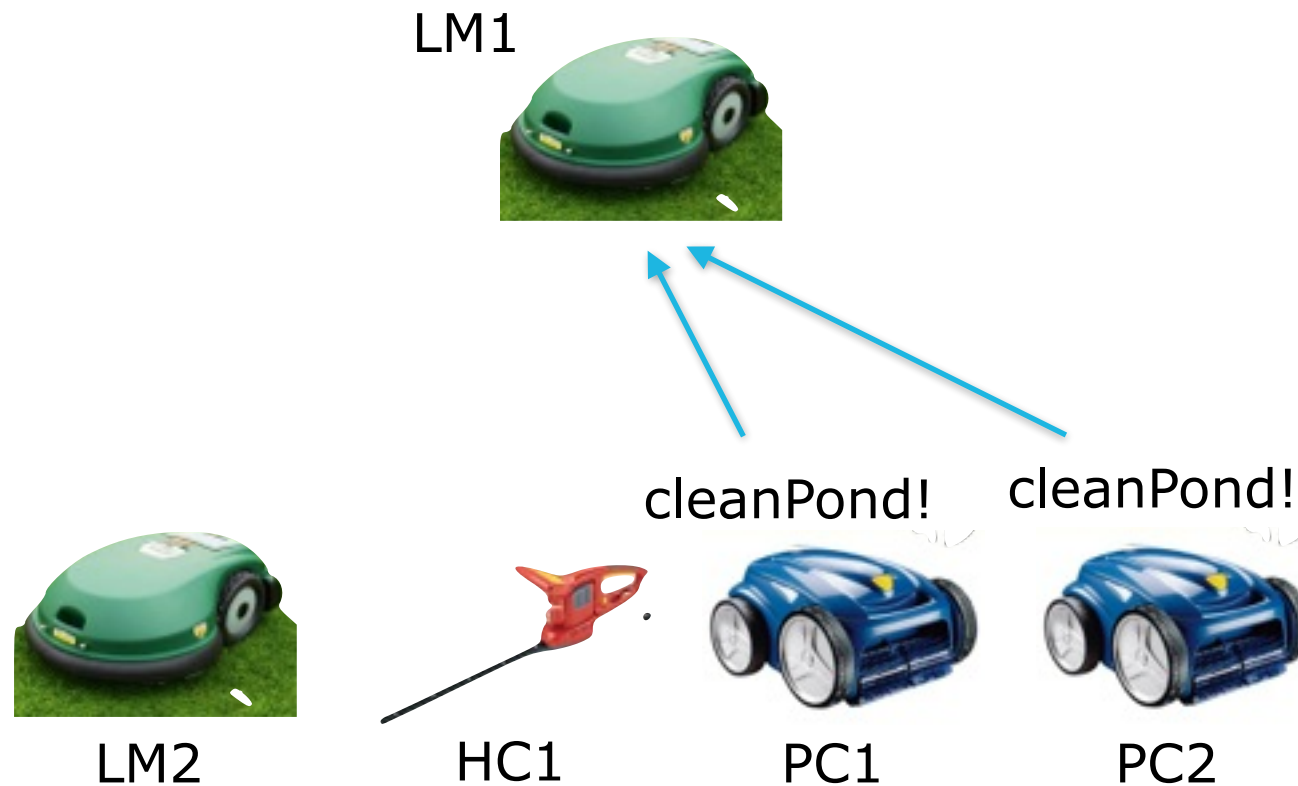
Announcement Revision



Task Sharing

Heterogeneous Systems

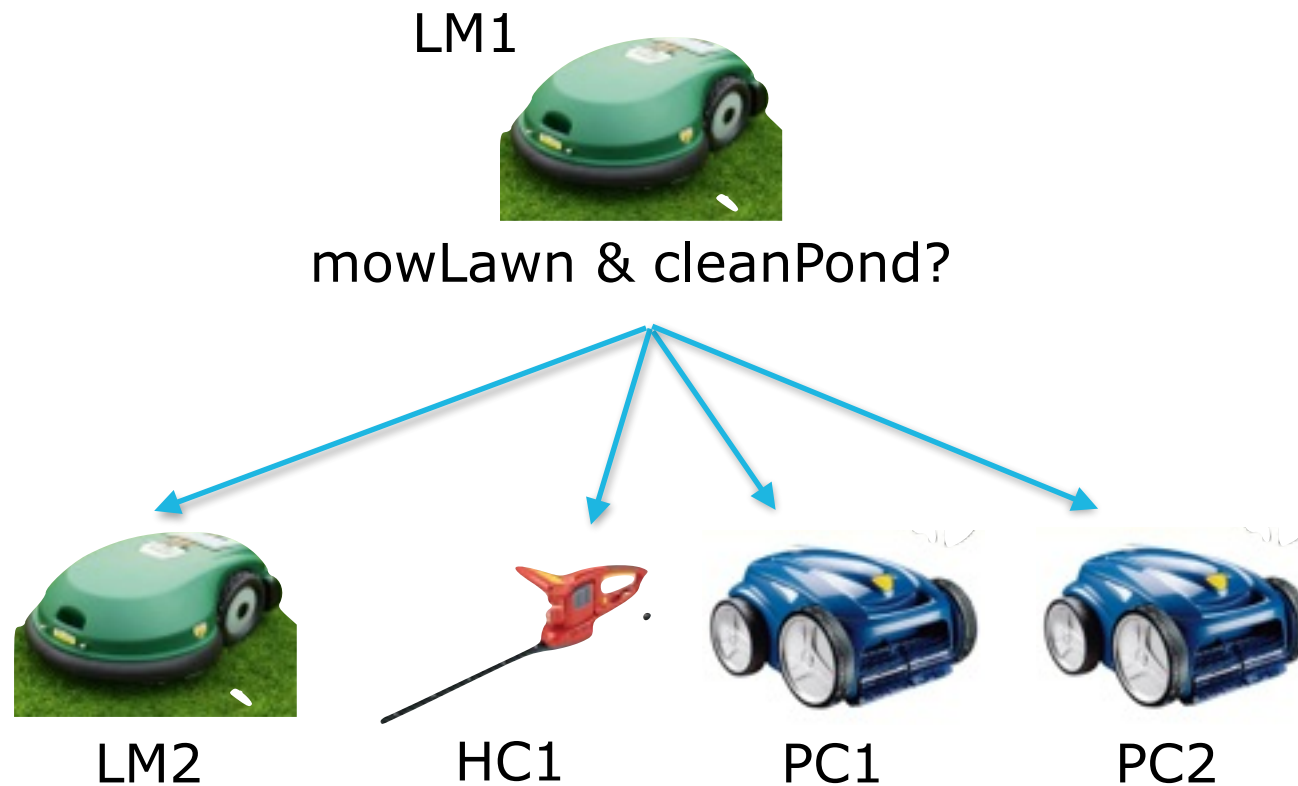
Announcement Revision



Task Sharing

Heterogeneous Systems

Alternative Decomposition



Task Sharing

Heterogeneous Systems

Alternative Decomposition

LM1



mowLawn & cleanPond?



