Real-Time Multi-Agent Planning and Scheduling in Dynamic Uncertain Domains *

Rajiv T. Maheswaran, Craig M. Rogers, Romeo Sanchez and Pedro Szekely
University of Southern California - Information Sciences Institute
4676 Admiralty Way #1001, Marina Del Rey, CA, USA

Abstract

Creating decision support systems to help people coordinate in the real world is difficult because it requires simultaneously addressing planning, scheduling, uncertainty and distribution. Generic AI approaches produce inadequate solutions because they cannot leverage the structure of domains and the intuition that end-users have for solving particular problem instances. We present a general approach where end-users can encode their intuition as guidance enabling the system to decompose large distributed problems into simpler problems that can be solved by traditional centralized AI techniques. Evaluations in field exercises with real users show that teams assisted by our multi-agent decision-support system outperform teams coordinating using radios.

Introduction

Teams of people need to coordinate in real-time in many dynamic and uncertain domains. Examples include disaster rescue, hospital triage, and military operations. It is possible to develop a plan *a priori* for these domains, but many parts must be left unspecified because people won't know exactly what needs to be done until they are executing the plan in the field. Additionally, requirements and tasks can evolve during execution.

Our work addresses a fundamental multi-agent systems endeavor of creating decision support systems that help humans perform better in real-time dynamic and uncertain domains. The technical challenges to compute good solutions for such domains have been well documented (Murphy 2004; Groen et al. 2007; Boutilier 1999). There are two main contributions in this paper: (1) we present a *generic methodology* for human guidance for planning and scheduling activities, and (2) we discuss an *extensive investigation* as to its usefulness in a thorough field exercise conducted by a third party.

In practice, it is possible to address specific domains with custom algorithms that use powerful heuristics to leverage the structures unique to that domain. These solutions are expensive to create as even these domains involve planning, uncertainty and distribution. The goal remains to develop *generic* approaches that produce good solutions that help human teams in many domains.

We introduce a new approach, STaC, based on defining collections of **S**ubteams each with **Ta**sks to perform and **C**onstraints on how they should be performed. The premise that people have good intuitions about how to solve problems in each domain and this approach both matches this intuition and can be matched to generic models of task allocation problems. The idea is to enable users to encode this intuition as guidance for the system and to use this guidance to vastly simplify the problems that the system needs to address.

The key to STaC is using the model and guidance to produce sufficiently smaller task structures that can be centralized so that a single agent can determine who does what, when and where with respect to these significantly simpler task structures. This mitigates the distribution challenge and enables using auxiliary solvers based on established AI techniques which produce good solutions at a smaller scale. These smaller task structures are solved independently assuming that the human guidance has addressed any significant dependencies. While this may not be the case in all domains, in many scenarios including ours, humans are far better at identifying effective structural decompositions than automated techniques.

STaC addresses tracking the dynamism in these task structures, the transitioning of agents assignment between these smaller task structures and the invocation of auxiliary solvers. Given that the task structures are treated independently and sufficiently small to be centralized, we call them sandbox reasoners. The sandbox reasoners required in each domain are different, so custom code must be written for each domain. However, the benefit of the approach is that sandbox reasoners are significantly simpler than the custom solvers required to produce a custom solution for a domain.

The paper is organized as follows. The next sections introduces the real-world domain where our approach was tested

*The work presented here is funded by the DARPA COORDINATORS Program under contract FA8750-05-C-0032. The U.S.Government is authorized to reproduce and distribute reports for Governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of any of the above organizations or any person connected with them. Approved for Public Release, Distribution Unlimited.

Figure 1: Field Exercise Images from Rome, New York, USA

followed by related work. We then describe the details of the STaC approach and the particular sandbox reasoners used in our example domain. We close with evaluation results, conclusions and directions for future work.

Field Exercises

The field exercises were based on a simulated disaster rescue domain. The challenge was to show that a human-team supported by intelligent agents could outperform a human team operating by themselves. The first two exercises were held in the city of Rome, New York, USA, and the second three were in Stanton Wood Park in Herndon, Virginia, USA. Images of the field exercise in Rome are shown in Figure 1 and a map of the sites and road network of Stanton Wood Park are shown in Figure 2. They were organized and evaluated by independent parties contracted by the DARPA (Defense Advanced Research Projects Agency) Coordinators program. The rules of the field exercise were created collaboratively by the teams building coordinator agents, the independent evaluation team, and subject matter experts. The specific instances or *scenarios* that comprised the test problems were chosen by the independent evaluation team.

