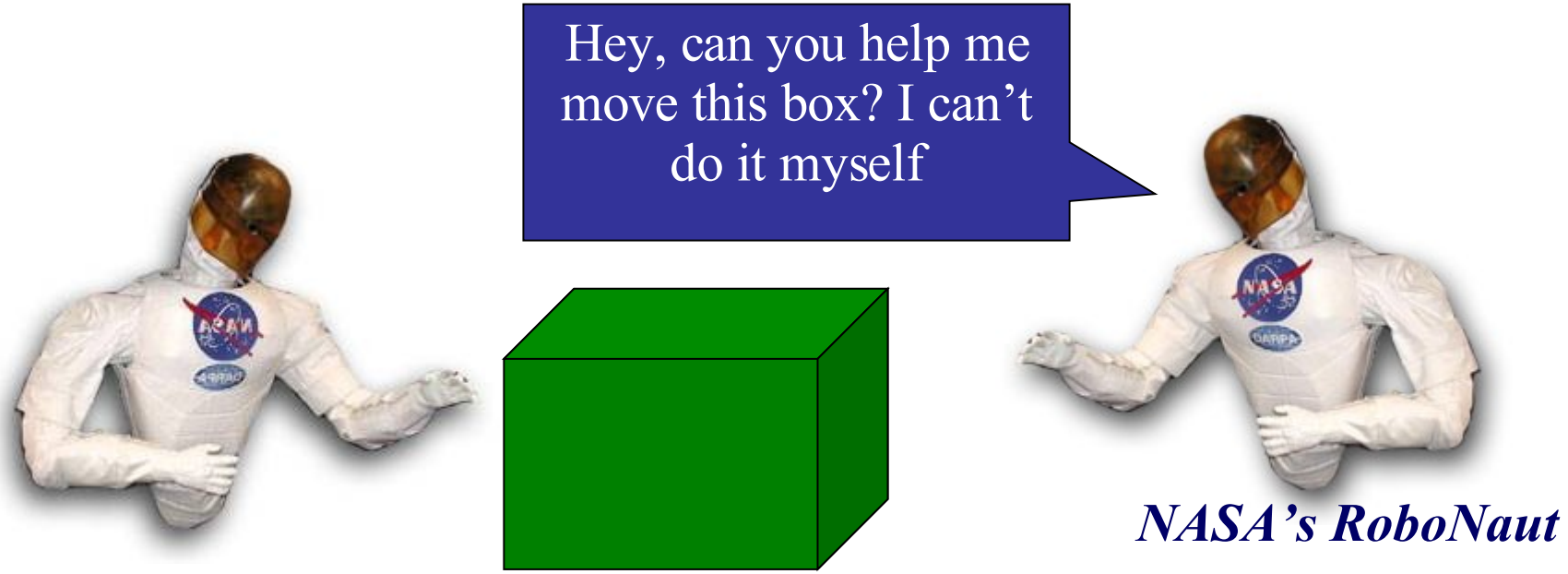
Multi-Robot Coordination
15-491 CMRoboBits

Manuela Veloso
And
Brett Browning

Fall 2007

Coordination

- The problem:
  - How do we make two or more robots work together to execute a task?
- We want to execute the task better than if robots do not coordinate

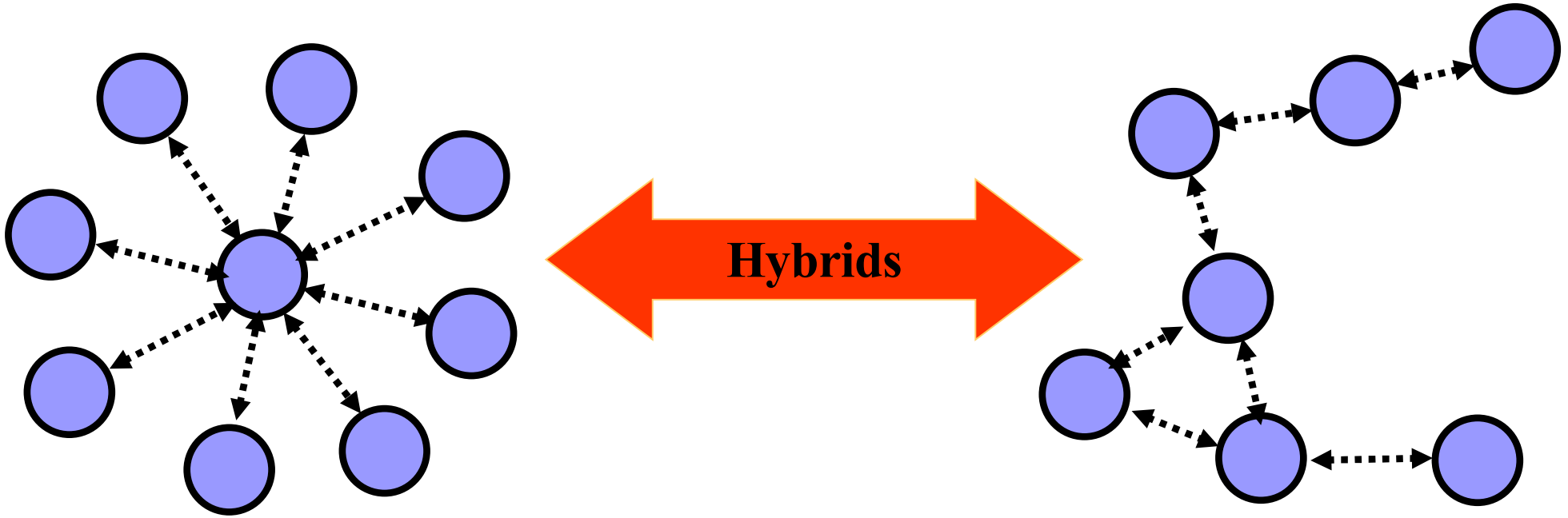


NASA's RoboNaut

Coordination Approaches

Fully Centralized

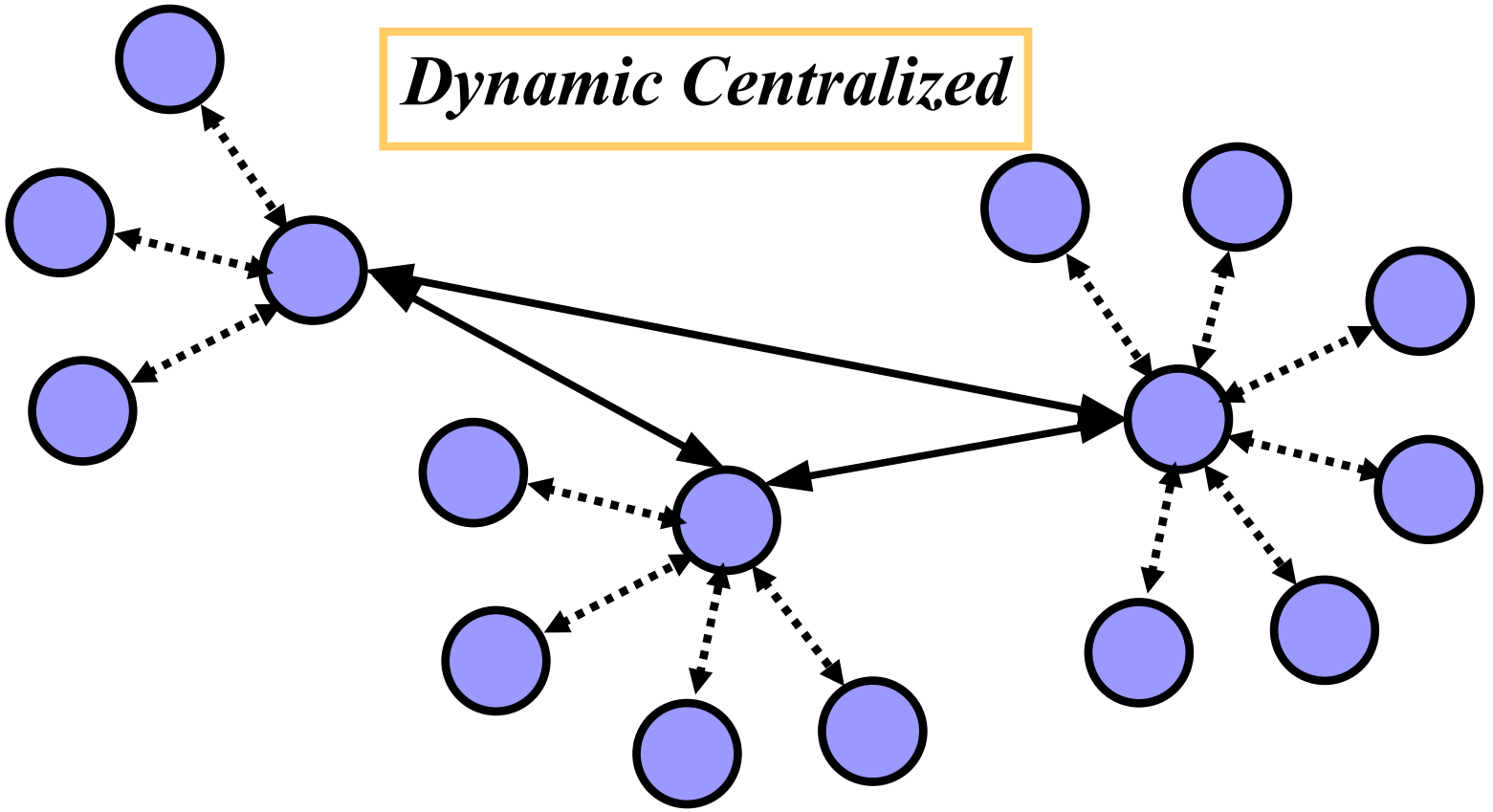
Fully Distributed



- optimal
- intractable
- brittle
- sluggish
- communication heavy

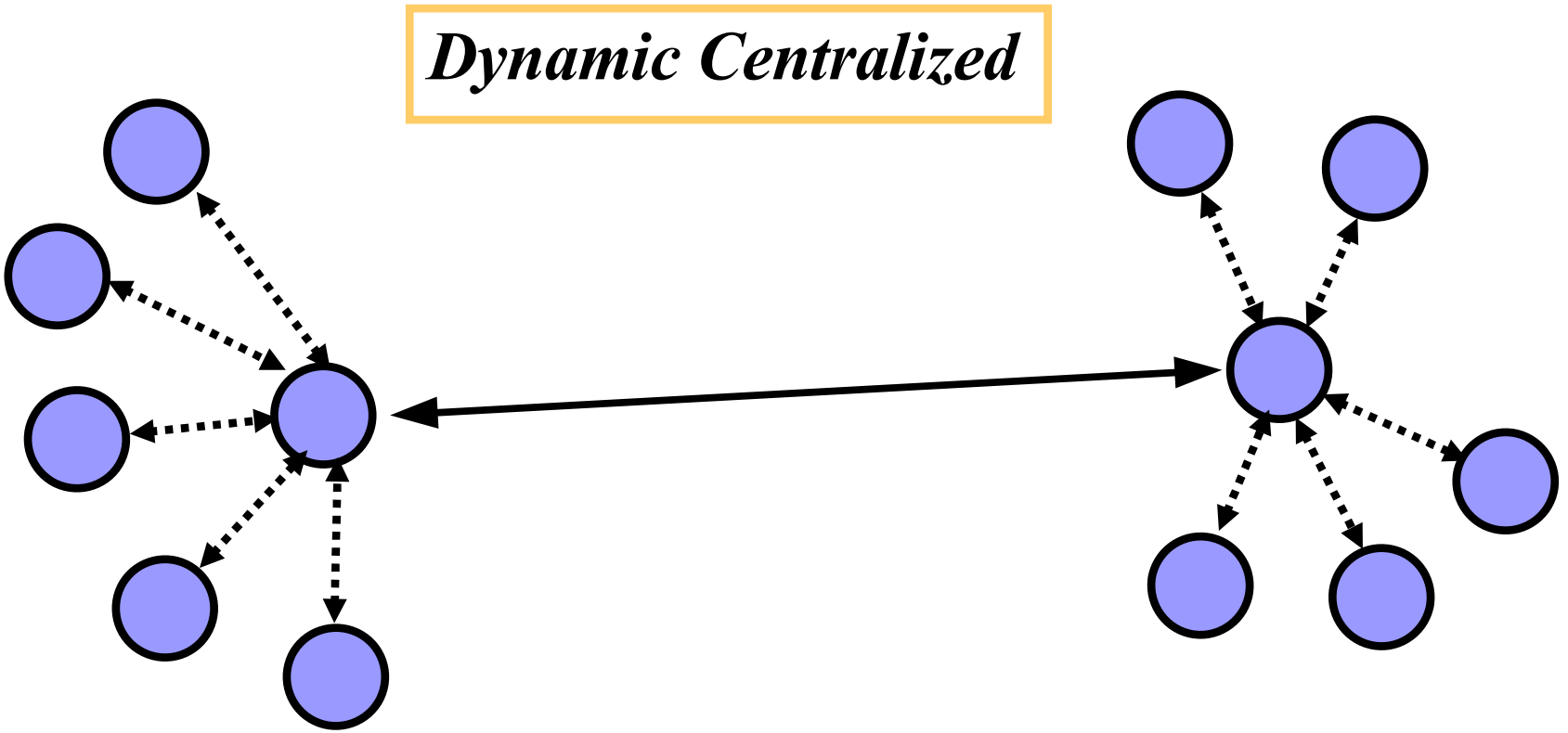
- suboptimal
- tractable
- robust
- nimble
- communication light

Hybrid Approaches



- **suboptimal (but better than distributed)**
- **tractable**
- **robust**
- **can be nimble**
- **communication medium**

Hybrid Approaches



- **suboptimal (but better than distributed)**
- **tractable**
- **robust**
- **can be nimble**
- **communication medium**

Today's Topics

- Role assignment
- Distributed Plays
- Combining distributed plays with role assignment

Role Assignment

- Fundamental problem
- Key idea:
  - Given N robots
  - Given M tasks
  - How to assign the M tasks to the N robots?

