Analysis of Multiagent Teams

Ranjit Nair, Milind Tambe

Computer Science Department

University of Southern California

Motivation

- Agent teamwork applied to several realistic domains
  - —Framework of beliefs, desires and intentions (BDI)
- How do we analyze the performance of these teams?
  - —*Performance critical: linked to loss of human life, etc.*
  - —Suggest improvements to the team plan?
  - —In particular, improvements to role (re)allocation

Disaster Rescue simulations

Battlefield simulations

Contributions

- Teams operate in uncertain dynamic domains
  - —Uncertainty sources: non-determinism, partial observability, multiple agents
  - —Hence, use decentralized POMDP model

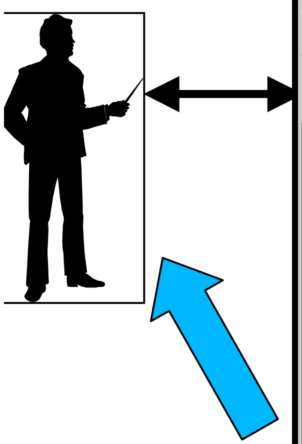
Key Contributions:

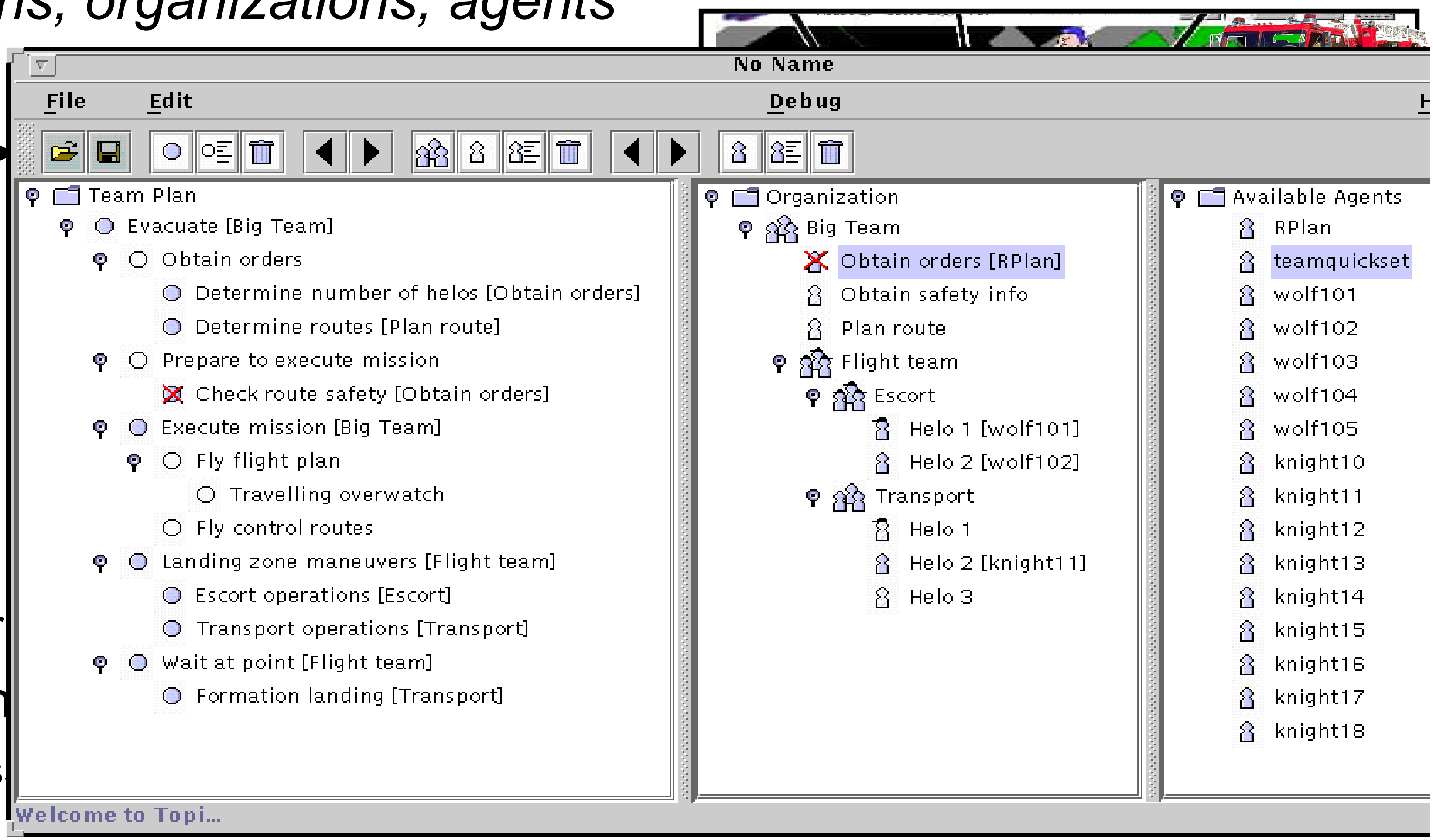
- *Analysis focused on only communication*
- **Approach:** Role-based Multiagent Team Decision Problem (RMTDP)
  - —Techniques for analysis of role allocation and reallocation
- *Analysis using decentralized POMDP model is difficult*
  - —Finding optimal Dec-POMDP policy is NEXP-Complete
  - —Even evaluating a policy is costly
- **Approach:** Methods for making analysis scalable
  - —Decomposition technique based on plan structure
  - —Heuristics for improving search time

