Research Review of Distributed Multi-Agent Planning

By Mohamed Bakr Ali

Introduction:

Having greatly enjoyed working with distributed platforms (Spark, Hadoop, etc.) in the past, for this research review, we chose to focus on the challenges of **distributed multi-agent planning (DMAP)** techniques. To guide us through this quick exploration, we relied in large parts on the work of B.K. Durkota as presented in [1].

DMAP Challenges:

The challenges of DMAP are many. If a particular agent is unable to solve a problem on its own and has its own actions, **how should its actions and the actions of the other agents be interleaved**? Coordination implies communication. In order not to incur a huge network overhead and slowing down the agents too much, **what is the best paradigm to use for coordination**?

DMAP Approaches:

- Distributed Constraint Satisfaction Problem solving for coordination of the agents and individual planning using local search [2],

- Multi-agent adaptation of A* with local heuristics [3],

- Distribution of the GraphPlan approach based on merging of planning graphs [4].

In [2], Nissim et al use the **DisCSP+Planning** algorithm to solve the multi-agent planning problem by separating the **public** and **private** aspects of the problems. In this paradigm, the public aspects (coordination between agents) is dealt with by the **coordination component** using a CSP searching for a sequence of **interaction points** between the agents and enforcing consistency requirements between actions. The local, internal aspect is dealt with by the **individual planning component** using a planner that handles the other types of constraints and encodes the local parts of the plan.

In [3], the authors based their work on the A* algorithm described in class. Similarly to regular A*, the **Multi-agent Distributed A* (MA-A*)** algorithm maintains open lists of unvisited states and closed lists of already visited states for all agents. Individual agents use local, potentially different, heuristics to decide which unvisited state should expand next. As in [2], actions must first be separated into public and individual actions. Agents send messages to each other to distribute the search at interaction points where the other agents can follow.

In [4], the authors use a distributed version of the planning graph data structure. They use the **Distributed Planning through Graph Merging (DPGM)** algorithm which first performs **global goal decomposition** (where each agent creates an individual goal). Then, it alternates **expansion** (where each agent builds new layers in their planning graphs) and **planning graph merging** (where agents share their actions until each of them reaches their individual goals). In the **individual plan extraction** phase, each agent extracts plans from its planning graph. The process ends with a **coordination** phase that yields a coordinated individual solution plan.

Results:

In [1], experimental results show DPGM to be efficient in domains which are not tightly coupled ("combinatorically easy"). The DisCSP+Planning algorithm is shown to be efficient in problems which are combinatorically hard from the perspective of individual planning. Finally, the implementation of MA-A* with set additive heuristics was shown to be most effective in highly coupled domains.

References:

[1] *Comparison of Deterministic Distributed and Multi-Agent Planning Techniques* (2013), by B.K. Durkota

[2] *A general, fully distributed multi-agent planning algorithm* (2010), by R. Nissim et al

[3] *Multi-Agent A* for Parallel and Distributed Systems* (2012), by R. Nissim et al

[4] *Distributed planning through graph merging* (2010), by D. Pellier