Utility Matrix Formulation

- Operations Research
- Uij: *utility* of assigning task j to robot i
- Utility is an objective function

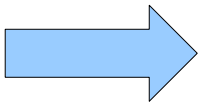
$$U = \begin{pmatrix} U_{1,1} & U_{1,2} & \dots & U_{1,M} \\ \dots & \dots & \dots & \dots \\ U_{N,1} & U_{N,2} & \dots & U_{N,M} \end{pmatrix}$$

Utility Matrix

$$A = \begin{pmatrix} 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & 1 \\ 1 & 0 & \dots & 0 \end{pmatrix}$$

Assignment Matrix

Maximize



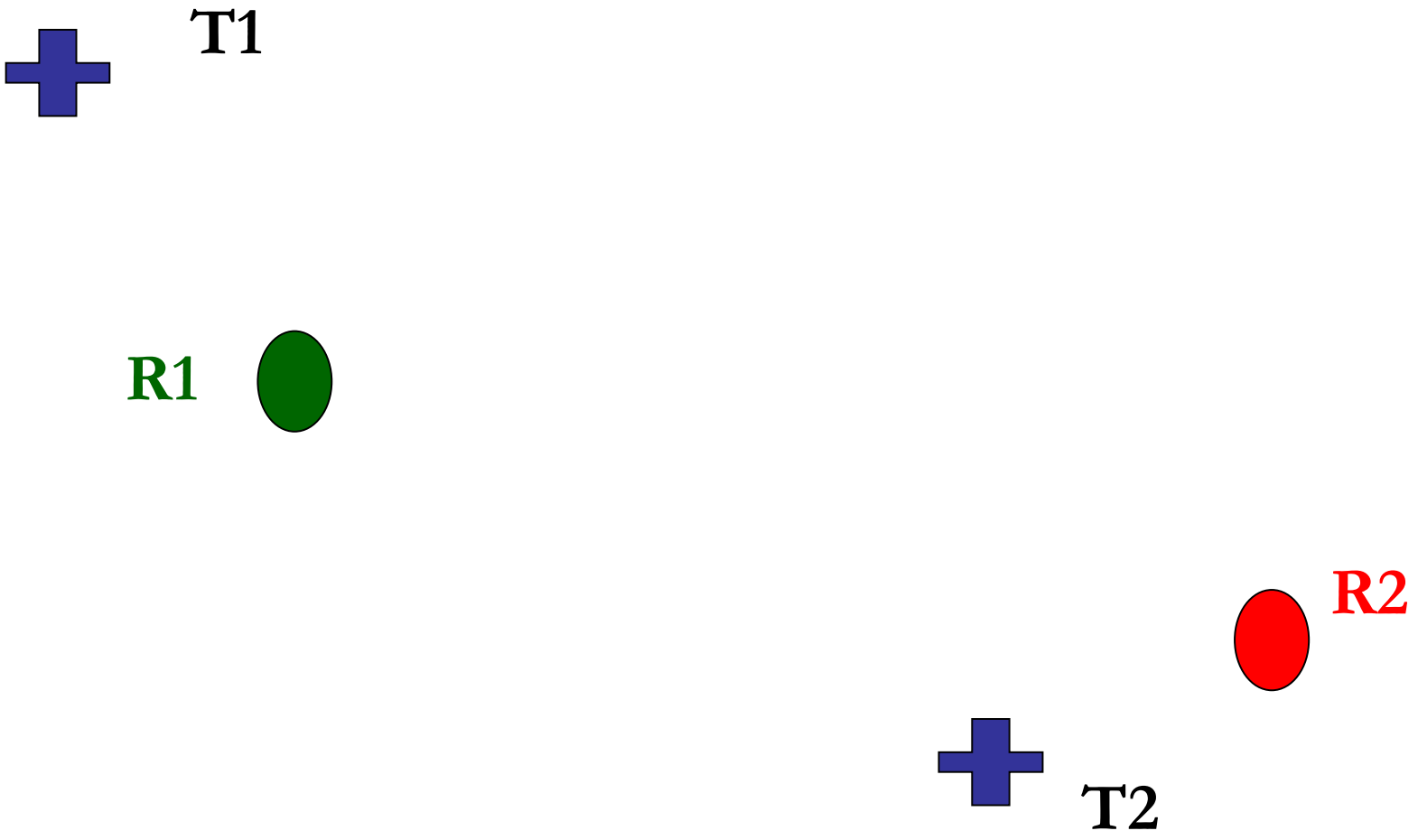
$$V = \sum \sum A_{i,j} U_{i,j} = \sum_{A_{i,j}=1} U_{i,j}$$

Maximizing Assigned Utility

- Key assumption
  - Maximizing individual utility maximizes overall utility (generally true, but not always)
- Optimal solutions exist
  - e.g. Hungarian Algorithm
- ...but, optimal solutions are worst case exponential
- Optimal solutions also assume a centralized model => we know everything!

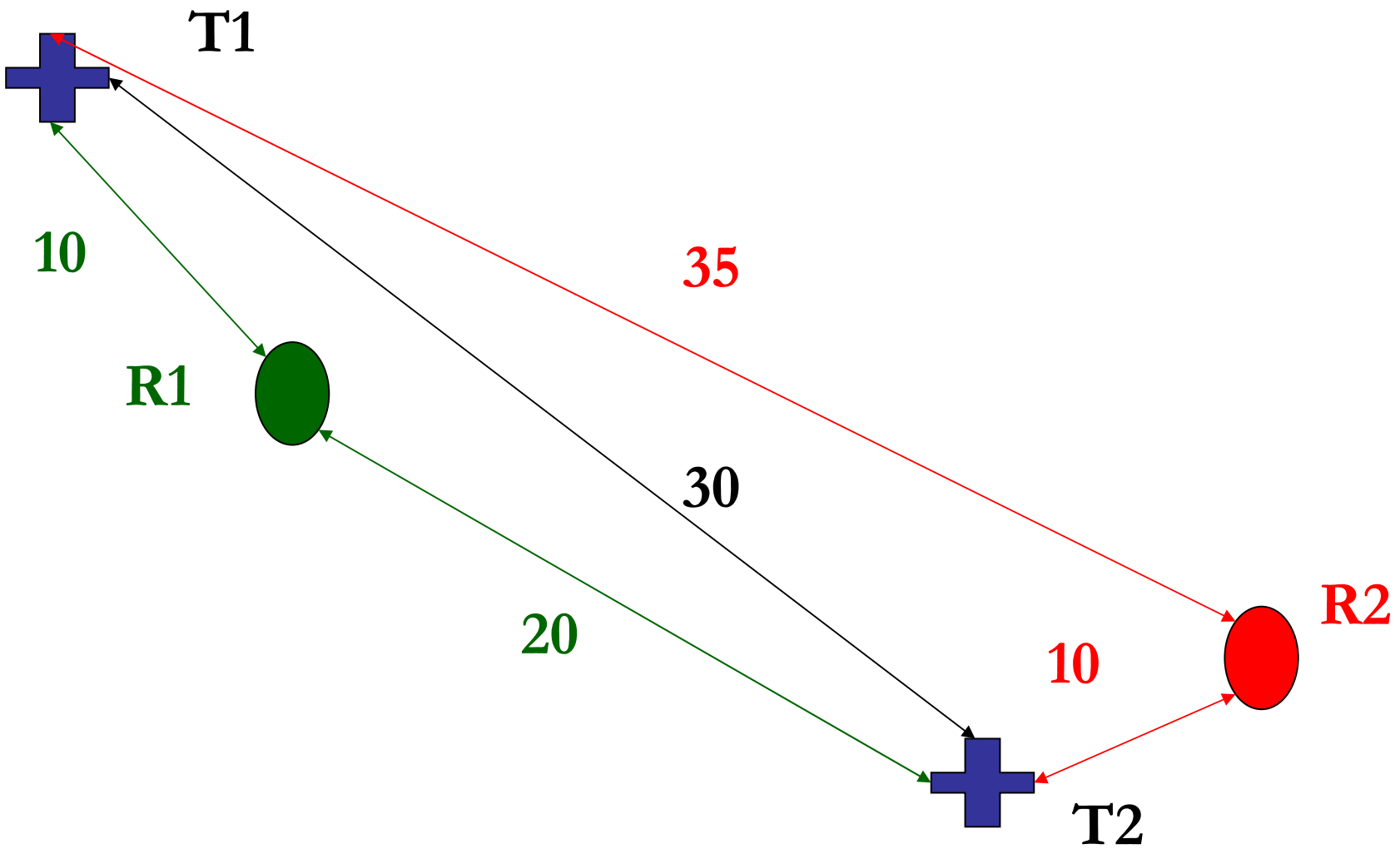
Common Scenario

2 robots and 2 tasks



Common Scenario

2 robots and 2 tasks

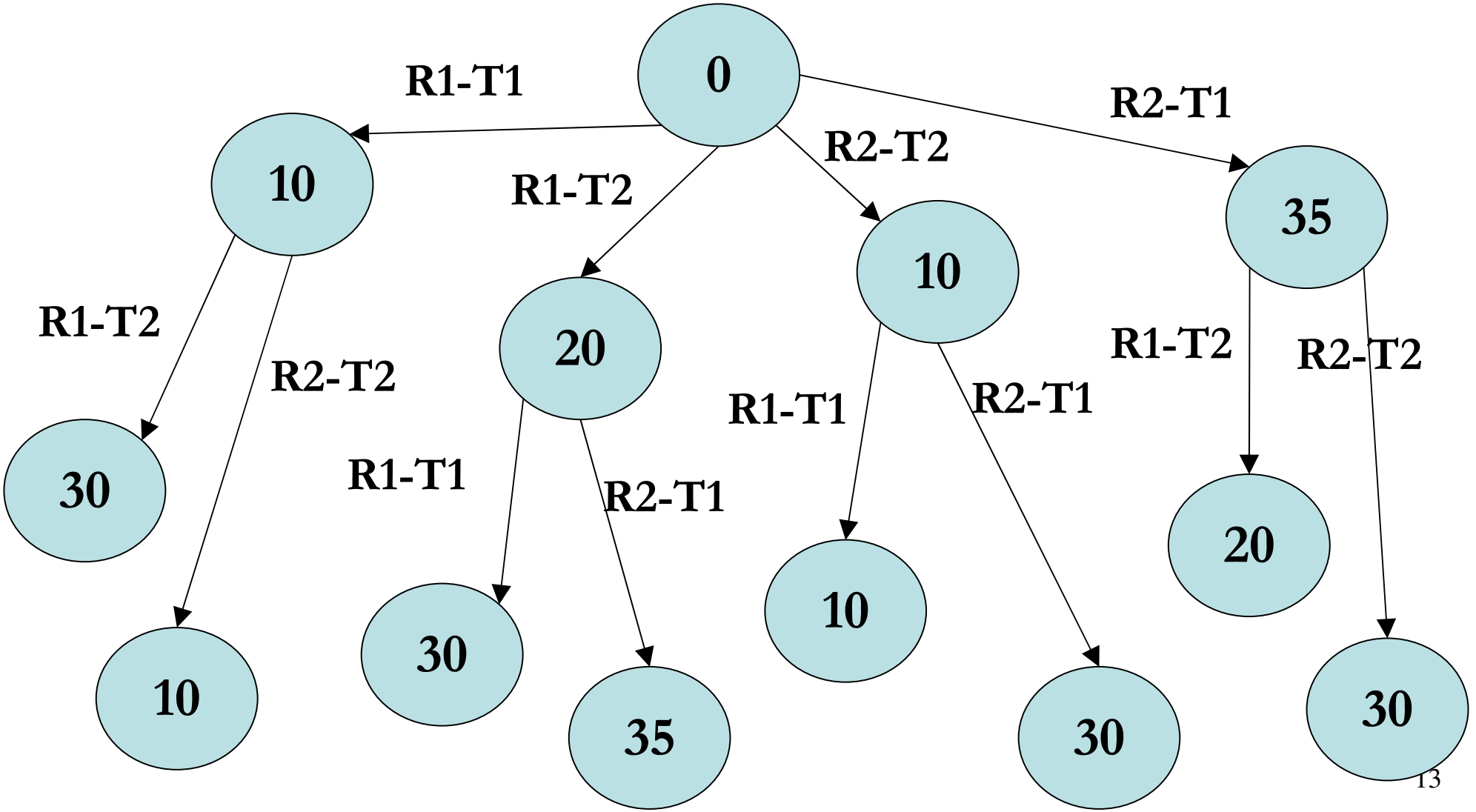


Example: Fully Centralized

- One robot in the team is the leader
- Leader produces optimal task allocation
- Centralized planner uses an exhaustive Depth First Search with some pruning