LM2



HC1



PC1

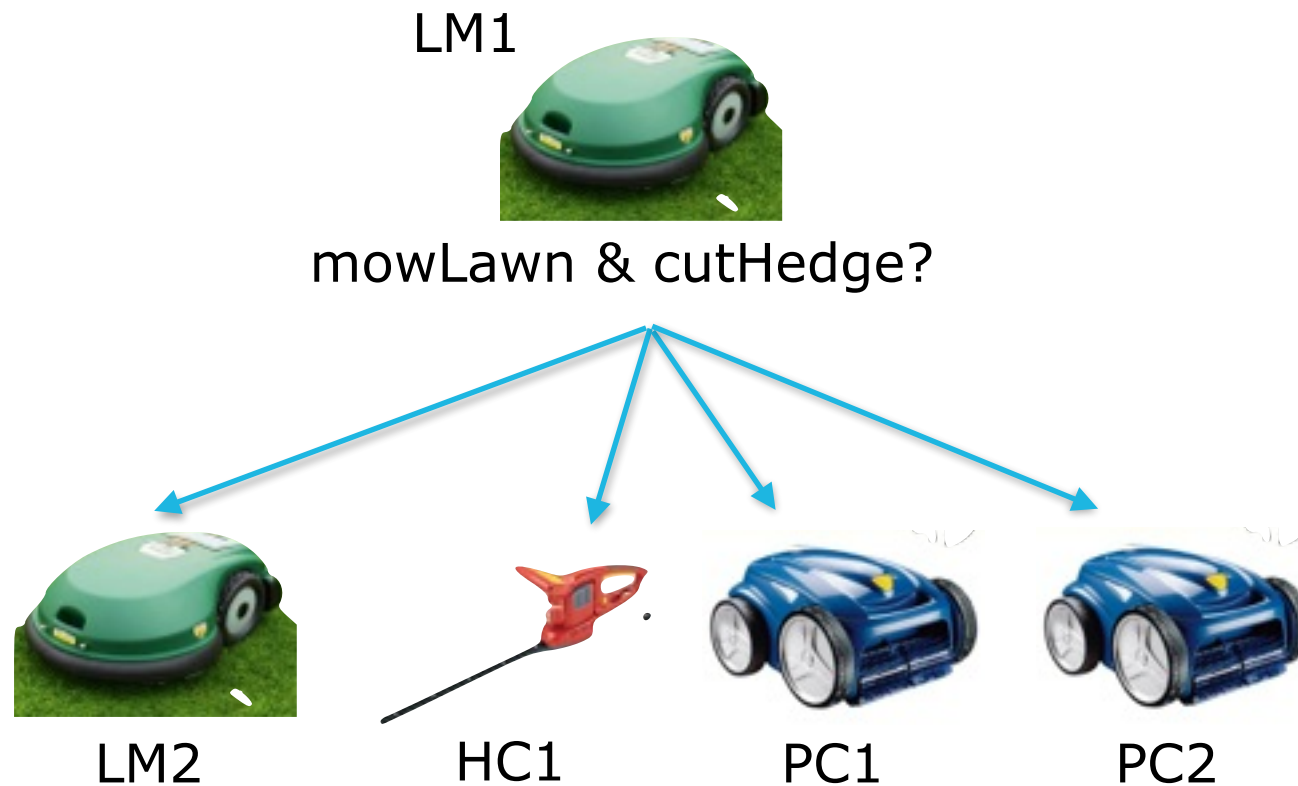


PC2

Task Sharing

Heterogeneous Systems

Alternative Decomposition



Task Sharing

Heterogeneous Systems

Alternative Decomposition



Result Sharing

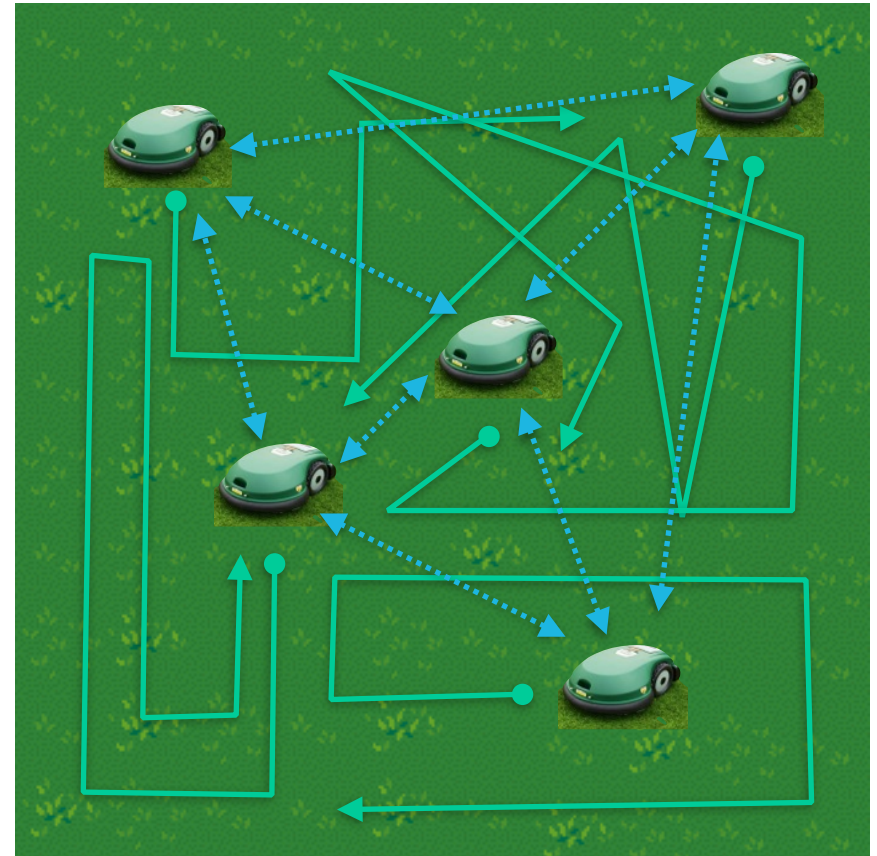
Why is Result Sharing necessary?

