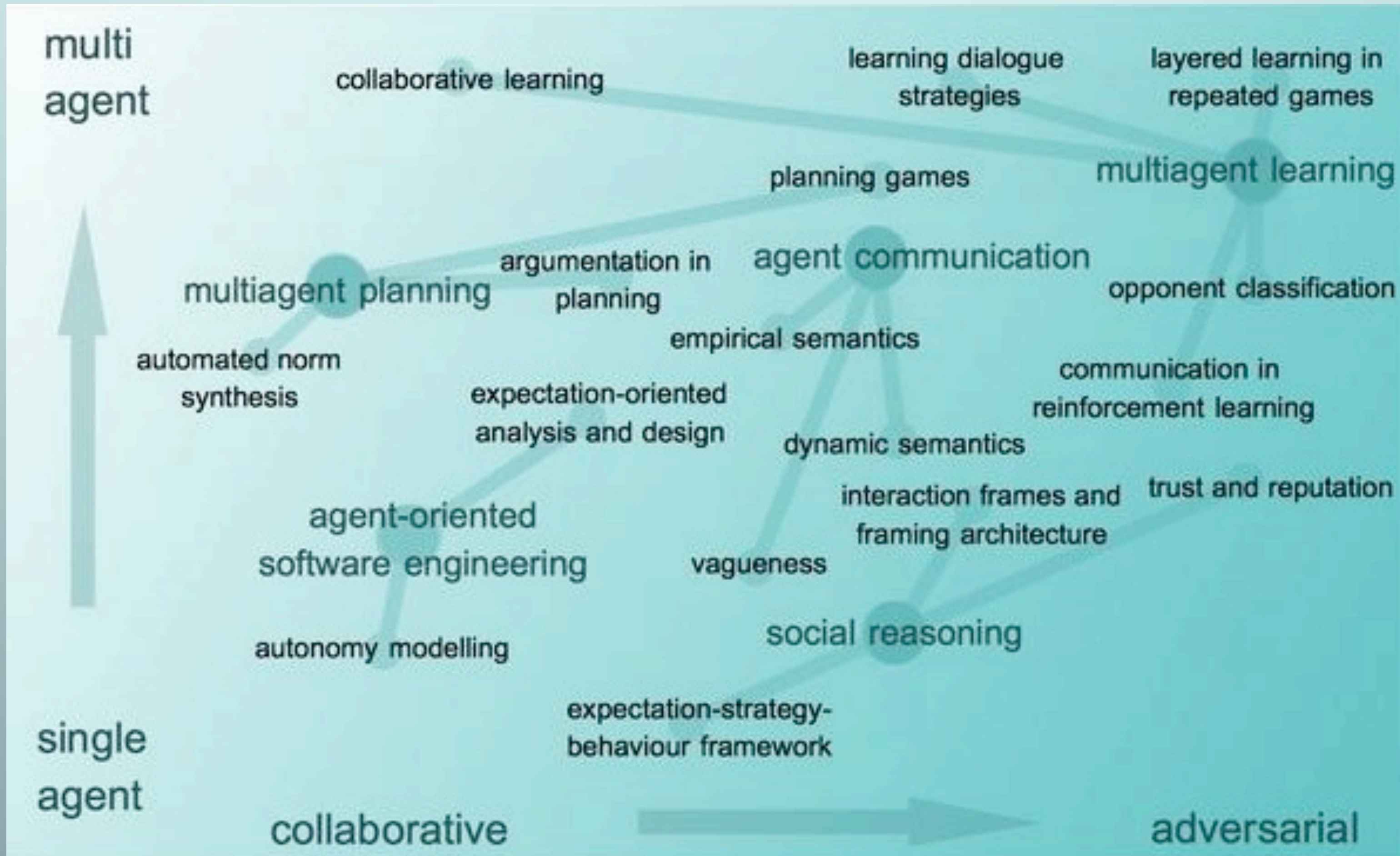
Knowledge-Based Reasoning about Strategic Interaction

(speculative talk)

Michael Rovatsos
(joint work with Matt Crosby, George Christelis, Alexandros Belesiotis)

University of Edinburgh
mrovatso@inf.ed.ac.uk

Background: Work in my group



Go to www.cisa.inf.ed.ac.uk/agents for details.