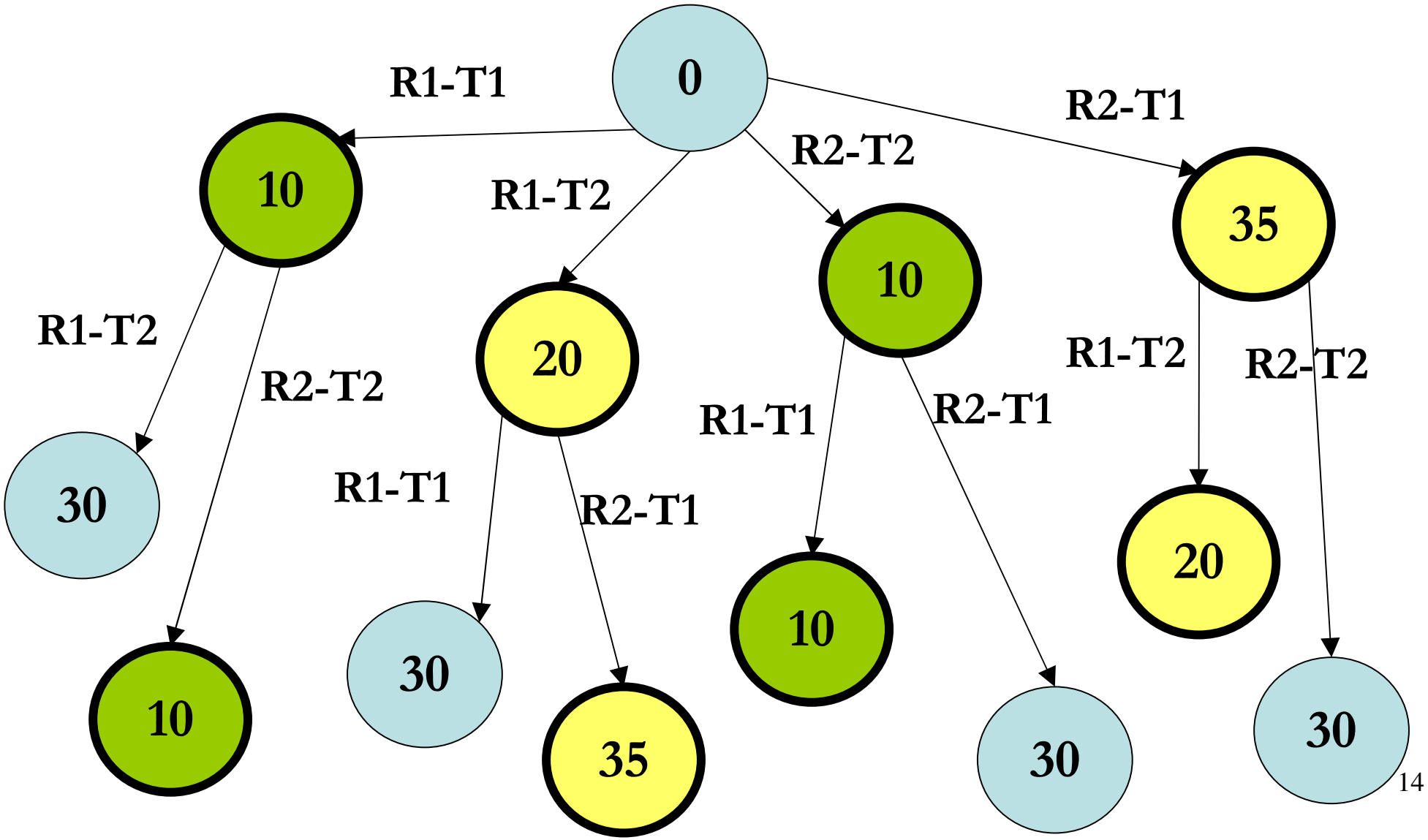
Fully Centralized Approach

2 robots and 2 tasks



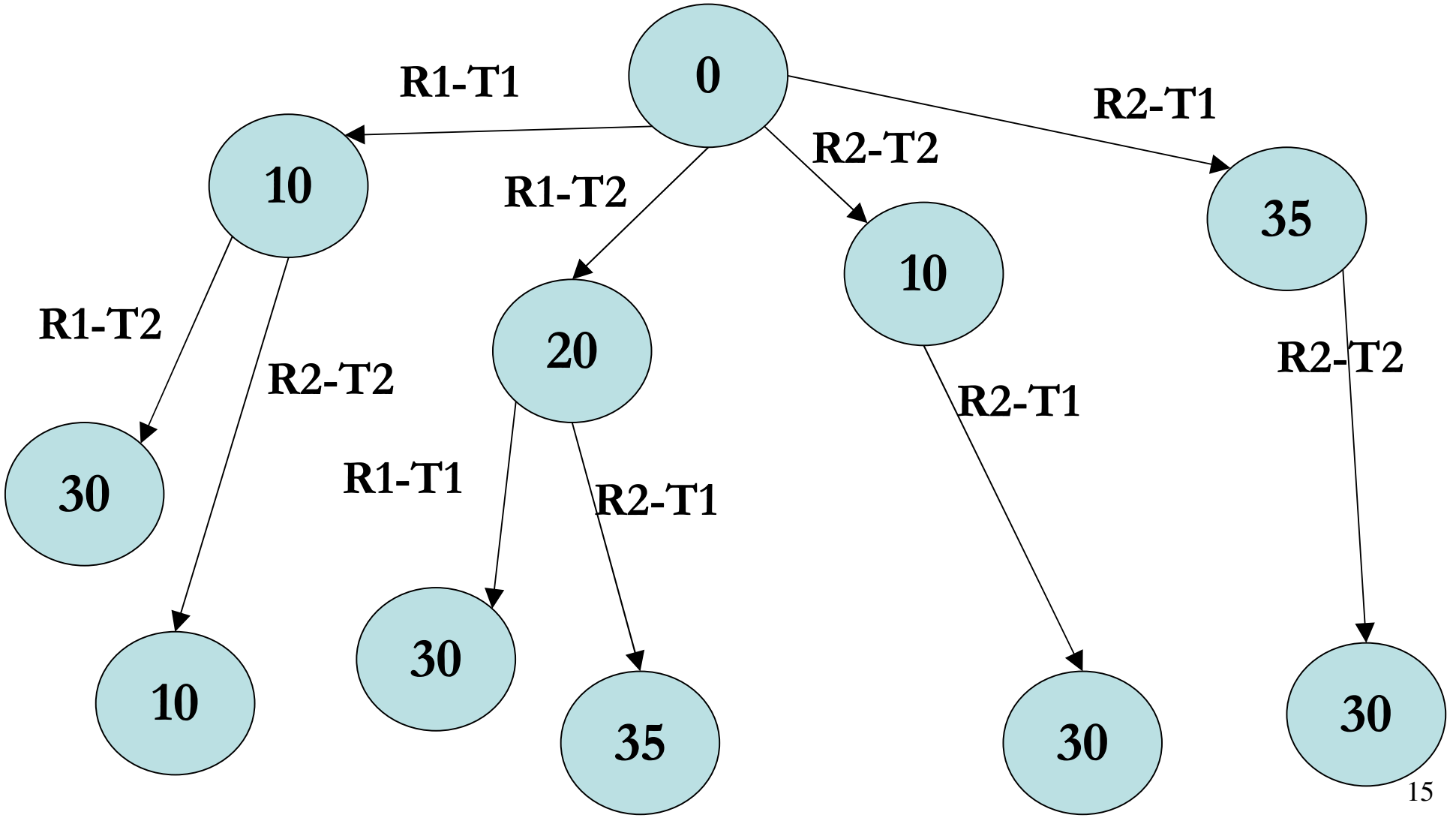
Fully Centralized Approach

2 robots and 2 tasks



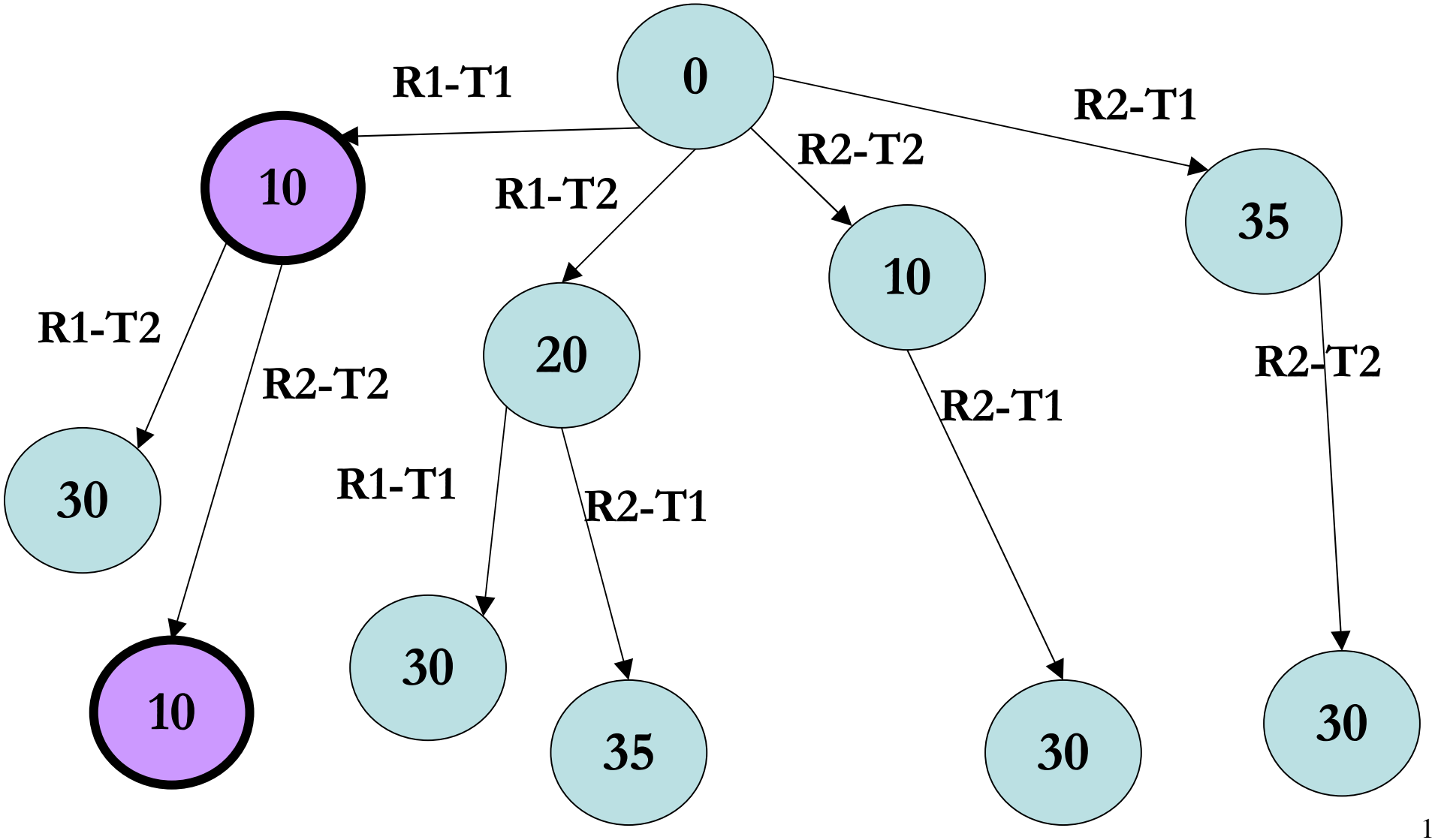
Fully Centralized Approach

2 robots and 2 tasks



Fully Centralized Approach

2 robots and 2 tasks

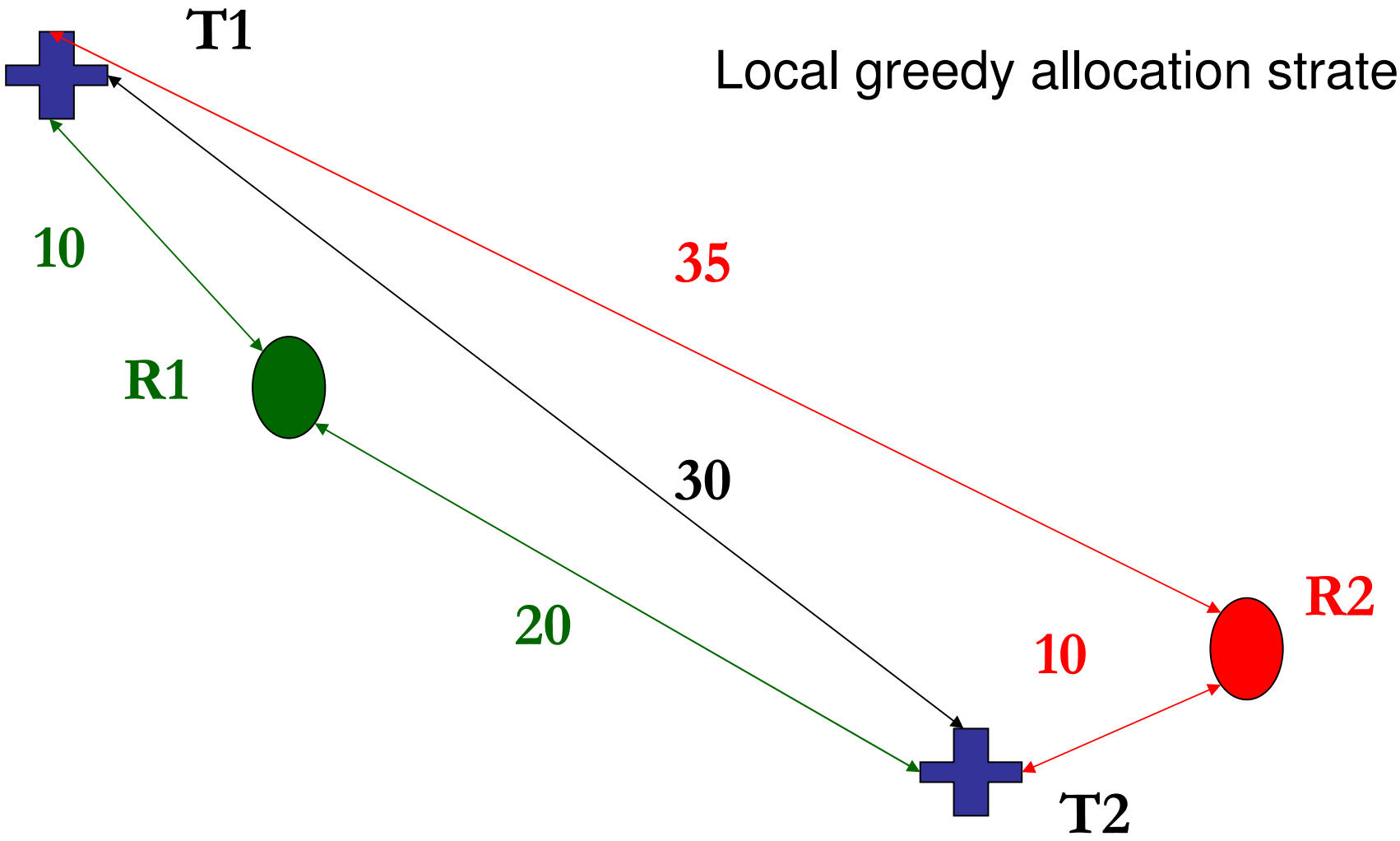


Example: Fully Distributed

- Each robot plans and executes actions independently
- All robots plan for ALL INCOMPLETE tasks (robust)
- All robots announce task completions
- Robots use same Depth First Search as in Centralized approach to optimize schedules

