ABSTRACT

In multiagent planning, it is often convenient to view a problem as two subproblems: agent local planning and coordination. Thus, we can classify agent activities into two categories: agent local problem solving activities and coordination activities, with each category of activities addressing the corresponding subproblem. However, recent mathematical models, such as decentralized Markov decision processes (DEC-MDP) and partially observable Markov decision processes (DEC-POMDP), view the problem as a single decision process and do not make the distinctions between agent local planning and coordination. In this paper, we present a synergistic representation that brings these two views together, and show that these two views are equivalent. Under this representation, traditional plan coordination mechanisms can be conveniently modeled and interpreted as approximation methods for solving the decision processes.

General Terms

Algorithms, Theory

Keywords

Multiagent Systems; Coordination, cooperation, and teamwork; Multiagent planning

1. INTRODUCTION

Decision theoretic studies of multiagent problem solving have recently become one of the most active fields in multiagent systems research. Most of the mathematical models proposed in this undertaking are based on some decentralized extensions to Markov decision processes (MDPs) or partially observable Markov decision processes (POMDPs). Multiagent decision processes, such as decentralized Markov decision processes (DEC-MDP) and decentralized partially observable Markov decision processes (DEC-POMDP) [12, 2, 8], have established a quantified framework for multiagent problem solving.

One problem here is that, these decision models are quite complex to solve. In general, problems specified in these models are in the NEXP-complete class [7]. Thus, developing approximation methods and heuristics has become an important direction, and so is the task of finding subclasses of problems that are of lower complexity levels. For example, one important subclass of DEC-MDP/POMDP is known as transition independent (TI) DEC-MDP [1] which made an early attempt to exploit the structural pattern in the state space and in agent interactions.

For complex problems, the decision process would be very complex. When the size of the system scales up, or when the system becomes open and dynamic, the approach that relies on solving the system-wide decision process altogether would have difficulty to scale up or adapt. In reality, for a complex system, it would be difficult to require all agents to know the overall decision problem, let alone solve it once-for-all. It is essential for each agent to retain some flexibility and autonomy in practical decentralized systems. In this regard, this approach of directly applying the decentralized decision models is somewhat in contrary to the concept of a truly decentralized system, in which the agents are considered autonomous rather than bound to a policy. Thus, it is critical for us to be able to maintain an agent-centric view rather than taking a process-centric view when studying decentralized models. In this paper, we will address this issue and provide a linkage between decentralized decision processes and the notion of agency.

An agent-centric model, such as the well-established BDI (Belief-Desire-Intention) architecture [6, 11] for agent design, represents agent plans and goals and reasons about agent interactions. Although not as quantified as decentralized decision models, it does allow quantitative modeling the agent’s plans and helps understanding multiagent cooperation via joint goals. In particular, the structural representation of agent plans via hierarchical task networks (HTNs) [9, 5] provides a compact yet powerful way to decompose the task into smaller tasks and specify constraints among the tasks.

TAEMS - a framework for task environment modeling [4], can be regarded as a much richer extension to HTN that can be used to represent multiagent tasks. TAEMS is able to represent complex interrelationships, sophisticated utility structures, and the environment that incorporates uncertain outcomes such as resulting quality, cost, and duration.

Under these representations, agent problem solving activities can be viewed as two intertwined processes: agent local problem solving, which strives to choose local actions that achieve the best utility; and coordination, which ensures that the activities in different agents are in harmony. This dichotomy facilitates the development of agent planners such as Design-to-Criteria [10], and more importantly, plan coordination mechanisms such as Generalized Partial
Global Planning [3]. The interaction between the local planning process and the coordination process can be summarized as the following: on one hand, the local planning process produces the local plan, which forms the basis for any pledges offered by the agent; on the other hand, the coordination process provides the context for local planning in a multiagent setting. The set of local solutions is now the equivalent of the set of local policies in the once-for-all model, but solving it involves only the local view, not the global view. This leads to reduction of problem size and complexity, as well as better autonomy and scalability. Of course, in practice, the tasks in different agents’ local views may be inter-related; therefore the agents generally cannot solve the planning problems associated with the local views independently.