- **dependency** on other tasks
- **confidence** of correct results
- **completeness** of tasks
- more **precise** solutions

Result Sharing

Functionally Accurate Cooperation

- formulate tentative results (functionally-accurate)
- iterative exchange of partial solutions (cooperation)
- impact on completeness, precision, confidence
- leads to overall solution



agent path
→

communication
↔

Result Sharing

Functionally Accurate Cooperation

Problem:

- communication overhead
- wasted computation
- too many shared results (distraction)
 - all agents do the same problem-solving actions

Solution:

- limiting communication (Shared Repository)

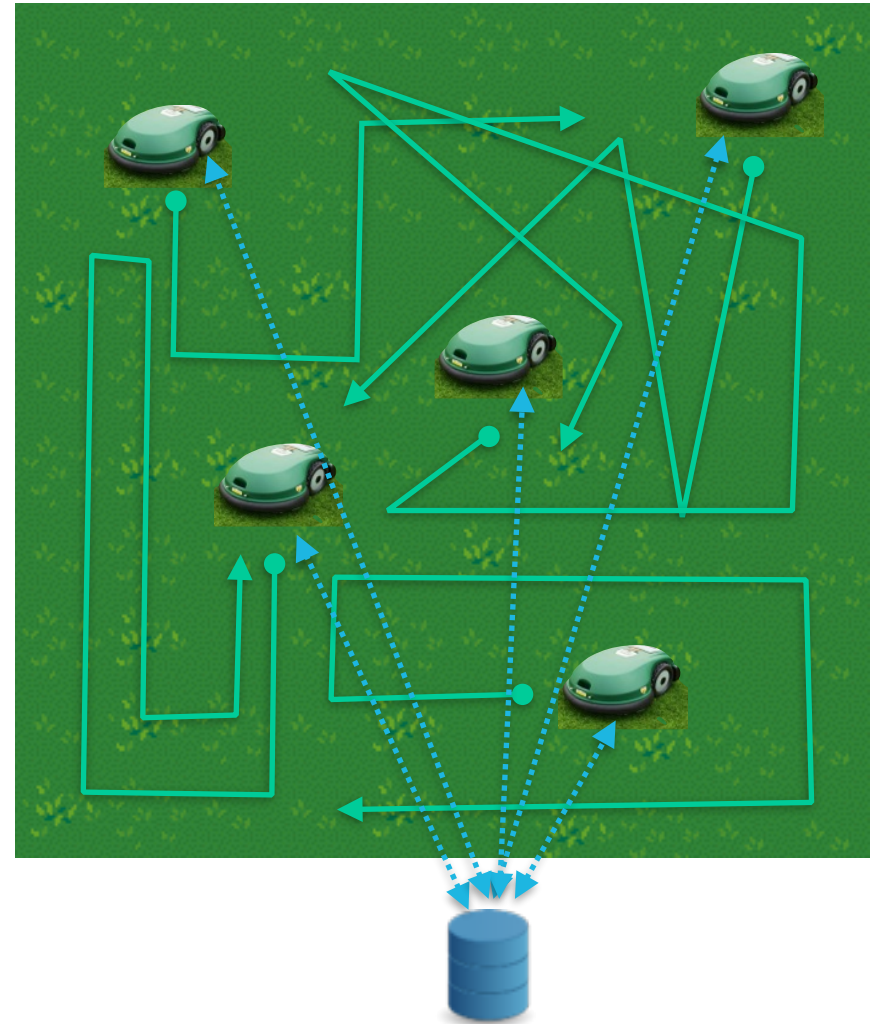
Result Sharing

Shared Repository

- store partial results in single shared repository (initiate)
- search through results
- extend & critique results
 - improve results
 - reject results
 - relax expectations (store rejected results)

agent path


communication

Result Sharing

Communication Strategies

- send all partial results to everyone
- only send results if complete
- only send results to interested agents at right time
 - too late -> delayed actions / not useful anymore
 - too early -> clutters memory
- only send results if requested

Result Sharing

How to detect lost messages?

- repeat sending messages until acknowledged or timeout
- predict & observe change of behavior of recipient

Distributed Planning

What is Distributed Planning?

Distributed Planning is a specialization of Distributed Problem Solving. The Problem to solve is to construct a plan.

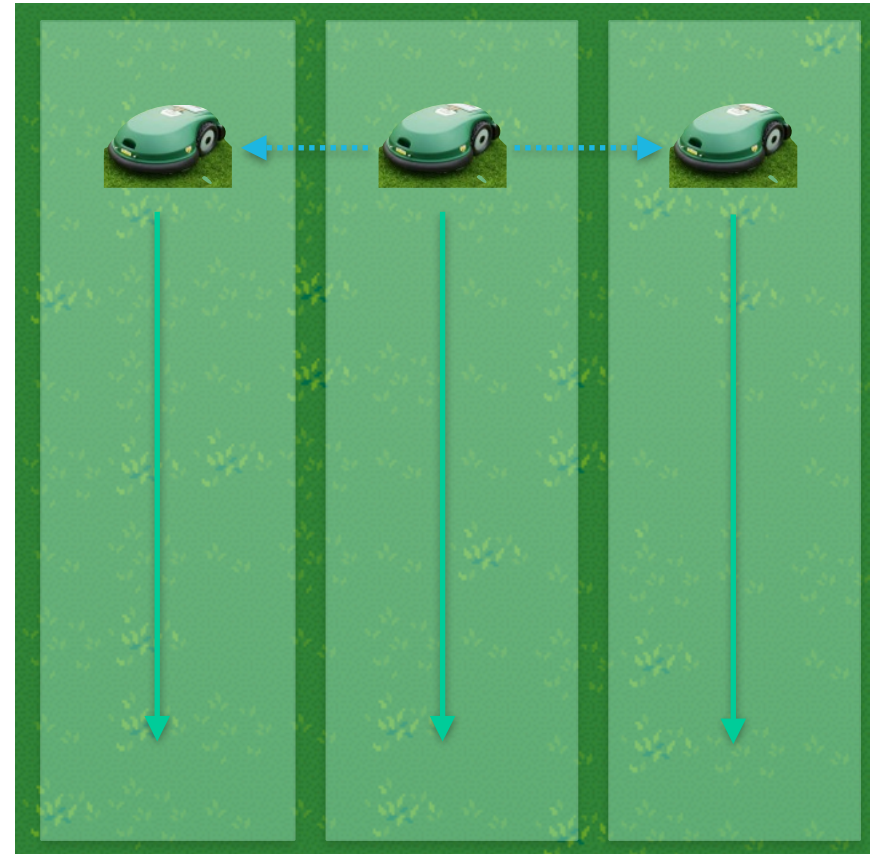
Which types of Distributed Planning exist?

- distribution of plans on execution systems
- distributing of the planning process
- distribution of the planning and execution process