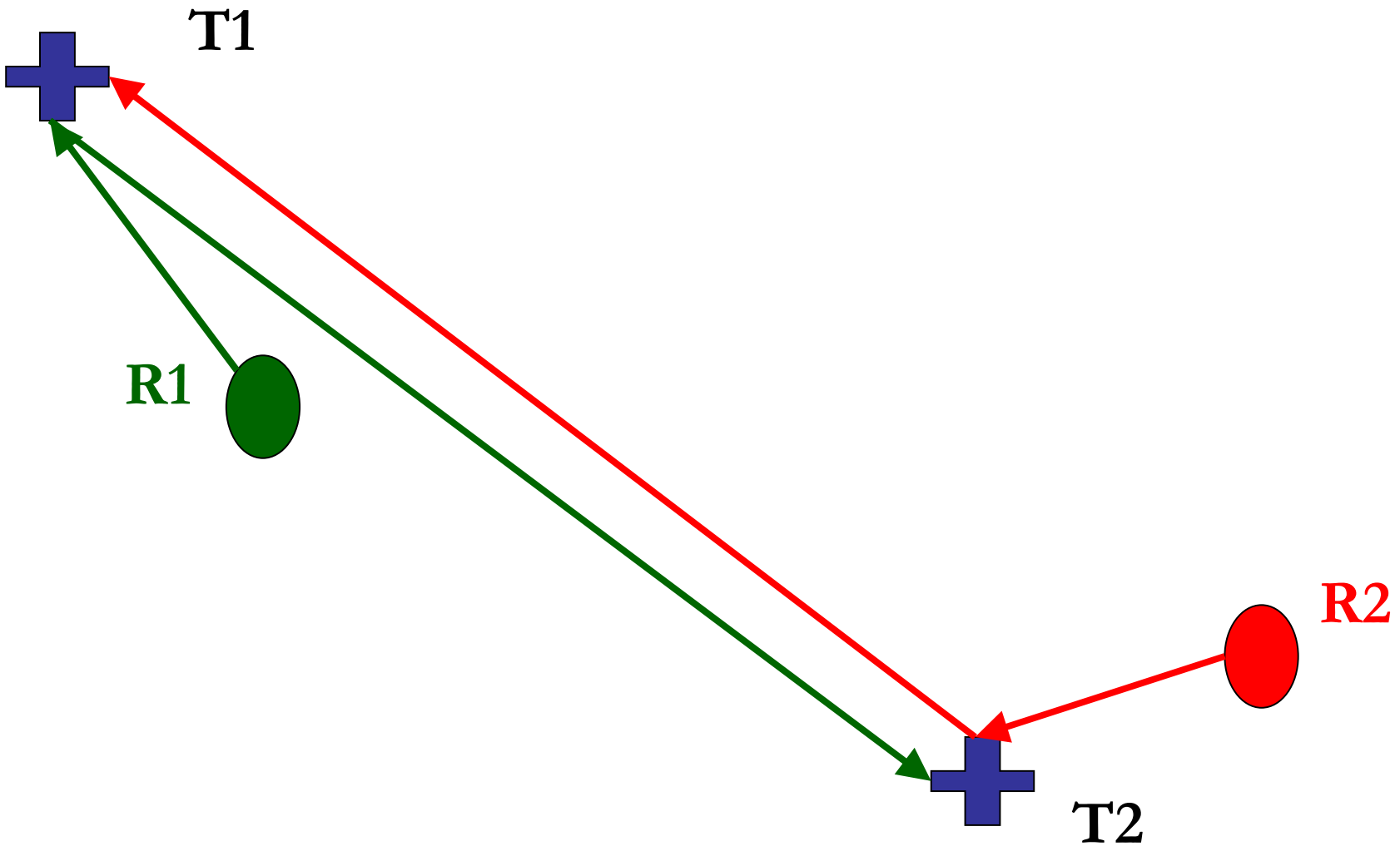
Fully Distributed Approach

2 robots and 2 tasks



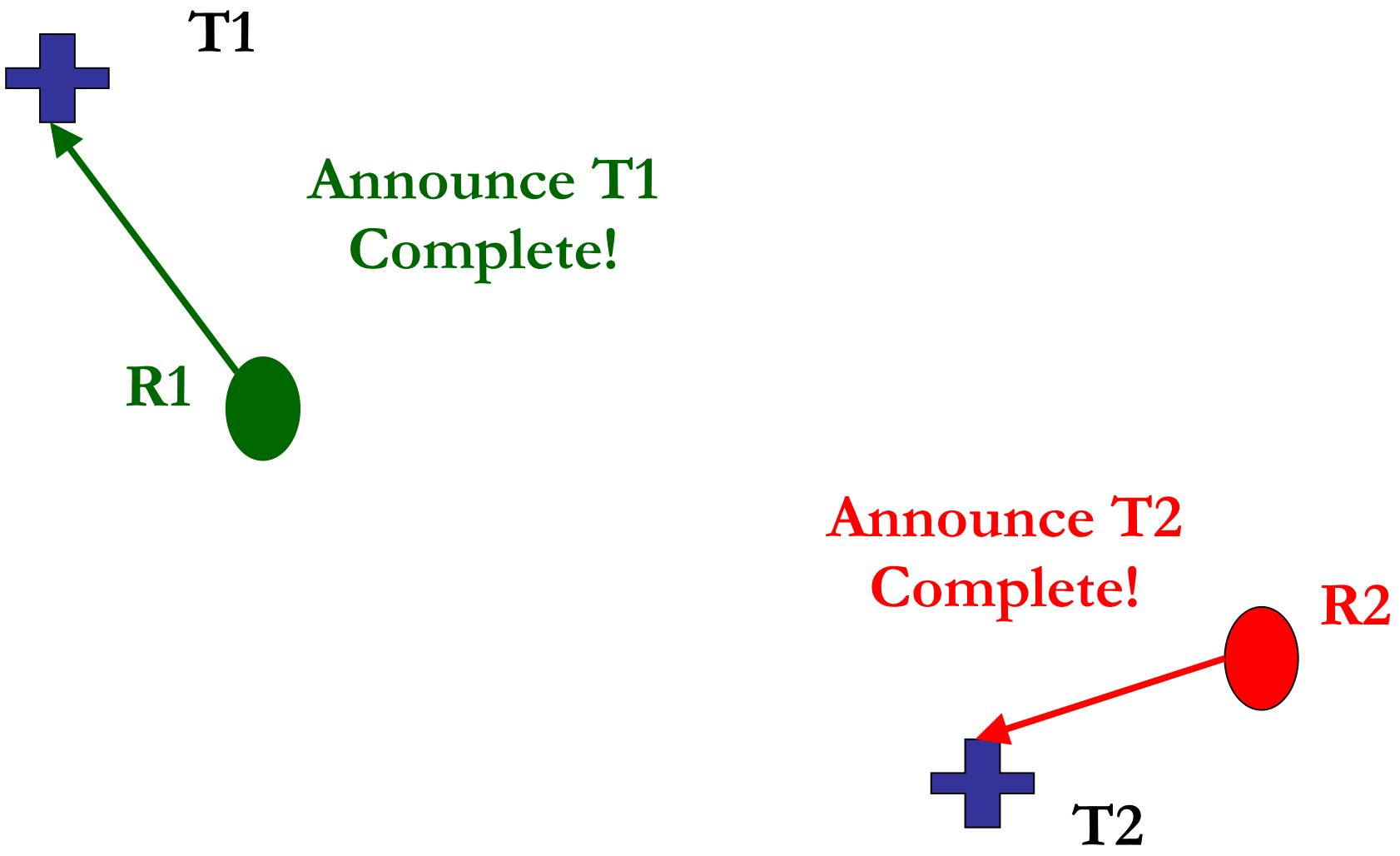
Fully Distributed Approach

2 robots and 2 tasks



Fully Distributed Approach

2 robots and 2 tasks

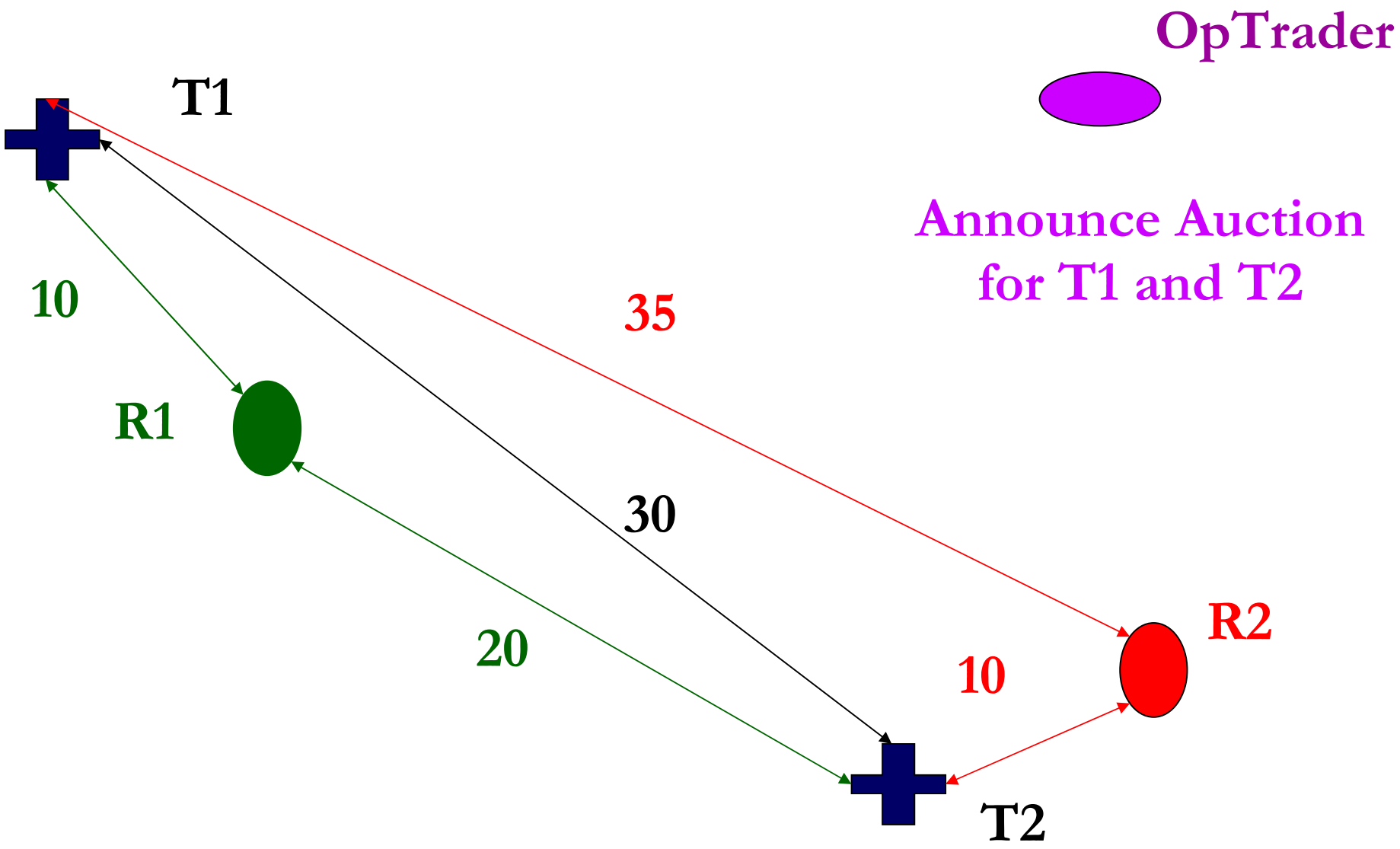


Example: Hybrid Market Approach

- Market based approaches have provided a good compromise
  - Dynamic centralized approach
- Key idea
  - OpTrader agent trades on behalf of operator
  - Robots estimate costs and make bids on tasks
  - OpTrader does initial assignment based on bids (greedy algorithm or otherwise)
  - Robots execute assignments and can re-trade as they execute
- Many key contributors: see reading for details