Overall Approach

- Applying decentralized POMDPs to the analysis of real world systems

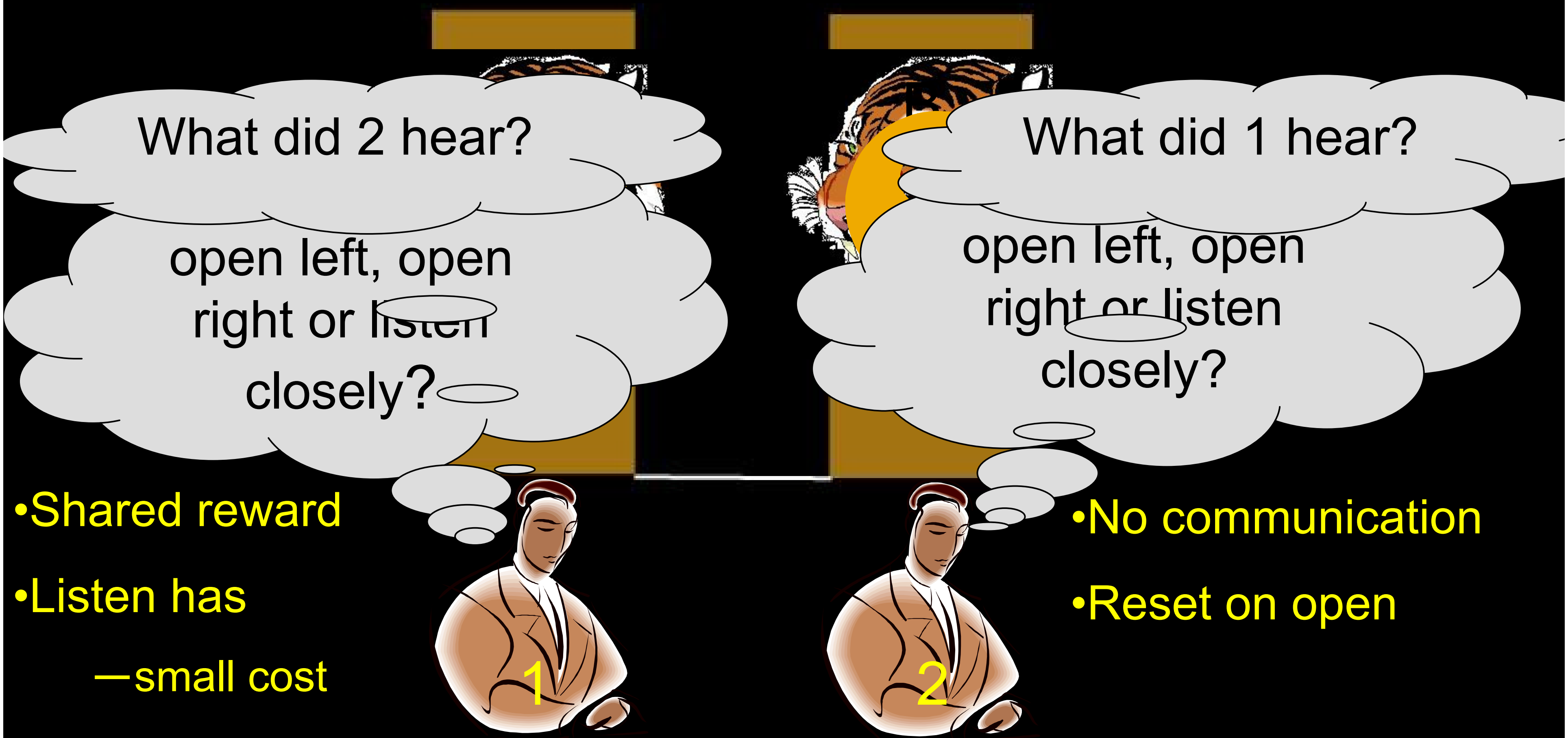
Team-oriented Programs:
Team plans, organizations, agents