Various locations were selected as *sites* and a feasible road network was constructed. If the site was *populated*, it could have injured people in either *critical* and *serious* condition. Populated sites would also have gas, power and water substations which may have been damaged. In addition, any site could have *facilities* such as a *hospital*, *clinic*, *warehouse*, *gas main station*, *power main station* and *water main station*. A team would obtain points by rescuing injured to hospitals or operational clinics (before a deadline associated with each injured person) and by repairing main stations and substations. The goal of a scenario was to accumulate as many points as possible before the scenario deadline.

The teams were composed of 8 field agents and 2 command agents. Each agent had a different set of skills. Three *specialists* in *gas*, *power* and *water* could perform *major* and *minor* repairs in their respective skill area. The *medical specialist* could load any type of injured person by themselves. The remaining four *survey specialists* could have any collection of skills involving minor repairs. The field agents could move throughout the field exercise area and perform actions. The command agents were located at a base where they helped to coordinate the activities of the team. The *Radio Team* communicated only with radios. Our *CSC Team* had ruggedized tablet computers on which our agents were loaded, in addition to radios. The tablets had cell modems and GPS.

Many outcomes were revealed during the game for which little or no likelihood information was given *a priori*, i.e., no probability distribution functions over outcomes. Teams did know the space of possible outcomes beforehand. A *survey for damage* at a main station or substation revealed the number and type of problems chosen from a set of known possible problems. A *survey for injured* at a populated site revealed the number, types and deadlines for the injured at that site. As the result of a survey, any team member might be injured, forcing them to go to an operational medical facility to recover before proceeding with any other action. A survey could also reveal that the vehicle of the agent doing the survey had failed and would require a vehicle repair before the agent could travel to any other site. While traveling, agents could encounter*road blocks* which could not be passed until fixed. Travel and repair times could vary and repairs could fail. These dynamic and uncertain events were planned parts of the exercise. In addition, the teams had to address uncertainties inherent in the environment, such as noisy radios, weather, and other activities in the public settings. Furthermore, most of these outcomes were only observable by the agent encountering the outcome.

The independent evaluation team chose the scenario from the space of possible exercises and informed the teams of the details below one day prior to the test: (1) the locations of populated sites and facilities, (2) the road network and ranges on travel times between sites, (3) a range for the total number of injured at each site, (4) the points for rescuing each type of injured, which could vary by type and site, (5) the points for repairing each substation or main station, which could vary by type and site, (6) potential problems after surveys for damage and corresponding repair options, (7) ranges on repair times, (8) likelihoods of failure for every repair activity, and (9) the skills of the survey specialist agents. The deadlines (for the scenario and injured) did not allow teams to do all possible repairs and rescues. The teams had one day to form a high-level strategy. The only element of uncertainty which could be modeled accurately with a probability density function was (8). When a team member completed a repair activity, they would call the evaluation team, which would report whether the repair was successful or a failure. The range in (3) was respected by the scenario designers, i.e., the number of injured did not fall outside the given range.

There were many rules and couplings that forced agents to coordinate. To do surveys, gas and power substations at the site had to be off, which required agents with those skills. Two agents had to be at the same location simultaneously to load a critically injured person or repair a road block. Repair options could involve multiple tasks and require two agents with certain skills to act in synchrony or in a particular sequence. Some repair options required kits

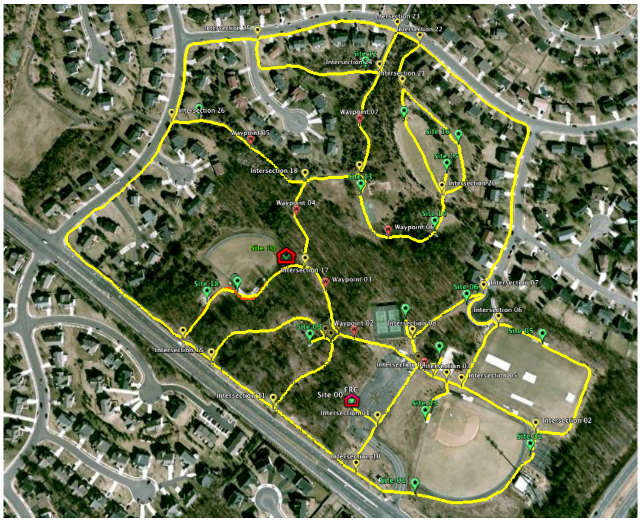

Figure 2: Stanton Woods Park, Herndon, Virginia, USA

which guaranteed their success, but kits were available only at warehouses. Agents could transport at most one entity, i.e, either a repair kit or a single casualty. A substation was considered repaired only if the corresponding main station was also repaired. A clinic was not operational until all substations at the site and all corresponding main stations were repaired. These are examples of rules that, along with the dynamism and uncertainty in outcomes mentioned earlier, created challenging real-time real-world distributed coordination problems.