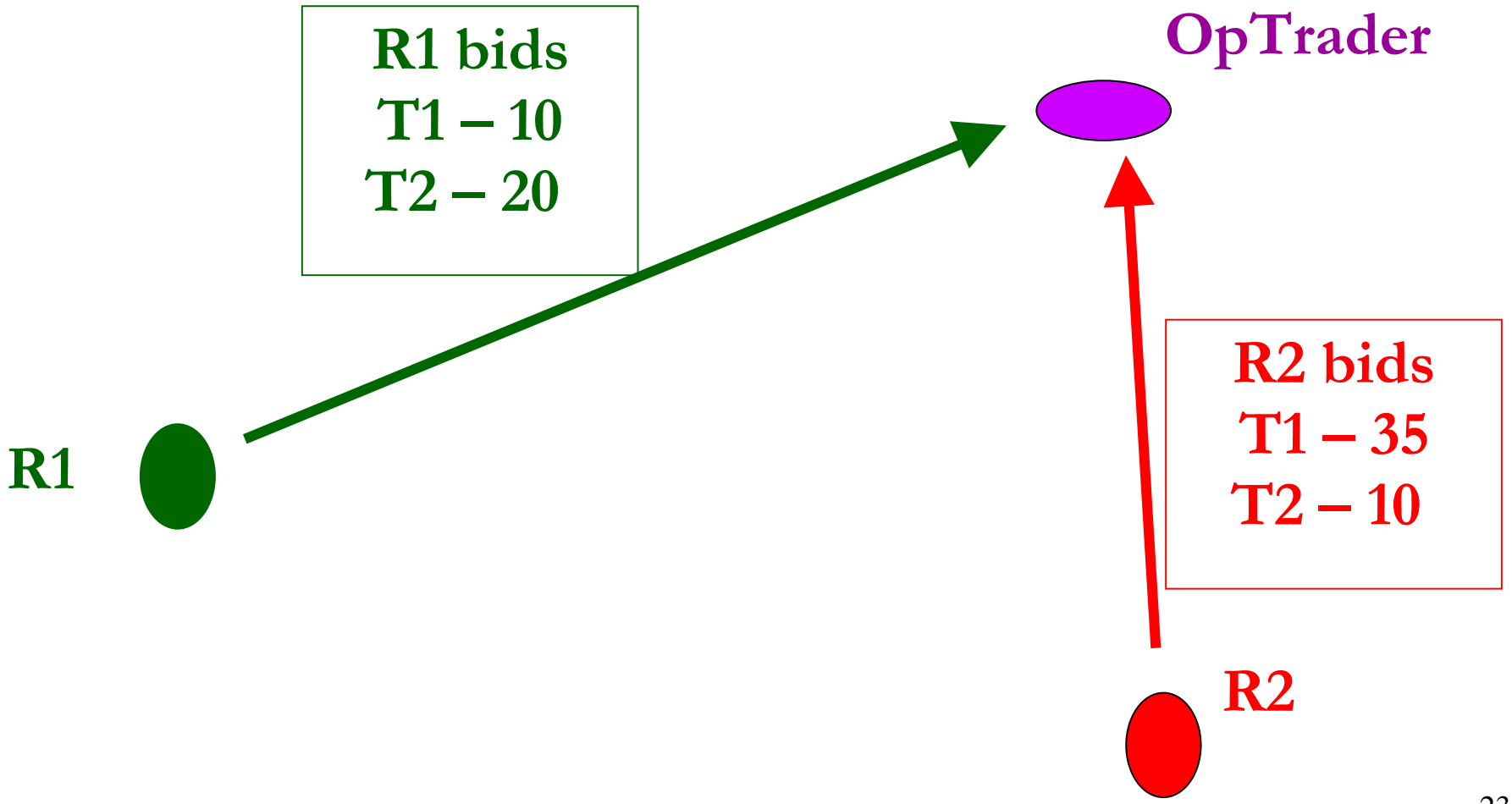
Market Approach

2 robots and 2 tasks



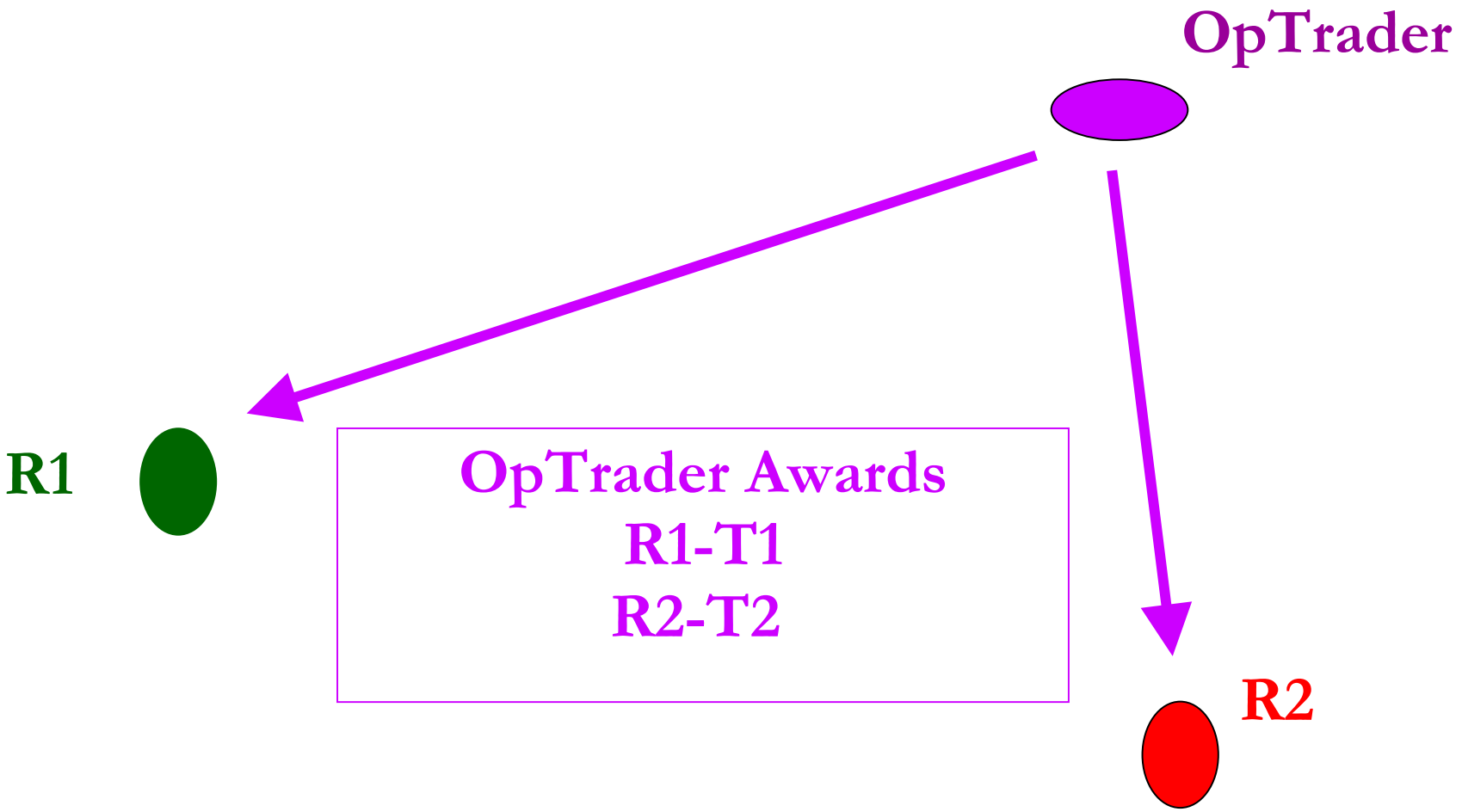
Market Approach

2 robots and 2 tasks



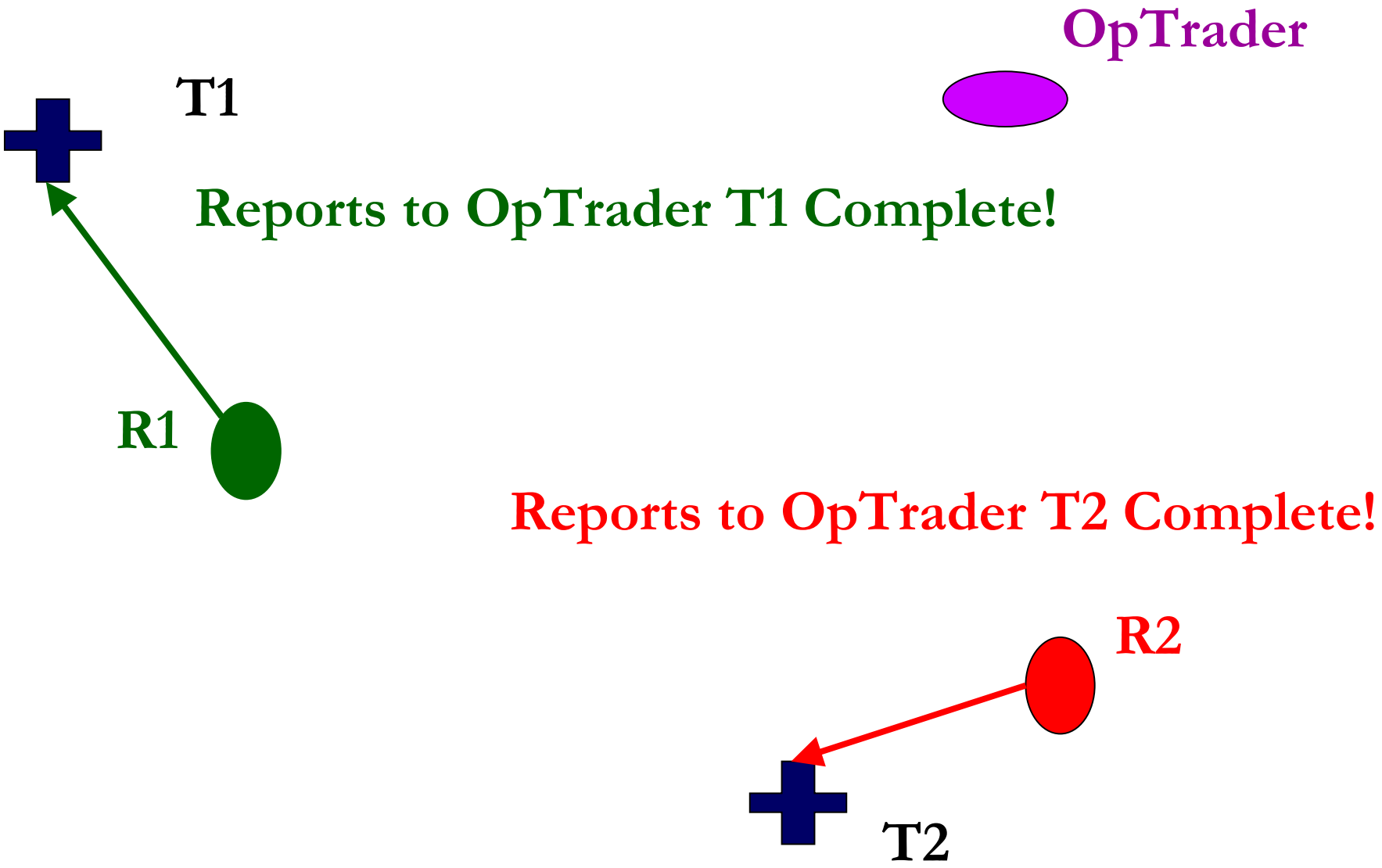
Market Approach

2 robots and 2 tasks

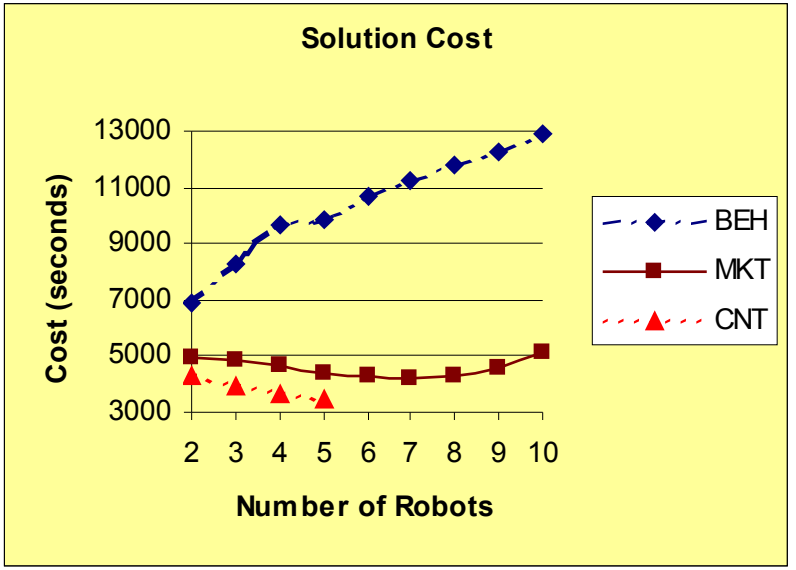


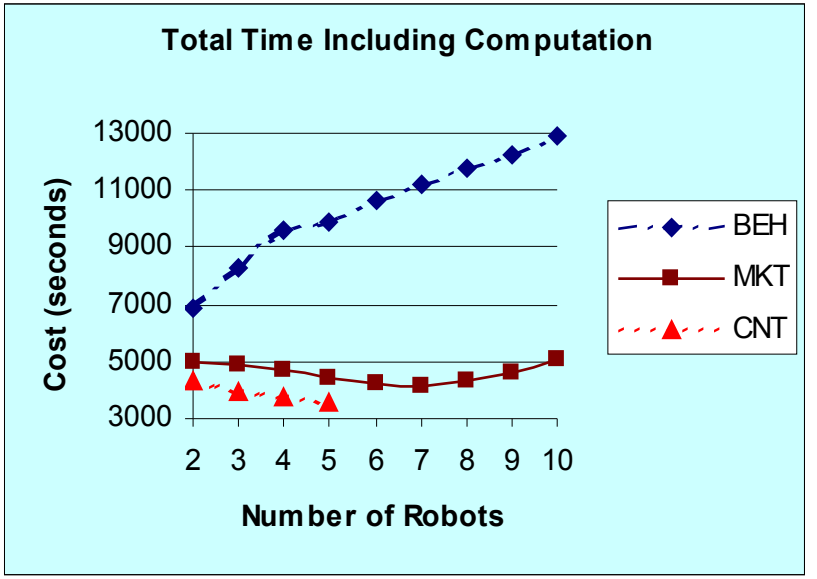
Market Approach

2 robots and 2 tasks

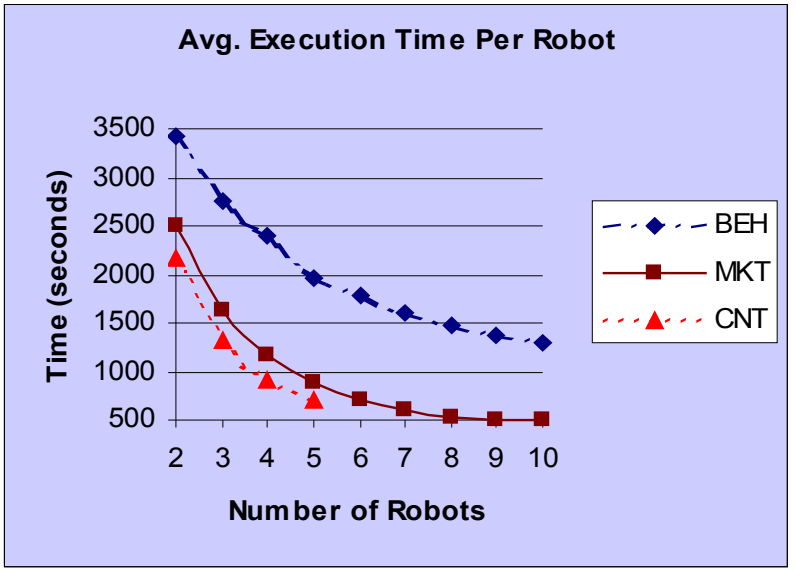


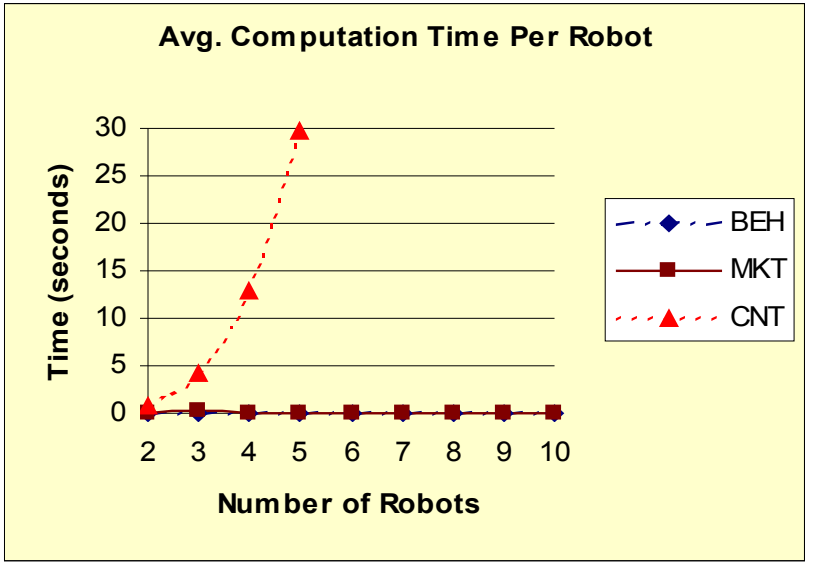
Comparison of Approaches





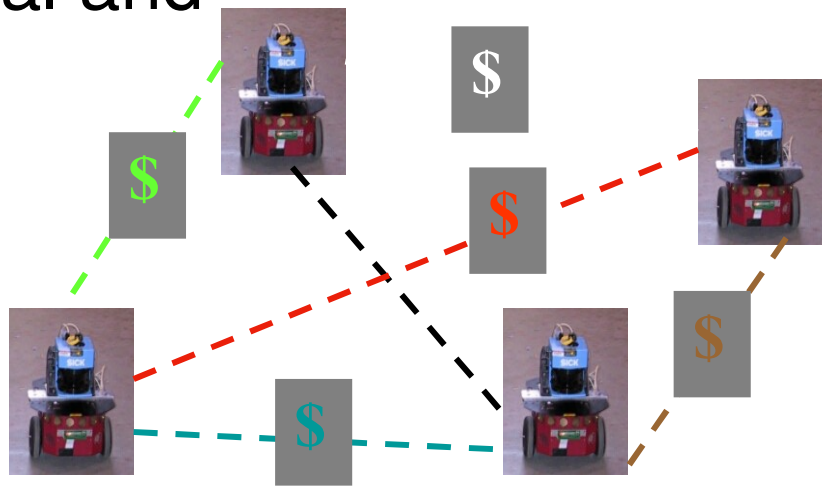
Ten Tasks





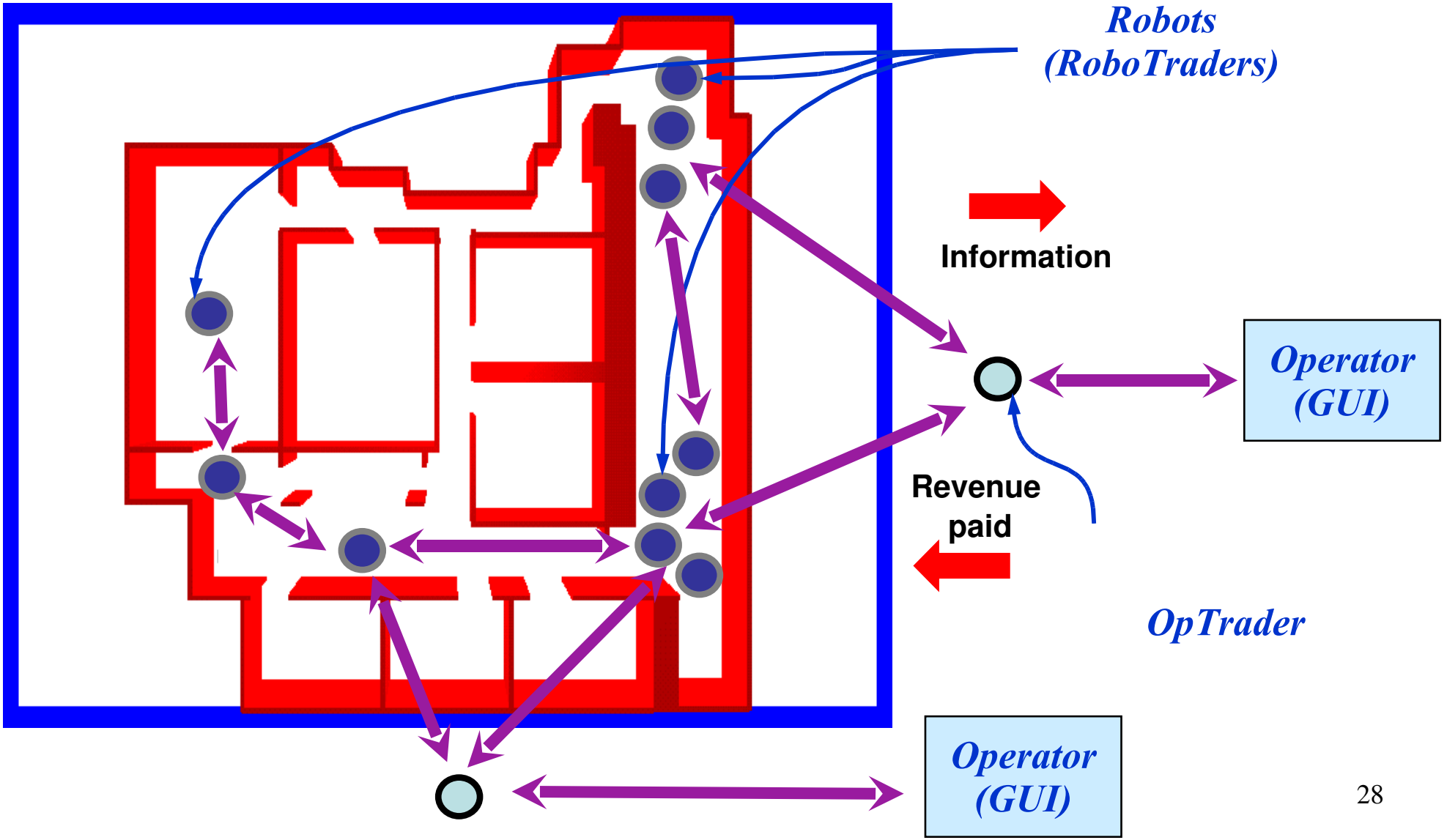
TraderBots

- Robots are organized as an economy
- Team mission is to maximize production and minimize costs