- Other
- Off-lin
  - As

Multiagent tiger problem



- Shared reward
- Listen has
  - small cost
  - unreliable

- No communication
- Reset on open

What is the best joint policy over horizon T?

Background: Example Domain

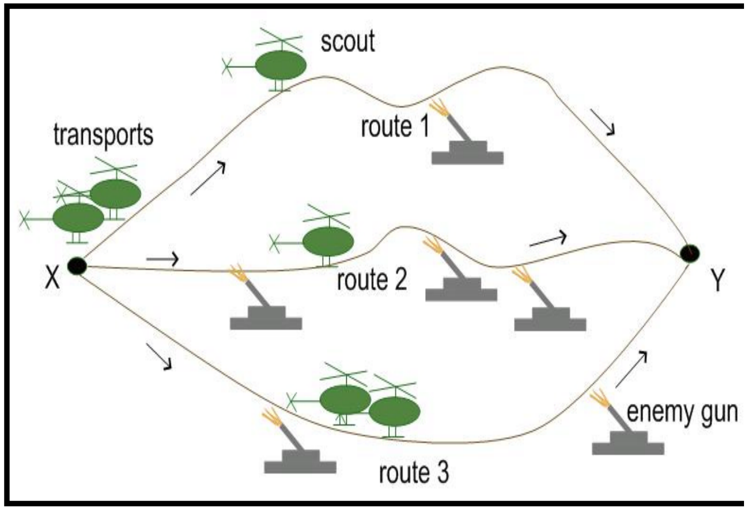
Task:

•Move cargo from X to Y along any route

•Helicopters need to be assigned to transport or scout role

•Scouts make assigned route safe while transports wait

•Once assigned, transport can become scout but not vice versa



Uncertainty:

•Each unscouted route: different failure probability and observability.

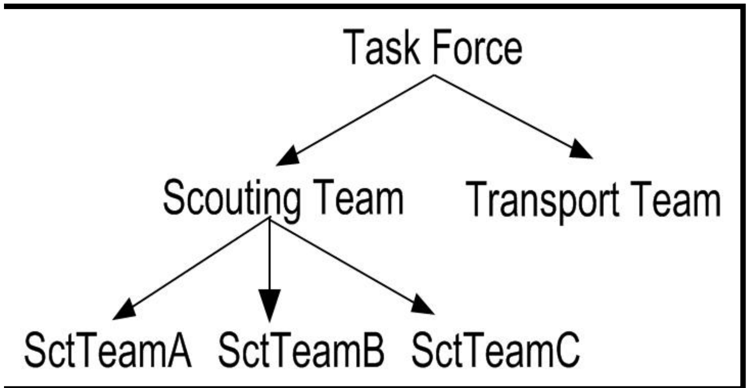
•Probability of failure depends on number of scouts

Goal:

