Multiagent Planning Problems and their Relevance to Next-Generation Transportation Systems

Michael Rovatsos
School of Informatics
The University of Edinburgh

Porto 12/07/2012

Introduction

- Planning is a key ability of intelligent systems, when these are distributed it becomes multiagent planning
- Algorithmically, a problem of generating action sequences that will bring about a certain goal
- Many other aspects: uncertainty, execution & monitoring, mixed-initiative planning
- Highly relevant to transportation & logistics domains
  - planning travel routes
  - planning portage tasks
  - planning collaborative transportation

Single-Agent Planning

- For single-agent planning, there exist simple and general formulations of planning problems
- STRIPS-based classical planning problem *P=<F,I,A,G>* with fluents *F*, initial state *I,* actions *A* and goal *G*
- Fluents are propositional properties of states, states are sets of these
- Actions have the form *<a,pre,eff>* where *pre*⊆*F* and *eff=(add,del)*
- Fluents *add*⊆*F* and *del*⊆*F* are added to/deleted from the current state when *a* is executed

Single-Agent Planning

- State transition function yields $S' = S \setminus del(a) \cup add(a)$ when *a* is executed in *S*
- Blocks' World example: action a=*Stack(X,Y), pre(a)= {Clear(Y), Holding(X)}, add(a)={On(X,Y), ArmEmpty} del (a)={Clear(Y), Holding(X)}*
- A plan $p=<a_1,\ldots,a_n>$ is a solution to planning problem *P* if execution of $a_1,\ldots,a_n$ from *I yields a state S* and $G \subseteq S$
- Various extensions to this:
  - conditional effects, disjunctive effects & preconds
  - uncertainty: conformant & conditional planning
  - concurrency and scheduling, temporal planning
  - planning with preferences and side conditions

Single-Agent Planning

- Planning made a lot of progress because of
  - common problem formulation
  - scalable algorithms
  - benchmarks for empirical evaluation
- In multiagent planning, additional complications:
  - concurrent action, different views, different goals
  - coordinating the planning activity itself
- This makes the problem much much harder...

Multiagent planning

- Differently from single-agent planning, a very fragmented area
- Problems addressed include:
  - centralised planning with concurrent actions
  - plan merging from individual agent plans
  - planning-time co-ordination of planning agents
  - centralised/decentralised strategic planning
  - continuous planning, execution, and co-ordination