If we can exploit the relatively small representations of the decision problem as perceived in each agent, we may not need to construct an exhaustive model for the overall decision problem and solve it once-for-all, but can hide from other agents the details that are pertinent only to one agent. This way, each agent would then have its own understanding of the overall problem solving process - the (somewhat expanded) local view, expressed in terms of a localized and simplified version of the decision problem.

In this work, we try to attack this problem by proposing an alternative framework to multiagent decision processes that establishes a connection between the multiagent decision process model and the agent-centric model. The key to our approach is to construct the corresponding local view of the overall decision process. This would allow us to re-introduce the notion of local problem solving and coordination in the decision process, to avoid the need to construct a complete decentralized decision model, and to be able to model the coordination mechanisms under this framework.

2. AN ALTERNATIVE MODEL

A decentralize decision process, such as DEC-MDP/POMDP, cannot be easily decomposed into a set of local decision processes (i.e. local MDP/POMDPs), because of the following three difficulties: (1) local information alone is insufficient for an agent to decide its actions, (2) an agent not only need information from others, but also need to inform others about its own action outcomes, and (3) the utility function is based on global states rather than local states, therefore cannot be easily decomposed into local functions.

If we assume factorized state space, our first attempt is to construct a model \((S, A, P)\), where \(S\) is the local state space, \(A\) is the set of local actions, and \(P\) the transition probability \(p(s'|s,a)\). This model is Markovian (memoryless), since the transitions are based only on the current state, not the entire history. However, in general, transition probability may also depend on some non-local information. One example would be an enables relationship defined in TAEMS, which represents a precedence relationship between tasks, i.e. task A may not start before task B finishes successfully. Thus, in general the transition probability may also include non-local information (nsls), so it may be represented this way: \(p(s'|s,a,nsls)\).

Note that this means that our model may be non-transition-independent, as defined in [7]. It also indicates that the model may also be observation independent, for example, when task B finishes, another agent who wants to perform task A would be able to notice that the enables relationship has materialized.

Thus, in order to address the first difficulty listed above, we need to also create representations of the non-local information in our framework. Because these information are based on other agents’ activities, their actual value changes in parallel to the local action process. Thus, we can represent these changes as a parallel process for each piece of nonlocal information. Since those information are not due to local actions (and the agent cares only about the actual information, not the action in other agents that causes the changes), the process can be modeled as a (stochastic) finite state machine, and as time evolves, the information transitions into another state.

For example, for an enables relationship, according to the enabling agent’s local process, the enables relationship has 20% chance of being materialized at time 10, and 40% chance of being materialized at time 15. Thus, if we use time as the X axis, and the percentage of chance of enablement as y axis, we will see a diagram like Figure 1.

![Figure 1: Enablement profile](image)

The set of nsls and their finite state machines together represents the information external to the agent that may be used in the agent’s local policy. Note that although both the state machines and local transitions are memoryless, this does not necessary mean that the agent’s local decision process would be Markovian - on the contrary, it is entirely possible for the agent to determine its policy based on past and current information contents. But, since the nsls are all information this agent cares about, it means that all other information could be hidden from this agent, therefore creates a far smaller local view than the whole decentralized decision process.

We call the set of nsls and their state machines an input profile for this agent, since it represents all inputs from other agents as stochastic processes, i.e. captures the dynamics of other agents for this agent. With an input profile, we addressed the first difficulty point above.

Similarly, for the second difficulty point above, we propose a structure called output profiles, which encodes the information this agent would transmit to other agents (to-share information), or to si, as stochastic processes. The relationship between input profiles and output profiles are obvious: information contained in the input profiles should all come from the output profiles.

The third difficulty point would be easier to solve if the global utility can be conveniently decomposed into a function of individual agents’ local utilities, which in turn would be based on local action outcomes as well as the actual nsls. Although these may well be the case for most problems (for example, most task-based problems), in general this may not be always easy to do. For the purpose of this model, this is not a problem, however: one can design such a scheme where all except one agent receives 0 utility for all activities it performs, and the remaining agent receives the global utility. Obviously, this requires that the remaining agent obtains all non-local information (hence an input profile of huge size) that may be used for the purpose of deriving the global utility, which defeats the purpose of creating a much smaller local view for all the agents.