Current research agenda

- Interested in knowledge-based reasoning about interaction
- Reasoning about interaction is by definition practical reasoning
- Vision: given a specification of the interaction problem, automatically synthesise behaviour
- Agents field has not many general things to say about the strategic case of this
- Specific approaches exist (I will give two examples in this talk), but research is fragmented

Why not game theory?

- Game-theoretic methods very popular currently and address the problem of reasoning about interaction
- Information in real-world domains available in relational terms (e.g. on the Web), not enumerated state actions as assumed in game theory
- Non-incremental: unable to express how a game changes when we incrementally change background knowledge
- Knowledge-based methods might be useful in lifting overly restrictive assumptions (full rationality, perfect knowledge, etc)
- Intuition: many large-scale games might be actually "easier" than we think *(this is speculative)*

Why planning?

- We are interested in building systems, not only specifying them formally
- Rational agents need to synthesise action plans to operate autonomously
- We want to tell them what to achieve, not how, abstraction desirable
- Planning is the interface between KR methods and practical reasoning
- Unfortunately, no standard frameworks for multiagent planning (let alone strategic MAP) exist

Current work

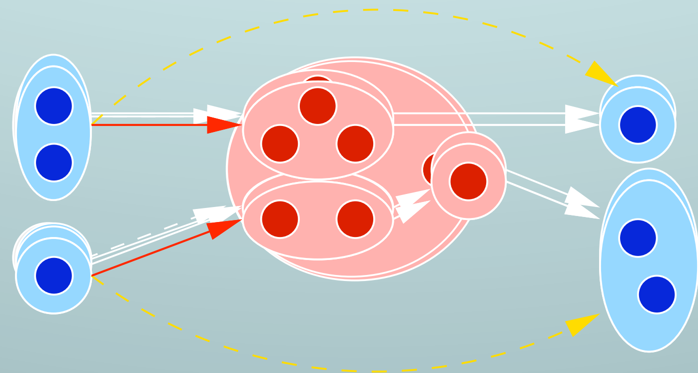
- Two examples of planning-related approaches for multiagent problems:
  - Automated norm synthesis
  - Argumentation-based conflict resolution
- These are popular problems in the multiagent systems community
- Our motivation is to make them more useful for automated reasoning about interaction
- Adapted them to planning environments
- Will give high-level overview, happy to discuss in more detail

Automated norm synthesis

- Norms ensure global conflict states are never entered by prohibiting certain actions in certain states
- At the same time agents' private goals should remain achievable
- Automated synthesis of such norms is NP-hard in enumerated state systems
- Existing methods don't exploit abstractions of propositional/first-order domain theories
- Our method: find "detours" around conflict states by local search in generalised state spaces

Automated norm synthesis

Iterated process of forward-backward search around conflict state specification:



- **Not better than full state-space search in the worst case but often get lucky**
- **With simple additional pruning techniques search can often be cut down drastically**
- **Currently working on synthesising sanctions**