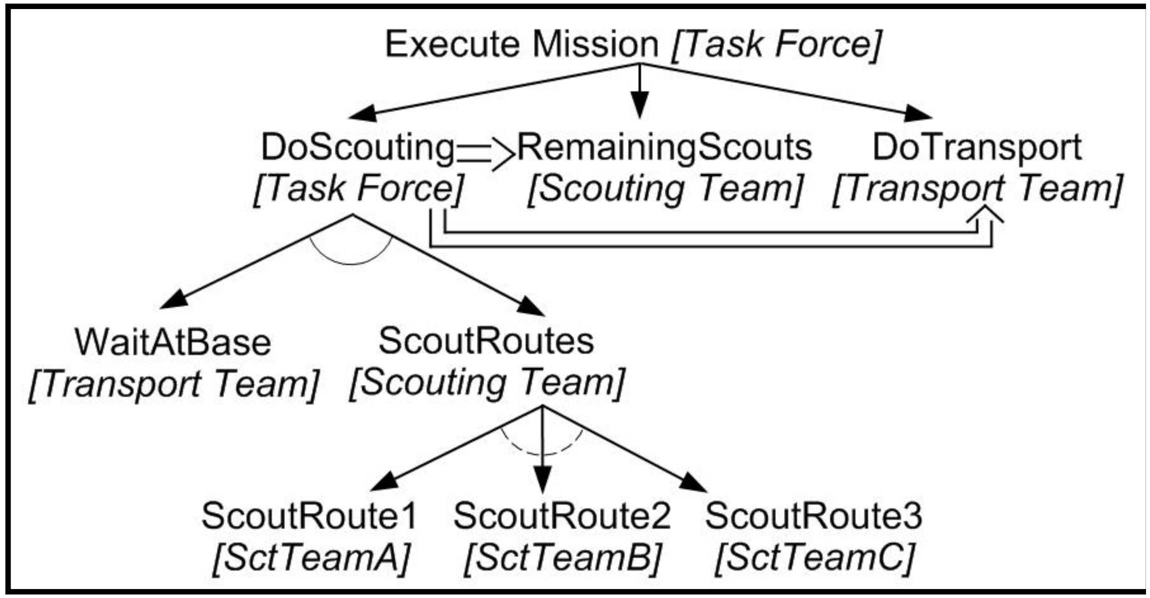
•Best role allocation? How many transports? How many scouts on each route?

•How should agents reallocate?

Background: Team-oriented Program



Organization hierarchy



Plan hierarchy

Given an assignment of subteams to subplans:

- Role Allocation: Best allocation of agents to roles (in organization hierarchy)?
  - —E.g. How many helicopters to each subteam?
- Role Reallocation: When and how should agents reallocate?
  - —Compare different reallocation strategies (see paper)

MDPs and POMDPs (background)

Markov Decision Problem (MDP) = <S, A, P, R>

S: States

A: Actions

Agent's actions have non-deterministic effects

P: Transition function

Obeys Markovian property

P(s,a,s') = Pr(s'|a,s)

R: reward function R:S X A $\rightarrow \mathcal{R}$

Goal: Find best action for each state (policy)

Partially observable Markov Decision (POMDP) = < S, A, P, O, Ω, R>

Agent has partial knowledge of state O(s,a,ω) = Pr(ω|s,a)

Goal: Find best action for each belief state

RMTDP model

Role-based Multiagent Team Decision Problem

- <*S, A*, P, Ω, O, R>: same as other DEC-POMDP models
- Separate out coordination actions that we wish to analyze i.e. role-taking
  - A= $\times_i A_i$: A_i is *role-taking* Y_i *or role-execution* Φ_i
- R: Reward; sub-divided based on action types
  - *Reward for role taking and for execution actions*

Policy π: Action selection of team is specified by joint policy

Joint policy:< $\pi_1, \ldots, \pi_n$>

Local policy for agent i, π_i :

- $\pi_{i\text{Role taking}}$: $\omega^1_i, \ldots, \omega^t_i \rightarrow$ role-taking action
- $\pi_{i\text{Role execution}}$: $\omega^1_i, \ldots, \omega^t_i \rightarrow$ role-execution action

Complexity Issues

- **Theorem1:** The decision problem of determining if there exist role-taking and role-execution policies that yield a reward at least K over finite horizon T is **NEXP-complete.** *(Policy Existence Problem)*

- What if we fix the role-execution policy?
- **Theorem2:** Policy Existence Problem for role-taking policy with a fixed role-execution policy is **NEXP-complete.**

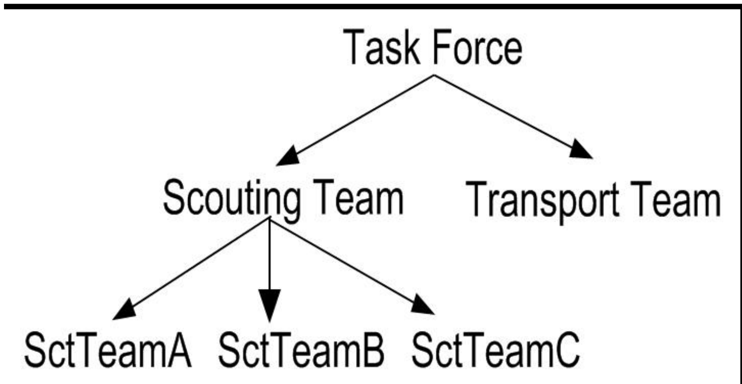
- Finding the globally optimal role-taking policy: intractable and likely doubly exponential

  —Brute force search requires $O\left[\left(|Role-taking|^{\frac{|\Omega|^T-1}{|\Omega|-1}}\right)^n\right]$ evaluations.