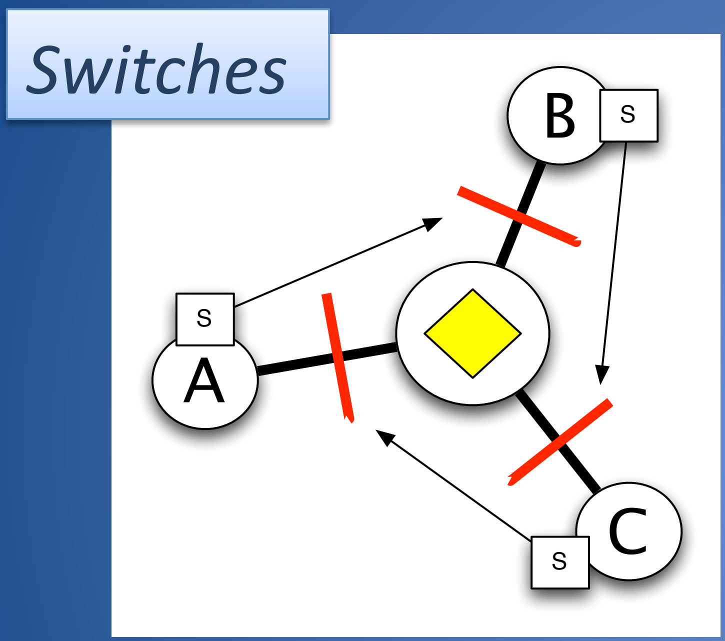
Our work in multiagent planning

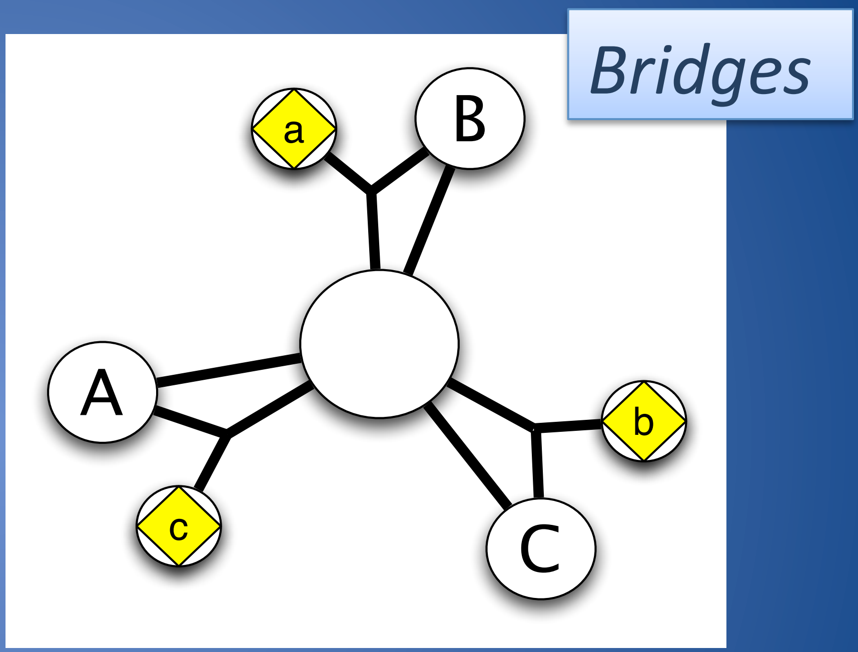
- We try to focus on multiagent planning problems that address different central issues
- I will discuss work on four problems and make connections to transportation domains:
  - concurrent centralised planning
  - multi-perspective planning
  - multi-objective planning
  - automated norm synthesis

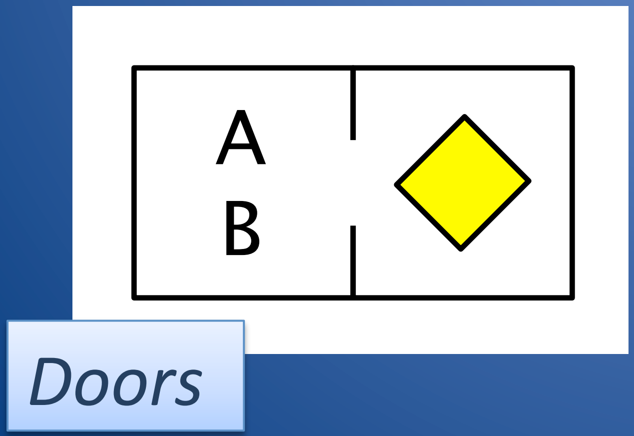
Concurrent planning

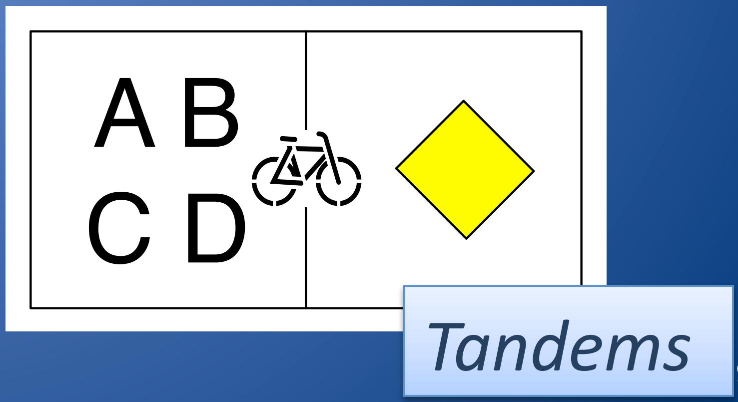
- Different execution models for multiagent systems
  - concurrent (synchronous/asynchronous)
  - sequential (synchronised/asynchronous)
- Concurrent model most expressive but leads to combinatorial explosion in action sets
- We focus on synchonised, concurrent problem $P=\langle F,I,A_1 \times \ldots \times A_n,G\rangle$
- Four different types of interaction between individual agents' activities (can occur in parallel)