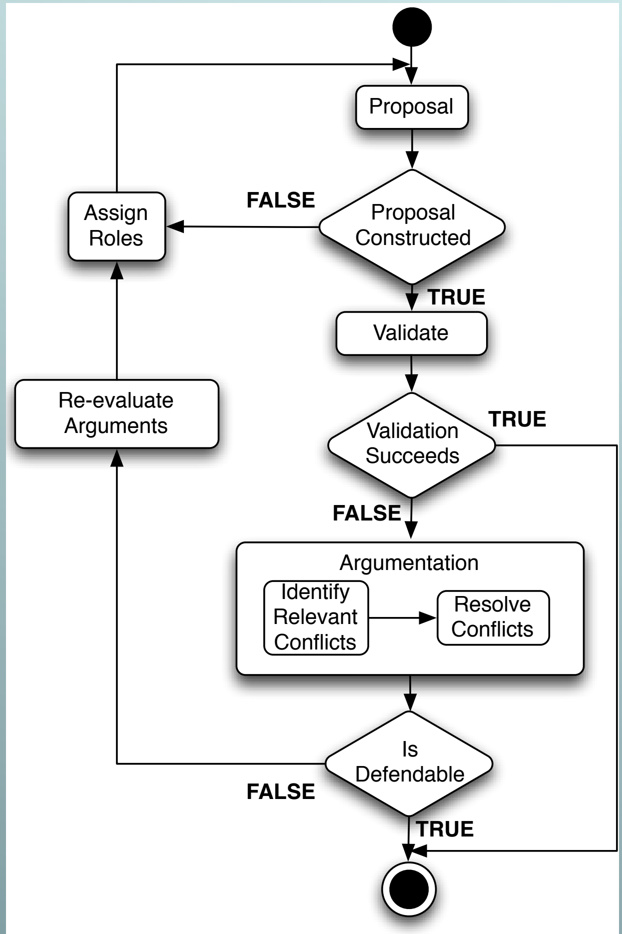
Argumentation-based conflict resolution

- **Argumentation** is a method for determining the status of propositions in the presence of conflicting information
- Different acceptability-based semantics and protocols that implement these
- Rarely used for reasoning about action, our intuition is that this can be done more efficiently due to domain structure
- Suggest framework for **acceptable** planning:

  A plan P is acceptable wrt (potentially conflicting) knowledge bases KB_1 and KB_2

  iff $KB_1 \models P$ and $KB_2 \models P$

Argumentation-based conflict resolution



- Plan proposal generated by singe agent (with any planner)
- Validation based on simple plan projection
- Dispute in case of disagree-ment, argumentation follows
- Ends in successful defence of initial proposal or rejection
- An alternative to generating one *P* that works under both *KBs*

Argumentation-based conflict resolution

- Planning domain represented in Situation Calculus
- Disagreement may exist regarding
  - initial state (including background knowledge)
  - planning operators (agreement on goal)
- Application of TPI-dispute protocol, but argument generation guided by plan structure
- Currently trying to extend method by updating local planning knowledge
- Also trying to extend method to planning with a defeasible planning theory
- Open problem: how to efficiently find plans that are possible using the combined knowledge of agents

Start again

- These examples show that it is useful to use planning representations for reasoning about interaction problems
- But are they interesting problems from a planning perspective?
- Want some more general AI problem that involves several agents and is genuinely novel
- Also should be more amenable to state-of-the-art planning techniques, not just use languages
- Are there any interesting general planning problems hidden in multiagent systems? How can we identify them?

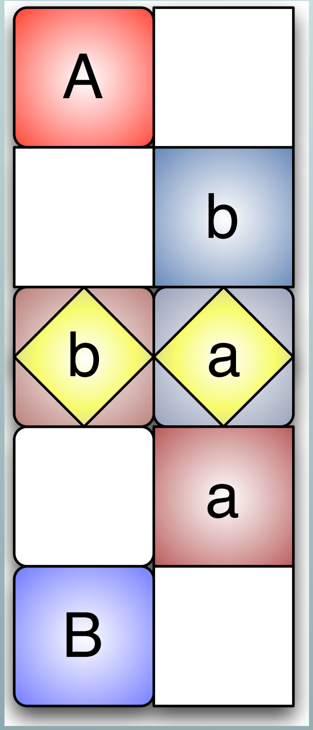
What is "multi" in multiagent planning (MAP)?

- Formulation
  - Multiple (conflicting?) goals, sometimes no to be shared
  - Multiple (conflicting?) views of the world (≈uncertainty)
  - Dependencies between actions, limited communication, distributed resources
- Synthesis
  - Negotiation in parallel with plan computation
  - Distributed computation, avoiding bottlenecks, interoperability
- Execution
  - Deception (due to asymmetry of knowledge, ≈uncertainty but continuous case interesting)
  - Distributed action and monitoring (basically like sensing)

Toward a more generic model

- Forget about execution for now, and focus on formulation/synthesis, and issue of multiple goals
- Need to address a number of issues to narrow down problem definition:
  - Q1: How to deal with (a)synchronous distributed execution?
  - Q2: What to assume about agent knowledge and rationality?
  - Q3: How to define what constitutes a solution?
- Start with minimal model: finite, fully observable, deterministic, static, no notion of time time, offline planning
- "Simple goals" assumption pointless in MAP, simplest model: goal utility=1, action cost=1

An example domain



- Parcel delivery in gridworld
  - Diamonds = parcels (source)
  - Uppercase = robots
  - Lowercase = depot (target)
- Actions: move, pickup, drop
- Action cost = 1
- Simplest interesting domain?
- Similar to TileWorld with added element of cooperation

We don't even have a solution concept for this simple example!