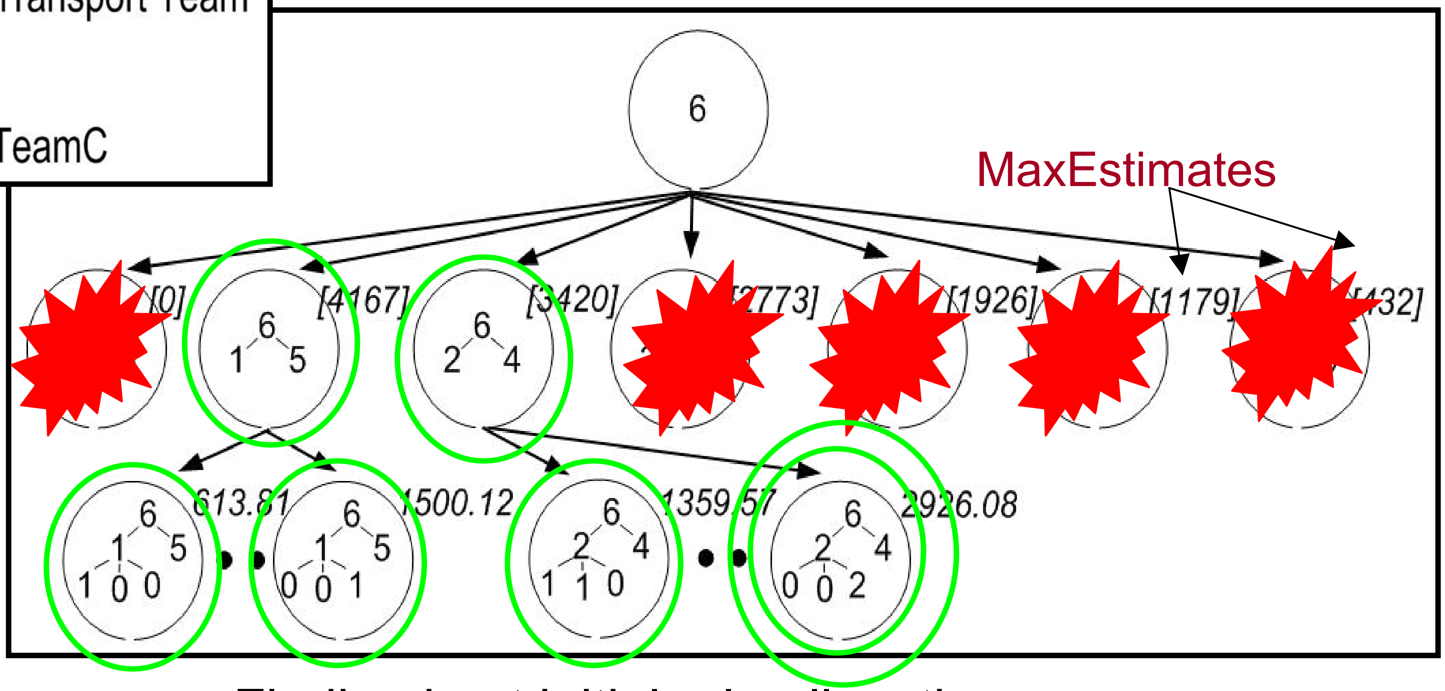
•Hence, separate role allocation and reallocation analyses

Analysis of Role Allocation

- Assumes fixed reallocation policy (e.g. STEAM) and fixed role-execution policy (from TOP)



