Doctoral Mentoring Program

AAMAS 2009

Eighth International Conference on Autonomous Agents and Multi-Agent Systems

Budapest, Hungary
May 10th, 2009

Stacy Marsella
Doctoral Mentoring Chair

Preface

AAMAS 2009 included a doctoral mentoring program intended for PhD students in advanced stages of their research. The program provided an opportunity for students to interact closely with established researchers in their fields, to receive feedback on their work and to get advice on managing their careers.

Specifically, the goals of the program were:

- To match each student with an established researcher in the community (who will act as a mentor).
- To allow students an opportunity to present their work to a friendly audience of other students as well as mentors.
- To provide students with contacts and professional networking opportunities.

The doctoral mentoring program afforded mentors and their students opportunities for interactions prior to the conference, as well as a one-day doctoral symposium on the first day of the conference.

This document is a compilation of the research abstracts by the students in the consortium.

Table of Contents

Increasing Scalability in Algorithms for Centralized and Decentralized Partially Observable Markov Decision Processes

(Extended Abstract)

Christopher Amato
Department of Computer Science
University of Massachusetts
Amherst, MA 01003 USA
camato@cs.umass.edu.com

ABSTRACT

Real-world problems contain many forms of uncertainty, but current algorithms for solving sequential decision making problems under uncertainty are limited to small problems due to large resource usage. In my thesis, I study methods to increase the scalability of these approaches such as using memory bounded solutions, sampling or taking advantage of domain structure. I also plan to explore other methods to improve scalability and generate more practical real-world domains on which to test these algorithms.

1. INTRODUCTION

Sequential decision making under uncertainty is a thriving research area. In these problems, agents must choose a sequence of actions to maximize a given objective function. The actions must be chosen based on imperfect information about the system state due to stochastic action results and noisy sensors. When multiple cooperative agents are present, each agent must also reason about the action choices of the others in order to maximize joint value while making decisions based solely on local information. Using single and multi-agent sequential decision making under uncertainty a wide range of single and multi-agent problems can be represented, but the computational complexity of solving these models presents an important research challenge.

As a way to address this high complexity, some topics that I study in my thesis include: optimizing agent performance with limited resources, achieving coordination without communication, exploiting goals in multi-agent coordination and using sampling to reason about the future. The models used to represent single and multi-agent problems are the partially observable Markov decision process (POMDPs) and decentralized POMDP (DEC-POMDP). POMDPs represent stochastic actions and uncertainty about the current system state. DEC-POMDPs extend the POMDP model to multiple cooperative agents.

I first discuss the work that I have completed towards studying these problems. I then describe the additional research that I expect to complete for my thesis. Note that because no communication is assumed in my work with the DEC-POMDP model, agents must plan without explicitly sharing information.

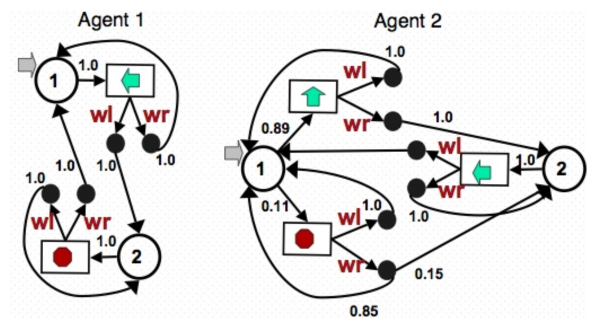

Figure 1: A set of two node stochastic controllers for a two agent DEC-POMDP.

2. OPTIMIZING CONTROLLERS FOR POMDPS AND DEC-POMDPS

Finite state controllers (depicted in Figure 1) have been shown to effectively model solutions for both infinite-horizon POMDPs [5] and DEC-POMDPs [4, 6]. This approach facilitates scalability as it offers a tradeoff between solution quality and the usage of available resources. That is, a controller may be optimized for a given amount of memory.

Unlike other controller based approaches for POMDPs and DEC-POMDPs, our formulation defines an optimal solution for a given size. This is accomplished by formulating the problem as a nonlinear program (NLP), and exploiting existing nonlinear optimization techniques to solve it. In the POMDP case, parameters are optimized for a fixed-size controller which produces the policy [2]. In the DEC-POMDP case, a set of fixed-size independent controllers is optimized, which when combined, produce the policy [1]. While an overview of how to solve these problems optimally is presented in the thesis, this would often be intractable in practice. As a result, we also evaluate an effective approximation technique using standard NLP solvers.

One premise of our work is that an optimal formulation of the problem facilitates the design of solution techniques that can overcome the limitations of previous controller-based algorithms and produce better stochastic controllers. The general nature of our formulation allows a wide range of solution methods to be used. This results in a search that is more sophisticated than those previously used in controller-based methods. Our approach also provides a framework for which future algorithms can be developed.

Two Agent Tiger Problem $|S| = 2$, $|A| = 3$, $|\Omega| = 2$

BFS	DEC-BPI	NLP	Goal-directed
-14.1, 12007s	-52.6, 102s	-1.1, 6173s	5.0, 75s

Meeting in a Grid Problem $|S| = 16$, $|A| = 5$, $|\Omega| = 2$

BFS	DEC-BPI	NLP	Goal-directed
4.2, 17s	3.6, 2227s	5.7, 117s	5.6, 4s

Box Pushing Problem $|S| = 100$, $|A| = 4$, $|\Omega| = 5$

BFS	DEC-BPI	NLP	Goal-directed
-2, 1696s	9.4, 4094s	54.2, 1824s	149.9, 199s

Rover Problems $|S| = 256$, $|A| = 6$, $|\Omega| = 8$

BFS	DEC-BPI	NLP	Goal-directed
x	-1.1, 11262s	9.6, 379s	26.9, 491s
x	-1.2, 14069s	8.1, 438s	21.5, 956s

Table 1: The values produced by each method along with controller size and time in seconds.

Our results demonstrate that local optimization of the NLP formulation provides concise high quality solutions. In POMDP domains, our technique was competitive in general and outperformed a leading approximate algorithm on a set of problems. In DEC-POMDP domains, our approach significantly outperformed other approximate algorithms, often producing the highest value while using the least amount of time. Further improvement in solution quality is likely as more specialized solution methods are developed.