One important property of the profiles is that the profiles have partial ordering: if one set of profiles matches or exceeds the value distribution of the same functions in another set, then we can say that one profile set dominates another set (notice that the function in the distributions (as illustrated earlier) are all non-decreasing functions).
Thus, for a given (complete) set of input profiles, if we compute a local policy for each agent based on this input profile, and then calculate the output profiles, and find out that the output profiles dominates the input profiles, then we can say that the local policies and the input profile are consistent, because all input profiles are matched or exceeded by output profiles. Thus, the “actual” input profile (i.e. the computed output profile) is stronger (dominates) the current input profile. Therefore, the actual expected rewards of the local policies (based on the “actual” input profile) will be at least as good as the expected values when computed with the stated input profile. Thus, we have the following lemma:

**Lemma 1** Assuming that the quality accumulation functions in the task network are monotonic, then if the set of local policies is consistent with the set of input profiles, then the expected reward of local policies calculated locally by each agent are underestimates of the true expected reward value of the local policies.

The input policy combined with the agent’s local policy may be viewed as a solution for the decision process represented by the agent’s local view, and the output profile may be viewed as the goal function when coordinating with other agents. The problem now is to ensure that the policies of all agents are consistent with the input/output profiles, and in the meanwhile achieve the best system reward.

**Theorem 1** Finding the optimal consistent input profiles is equivalent, i.e. has the same complexity, to solving the overall decision problem for the task network.

**Proof Sketch:** if we have the optimal input profiles, which means we can easily use a local decision process to find the optimal local policies, which involves solving a single agent MDP/POMDP and is of a lower complexity class than solving the overall DEC-MDP/POMDP. Conversely, if we have an algorithm to find the optimal local policies, then it is straightforward to compute the output profiles - the values of the variables in the output profile are updated during the dynamic programming process of calculating the quality values for those task nodes, which is a process taking time polynomial to the number for nodes in the task network. Obviously, this output profiles is the optimal input profiles we are seeking (which means, in this case, the input profile and the output profile should be a fixed point.)

3. **SUMMARY AND FUTURE WORK**

The above result means that our alternative model is essentially equivalent to the decision process, and as expected, there is no easy way to find the optimal input profiles since the complexity level is the same. However, the separation of input profiles and the local decision process allows explicit reasoning of at goal level, rather than having to directly deal with flat state spaces. For this, we propose an iterative process that adjusts the input profiles dynamically. This corresponds to the coordination process: from a set of input profiles, we can calculate the local policy independently, then we can calculate the output profiles, and the output profiles may be used to update the input profiles. This process is completely decentralized and can happen in parallel. Considering that the local views are small, this method can be very flexible and much more scalable than trying to find the overall decision model and solve it once-for-all.

Because the profiles describe only the directly relevant information that may impact the agent’s local policy, and therefore avoid the combinatorial explosion when other agents’ policies have to be considered in order to reason about the performance of a local policy, this framework is much more scalable in system design and is able to deal with open and dynamic systems, and better reflects the nature of decentralization, where agents usually have only partial views of the overall problem. In addition, hierarchical structure of the problem is preserved when trying to solve the decision processes associated with the local views.

Overall, by showing the equivalence of our model and the decentralized decision process models, we provided a bridge that connects the state-space representation with a more agent-centric representation of multiagent cooperation problems. For traditional multiagent planning problems that are based task network representations, our model provides a way to map this representations into the decision process framework, yet at the same time preserved the local planning/coordination dichotomy. Thus, our model paves the way for better integration of techniques developed in both MDP worlds and task network worlds. In particular, the role of coordination can be described in terms of determining the consistent input/output profiles. We believe that this work is an initial step toward unifying these two representations - an important future direction for multiagent planning.

4. **REFERENCES**