Distributed Planning

Centralized Planning for Distributed Plans

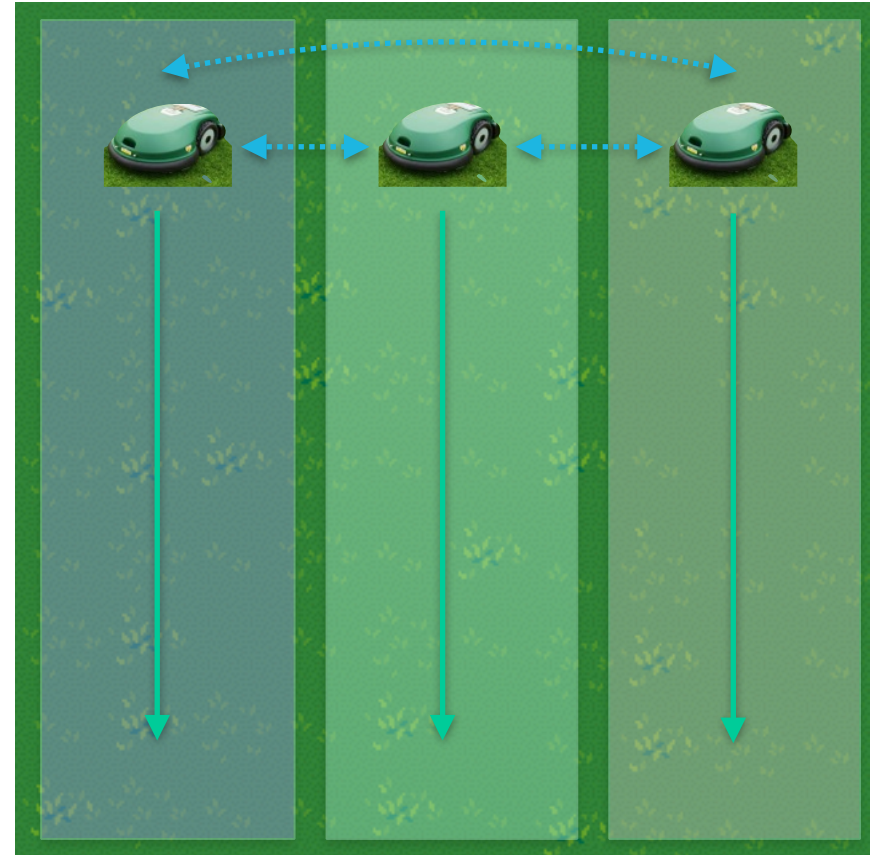
- centralized coordinator agent
 - find plan with few dependencies or ordering constraints
 - split up plan
 - synchronize actions
 - assign plan pieces to agents
 - trigger & monitor plan execution



Distributed Planning

Distributed Planning for Centralized Plans

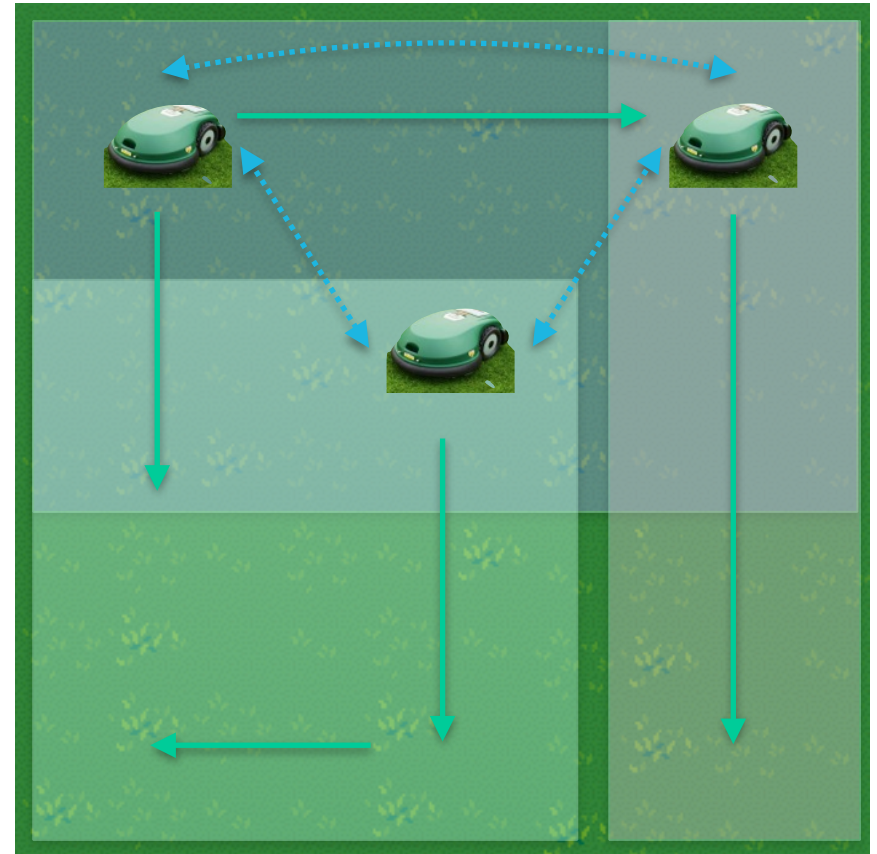
- collaboration among cooperative planning specialists
- generate partial-specified plans in parallel
- exchange & share plans
- merge partial plans to complete plan
- backtrack if stuck on subtask



Distributed Planning

Distributed Planning for Distributed Plans

- multiple agents formulate plans for themselves
- ensure that plan can be executed without conflict
- agents should preferably help each other
- Challenge: identify & resolve conflicts
- centralized plan coordination approach (one agent plans planning)

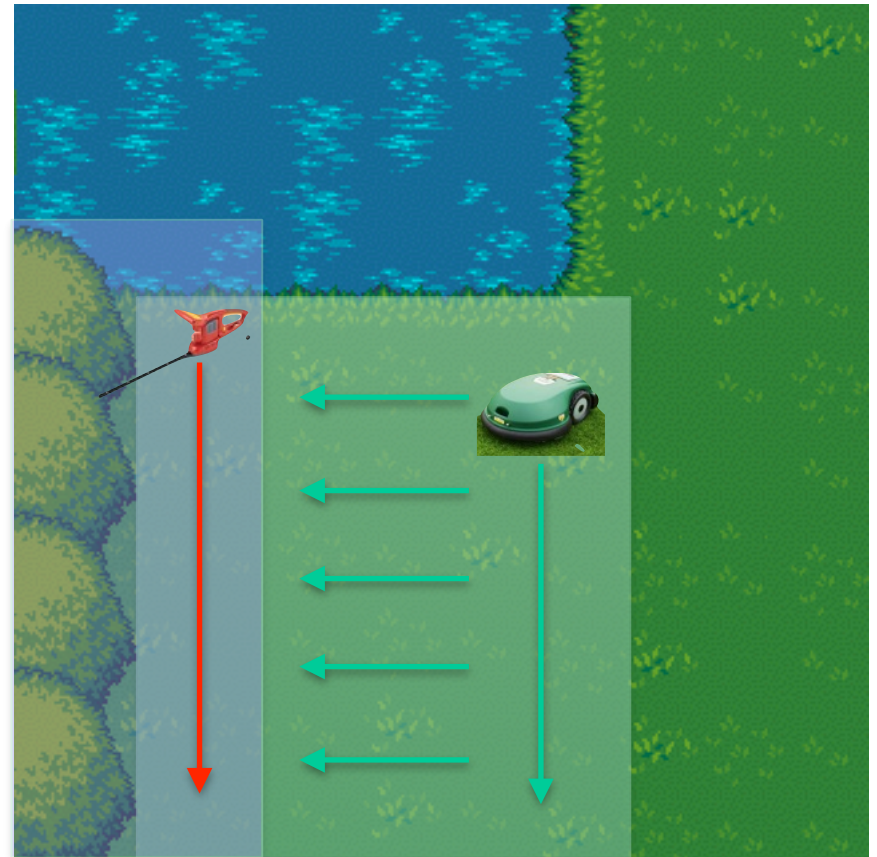


Distributed Planning

Distributed Planning for Distributed Plans

Plan Merging

- analyze interactions between pairs of actions (different agents)
 - independent actions -> no conflict
 - dependent actions -> switch execution order
 - conflicted actions -> restrict or suspend actions
- plan synchronization
 - messages as signals
 - schedule with time frame