The goal was to see if humans operating with radios and a multi-agent decision-support system could outperform humans operating with only radios. While some aspects of a real-world disaster scenario were abstracted, we believe the field exercises closely approximated the challenges of helping a human team solve difficult real-world problems.

Related Work

The STaC framework was developed during the DARPA Coordinators program. In the first two years, DARPA ran competitive evaluations on simulated scenarios, and CSC (Criticality-Sensitive Coordination), the underlying system behind the STaC framework, won such evaluations by considerable margins against two competing approaches based on Markov-Decision-Processes (MDPs) (Musliner et al. 2006) and Simple Temporal Networks (STNs) (Smith et al. 2007).

The MDP-based (Musliner et al. 2006) approach addressed the infeasibility of reasoning over the joint state space by setting the circumstance set to a subset of local state space that is reachable from the current local state, unrolling the state space by doing a greedy estimation of boundary values. It biased its local reward function on the commitments made by the agents during execution. However, such approximations lose critical information, exploring state spaces that are far from good distributed solutions.

The STN framework (Smith et al. 2007) addressed temporal uncertainty by using a time interval (instead of a point) as the circumstance that denoted feasible start times for a method to be executed. The system used *constraint propagation* to update the start intervals of the agents' activities during execution. A policy modification phase was triggered if execution was forced outside the given set of intervals. One of the problems of this approach is that agents tried to maintain consistency and optimize their local schedules, losing information that was needed to timely trigger policy modifications for their schedules.

We encoded scenarios of the field exercise as planning problems using PDDL (Planning Domain Definition Language) (Fox and Long 2006). The motivation was to identify to the extent to which current automated planning technology can address complex distributed, resource-driven, and uncertain domains. Unfortunately, this proved to be extremely difficult for state-of-the-art planning systems. From the set of planning systems tried, only LPG-TD (Gerevini et al. 2005), and SGPLAN (Chen, Wah, and Hsu 2006) solved a few simplified problems, after uncertainty, dynamism, non-determinism, resource-metrics, partial observability and deadlines were removed. Planners were unable to scale to more than 5 sites. LPG-TD produced solutions more efficiently but less optimally.

In general, mixed-initiative approaches where humans and software collaborate can often produce better solutions for complex problems. Mixed-initiative planning systems have been developed where users and software interact to construct plans. Users manipulate plan activities by removing or adding them during execution while minimizing the changes from a reference schedule (Ai-Chang et al. 2004; Hayes, Larson, and Ravinder 2005; Myers et al. 2003). Most of these systems are centralized, so humans and systems are fully aware of the entire plan, and of the consequences of updating it. In our scenario, agents (including humans) have subjective views of the world, and any decision may trigger many unknown global effects.

Multi-agent systems for disaster domains have been studied in the context of adjustable autonomy. The idea is to improve limited human situational awareness that reduces human effectiveness in directing agent teams by providing the flexibility to allow for multiple strategies to be applied. A software prototype, DEFACTO, was presented and tested on a simulated environment under some simplifications (e.g., no bandwidth limitations, reliable communications, omnipresence) (Schurr et al. 2005).

Conclusions and Future Work

Our 18-month experience working on a system to compete against radio teams in the field exercises provided evidence for the benefits of our approach. Our starting point was our generic CSC system developed during the previous two years to solve generic, synthetically generated problem instances specified in CTAEMS. Even though the synthetically generated problem instances were generated according to templates that combined "typical" coordination situations, the resulting problems were not understandable by humans. In contrast, the field exercise problems are more natural, and appeal to our lifetime of experience coordinating every day activities. Intuitions about space, distance, time, importance and risk all came into play, enabling teams of humans to devise a sophisticated strategy with a few hours of brainstorming. It became obvious early on that the generic CSC system would not be able to produce solutions comparable to the desired sophisticated, coordinated behavior of human-produced strategies.

Our existing system had performed extremely well in Phase 2 by using our Predictability and Criticality Metrics (PCM) approach. In the PCM approach, the policy modifications that agents consider are limited to those that can be evaluated accurately through criticality metrics that capture global information. These policy modifications were simple and thus the reasoners that implemented them were simple too.

