Divide and Conquer in Multi-agent Planning

Eithan Ephrati & Jeffrey S. Rosenschein
1994

Speaker: Denis Sheynikhovich
November 28, 2002
- Planning problem
- Simple example
- Multi-agent planning algorithm
- Planning and execution
- Advantages & Drawbacks
- Conclusion
Let $d$ be the depth of problem

Let $b$ denote the branching factor

Time complexity of the problem is $O(b^d)$

If we add to the problem $n$ agents that carry out any operation on each step (with different costs) then for central planner the problem will be $O(nb^d)$

If the problem can be divided into $n$ independent sub-goals $g_1...g_n$ and agents can plan in parallel, the problem complexity can be reduced to $\max_i (b_i^{d_i})$ where $b_i = b/n$, $d_i = d/n$
Example: blocks world

Relations:

- \( C(x) \) - block/position \( x \) is clear (no block on it)
- \( O(x, y, V/H) \) - block \( x \) is on \( y \) (Vertically/Horizontally)
- \( A(x, pos) \) - left edge of \( x \) (agent or block) at position \( pos \)

Operators:

- \( T(b, x, y) \) – take block \( b \) from \( x \) to \( y \)
- \( R(b) \) – rotate block \( b \)
- \( M(x, y) \) - agent moves from \( x \) to \( y \)
Algorithm (1/2)

Each agent is assigned *a priori* defined sub-goal

Each sub-goal is simply a desirable world state (from the local point of view of an agent)

Agent generates sub-plans that achieve the sub-goals

Several sub-plans can be necessary to ensure that all sub-plans can be merged into one global plan.

\[ G = g_1 \cup g_2 \cup g_3 \]

\[ g_1 = \{ A(C, 1), O(C, 1, V), C(C) \} \]

\[ g_2 = \{ A(B, 3), O(B, 3, V), C(B) \} \]

\[ g_3 = \{ A(D, 1), O(D, C, H) \} \]
Algorithm (2/2)

- Each plan is an ordered set of propositions that describe how the sub-goal can be achieved (each proposition can be executed by different operations with different costs)
  - For example, plan can be
    \[
    \{ A(a[i],2) \} \quad \{ C(C) \} \quad \{ O(C,1,V) \}
    \]

- On each step of the algorithm each agent give a subset of their propositions (subset is identified by one operation, executed on the current state)

- All proposition subsets construct aggregated set

- Evaluation function \( f' \) is calculated for each possible execution of the aggregated set.
  - \( g \) is a cost for already constructed plan.
  - \( h'[i] \) are costs for each \( a[i] \) of executing remaining sub-plans

- Actual sequence of operators is chosen based on the minimal cost

\[
\begin{align*}
  f' &= g + h' \\
  h' &= \Sigma h'[i]
\end{align*}
\]
Back to example

Step 1

Step 2

Step 3

Step 4
Since the algorithm is iterative, it is possible to interleave planning and execution (there is a look-ahead factor $L$ that defines the number of planning steps before the actual execution).

In case of dynamical changing of the goal, $L$ last steps can be revised to satisfy the new goal.

The new goal must be divided into sub-goals. Some sub-goals may remain the same, other may change. Agents with changed sub-goals must revise the plan, while others should not.
Advantages & Drawbacks

- No central planner is required
- Agents work in parallel reducing complexity and planning time
- The algorithm guarantees cost-optimized global plan
- Iterative process allows interleaving execution and planning
- Algorithm is suitable for dynamically changed global goal
- The algorithm requires *a priori* division on sub-goals
- It is assumed that each agent knows how many plans it has to generate to ensure proper merging
- The evaluation function $f'$ can be overestimate or underestimate. Sometimes it will require additional iterations to ensure the plan is optimal
- There is no description of tasks for which this algorithm is not applicable
Conclusions

- Proposed multi-agent algorithm significantly reduces the planning time if the global goal can be divided into independent sub-goals.

- The algorithm iteratively finds optimal plan (optimal in the sense of execution cost).

- The algorithm allows interleaving planning and execution, making it useful for application where execution is urgent or the global goal can change.

- *A priori* division on sub-goals is required.