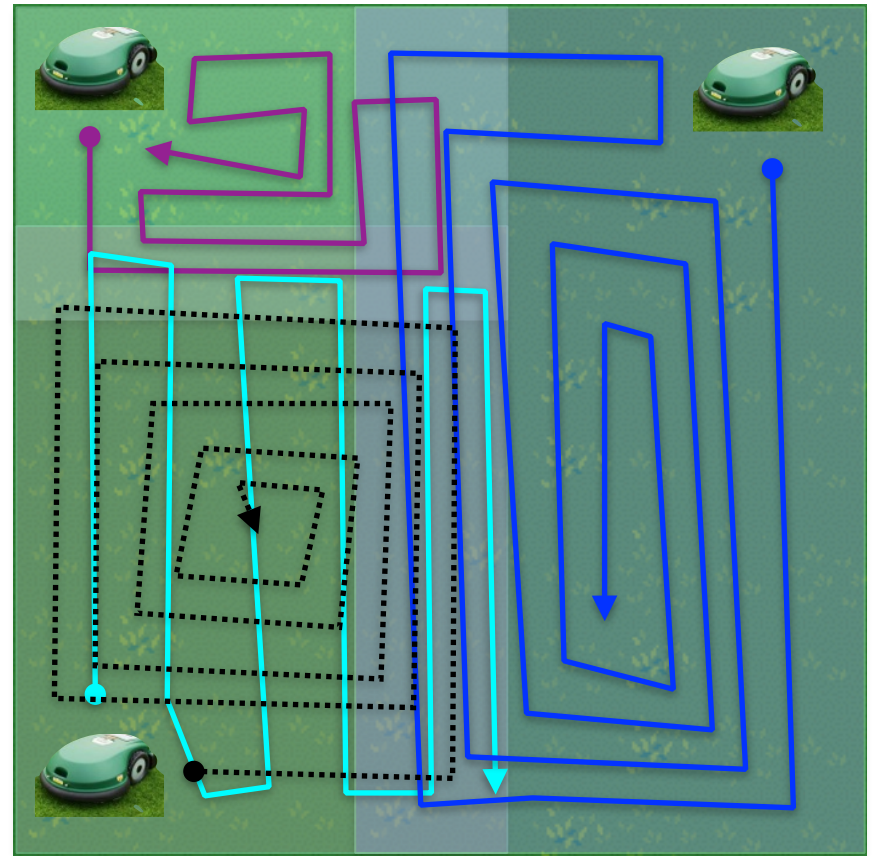


Distributed Planning

Distributed Planning for Distributed Plans

Iterative Plan Formation

- every agent constructs a set of feasible plans to reach its goal

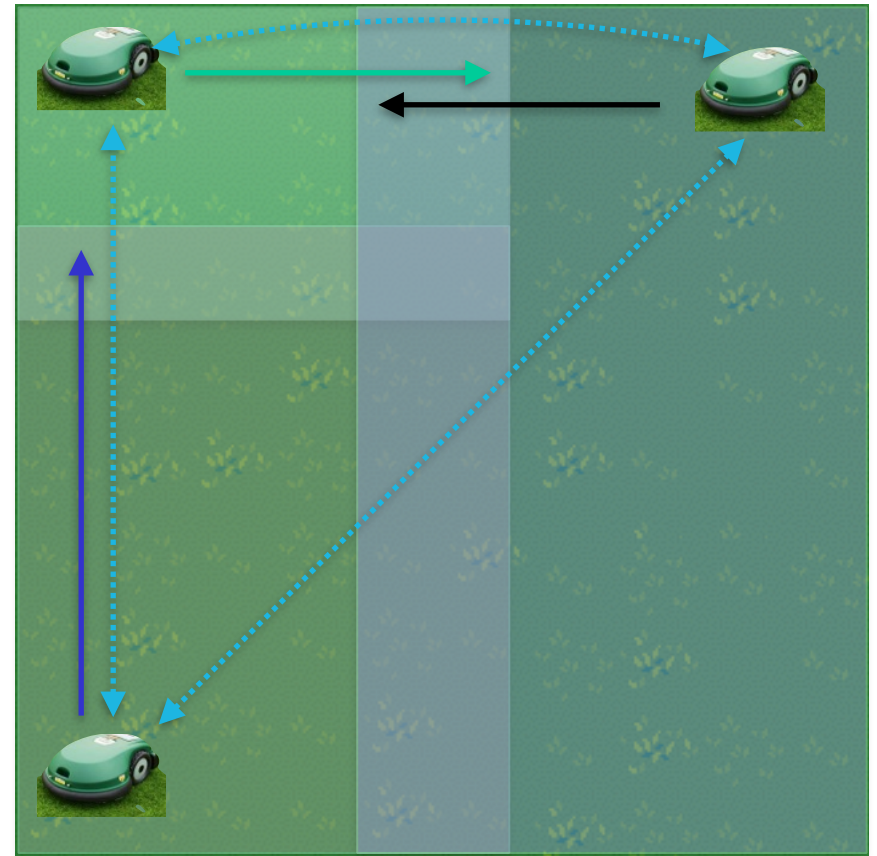


Distributed Planning

Distributed Planning for Distributed Plans

Iterative Plan Formation

- every agent constructs a set of feasible plans to reach its goal
- agents propose single action to other agents