- Robots exchange money for tasks to maximize individual profit
- System is designed to align local and global profit maximization

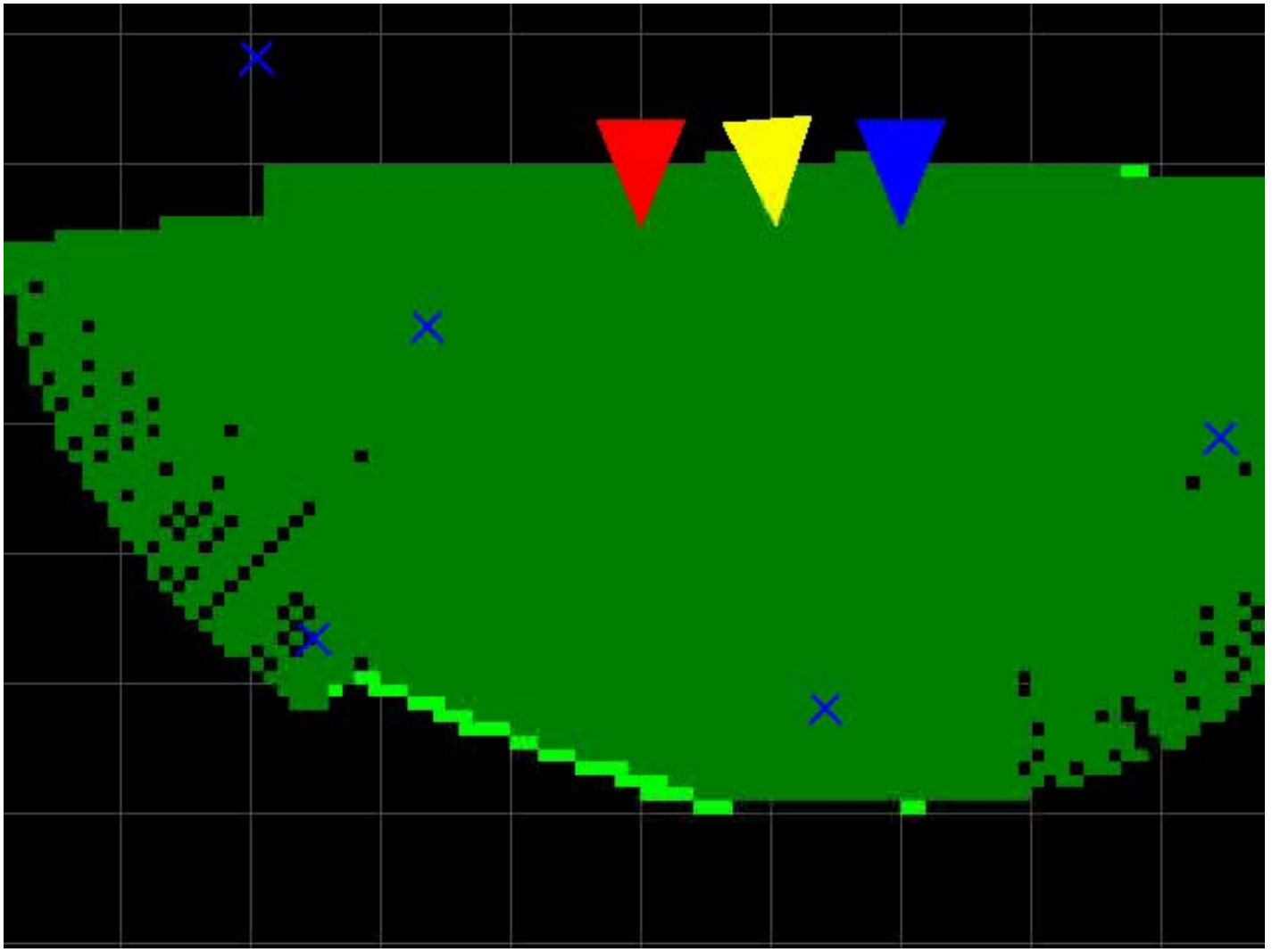


[Dias et al., 07][Stentz et al. 98]

Trader Interaction



Overall Performance

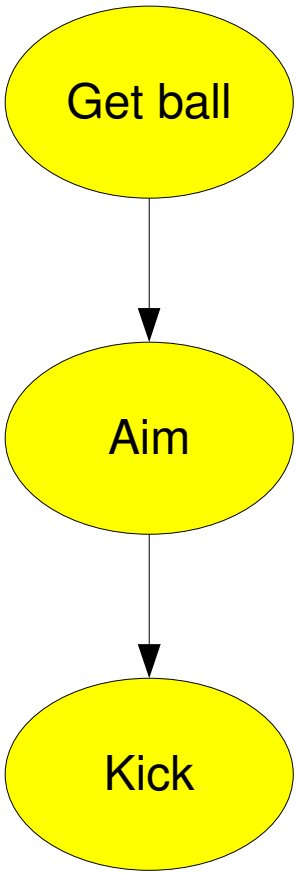


Complex Tasks?

- Complex tasks can require robots to coordinate *while* executing the task
  - Task allocation does not solve the problem!
- How do we coordinate robot execution?
  - e.g. carrying a ladder, passing in soccer
- We saw last time that plays (team plans) provide a way to coordinate actions
- How do we incorporate plays into a hybrid framework?
  - We can exploit the dynamic centralized approach