3. ACHIEVING GOALS IN DEC-POMDPS

Another method of improving scalability is to take advantage of structure inherent in domains. One such structure is the achievement of goals, after which the problem terminates. We have demonstrated that when certain goal conditions are present in DEC-POMDPs, this structure can be used to improve scalability and solution quality [3].

To demonstrate this, we have extended the indefinite-horizon framework to decentralized domains using common assumptions – that terminal actions exist for each agent and rewards for non-terminal actions are negative. Under these assumptions we showed that dynamic programming could be adapted to solve the indefinite-horizon problem. We also developed a sample-based algorithm which is able to solve problems with more relaxed goal conditions. For this algorithm, we provided a bound detailing the number of samples required to ensure that the optimal solution is approached. As shown in Table 1 this algorithm was often able to significantly outperform other DEC-POMDP approximate algorithms on a range of goal-directed problems. The approach also provides the framework for sample-based methods to be extended to other classes of decentralized problems.

4. FUTURE CONTRIBUTIONS

In addition to the work above, I also plan to work on the following projects for my thesis.

Incremental policy generation

We are currently developing a method to improve optimal DEC-POMDP algorithms by reducing the necessary search space. This will allow larger problems to be solved optimally and better solutions to be found for other problems. This is accomplished by determining what states are reachable after different action choices are made and observations are seen by the agents. Because not all states will be reachable, not all states will need to be considered to determine a solution. An example of this is a robot observing a wall to its left. The exact system state may not be known, but it can be limited to those states in which the agent has a wall to its left. If the number of solutions can be sufficiently limited, we may be able to identify natural lower complexity subclasses. This approach can also be incorporated in a number of approximate methods, which will improve their performance as well.

Attribute-based planning

We are also working on other ways to make use of domain information to simplify the planning process in DEC-POMDPs. This approach would utilize user generated or learned information in the form of attributes or landmarks that serve to summarize parts of agent histories. These attributes could include the last location of a wall seen or the number of steps since another agent was observed. Agents could remember only these attributes, allowing planning to be conducted over a smaller set of attributes rather than over all possible histories.

5. CONCLUSIONS

In conclusion, my thesis work improves scalability and solution quality for solving uncertain single and multi-agent domains. This is accomplished by such methods as determining the optimal use of a fixed solution space and utilizing domain structure to improve solution search. These approaches perform well in a wide range of problems. In the future, we plan to further improve scalability and solution quality while applying our methods to real-world domains such as e-commence, manufacturing or medical diagnosis.

6. REFERENCES

[1] C. Amato, D. S. Bernstein, and S. Zilberstein. Optimizing memory-bounded controllers for decentralized POMDPs. In *Proceedings of the Twenty-Third Conference on Uncertainty in Artificial Intelligence*, Vancouver, Canada, 2007.

[2] C. Amato, D. S. Bernstein, and S. Zilberstein. Solving POMDPs using quadratically constrained linear programs. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence*, pages 2418–2424, Hyderabad, India, 2007.

[3] C. Amato and S. Zilberstein. Achieving goals in decentralized pomdps. In *Proceedings of the Eighth International Joint Conference on Autonomous Agents and Multiagent Systems*, Budapest, Hungary, 2009.

[4] D. S. Bernstein, E. Hansen, and S. Zilberstein. Bounded policy iteration for decentralized POMDPs. In *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence*, pages 1287–1292, Edinburgh, Scotland, 2005.

[5] P. Poupart and C. Boutilier. Bounded finite state controllers. In *Advances in Neural Information Processing Systems*, 16, Vancouver, Canada, 2003.

[6] D. Szer and F. Charpillet. An optimal best-first search algorithm for solving infinite horizon DEC-POMDPs. In *Proceedings of the Sixteenth European Conference on Machine Learning*, Porto, Portugal, 2005.

Modelling Rational Agents in Multi-Agent Systems

(Extended Abstract)

Nils Bulling
Department of Informatics, Clausthal University of Technology, Germany
bulling@in.tu-clausthal.de

ABSTRACT

This extended abstract gives an overview about my current and future research as well as a summary of my PhD thesis. The thesis is about rational agents in multi-agent systems where the main focus is on formal methods that allow for modelling and reasoning about such systems and its comprised agents. Several aspects of rational agency are treated, for instance, rational agents' behaviors, coalition formation processes, communication among rational agents, and acting with limited resources. The main questions which are tried to be answered are of the following nature: How do rational agents behave under various restrictions and settings? Complexity issues are considered as well, mainly with respect to model checking.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent Systems*; I.2.4 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods—*Modal logic, Temporal logic*

General Terms

Theory, Logical foundations

Keywords

multi-agent systems, argumentation, coalition formation, game theory, temporal logic

1. MOTIVATION AND OVERVIEW

A vast amount of software systems can be considered as multi-agent systems (MAS) [10, 11], even if the notion of an agent is not explicitly used. Such systems (e.g. online shops, distributed systems, web services, and computer games) often require human or rational decision making to provide a good and up-to-date service. Computer games are excellent examples of software programs with an increasing demand for such rational decision techniques; the same is valid for a variety of commercial applications. Arguably, basic needs of most of those programs include *knowledge representation techniques* and interfaces that allow to *query* or *infer data* from the information stored. The logic-based approaches for rational agents we present here provide means for both points mentioned above. We discuss rationality in MAS from a more theoretical point of view: The focus is on *modelling*, *specifying*, and *verifying* the behavior of rational agents, issues important to guarantee that software is reliable and to ensure that it does what it is supposed to do.

Apart from these cases we consider formal logics as tools to *speak about* and to better understand the complex interactions taking place in MAS, again, focussing on rationality. The main questions we try to answer are of the following form: *How to model and to reason about rational agents?*; *How do rational agents cooperate and communicate?*; or *How do rational agents act under incomplete information and limited resources?*

In addition to the main part, in which we take on a *model theoretic point of view*, we also discuss how rationality aspects can be analyzed and implemented in the more practical setting of *agent programming languages*. Here we especially focus on *communicative acts* and how to interpret them. We also sketch how these tools might be used for reasoning *within* agents; clearly, agent programming languages prepare an appropriate ground for that.