Role Allocation space

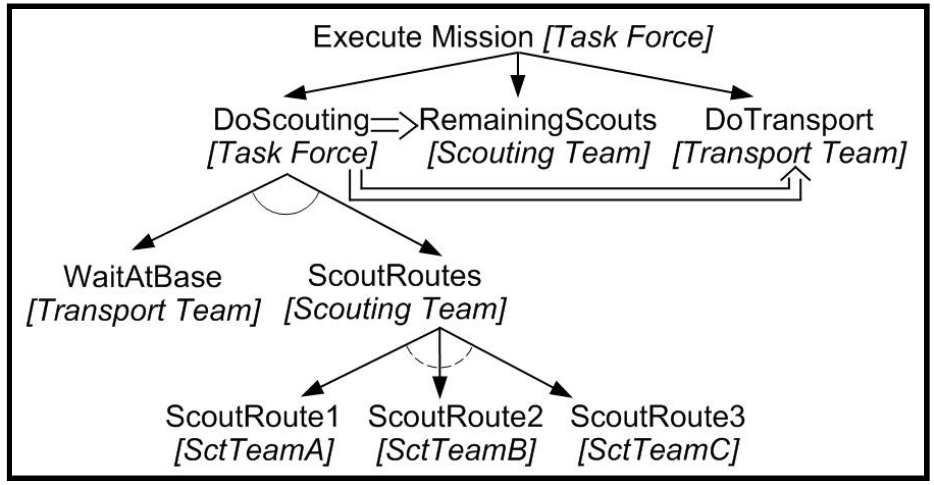


Finding best initial role allocation

- Branch-and-bound search using MaxEstimates of parent nodes for pruning

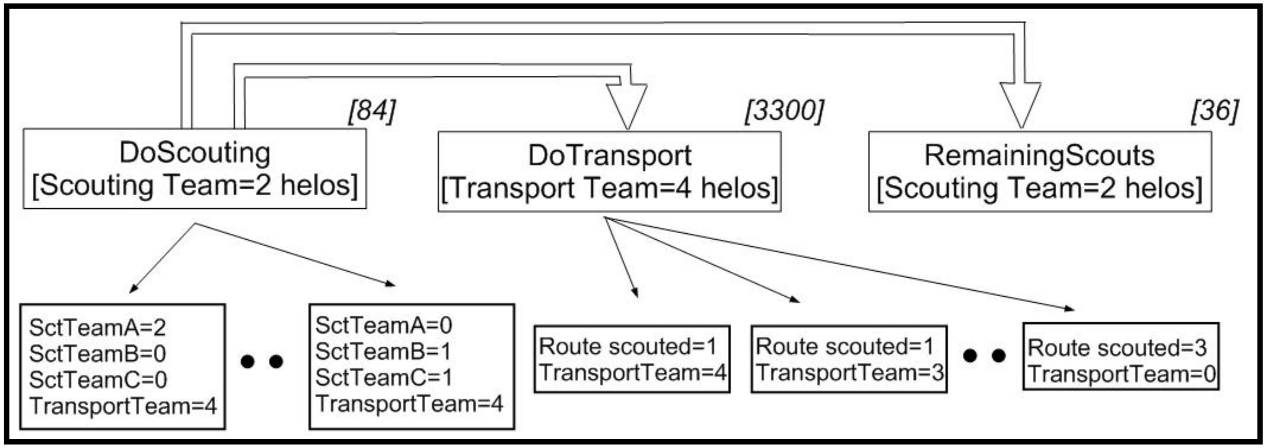
RMTDP Decomposition

- Identify components in plan hierarchy to decompose RMTDP
  - —Components with temporal constraints.
  - —Independent components
  - —Obtain smaller RMTDPs for each component
    - Provided by domain expert



Decomposition allows fast component-wise computation of max estimates.

Component-wise Max Estimate



3rd parent in allocation space

- Obtain start states and starting observation histories
  - —1st component: start states= all possible initial allocations
  - —Otherwise: start states = end states of previous component; similarly for observation histories
- Obtain maximum expected utility (MEU) of each component over all start states and observation histories
- MaxEstimate = $\Sigma_{\nabla j}$ MEU_j
  - *Called MAXEXP* = 84+3300+36=3420

Component-wise Max Estimate

Savings due to decomposition:

- Component-wise evaluation avoids duplication of evaluation
  - Combine end states before determining next component's start states
- Not all variables of a component are relevant to resulting components
  - Remove irrelevant variables from end states
  - Delete resulting duplicate states → fewer start states
- Similarly, with observation histories
  - Irrelevant observations can be removed