Bowling/Jensen/Veloso (IJCAI 2003)

- Synchronous action execution (cross-product of action sets)
- Initial state may be unknown to the agent (but taken from a fixed set)
- Define different types of solutions (*I* an initial state, *G* i's goal state, *S* any other state):
  - weak iff from any *I* we can reach *G*
  - strong cyclic iff *G* can be reached from any *S* that can be reached from any *I*
  - strong iff all paths from any *I* contain *G*
  - perfect iff all paths from any *I* reach *G* in a finite number of steps and maintain it forever after

Bowling/Jensen/Veloso (IJCAI 2003)

Equilibrium defined in terms of these types:

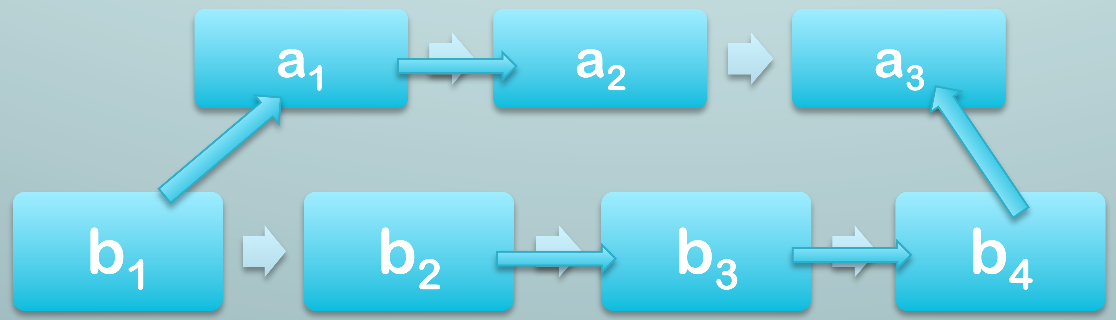
situation in which no agent can unilaterally switch to a better type of solution

Good: useful classification, clear equilibrium notion, basic algorithm

Bad: complexity (calculate all joint plans), equilibrium need not exist

Ben Larbi/Konieczny/Marquis (ECSQARU 2007)

- Consider shuffle sets to deal with issue of synchronisation



- Problem: agent cannot know whether deal will achieve its private goal
- Set of plans known, closed under sub-plans

Ben Larbi/Konieczny/Marquis (ECSQARU 2007)

Preference over shuffle set properties:

- **always satisfied**: all elements of the shuffle set achieve agent's goal
- **mutual interest**: one element of the shuffle set achieves everyone's goals
- **dependence**: agent can receive help that doesn't hurt the helper
- **antagonism:** agents cannot be jointly satisfied
- **always dissatisfied:** no element of the shuffle set achieves the agent's goal

Ben Larbi/Konieczny/Marquis (ECSQARU 2007)

Equilibrium defined in terms of these categories:

situation in which no agent can unilaterally switch to a better shuffle set category

- Good: useful categories, robust solution, basic solution algorithm
- Bad: complexity (calculate shuffle set), extreme categories not useful, conditions on ordering too extreme

Brafman & Domshlak (ICAPS 2008)

- Avoid problem of calculating all plans and then converting them to a game
- Synchronisation through time stamps
- Measures of coupling:
  - number of agents affected by own actions
  - number of necessary “interacting” actions
- Based on distinction between public and private actions
- Solution based on CSP over coordination points (constraints = valid subplan + coordination point preconditions met on time)

Brafman & Domshlak (ICAPS 2008)

Contributions: formulation as standard CSP, identification of degree of interaction as determining factor for complexity, not number of agents

Good: complexity results, distinction between public and private actions

Bad: Are most systems loosely coupled?

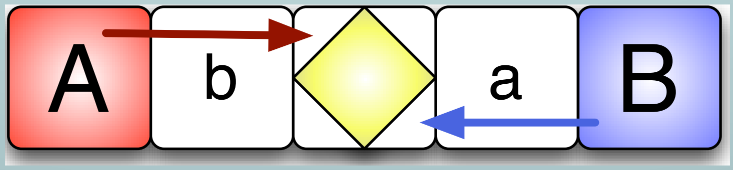
Only ones with heterogeneous agents.