Finally, throughout the thesis we are interested in the computational aspect of the presented formal frameworks, in particular in the analysis of the model checking complexity.

2. RATIONALITY ASPECTS IN MAS

In the past it has been shown that modal logics are applicable to a great many of heterogeneous systems. *Epistemic logics*, for instance, are used to model and to reason about knowledge of agents; *temporal logics* allow to verify temporal properties of systems. *Strategic logics* have attracted quite some interest in recent research. They describe what agents can enforce and what power coalitions have. Among these logics *Alternating-Time Temporal Logic* (ATL for short) [1] is one of the most influential; it combines temporal concepts with basic game theoretic ones. ATL is very flexible regarding extensions by other modal concepts, e.g. by epistemic logic, which often result in powerful and interesting logics applicable to various areas of MAS.

In this thesis we analyze how ATL can be extended in such a way that it is suitable for the modelling of various *rationality aspects* in MAS as pure ATL does not allow to speak about sensible strategies *per se* rather about *all* possible behaviors of agents regardless whether they make sense or not.

One of the main questions addressed is the following: *How do agents behave if they act according to a given* plausibility *or* rationality *assumption?* Apart from the epistemological gain about agents' rational behavior the answer provides, there also is a more practical aspect: In many games, from a game theoretic point of view, the number of all possible outcomes is infinite, although only some of them "make sense"; hence, a notion of rationality (like subgame-perfect Nash equilibrium) allows to discard the "less sensible" ones, and to determine what should happen had the game been played by ideal players. For this purpose we extend ATL with a notion of *plausibility* [5] and refer to the logic as ATLP. This extension of ATL enables us (1) to *express* various rationality assumptions of intelligent agents; (2) to *specify* sets of rational strategy profiles *in the object language*; and (3) to reason about agents' play if only those strategy profiles were allowed. For example, we may assume the agents to play only Nash equilibria, Pareto-optimal profiles or undominated strategies, and ask about the resulting behaviour (and outcomes) under such an assumption. The logic also gives rise to generalized versions of classical solution concepts through characterizing patterns of payoffs by suitably parameterized formulae of ATLP. We investigate the complexity of model checking for several classes of formulae: It ranges from $\mathbf{\Delta_3^P}$ to **PSPACE** in the general case and from $\mathbf{\Delta_3^P}$ to $\mathbf{\Delta_4^P}$ for the most interesting subclasses, and roughly corresponds to solving extensive games with imperfect information.

We do also propose a version of ATLP for *imperfect information games* as "pure" ATLP is for *perfect information games* only. The resulting logic *Constructive Strategic Logic with plausibility* (CSLP) [7, 9] can be used in the same way as ATLP but now for perfect *and* imperfect information games. Moreover, the logic is more than just an independent combination of ATLP with epistemic operators, the plausibility concept allows to defined a neat doxastic notion, *rational beliefs*, on top of knowledge (similar to [4]). We show that beliefs satisfy axioms **KD45**. In summary, CSLP can be used to reason about rational play and rational beliefs under *uncertainty*.

The previous extension is about classical indistinguishability between states, however, there is another interesting angle to incomplete information. Where in ATL *the worst* possible response from the other agents is assumed we consider the case in which agents communication and cooperation abilities are limited such that it is not very likely that the "worst case" will happen. The presented logic ATL *with probabilistic success* [6] tries to soften the rigorous notion of success that underpins ATL and allows to reason about the likelihood that agents have a successful strategy to enforce their goals.

Undeniably, cooperation among agents plays a decisive role in strategic logics, however, in ATL it is only present implicitly. What we would like to analyze is *why* agents should cooperate with other agents. For this purpose we combine an argumentation-based approach to coalition formation [2] into the semantics of ATL. The proposed logic *Coalitional* ATL [3] allows to reason and to model the formation process of *rational* coalitions and their power.

Finally, we identify two further ingredients important for the modelling of rational agents. Firstly, we argue that *resources* play a decisive role in the selection of the right strategy as agents are usually confronted with a limited amount of them what should be reflected in the choice of strategies and in the selection of agents to cooperate with. For this purpose a combination of ATL with a variant of *Linear Logic* [8] is proposed where resources are treated as first-class citizens. Finally, we consider the *communication process* among agents in the more practical setting of agent oriented programming languages. We would like to note that both of the latter topics are part of our ongoing research.

3. CONCLUSIONS

We have presented several logics to model and to analyze rationality aspects in MAS; each of them suitable to be used for a specific aspect of rationality. Our main focus was on a model theoretic analysis where logics can be used to reason about a previously built *model*. This allows for the verification and specification of MAS. In consequence the model checking complexity was important throughout this thesis. In the case of ATL with limited ressources however we exemplarily showed[1] how these logics can also be used from a deductive point of view, e.g. as inference systems *within* agents.

4. REFERENCES

[1] R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time Temporal Logic. *Lecture Notes in Computer Science*, 1536:23–60, 1998.

[2] L. Amgoud. An argumentation-based model for reasoning about coalition structures. In *ArgMAS*, pages 217–228, 2005.

[3] N. Bulling, C. Chesñevar, and J. Dix. Modelling coalitions: ATL+argumentation. In *Proceedings of AAMAS'08*, Estoril, Portugal, May 2008. ACM Press.

[4] N. Bulling and W. Jamroga. Agents, beliefs, and plausible behavior in a temporal setting. In *Proceedings of AAMAS'07*, pages 570–577, Honolulu, Hawaii, USA, 2007. ACM Press.

[5] N. Bulling, W. Jamroga, and J. Dix. Reasoning about temporal properties of rational play. *Annals of Mathematics and Artificial Intelligence*, 2009.

[6] N. Bulling and W.Jamroga. What agents can probably enforce. In *Proceedings of the International Workshop on Concurrency, Specification and Programming CS&P'08*, Gross Vaeter, Germany, 2008.