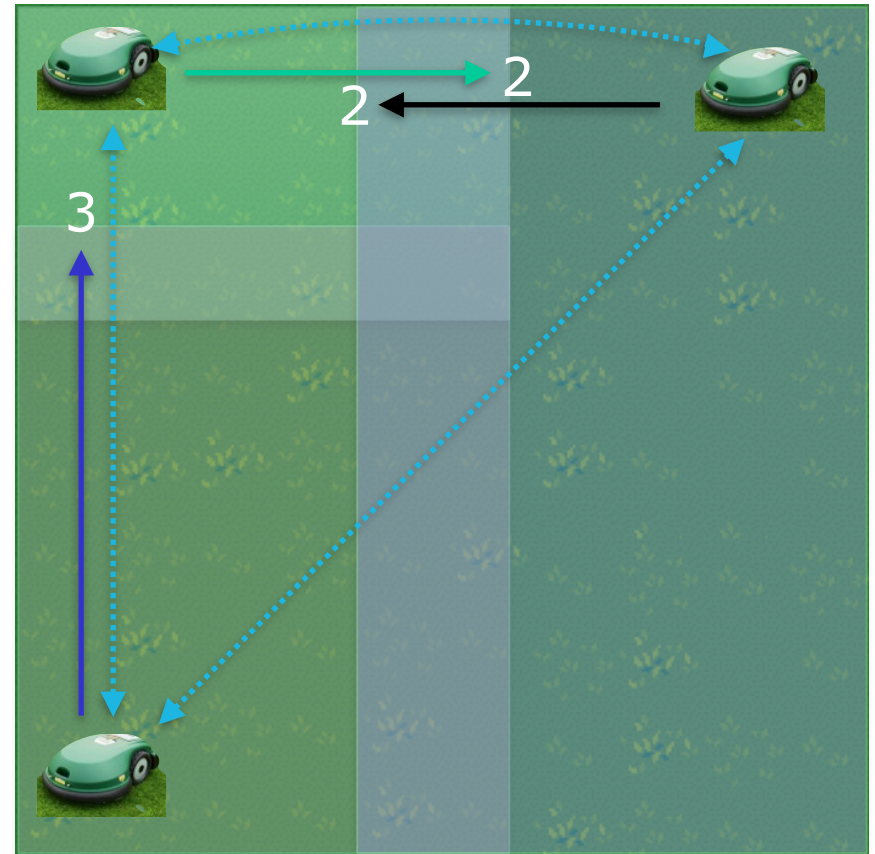


Distributed Planning

Distributed Planning for Distributed Plans

Iterative Plan Formation

- every agent constructs a set of feasible plans to reach its goal
- agents propose single action to other agents
- all actions will be rated
- best rated action will be chosen

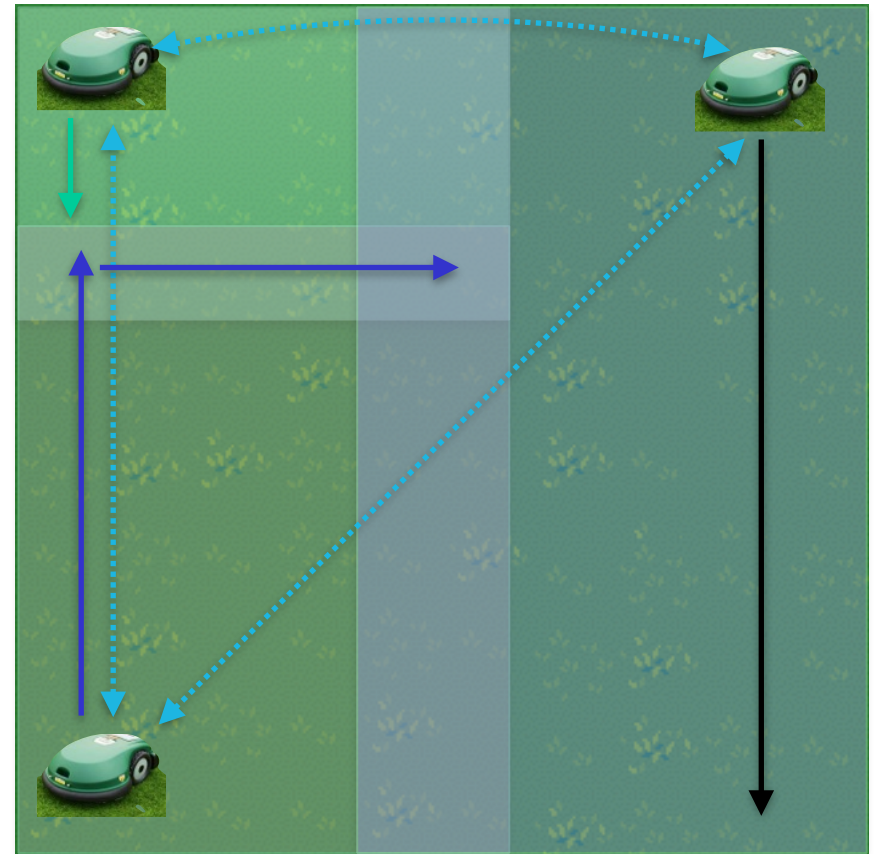


Distributed Planning

Distributed Planning for Distributed Plans

Iterative Plan Formation

- every agent constructs a set of feasible plans to reach its goal
- agents propose single action to other agents
- all actions will be rated
- best rated action will be chosen
- refine subset based on chosen action

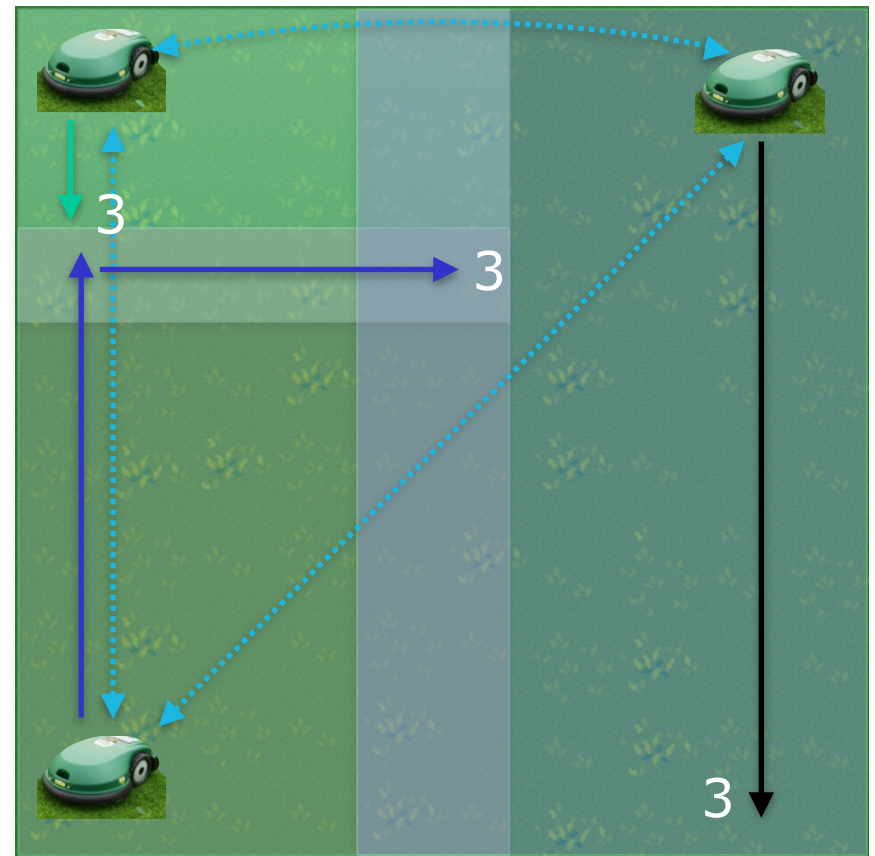


Distributed Planning

Distributed Planning for Distributed Plans

Iterative Plan Formation

- every agent constructs a set of feasible plans to reach its goal
- agents propose single action to other agents
- all actions will be rated
- best rated action will be chosen
- refine subset based on chosen action
- repeat proposing until no further changes needed