Example domains









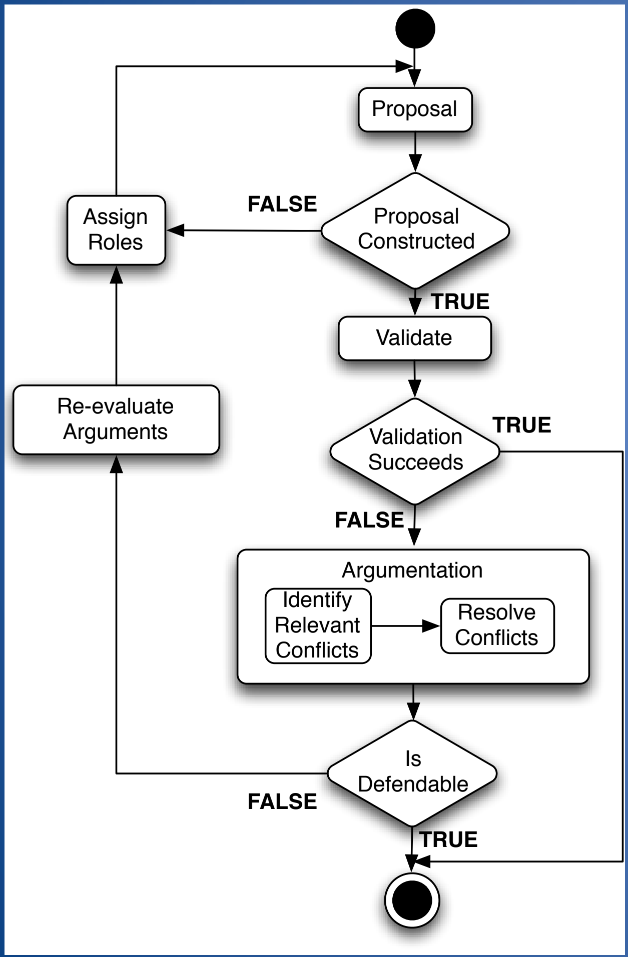
Concurrent planning

- Crosby has developed novel heuristics to make centralised planning much more scalable
- Based on building separate planning graphs for individual agents
- Helpful actions are assumed to be provided by others when a fluent cannot be achieved
- Highly relevant for transportation problems with resource sharing, contention, side effects:
  - urban traffic management and optimisation
  - cooperative multi-modal logistics
  - non-replenishing resources (e.g. flight tickets)

Multi-Perspective Planning

- Agents disagree about initial state and action definitions, but share goal: $P=\langle F, A_i, I_i, G \rangle$
- Acceptable planning problem: *P* is acceptable wrt KB_1 and KB_2 iff $KB_1 \models P$ and $KB_2 \models P$
- Belesiotis' argumentation-based method of determining winning arguments based on evaluating individual agents' proposals
- Scalability achieved by using off-the-shelf single-agent planner for sub-tasks in the argumentation process