[7] N. Bulling and W.Jamroga. Rational play and rational beliefs under uncertainty. In *Proceedings of AAMAS'09*, Budapest, Hungary, may 2009.

[8] J.-Y. Girard. Linear logic: Its syntax and semantics. In J.-Y. Girard, Y. Lafont, and L. Regnier, editors, *Advances in Linear Logic (Proc. of the Workshop on Linear Logic, Cornell University, June 1993)*, number 222. Cambridge University Press, 1995.

[9] W. Jamroga and T. Ågotnes. Constructive knowledge: What agents can achieve under incomplete information. *Journal of Applied Non-Classical Logics*, 17(4):423–475, 2007.

[10] G. Weiss, editor. *Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence*. MIT Press: Cambridge, Mass, 1999.

[11] M. Wooldridge. *An Introduction to Multi Agent Systems.* John Wiley & Sons, 2002.

[1]More precisely, we will show this as part of our current research.

Agents with Emotional Intelligence for Storytelling

João Dias, Ana Paiva INESC-ID
Av. Prof. Cavaco Silva, TagusPark
2780-990 Porto Salvo, Portugal
joao.dias@gaips.inesc-id.pt, ana.paiva@inesc-id.pt

1. INTRODUCTION

In trying to build increasingly believable autonomous characters for virtual environments, researchers successfully explored the use of emotional systems to endow their agents with emotional capabilities. Of relevant notice are the ability to experience and express emotions, and the ability to use its emotions to influence decision-making. These systems are often based in appraisal theories, such as OCC [5], which claim that emotions are a result of subjective evaluations (appraisals) of events and situations.

However, agents with full Emotional Intelligence have been largely unaddressed so far. By Emotional Intelligence we mean the definition proposed by Salovey and Mayer[6, 4]:

> "Emotional intelligence refers to an ability to recognize the meanings of emotion and their relationships, and to reason and problem-solve on the basis of them."

Although the ability to monitor feelings and emotions has been addressed, having explicit knowledge about the appraisal process and other's emotions, and using that knowledge to reason about emotions and build plans of actions, has not. This is due in part to the fact that one cannot address this problem without first tackling the other components. Only now we are ready to start addressing it.

We believe that Emotional Intelligence is an important component to achieve more human-like and believable behaviour, especially in a Storytelling scenario where social interaction and emotional conflicts take a major role. It is true that it is still possible to achieve believable behaviour without this component in a storytelling scenario (as in FearNot![2]), because we can author the characters in a way to portray such emotional intelligence. However, when facing an interacting user, this is much harder to do without a complete Emotional Intelligence. Thus, Emotional Intelligence offer us more flexibility in achieving believable social behaviour.

2. RELATED WORK

Continuing their research in Emotional Intelligence, Mayer and Salovey put forward a four branch model that divides Emotional Intelligence in four main skills:

- **perceiving emotions in oneself and others -** has to do with the perception and expression of emotion through gestures, facial expressions, or other communication mechanisms. This area was the first one to be addressed by researchers, and is still an important subject of research in IVAs and ILEs.
- **using emotions to facilitate thought -** the second most researched skill, focus on using emotions to guide cognitive processes, such as learning and adaptation, attention and decision making.
- **understanding emotions -** the idea here is that emotions convey information. For instance, Anger indicates a possible intention of harming other. Therefore, understanding emotions involves understanding the meaning of emotions, together with the capacity to reason about those meanings. This skill together will be the focus of our work.
- **managing emotions -** once a person understand emotions, it can manage one's own and other's emotions in order to promote social goals. For instance, one can go see a movie when distressed in order to feel better, or do something pleasant to help a friend come out of a bad mood. To some extent, this skill was addressed by the work of Marsella and Gratch in EMA[3], where they apply emotion coping strategies to deal with one's negative emotions.

3. MODEL

The proposed model will be integrated and implemented in an existing emotional agent architecture, named FAtiMA[2, 1]. In FAtiMA, emotions result from a subjective appraisal of events according with OCC Theory. The architecture is divided in two main layers, a reactive and a deliberative one. The first layer is responsible for the agent's reactive behaviour and is composed by: a set of emotional reaction rules that define OCC's appraisal variables such as desirability which are then used to generate emotions; and by a set of action tendencies (AT) that represents the character's impulsive actions (e.g. crying when very distressed). The deliberative layer is responsible for the agent's goal-oriented behaviour and means-ends-reasoning. It also has an appraisal component that generates emotions from the state of plans in memory. The Knowledge Base and the Autobiographic Memory are the main memory components. The top of Figure 1 shows a simplified diagram of the architecture.

In order to extend FAtiMA with the ability to understand emotions, we must first endow the planner with explicit knowledge about the Appraisal Process. This can be done, by translating the emotional reaction rules into planning operators, which use a STRIPS notation. Then, the OCC rules used to create emotions from the appraisal variables must also modeled as planning operators. For instance, the rule

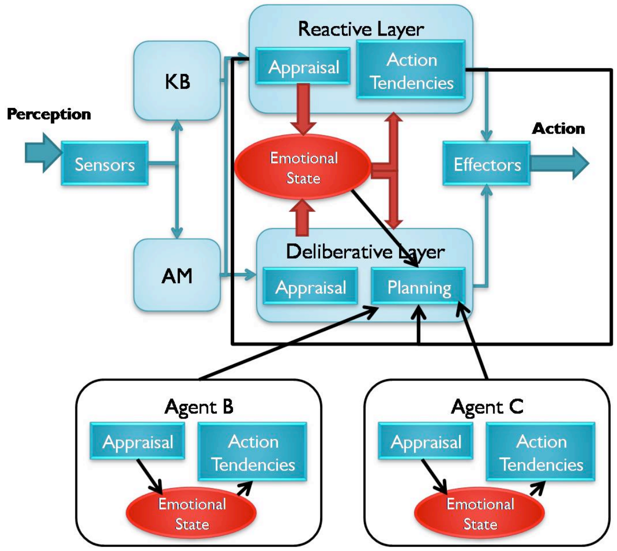

Figure 1: Extending FAtiMA with the skill of understanding emotions

that maps an undesirable event into the emotion of distress is translated into an operator which has the precondition that an event is undesirable and has the effect of causing distress. Additionally to the Appraisal Process, is also important to model actions that are triggered from particular emotional states, i.e. Action Tendencies. The translation is achieved by modeling action tendencies as operators where the preconditions correspond to the emotional state that triggers the AT. Finally, the planner must also have knowledge about the agent's emotional state. These connections are represented in Figure 1 by the black arrows coming out of the appraisal and AT processes.