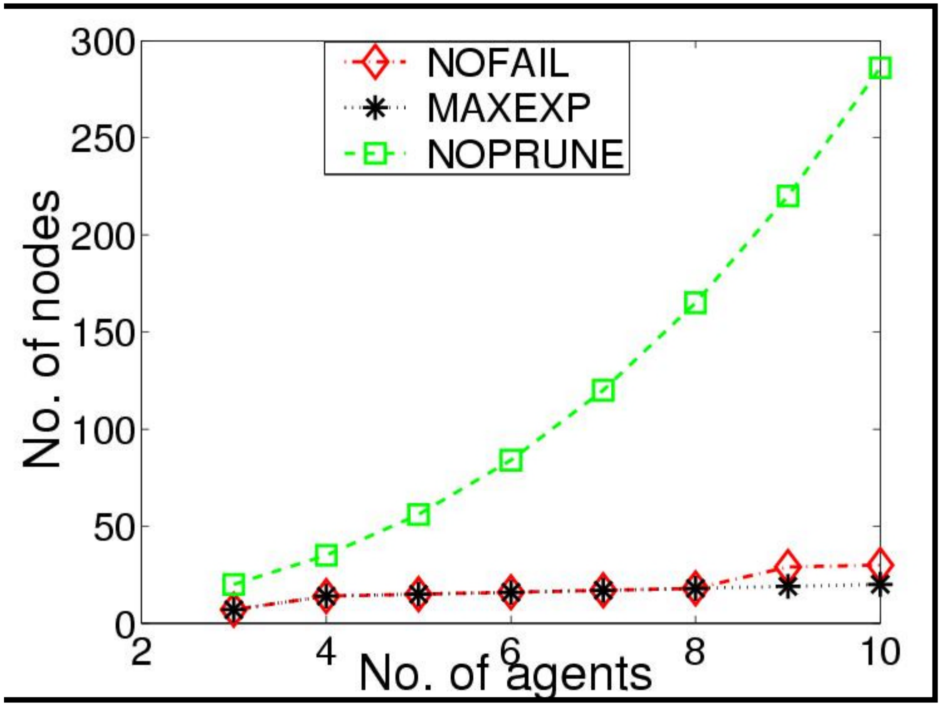
NOFAIL heuristic:

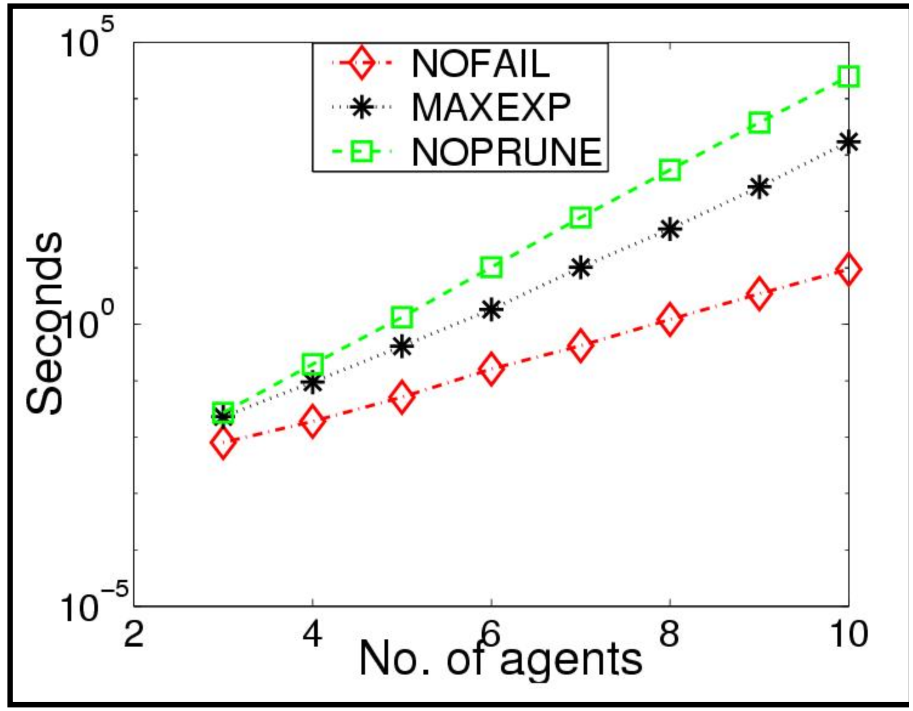
- Similar to MAXEXP
- Assumes agents don't failure (only for computation of max estimate)
- Results in less branching in evaluation
- Valid for some domains