For the field exercises, we extended our approach so that policy modifications would be constrained using the guidance provided by the users. This guidance was in the form of a sequence of sites to visit. The system was left to make decisions that we believed it could evaluate accurately (e.g., how to perform repairs or rescue injured at a single site). The system relied on the TCR-set criticality metric to determine how to move agents along the list of guidance elements. The approach worked well. Our users outperformed the radio team because they were able to communicate their strategy to their agents, and the system optimized the execution of the strategy, adapting it to the dynamics of the environment.

The field exercises in Rome used a simpler language for specifying guidance. It had a single guidance group consisting of the entire set of agents. Also, it did not support constraints to control the capabilities within a guidance element. In that evaluation, our system remained competitive with the radio team, but lost in two out of the three scenarios.

The final language for guidance was inspired by our observations of the radio-team strategies, extensive discussions with subject matter experts and extensive numbers of simulations. We noted that while the human team could not execute a strategy as well as we could, the space of strategies that they were able to engage were far more sophisticated than ours. This led to the creation of a the more sophisticated formalism for capturing human strategic guidance.

We have taken the first step towards generic coordination technology that end-users can tailor to specific problem instances. The approach was validated in one domain thanks to the extensive and expensive evaluations carried out by the DARPA Coordinators program. In the future, we hope to be able to apply this approach to other application domains. One key area that needs to be investigated is extensions to allow human users to make guidance adjustments *during* execution. There are situations where a series of outcomes either invalidates an assumption when creating the *a priori* guidance or creates an opportunity to improve on that guidance. Addressing this requires the ability for human users to quickly and easily understand and modify the guidance while it is being executed. Even more advanced steps would be evaluating and ultimately generating appropriate online guidance modifications.

References

Ai-Chang, M.; Bresina, J.; Charest, L.; Chase, A.; jung Hsu, J. C.; Jonsson, A.; Kanefsky, B.; Morris, P.; Rajan, K.; Yglesias, J.; Chafin, B. G.; Dias, W. C.; and Maldague, P. F. 2004. Mapgen: Mixed-initiative planning and scheduling for the mars exploration rover mission. *IEEE Intelligent Systems* 19(1):8–12.

Boutilier, C. 1999. Multiagent systems: Challenges and opportunities for decision-theoretic planning. *AI Magazine* 20(4):35–43.

Chen, Y.; Wah, B.; and Hsu, C.-W. 2006. Temporal planning using subgoal partitioning and resolution in sgplan. *Journal of Artificial Intelligence Research (JAIR)* 26(1):323–369.

Fox, M., and Long, D. 2006. Modelling mixed discrete-continuous domains for planning. *Journal of Artificial Intelligence Research (JAIR)* 27.

Gerevini, A.; Saetti, A.; Serina, I.; and Toninelli, P. 2005. Fast planning in domains with derived predicates: an approach based on rule-action graphs and local search. In *Proceedings of the 20th national conference on Artificial intelligence (AAAI)*, 1157–1162. AAAI Press.

Groen, F. C.; Spaan, M. T.; Kok, J. R.; and Pavlin, G. 2007. *Real World Multi-agent Systems: Information Sharing, Coordination and Planning*, volume 4363/2007 of *Lecture Notes in Computer Science. Logic, Language, and Computation*. Springer-Verlag.

Hayes, C.; Larson, A.; and Ravinder, U. 2005. Weasel: A mipas system to assist in military planning. In *ICAPS05 MIPAS Workshop (WS3)*.

Murphy, R. R. 2004. Human-robot interaction in rescue robotics. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on* 34(2):138–153.

Musliner, D. J.; Durfee, E. H.; Wu, J.; Dolgov, D. A.; Goldman, R. P.; and Boddy, M. S. 2006. Coordinated plan management using multiagent MDPs. In *Proceedings of the 2006 AAAI Spring Symposium on Distributed Plan and Schedule Management*.

Myers, K. L.; Jarvis, P. A.; Mabry, W.; Michael, T.; and Wolverton, J. 2003. A mixed-initiative framework for robust plan sketching. In *Proceedings of the 13th International Conference on Automated Planning and Scheduling (ICAPS)*.

Schurr, N.; Marecki, J.; Scerri, P.; Lewis, J.; and Tambe, M. 2005. *The DEFACTO System: Coordinating human-agent teams for the future of disaster response*. Springer Press. chapter Programming multi-agent systems: Third international workshop.

Smith, S.; Gallagher, A. T.; Zimmerman, T. L.; Barbulescu, L.; and Rubinstein, Z. 2007. Distributed management of flexible times schedules. In *Proceedings of the Sixth International Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS 2007)*.