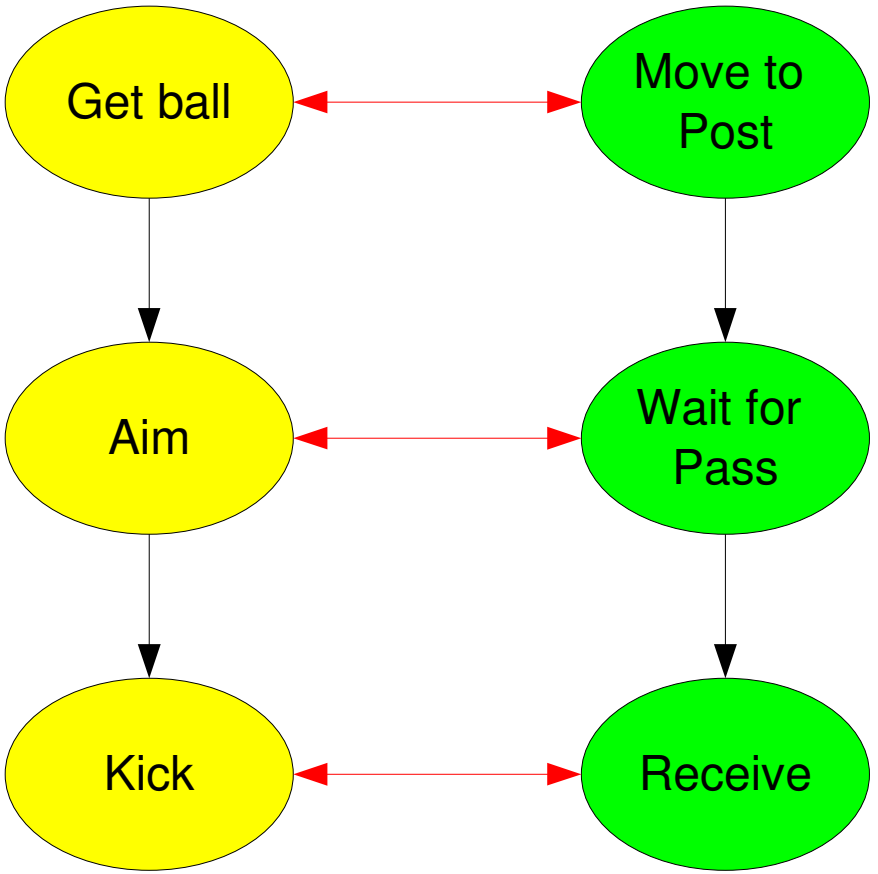
Plays Recap

- For a single role, we can write a *program* for executing a task



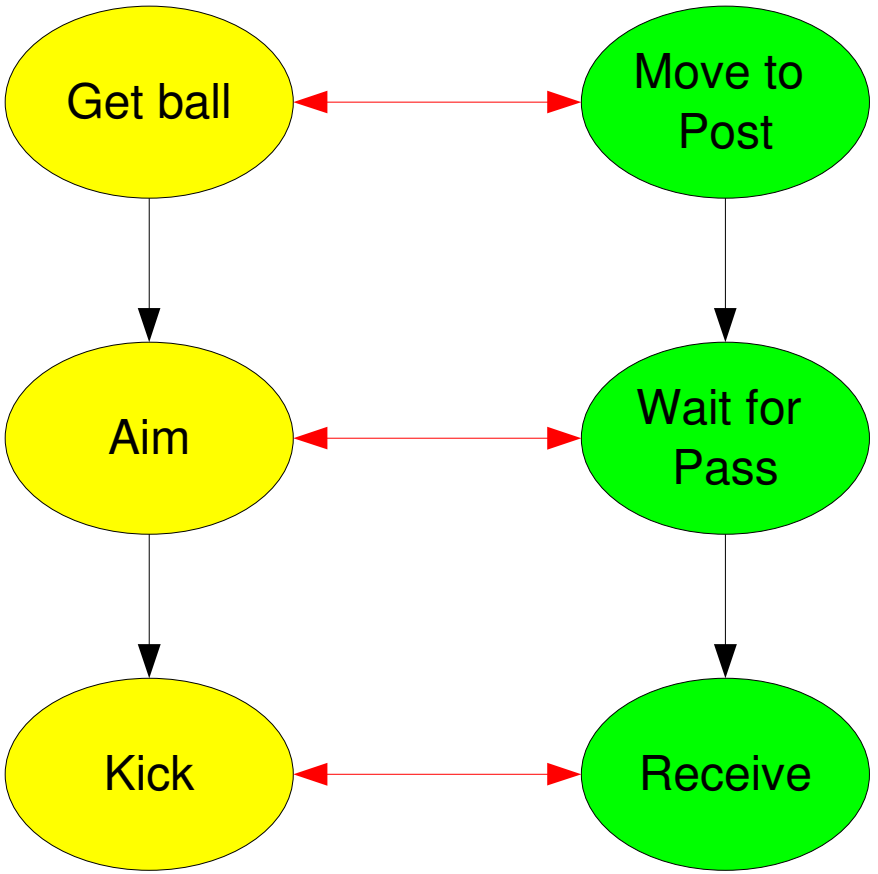
Plays Recap

- For a multiple roles, we can *couple* the execution of the programs



Plays Recap

- Applicability: Conditional execution of a play for a given world state

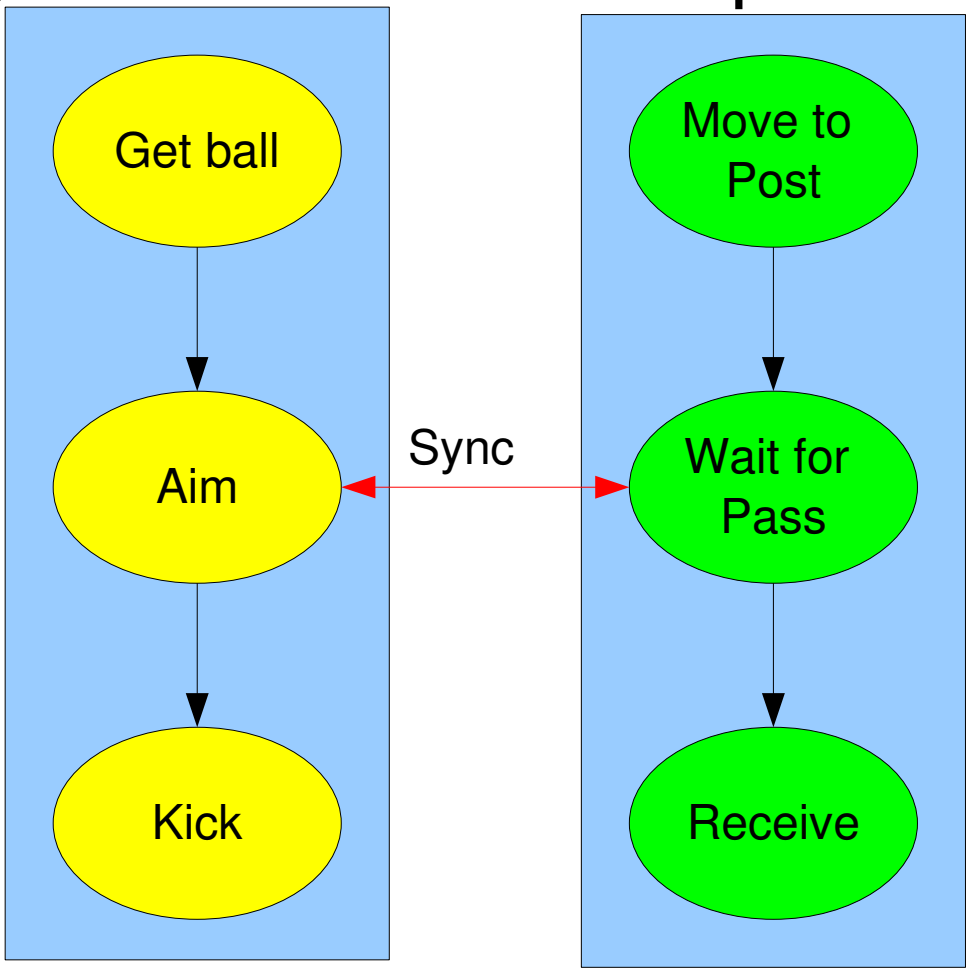


Distributed Plays

- Key ideas
  - Role programs execute on assigned robot
  - Only minimal coupling is required
  - Play execution needs to be monitored
- Key challenges
  - How to assign roles?
  - How to select plays?