Other heuristics that guarantee correctness possible

Role Allocation Results

•NOPRUNE: brute force evaluation of all leafs





•20 fold reduction in number of nodes for 10 agents

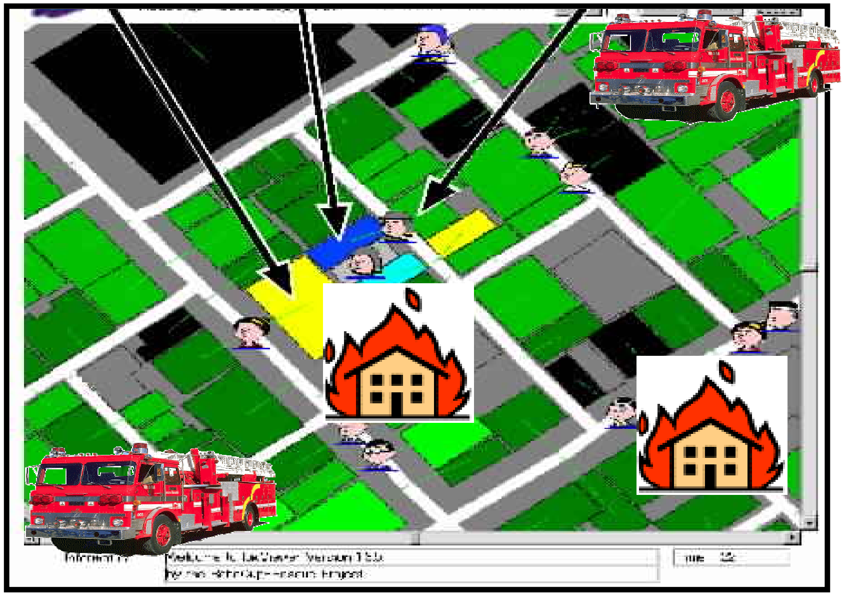
•MAXEXP evaluates fewer nodes than NOFAIL

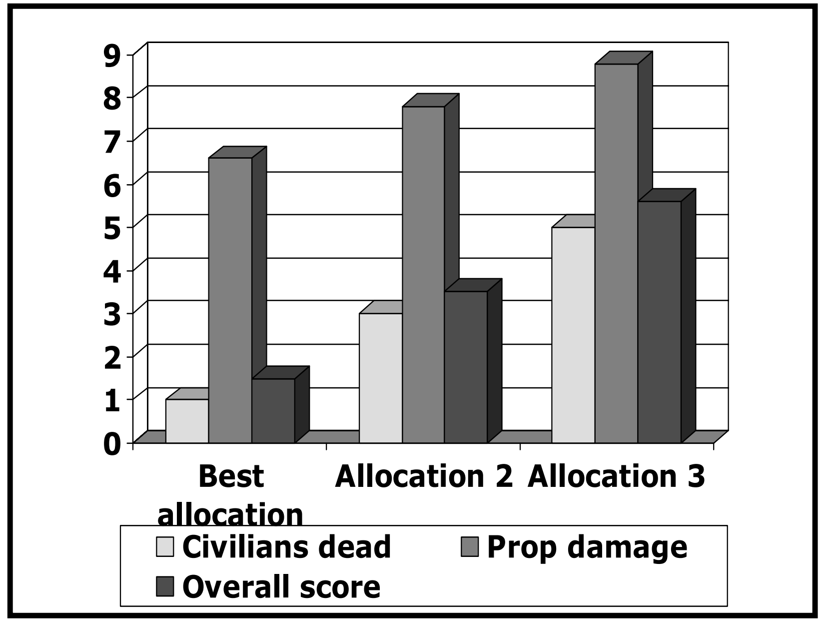
•MAXEXP:14-fold speed up over NOPRUNE for n=10

•NOFAIL: 140-fold speed up over MAXEXP for n=10

USC University of Southern California
agents@USC

RoboCupRescue Results





<u>Allocate fire-engines & ambulances in RoboCupRescue</u>

- 7 fire brigades and 5 ambulances
- 2 fires with trapped civilians

• Best allocation: saved 6/7 civilians and resulted in less property damage

•Allocation 2: good evaluation but significantly lower than best

•Allocation 3: predicted to perform badly

Summary

- BDI-based team plans need analysis tools
  - *Marry BDI and POMDP approaches*
- RMTDP Model for analysis of role (re)allocation
  - Useful for evaluating a TOP
- Finding best initial role allocation
  - Novel decomposition technique
- Comparing role reallocation strategies (in paper)
  - Family of locally optimal perturbations

Thank You