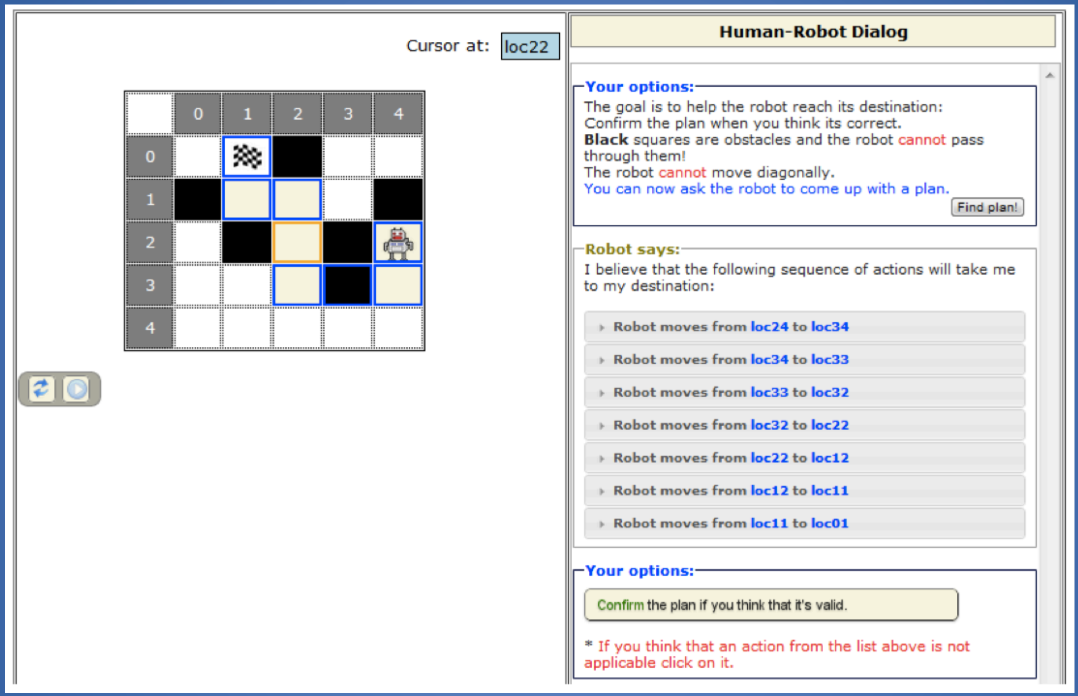
Argumentation-based conflict resolution



- Plan proposal generated by single agent (with any planner)
- Dispute in case of disagreement, argumentation follows
- Ends in successful defence of initial proposal or rejection
- An alternative to generating one *P* that works under both *KBs*
- Can be used in single-agent way to make decisions under conflicting infomation

Application: ArguDem

- Moralis' demonstrator uses this method to help a navigating robot:



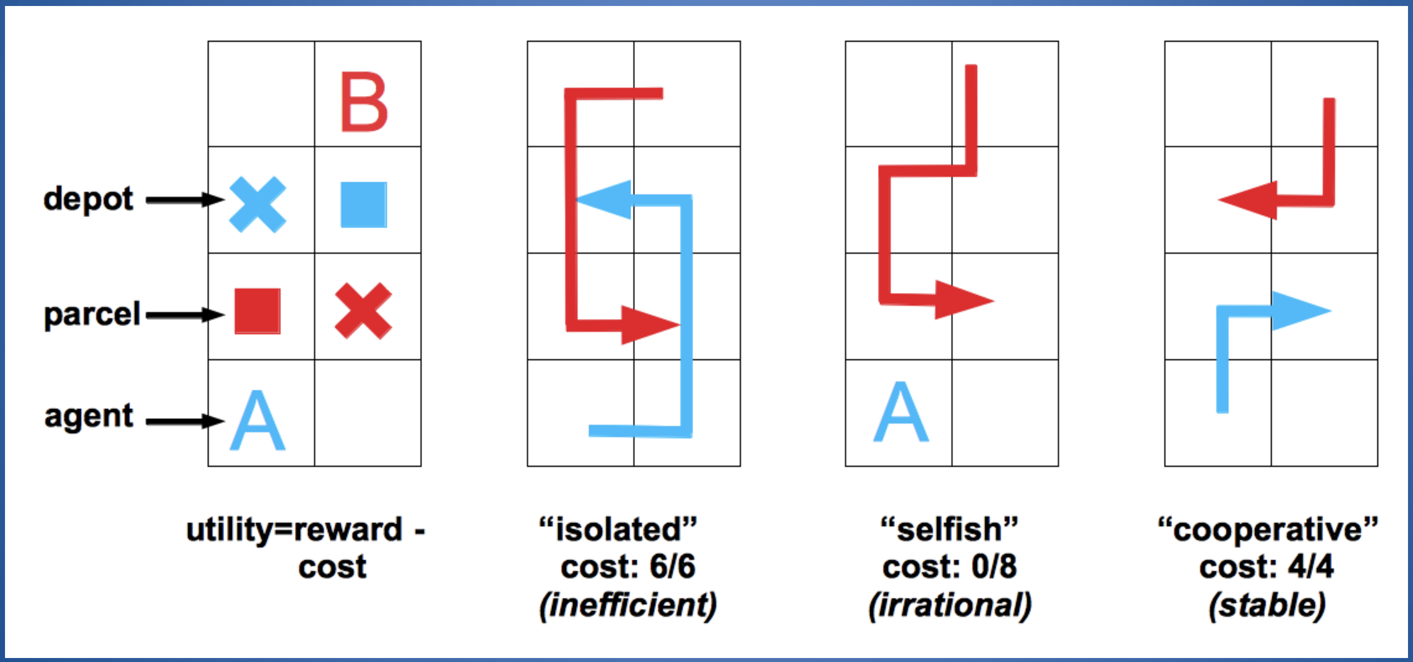
- Relevant in domains with conflicting information:
  - autonomous vehicles with different local information
  - remote sensing (also via input from people)
  - transportation planning under uncertainty

Multi-Objective Planning

- Agents have independent goals: $P=<F,A,I,G_i>$
- Strategic problem, acceptability based on notions of stability and equilibrium
- Problem depends on whether contracts can be enforced and utility can be transferred
- Like concurrent planning with additional constraints on plan cost to individuals
- Hard to define meaningful solution concepts if goals incompatible or agents untrustworthy