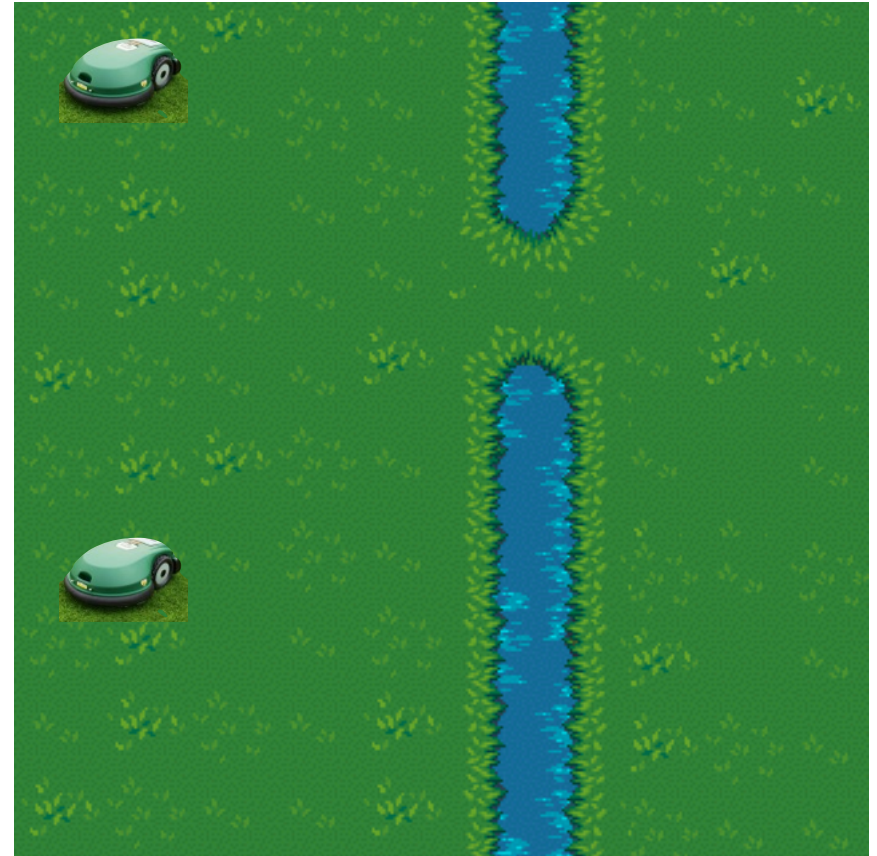


Distributed Planning

Distributed Planning for Distributed Plans

Hierarchical Behavior-Space Algorithm

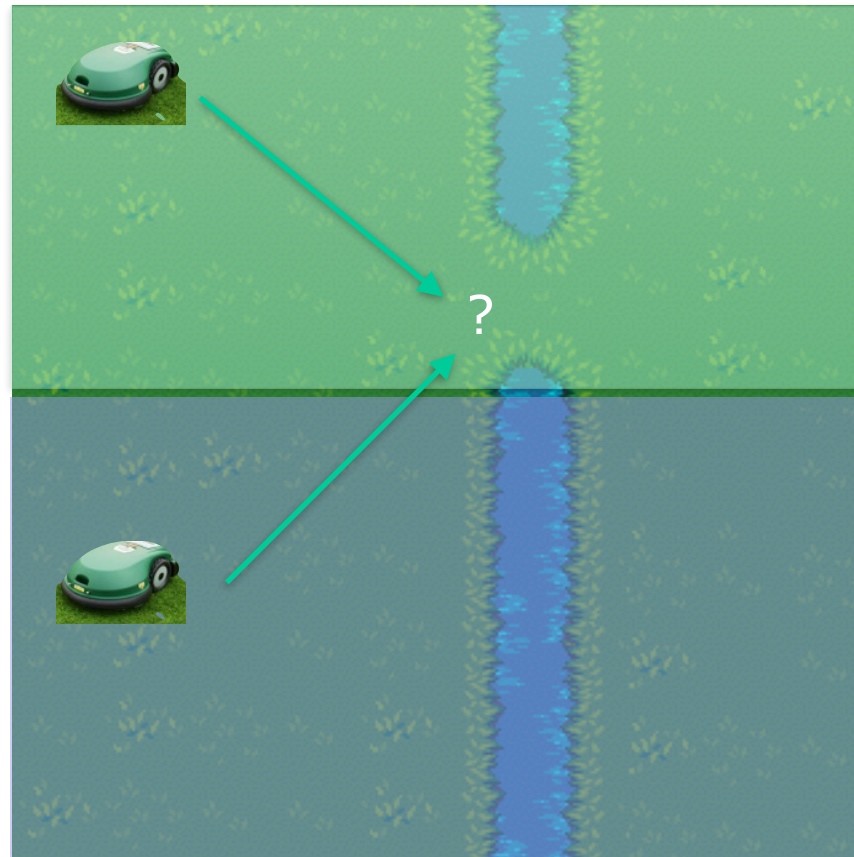
- agents represent local planned behaviors at multiple levels of abstraction
- agents decide if conflicts should be resolved at particular abstraction level or passed to more detailed levels



Distributed Planning

Distributed Planning for Distributed Plans

Hierarchical Behavior-Space Algorithm

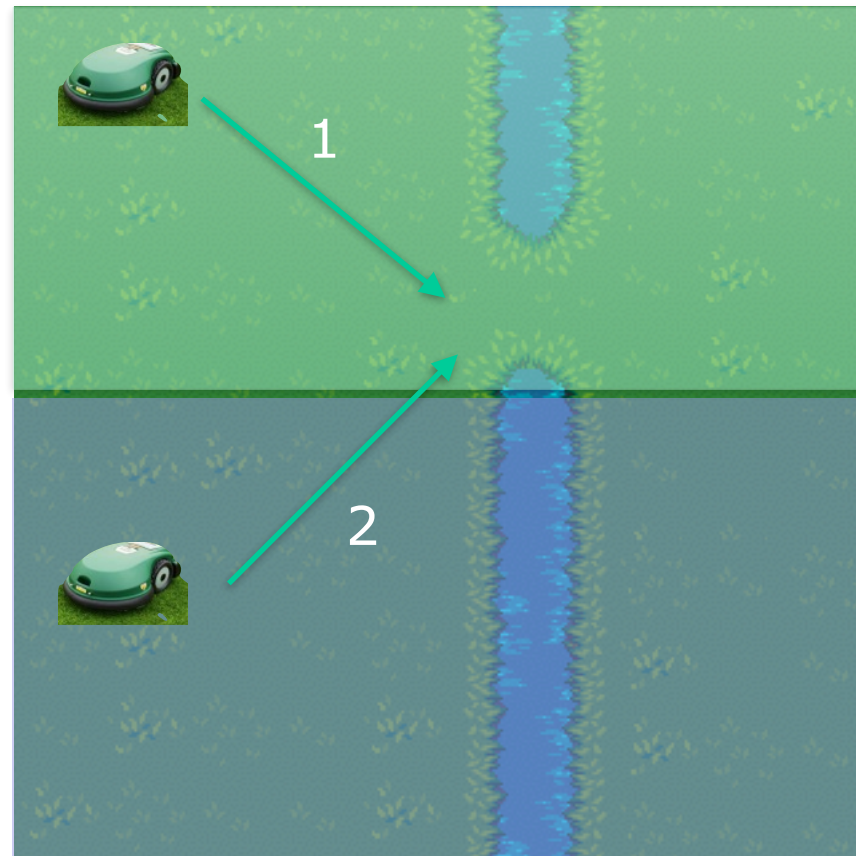


resolve by space (abstract level)