This first step gives our agent the capability to reason about his own's emotions but only partially about other's emotions. Although it is true that the agent can assume that others are like him and use his information to predict how others will feel, this will often lead to wrong assessments. Given the subjective nature of appraisal, the agent must build a model of how others appraise events and react to a given emotion. So, if the agent A knows two other agents B and C he will, additionally to its own structures it will model the other agents' emotional reaction rules, emotional state and action tendencies (as seen in bottom of Figure 1). Initially, when the agent first meets another agent he will start with a model equal to his own (he assumes that others are like him). But as time goes by, the agent will refine the model it has about that new agent. For instance, if a given event is thought to be undesirable to another agent, but that agent happens to express joy or happiness, the agent will have to update the desirability value for that event.

4. ILLUSTRATIVE EXAMPLE

We will give a brief example of what kind of reasoning an agent can do with this emotional information. In order to model a bullying scenario, we can model a bully character with a high level goal of making the victim cry. In order to achieve this goal, the bully knows that the victim cries when it's very distressed (an AT), so the planner will try to force the victim to become distressed. The planner will also know that distress is caused by an undesirable event, and will consider all actions undesirable for the victim (kicking, pushing, insulting, etc). The deliberative layer will then select one of the alternatives and execute it. If everything goes as planned the bully will succeed and become satisfied. However, if the victim doesn't cry but seems happy instead, he will either try something else or eventually fail to bully the victim.

5. FINAL CONSIDERATIONS

The proposed extensions will have strong implications in some of the core components of the architecture. In the current architecture there is an initial deliberation where a goal is selected, and then the means-ends reasoning takes full control of the rest. With the proposed model, there will be several levels of deliberation and commitment, interleaved with planning. Moreover, by modeling behaviour with higher-level goals, which can expand to a wide number of alternative solutions, we will increase the search space and planning may become intracktable. We believe that this problem can be solved by using emotional information as a heuristic to guide and constrain means-ends reasoning, which actually corresponds to the second skill in Mayer's model. Thus, on an ending note, we point out that in order to tackle the last two skills and build agents with Emotional Intelligence we need to address all the four skills.

6. ACKNOWLEDGEMENTS

This work was supported by a scholarship (SFRH BD/19481/2004) granted by the Fundação para a Ciência e a Tecnologia (FCT) The authors are solely responsible for the content of this publication. It does not represent the opinion of the FCT, which are not responsible for any use that might be made of data appearing therein.

7. REFERENCES

[1] J. Dias. Fearnot!: Creating emotional autonomous synthetic characters for empathic interactions. Master's thesis, Universidade Técnica de Lisboa, Instituto Superior Técnico, Lisboa, 2005.

[2] J. Dias and A. Paiva. Feeling and reasoning: a computational model for emotional agents. In *Proceedings of 12th Portuguese Conference on Artificial Intelligence, EPIA 2005*, pages 127–140. Springer, 2005.

[3] G. J. and M. S. A domain-independent framework for modeling emotion. *Cognitive Systems Research*, Volume 5(4), 2004.

[4] M. J., C. D., and P. Salovey. Emotional intelligence meets traditional standards for an intelligence. *Intelligence*, 27:267–298, 1999.

[5] A. Ortony, G. Clore, and A. Collins. *The Cognitive Structure of Emotions*. Cambridge University Press, UK, 1998.

[6] P. Salovey and J. Mayer. *Imagination, Cognition and Personality*, chapter Emotional Intelligence, pages 185 – 211. 1990.

Trustworthy Service-Oriented Computing

Chung-Wei Hang
chang@ncsu.edu
Dept. of Computer Science
North Carolina State University
Raleigh, NC 27695-8206, USA

1. THE PROBLEM

In service-oriented computing environments, computing resources are managed as *services*, which can be used directly or composed into larger services. Service-oriented architecture has been widely adopted in modern distributed environments such as for cloud computing. However, the problem of finding desired services has arisen. Finding desired services can be divided into two sub-problems: *service discovery* and *service selection*. The former one emphasizes how to find services that match consumers' requirements. The latter focuses on how to select *best* matched services. Service discovery usually finds services based on static functional attributes (e.g., service descriptions), whereas service selection tends to capture the dynamism of nonfunctional properties. For example, suppose a traveler is looking for flight tickets from Raleigh to Budapest. Service discovery returns itineraries provided by various airline companies. Service selection, on the other hand, selects best quality itinerary in terms of in-flight service, delay record, etc.

Traditional service discovery approaches, such as, *Web Service Definition Language* or *WSDL*, and *Service level agreement* or *SLA*, describe the functional configurations of services. However, these approaches lack mechanisms to monitor and track the nonfunctional properties like quality of service (QoS) dynamically. For instance, suppose an United flight delays due to technical difficulties, how this affects the airline company the consumers choose next time? The QoS assessments should be able to reflect the expected outcome of future behavior and affect consumers' willingness to select that service, even though the service matches their requirements.

2. QOS-BASED SERVICE SELECTION

We introduce the idea of QoS-based service selection approach to address the problem of selecting services based on both functional and nonfunctional properties. There are two main problems to be solved.