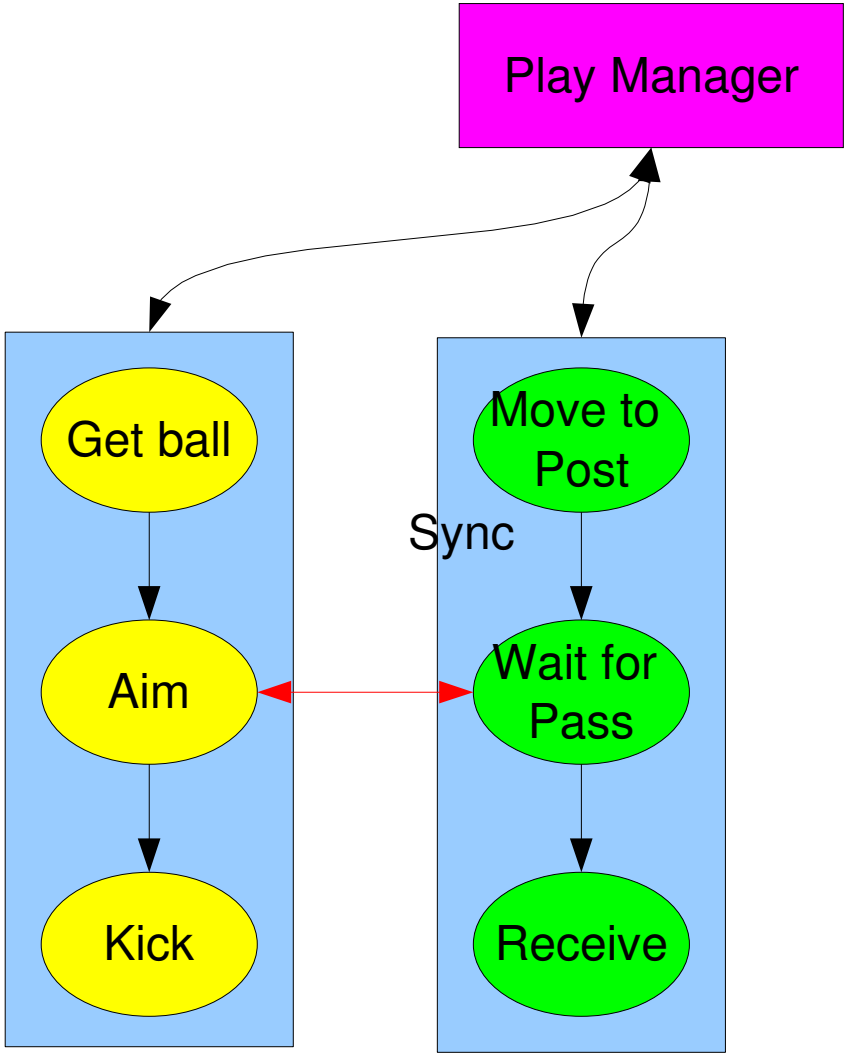
Distributed Plays

- Only communicate when synchronization is required, otherwise run independently



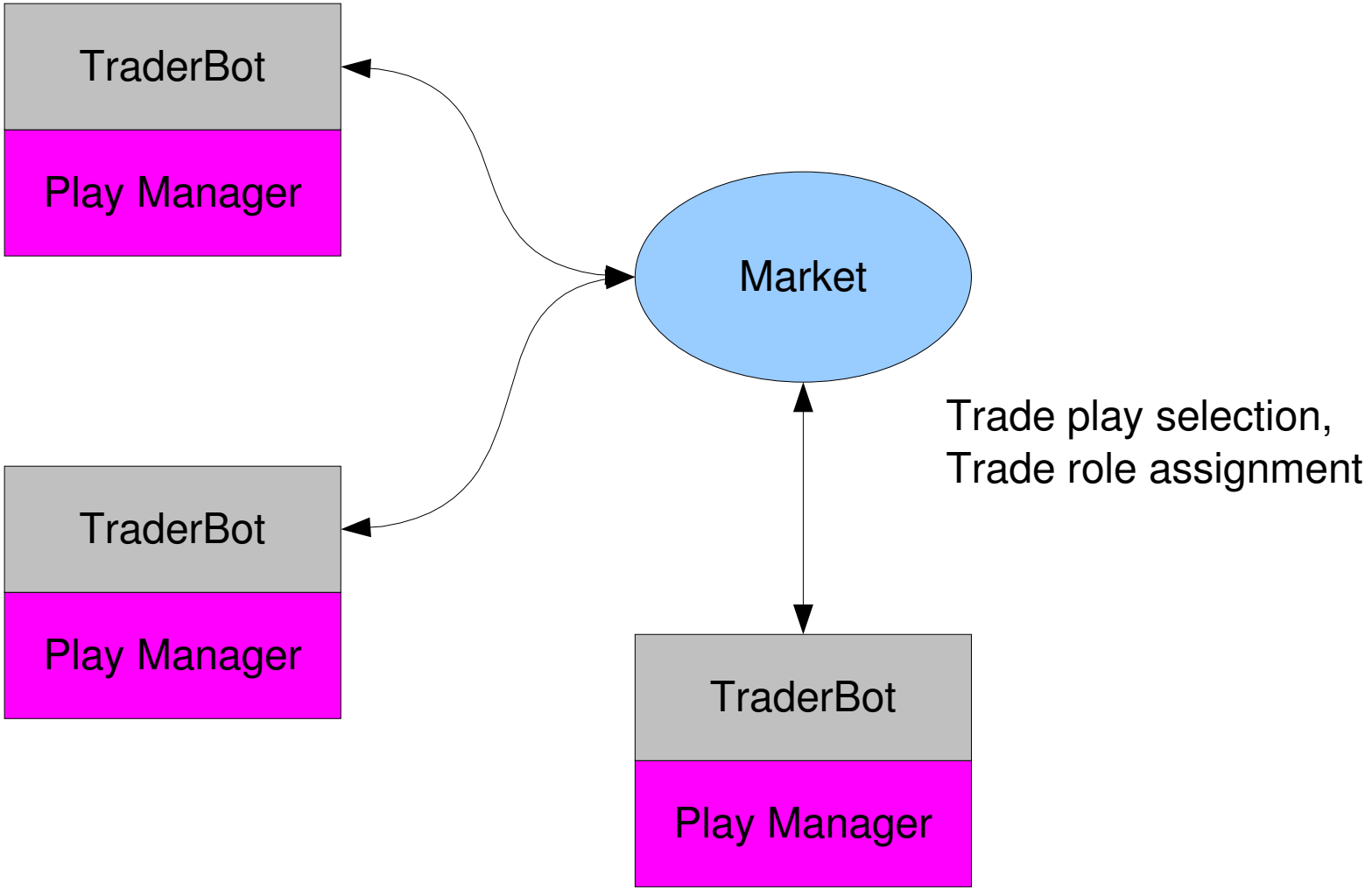
Distributed Plays

- Dynamic centralized agent monitors execution

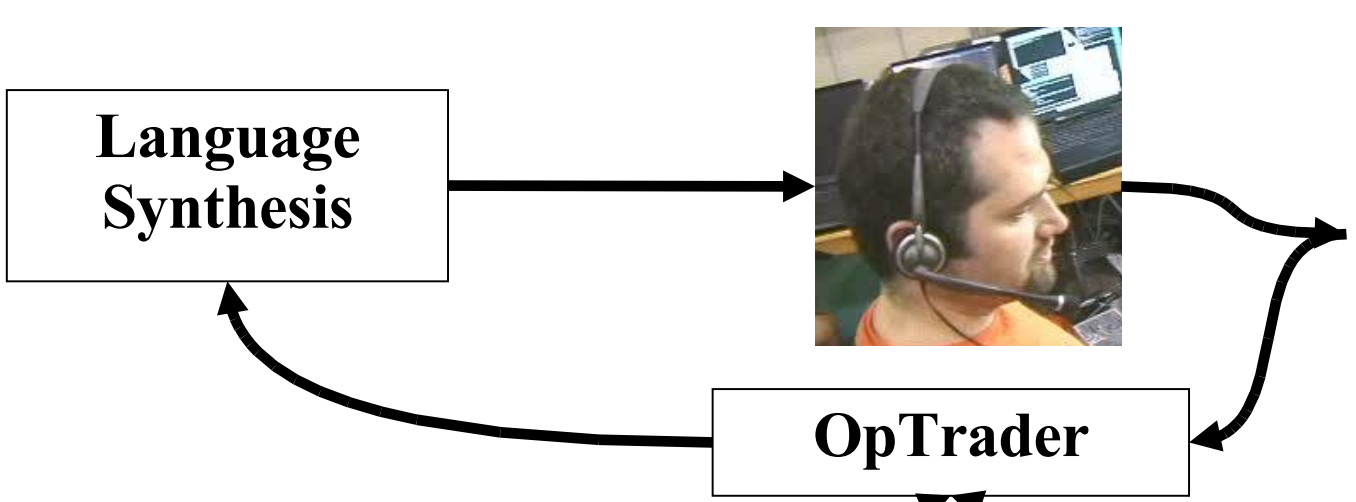


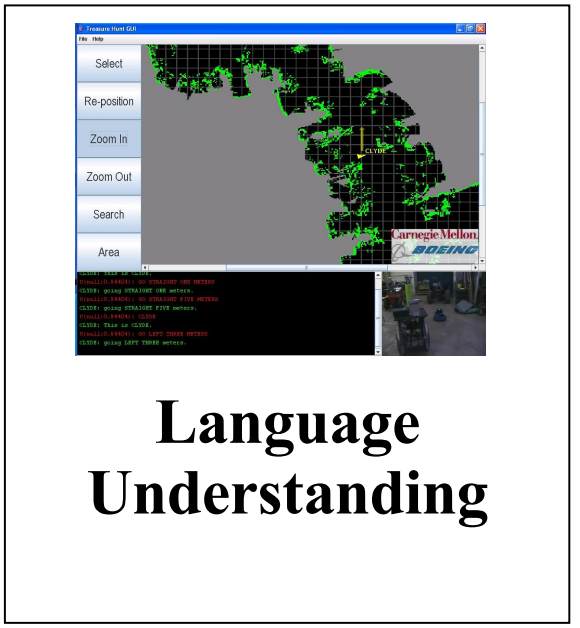
Market-based Role Assignment

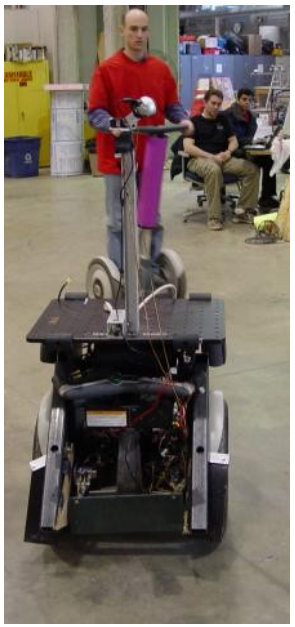
- Dynamic centralized agent monitors execution

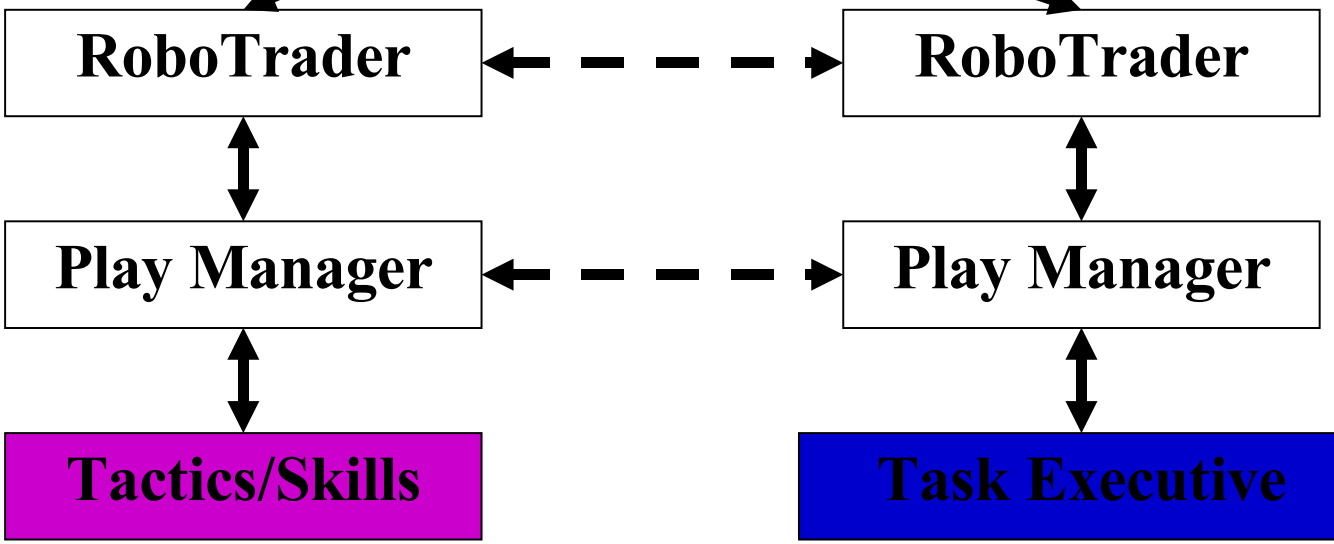


Boeing TreasureHunt

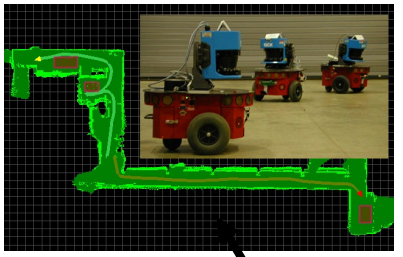


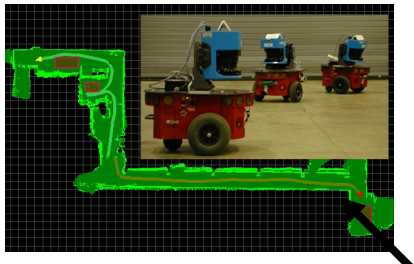






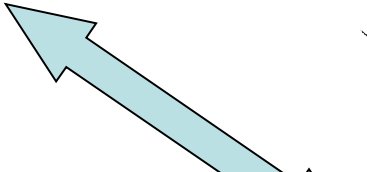
Task Allocation

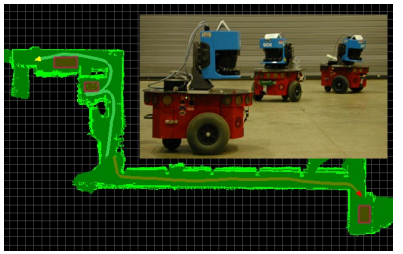


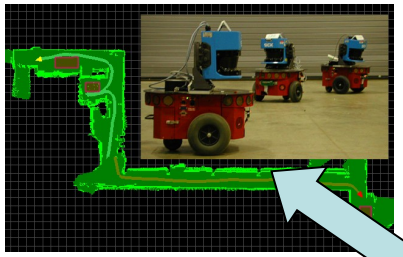


Task Allocation between Sub-Teams and Role Allocation within Sub-Teams

Tight Coordination



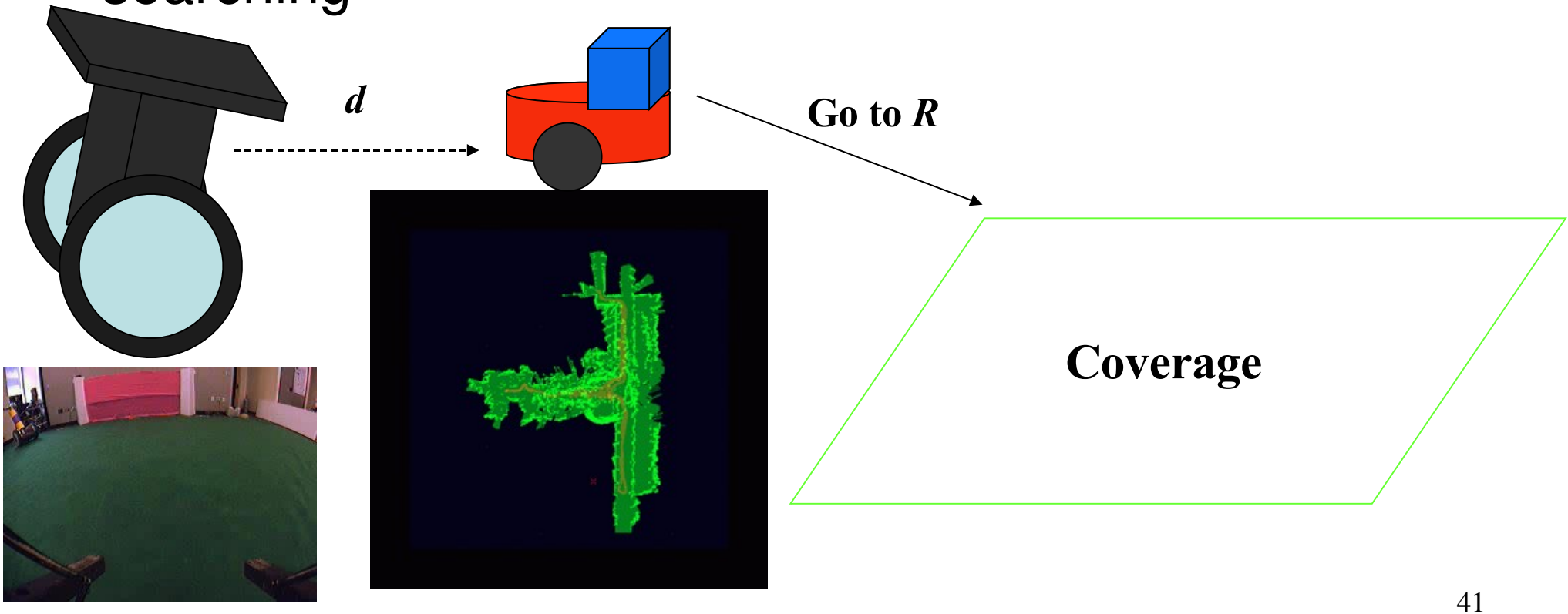




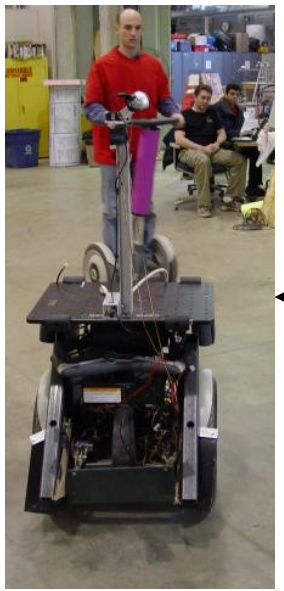
Team Action Coordination within Sub-Teams for Task Execution

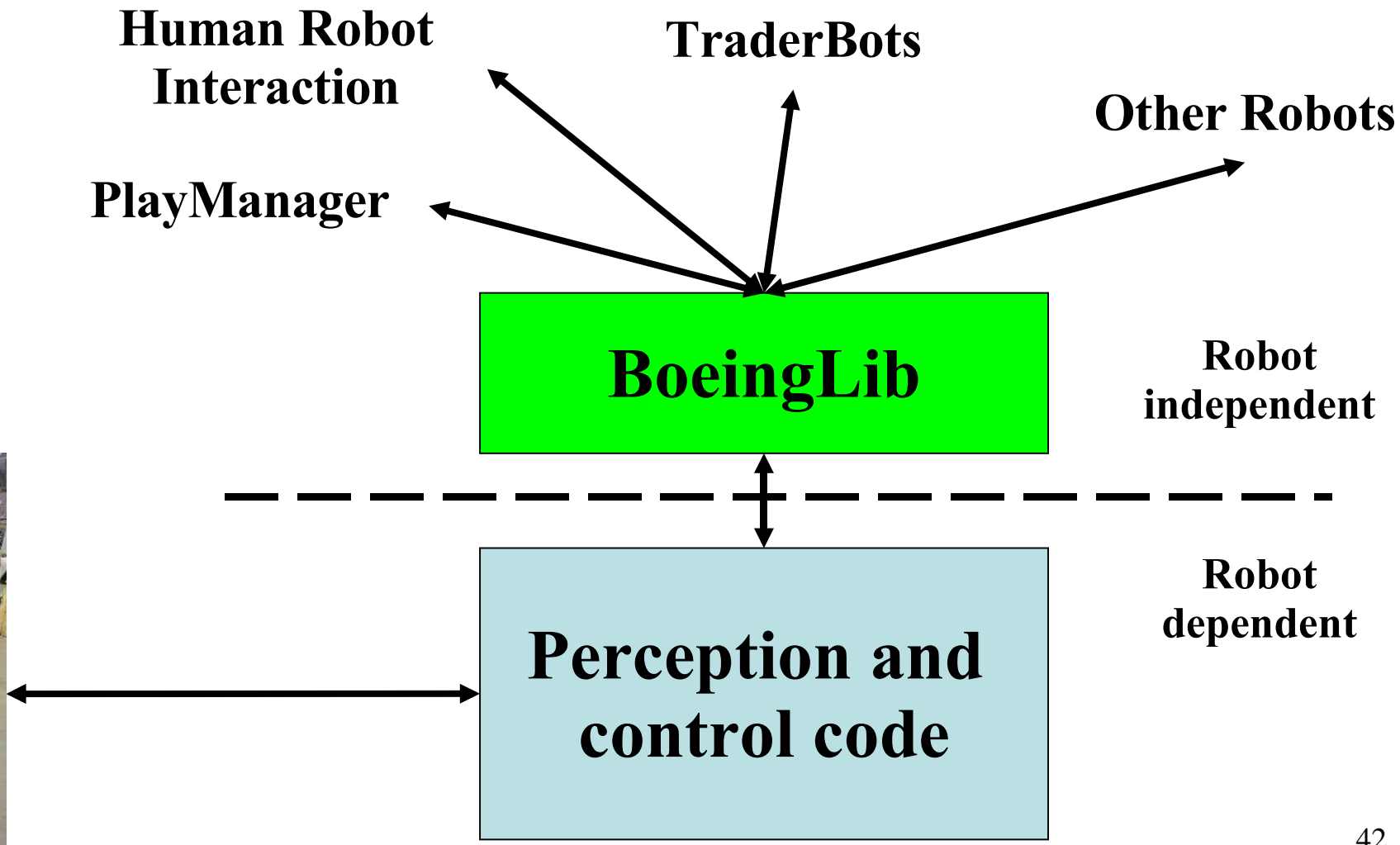
Search for Treasure

- **Human**: Overseas search for treasure
- **Pioneer**: Uses maps to plan coverage
- **Segway**: Uses camera to follow pioneer while searching



Pickup Teams…





Questions?