AI Engineer Nanodegree
Udacity

July 19, 2017

Research Review

Distributed Multi Agent Planning (DMAP)

- **What is DMAP**: DMAP is abbreviation of Distributed Multi Agent Planning. There are several distributed computing platforms like Spark, Hadoop, etc. in the market. 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 Challanges**:
  There are many challenges for DMAP. The main challange is Communication between multiple agent through Coordination. 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? 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**:
  As listed in [1], three recent approaches include:

  - Distributed Constraint Satisfaction Problem solving for coordination of the agents and individual planning using local search [2],
  - Multiagent adaptation of A* with local heuristics [3],
  - Distribution of the GraphPlan approach based on merging of planning graphs [4].

- **DMAP with DisCSP+Planning**:

  In [2], Nissim et al described fully distributed multi-agent planning algorithm. Their methodology uses distributed constraint satisfaction to coordinate between agents, and local planning to ensure the consistency of these coordination points. To solve the distributed CSP efficiently, they modified the existing methods to take advantage of the structure of the underlying planning problem. In multi-agent planning domains with limited agent interaction, their algorithm empirically shows scalability beyond state of the art centralized solvers. Their work also provides a novel, real-world setting for testing and evaluating distributed

constraint satisfaction algorithms in structured domains and illustrates how existing techniques can be altered to address such structure.

To solve the multiagent planning problem, they separated 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.

- **DMAP with Multiagent Distributed A* (MAA*)**:

  In [3], the authors based their work to adapt A* to the multiagent setting, focusing on multi-agent planning problems. They provided a simple formulation of multi-agent A*, with a parallel and distributed variant. Their algorithms exploit the structure of multi-agent problems to not only distribute the work efficiently among different agents, but also to remove symmetries and reduce the overall workload. Given a multi-agent planning problem in which agents are not tightly coupled, their parallel version of A* leads to super-linear speedup, solving benchmark problems that have not been solved before. In its distributed version, the algorithm ensures that private information is not shared among agents, yet computation is still efficient sometimes even more than centralized search despite the fact that each agent has access to partial information only.

  Similarly to regular A*, the Multiagent Distributed A* (MAA*) 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.

- **DMAP with Distributed Planning through Graph Merging (DPGM)**:
  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 MAA* with set additive heuristics was shown to be most effective in highly coupled domains.

References:

1. Comparison of Deterministic Distributed and MultiAgent Planning Techniques (2013), by B.K. Durkota

2. A general, fully distributed multiagent planning algorithm (2010), by R. Nissim et al

3. MultiAgent A* for Parallel and Distributed Systems (2012), by R. Nissim et al

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