The advantage of service-oriented computing is that we can compose services to create new ones. This is called *service composition*. Much attention on service composition focuses on lower-level solutions, such as, BPEL, OWL-S, π-calculus, and Petri nets [5]. These methods capture the composition configurations, but, similar to WSDL, fail to take nonfunctional properties into consideration. Services are composed into larger services. However, these underlying services may not be directly exposed to the consumers. Service composition can be divided into many scenarios [4] and these scenarios can be nested. These scenarios make QoS metrics hard to collect and evaluate. For example, a traveler books an itinerary from a travel agent without knowing which hotel agent is behind. Thus, service selection becomes more complicated because the consumers may not even know with whom they are interacting. Existing service selection approaches deal with service composition poorly because they mostly either not consider service composition, or assume the composition information is fully observable. Therefore, how to collect QoS metrics, and how to evaluate the underlying services behind composition must be addressed in our service selection solution.

Another challenge is, even if a service can be evaluated based on past experience, consumers may lack past experience of unknown parties. One common solution is to bootstrap the unknown parties by assigning initial assessments as new comers. Better solution is to introduce referral networks. One may ask others for referrals of an unknown party, which is called the target. The referrals contain either direct information with the target, or further references if the referrers have no experience themselves. The initial party can follow the *referral chains* until certain criteria are met, say, until a certain depth. After collecting the referrals, the initial party can *aggregate* all information gathered as the experience of the target, also evaluate the *sociability* of referrers. The sociability is the ability of providing accurate referrals.

3. SOCIAL TRUST MODEL

Trust modeling in artificial intelligence provides us a promising solution to above questions. Trust is a basis of interactions, indicating the relationships between parties in large, open systems. Two parties must trust each other sufficiently to be willing to carry out desired interactions. In a service-oriented context, a party Alice trusts another party Bob, because Alice expects Bob will provide desired service. In general, an ideal trust model should contain following functionalities: *trust representation*, *trust propagation*, and *trust update*.

The trustworthiness of a party should be represented as not only a *probability*, but also the *confidence* of the probability. An ideal trust representation should satisfy: (a) the confidence goes up as the evidence increases given a fixed probability, and (b) the confidence drops if conflicts occur given a fixed amount of evidence.

Trust propagation defines how trust information is propagated. There are two basic cases. First, how indirect trust information should be discounted? For example, Alice trusts Bob who trusts Charlie. Alice should not consider the trust information of Charlie from Bob totally. Instead, Alice should discount Bob's trust in Charlie by her trust in Bob. Second, how trust information should be combined? For example, Alice collects trust information of Dave from both Bob and Charlie. A trust model should define how trust information from different sources is aggregated.

The third component is trust update. As we gain more experience with the target, trust should be updated in the way that updated trust can predict the target's behavior more accurately. For example, Alice asks Bob for referrals of Charlie. When Alice has better knowledge of Charlie, how she updates her trust in Bob about his sociability? Generally, given estimated trust and actual knowledge, trust update defines how accurate the estimation is.

Our previous work [1, 2] provides a complete solution to trust modeling. We adopt the trust representation from [6, 7], which defines trust in both evidence and belief space. For example, the trust in evidence space $\langle r, s\rangle$ represents how much good and bad evidence we have with the target. The probability is defined by $\frac{r}{r+s}$. In belief space, $\langle b, d, u\rangle$ corresponds to *belief* (belief of trust), *disbelief* (belief of distrust), and *uncertainty*, respectively. The trust can be translated between evidence and belief spaces. The definition of uncertainty satisfies the two requirements of confidence. We also define trust update by comparing the difference of probability-densities of the estimation and the actual trust. Finally, our trust model provides three trust propagation operators: *concatenation*, *aggregation*, and *selection*. The concatenation operator defines how indirect information should be discounted, whereas the aggregation operator is used to combine trust evidence from different sources. The selection operator exempts trust propagation from *double counting*. Additionally, our trust model is verified via simulations and social network data.

4. QOS-BASED TRUSTWORTHY SERVICE SELECTION

We aim to provide a QoS-based trustworthy service selection method based on our trust model. There are three main components as follows:

1. Developing an ontology that include classes, relationships, and attributes required to characterize services and their uses in service-oriented environments.

2. Formalizing rich service composition models built on trust attributes specified in the above ontology.

3. Developing approaches for agents to monitor and explore desired service compositions dynamically.

We refine and enhance an existing QoS ontology from [3] to fit it into our approach. This ontology will be able to capture SLAs as well as the requirements of consumers and advertisements from providers. Both domain-independent and domain-specific QoS properties can be defined in our ontology. We model the service-oriented environments by a directed graph. The graph can capture the relationships between services in service composition. Then, QoS properties are monitored and collected from direct experience and indirect evidence (i.e., referrals). The QoS assessments are represented as trust. The trustworthiness of a QoS attribute can be inferred by trust propagation. Also, we can further evaluate the QoS properties, by comparing the QoS metrics and SLAs, and the sociability of referrers by trust update. Knowing the sociability can yield more accurate trust information from referrals. Finally, we will apply *multiattribute utility theory* for decision-making, based on the trustworthy QoS assessments.

5. CHALLENGES

Our main challenge is how to capture the relationships in service compositions so that the trustworthy QoS assessment can accurately reflect the QoS of services. For example, a traveler books an itinerary from a travel service, which interacts with a flight service, a hotel service, and a car rental service. Suppose the availability of the car rental service is not satisfiable. This ends up with bad availability of the travel service. Given the fact that the traveler is not aware of the services behind, an appropriate mechanism is needed in order to punish the car rental service, and the travel service (because it selects the car rental service), rather than the flight and hotel services.

6. CONCLUSION

This work aims to provide a QoS-based trustworthy service selection model in service-oriented environments. The model provides an ontology to capture consumers' requirements and providers' advertisements dynamically. We formalize a graphical service composition model to capture the relationships between services, develop approaches for consumers to monitor and explore desired services and service compositions. Our trust model, built on [1, 2], estimates trustworthiness of services in term of QoS properties, from both direct experience and indirect referrals for consumers to select desired services.

7. REFERENCES

[1] C.-W. Hang, Y. Wang, and M. P. Singh. An adaptive probabilistic trust model and its evaluation. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 1485–1488, 2008.

[2] C.-W. Hang, Y. Wang, and M. P. Singh. Operators for propagating trust and their evaluation in social networks. In *Proceedings of the 8th International Joint Conference on Autonomous Agents and Multiagent Systems (To appear)*, 2009.