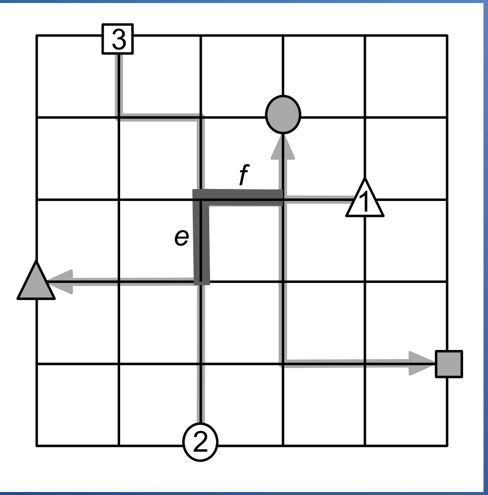
Example

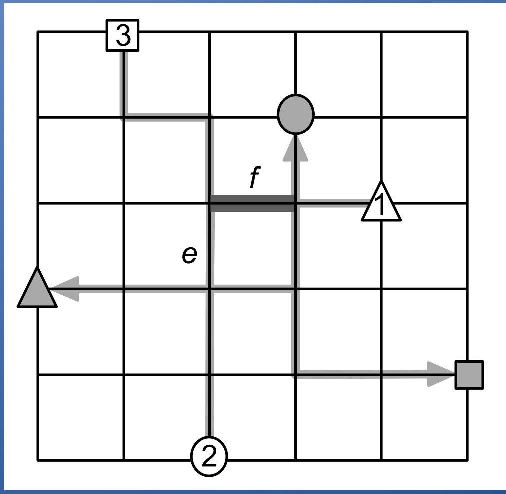
- Parcel delivery domain



Multi-Objective Planning

- Best-Response Planning (Jonsson & MR):
  - iterative method of optimising agents' individual plans without breaking others' plans
  - computes equilibrium plans fast in congestion games, restricted to interactions of cost
  - useful for plan optimisation in other domains
- Network routing example:





Application: Travel Sharing

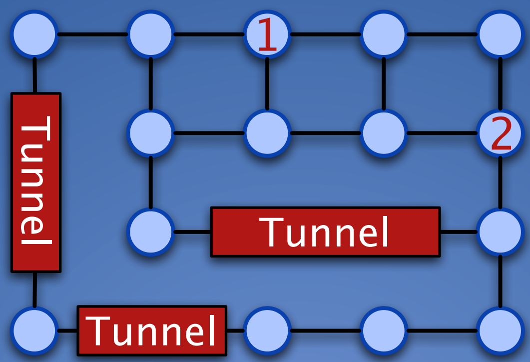
- Hrncir's system uses BRP to determine joint travel routes using real-world UK public transportation data (>200,000 connections)

- Relevant for transportation problems with conflict of interest:
  - calculating routes for trip sharing (with no incentives to deviate)
  - agreeing on cross-organisational logistics collaboration
  - recommender systems for congestion control

Automated Norm Synthesis

- Avoiding undesirable states in a system regardless of agents' planning activities
- Given a planning domain, calculate a set of prohibitions for agents that avoid conflict states
- Christelis developed CRS algorithm based on forward-backward search around conflict states guaranteeing full goal accessibility
- Pruning techniques and use of single-agent performant planners result in highly scalable methods

Tunnel World Example



- Inside tunnels no change of direction or stopping
- Our algorithm solves this by computing a general norm
  "if you are next to a tunnel and another agent is at the opposite end, don't enter the tunnel"
- Relevant for transportation domains with soft or hard safety constraints:
  - design of traffic rules
  - non-disruptive roadwork and maintenance planning
  - congestion avoidance recommender systems

Conclusion

- Combining single-agent planning technology with novel ideas can help solve hard problems
- These problems are highly relevant to transportation domains
- Their contribution is automated reasoning about complex domains
  - to influence behaviour
  - to ensure safety
  - to optimise resource allocation
  - to balance different objectives

Conclusion

- There are many opportunities to exploit the potential of these methods in the real world:
  - "open data" gives us easy access to real-world information about transportation domains
  - mobile technologies provide multi-perspective input, "human-based" computation, novel interaction capabilities
  - novel transportation and vehicle technologies permit more automation of control
  - transportation-style problems are the most explored (and probably most suitable) domains for planning
  - huge potential of mixed-initiative "human-in-the-loop" technologies and "social computation" unexplored

Questions?