Brafman/Domshlak/Engel/ Tennenholtz (IJCAI 2009)

- Introduce notion of coalition-planning game (reward for goal, cost for plan, no action = 0)
- Solution stable if no set of agents can increase utility by jointly adopting other plan
- Formally: plan π *stable* for iff no plan π exists for any subset Φ' of agents Φ such that $u_\phi(\pi') > u_\phi(\pi)$ for all ϕ in Φ'
- While conceptually clear, problematic in practice

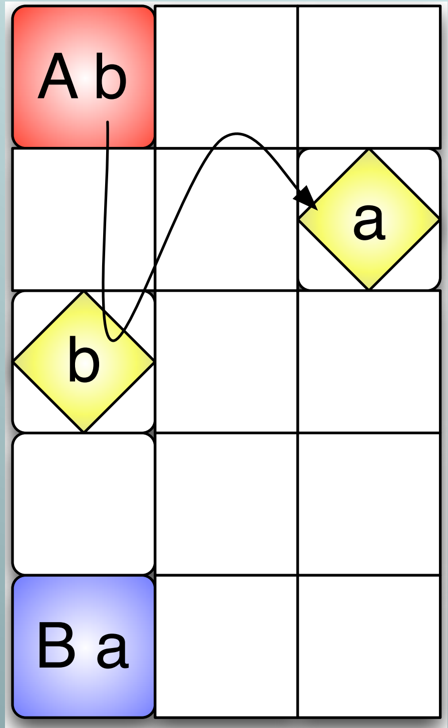
Brafman/Domshlak/Engel/ Tennenholtz (IJCAI 2009)

- Any agent not changing to a different plan is assumed to perform no actions



- Agents best off helping each other
- If A deviates and does nothing instead, B won't do anything either
- Assumption makes sense as otherwise agents could provide help for no reason

Brafman/Domshlak/Engel/ Tennenholtz (IJCAI 2009)



- But here is a situation where non-deviating agents have a reason to help
- A makes a detour to deliver B's parcel, while B delivers A's
- Stable solution although shorter path from b to a exists, because not all agents would benefit from deviation

Brafman/Domshlak/Engel/ Tennenholtz (IJCAI 2009)

- Give an algorithm for finding a stable strategy
- Based on iteratively eliminating strategies in an interaction graph
- Works only in acyclic interaction graph, very limiting assumption
- So not even two agents can mutually depend on each other, let alone cycles with more than two agents
- We are currently working on modifying solution concept to include concept of “not destroying preconditions of other agents”
- Also enumeration of strategies not realistic, have to look at heuristics/approximations

Interesting problems

Three general problems seem interesting:

- How to compute acceptable plan given a solution criterion (in particular adapting existing planning heuristics)
- How to search plan space incrementally for generating proposals during negotiation
- How to use background knowledge to guide plan recognition and optimal response generation

Evaluation

- No good benchmarks for MAP exist because research is fragmented
- Too many different potential problems to be accommodated
- Single-agent planning benchmarks can be adapted but is this useful?
- Multiagent systems people also interested a lot in continuous planning
- But performance metrics domain-dependent in this case

Integrated applications?

- Dialogue planning metaphor covers synthesis, negotiation, and execution aspect
  - If communication actions are interpreted in a planning-based way, we should be able to plan them just like physical actions
  - But hard to decide about communication strategy before having synthesised collaborative plans
  - Actions planned for deception detection ahead of execution may affect suggested deals

Examples

BUYER-SELLER

B: I would like an art history book.

S: Good art history books range from $35-$55.

B: I would like something cheaper.

S: There's "Art for Kids" at $15.

B: I want a book for adults.

S: There's "Art History for Dummies" at $25.

B: Great, I'll take that.

(execution follows, including payment, delivery, etc)

PEER-TO-PEER

P: I'd like to stream a music concert in high quality tomorrow night.

Q: Who will be performing?

P: It's a "best-of" transmission from a festival.

Q: I don't like watching concerts unless I know what bands are playing.

P: Could I still borrow your bandwidth?

Q: OK, if you grant me prioritised access to yours for seven days after that.

(*execution follows, including settings to preference in P2P system, actual streaming actions, etc)*

Conclusions

- Reasoning about interaction crucial to multiagent systems
- Must involve planning one way or another, but no standard simple frameworks for MAP
- Some of our own work shows that planning formalisms are useful
- To develop more generic problems need convincing, simple examples
- Looking at multiple goals is (in my opinion) the strongest thing that multiagent perspective can add to single-agent planning
- Current solution concept proposals lead aoverly complex, more approximate methods needed

Thank you. Questions?