[3] E. M. Maximilien and M. P. Singh. A framework and ontology for dynamic web services selection. *IEEE Internet Computing*, 8(5):84–93, Sept. 2004.

[4] D. A. Menascé. Composing web services: A QoS view. *IEEE Internet Computing*, 8(6):88–90, 2004.

[5] N. Milanovic and M. Malek. Current solutions for web service composition. *IEEE Internet Computing*, 8(6):51–59, 2004.

[6] Y. Wang and M. P. Singh. Trust representation and aggregation in a distributed agent system. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI)*, pages 1425–1430, Menlo Park, 2006. AAAI Press.

[7] Y. Wang and M. P. Singh. Formal trust model for multiagent systems. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1551–1556, Detroit, 2007. IJCAI.

Integrating Value Function-Based and Policy Search Methods for Sequential Decision Making

(Extended Abstract)

Shivaram Kalyanakrishnan
Department of Computer Sciences, The University of Texas at Austin
shivaram@cs.utexas.edu

Keywords

Reinforcement learning, Temporal difference learning, Policy search, Function approximation.

1. THESIS TOPIC

Sequential decision making from experience, or reinforcement learning (RL), is perfectly suited to autonomous agents that are situated in an unknown environment, and which must use their interactions with the environment to learn behavior that maximizes long-term gains. In general, this setting can be treated as a Markov Decision Problem (MDP), comprising a set of states S, a set of actions A, a reward function $R : S \times A \times S \rightarrow \mathbb{R}$, and a transition function $T : S \times A \times S \rightarrow [0, 1]$. In an MDP, the objective is to find a policy $\pi : S \rightarrow A$ that maximizes the expected long-term reward from every state $s \in S$. This can be done by determining the optimal action value function $Q^* : S \times A \rightarrow \mathbb{R}$, from which the optimal policy, denoted π^*, can be derived as: $\pi^*(s) = \text{argmax}_a Q^*(s, a), \forall s \in S$.

Classical approaches such as temporal difference learning [6], which proceed by successively refining the action value function based on observed experiences, provide efficient solutions to MDPs with finite sets of states and actions. Yet, a predominant number of sequential decision making problems that arise *in practice* have continuous (or very large) state spaces, which force the use of function approximation. Further, in many applications, sensor noise corrupts the state signal. As a consequence, nearly every RL problem in practice corresponds to a Partially Observable MDP (POMDP), to which most of the theoretical guarantees of value function-based (VF) methods fail to extend. Coping with partial observability in a principled manner has merited considerable attention in the literature [2], but is yet to scale to complex tasks with continuous state spaces.

Policy Search (PS) methods [1, 7] are optimization methods that directly seek to find parameters $\mathbf{w}^*$ of the optimal policy π^* by searching through the space of parameters W. In so doing, they do not necessarily compute the value function of the policy, and consequently, are likely to be less sample-efficient than VF methods. At the same time, their asymptotic performance is likely to be affected less by function approximation and partial observability. For most sequential decision making problems that arise in practice, there exist no theoretical bounds for the sample efficiency or asymptotic performance of either VF or PS methods; it is left to empirical devices to ascertain how these contrasting method perform.

This thesis aims to develop learning methods for practical sequential decision making tasks by integrating VF and PS methods, with the objective of achieving both sample efficiency and superior asymptotic performance.

2. COMPLETED WORK

2.1 Empirical Analysis of VF and PS Methods

As the first step towards combining the merits of VF and PS methods, we conduct a systematic empirical study to examine their relative strengths and weaknesses [3]. To do so, we devise a suite of "grid world" domains that can be varied for four parameters: problem size s, action noise p, expressiveness of function approximation χ, and state noise σ. Across a broad range of parameters settings (1250 in total), we record the performance of Sarsa, a classical VF method, and cross entropy optimization, a PS method.

We see clear patterns in the domain characteristics for which each class of methods excels. Our experiments illustrate that VF methods enjoy superior sample complexity and asymptotic performance when provided precise function approximators and complete state information. However, with inadequate function approximation and noisy state information, their performance drops significantly, and indeed below the asymptotic performance achieved by PS methods. With fixed values of $s = 10$, $p = 0.3$, and $\sigma = 0$, we observe the effect of varying the function approximation paramater χ in Figures 1(a) and 1(b). At $\chi = 1$ (exact representation of state space), VF indeed converges to the optimal policy, and at a much quicker rate than PS. Yet, under a deficient representation ($\chi = 0.1$), VF performs very poorly when compared to PS, which does not show such a drastic drop in asymptotic performance. Increasing the state noise σ adversely affects the asymptotic performance of both VF and PS methods, although the decline is more gradual for PS.

We implement a simple scheme to integrate VF and PS, which we enforce to share the same representation. In this integrated method, VF+PS, the learned representation of VF after a certain number of episodes of learning is transferred to PS. As visible in Figures 1(a) and 1(b), VF+PS inherits both the superior sample efficiency of VS and the high asymptotic performance of PS. Not only does VF+PS achieve higher asymptotic performance than both VF and

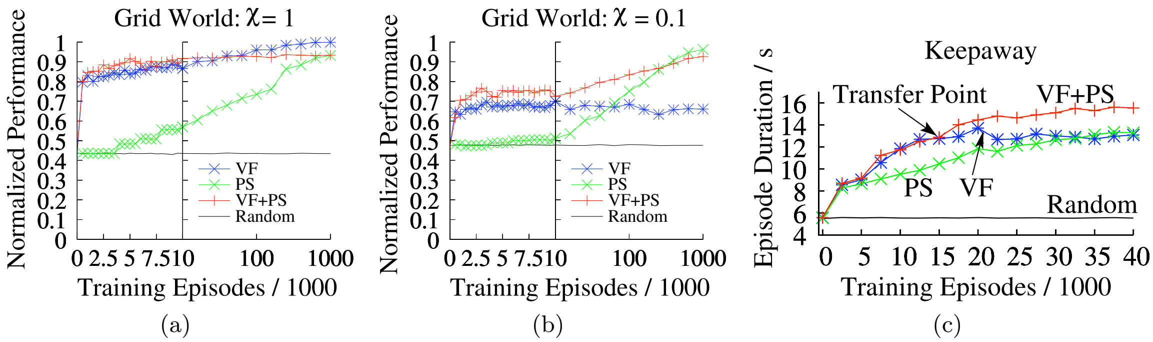

Figure 1: Empirical Analysis of VF and PS methods. In (a) and (b), note the break in the x axis at 10,000 episodes, beyond which a log scale is adopted. Descriptions are provided in text.