Distributed Planning

Distributed Planning for Distributed Plans

Hierarchical Behavior-Space Algorithm

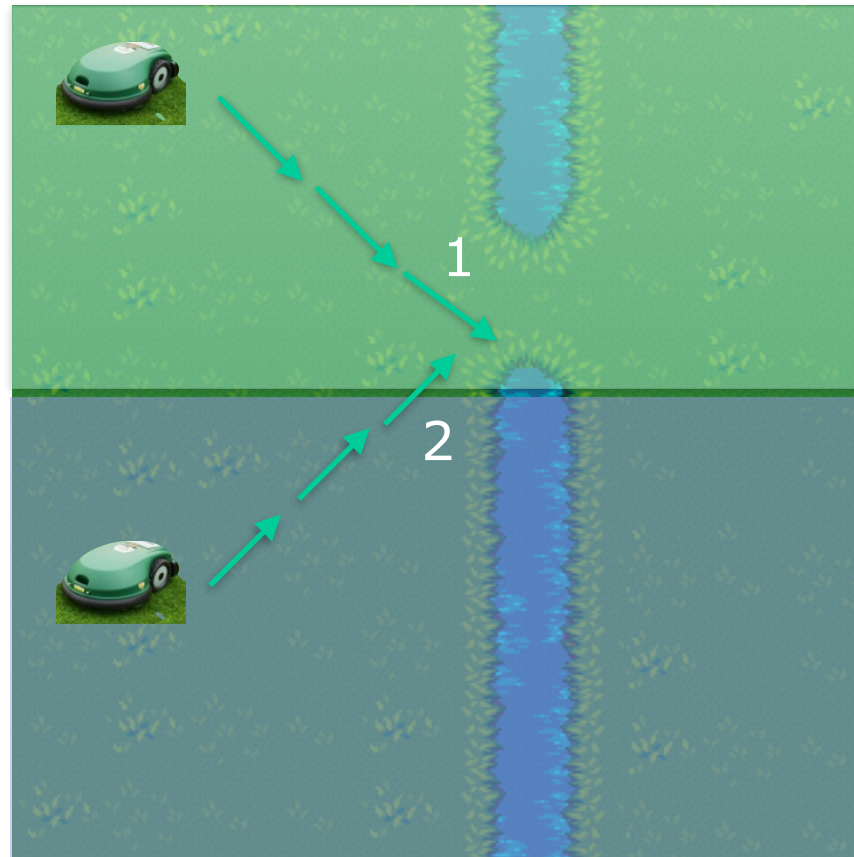


resolve by time (abstract level)

Distributed Planning

Distributed Planning for Distributed Plans

Hierarchical Behavior-Space Algorithm

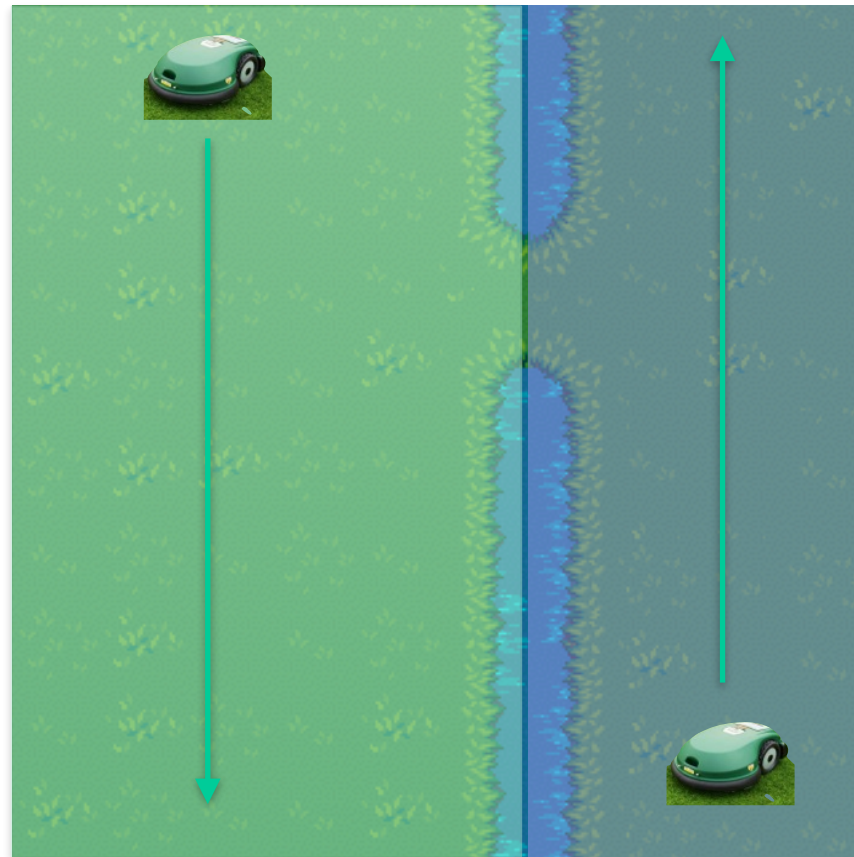


resolved by time (more detailed level)

Distributed Planning

Distributed Planning for Distributed Plans

Hierarchical Behavior-Space Algorithm



resolve by space (other solution)

Distributed Planning

Negotiation in Distributed Planning

Which agent should revise its plan?

- exchange description of options
- change plan when...
 - with most other options
 - having the least effort to change plan
 - plan change has least negative influence on other agents

Planning & Execution

Pre-Planing Coordination

- Avoid undesirable states of the world
 - backtrack from undesirable states
 - restrict actions which led to these states
 - Challenge: find restrictions without restricting too much
- prefer actions from which most agents benefit
 - even if not directly relevant to agents goal

Planning & Execution

Pre-Planing Coordination

Task Decomposition

- avoid overloading critical resources
- assign tasks to agents with matching capabilities
- agents with wide view should assign tasks
- assign overlapping responsibilities (coherence)
- assign highly interdependent tasks to agents in proximity

Planning & Execution

Post-Planing Coordination

What if agent fail to progress as expected?

- each agent has to formulate alternative plans
- each agent monitors its own plan execution
 - on fail, stop all agents progress
 - plan, coordinate & execute again
- try to address problems at local level (no coordination with others)
- re-assign tasks if necessary to complete urgent tasks

Summary

- Introduction
- Task Sharing
 - Homogeneous Systems
 - Heterogeneous Systems
 - Task Assignment & Addressing
- Result Sharing
 - Functional Accurate Cooperation
 - Shared Repository
 - Communication Strategies

Summary

- Distributed Planning
 - Centralized Planning for Distributed Plans
 - Distributed Planning for Centralized Plans
 - Distributed Planning for Distributed Plans
 - Plan Merging
 - Iterative Plan Formation
 - Hierarchical Behavior-Space Algorithm
- Planning & Execution
 - Pre-/Post-Planning