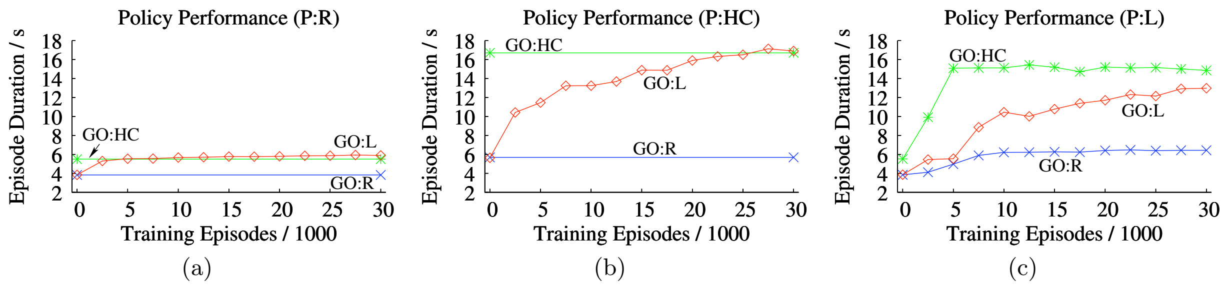

Figure 2: Keepaway Pass and GetOpen. The graphs shows three GetOpen policies (Random (GO:R), Hand-coded (GO:HC) and learned (GO:L)) when paired with a Pass policy that is Random (P:R (a)), Hand-coded (P:HC (b)), or Learned (P:L (c)).

PS on a majority of our test settings, we also demonstrate its effectiveness on the more complex Keepaway task in robot soccer [5] (Figure 1(c)).

2.2 VF+PS for a Complex Multiagent Task

Whereas previous successful results in the Keepaway task have limited learning to an isolated, infrequent decision that amounts to a turn-taking behavior among players (PASS), we expand the agents' learning capability to include the more ubiquitous action of moving without the ball (GETOPEN) [4]. GETOPEN induces a complex MDP, which is not suitable to be learned by VF approaches, such as the one employed by Stone *et al.* for learning PASS. Unlike PASS, there are multiple players executing GETOPEN at any instant of time. We provide a PS method for learning GETOPEN. As a result, we learn a composite behavior (PASS+GETOPEN) in which multiple agents execute *learned* policies simultaneously.

As reported in Figure 2, the learned GETOPEN policy (GO:L) matches the best hand-coded policy for this task (GO:HC) when paired with a hand-coded PASS policy (P:HC). Indeed GO:L outperforms GO:HC when paired with a random PASS policy (P:R). Importantly, we notice that PASS and GETOPEN can be learned simultaneously, signifying that a very complex multiagent task can be completely learned by decomposing it into components that are learned separately by VF and PS methods (Figure 2(c)).

3. PROPOSED WORK

In our empirical analysis, we identify three relevant classes of methods to include in our study: actor-critic algorithms, policy gradient methods, and VF methods using eligibility traces [3]. All these methods show some degree of resistance to deficient function approximation and partial observability; we aim to include them in our comparison of VF and PS methods. Intelligently determining the "transfer point" in our VF+PS algorithm, i.e., when to stop applying VF and switch to PS, constitutes yet another problem for proposed research.

One of the reasons PS methods such as evolutionary algorithms are not sample-efficient is because they have to negate the stochasticity in fitness estimates of candidate solutions by taking an average over multiple evaluations. Currently we are currently working on a statistical technique to reduce the number of such evaluations needed to get reliable estimates. Needless to say, we seek to extend our results from the Keepaway domain to other complex, realistic sequential decision making tasks.

4. REFERENCES

[1] S. Bhatnagar, R. Sutton, M. Ghavamzadeh, and M. Lee. Incremental natural actor-critic algorithms. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 105–112. MIT Press, Cambridge, MA, 2008.

[2] A. R. Cassandra, L. P. Kaelbling, and M. L. Littman. Acting optimally in partially observable stochastic domains. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, volume 2, pages 1023–1028, Seattle, Washington, USA, 1994. AAAI Press/MIT Press.

[3] S. Kalyanakrishnan and P. Stone. An empirical analysis of value function-based and policy search reinforcement learning. In *Proceedings of the Eighth International Conference on Autonomous Agents and Multiagent Systems*, May 2009. To appear.

[4] S. Kalyanakrishnan and P. Stone. Learning complementary multiagent behaviors: A case study. In *Proceedings of the Eighth International Conference on Autonomous Agents and Multiagent Systems*, May 2009. To appear as short paper.

[5] P. Stone, R. S. Sutton, and G. Kuhlmann. Reinforcement learning for RoboCup-soccer keepaway. *Adaptive Behavior*, 13(3):165–188, 2005.

[6] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.

[7] S. Whiteson and P. Stone. Evolutionary function approximation for reinforcement learning. *Journal of Machine Learning Research*, 7:877–917, May 2006.

Learning a Model of Speaker Head Nods using Gesture Corpora

Jina Lee
Institute for Creative Technologies
University of Southern California
13274 Fiji Way, Marina del Rey, CA 90292 USA
jlee@ict.usc.edu

During face-to-face conversation, the head is constantly in motion, especially during speaking turns [2]. These movements are not random; research has identified a number of important functions served by head movements [7] [5] [3] [4]. Head movements provide a range of information in addition to the verbal channel such as nods to show our agreement or shakes to express disbelief.

The goal of our work is to build a domain-independent model of speaker's head movements and use the model to generate head movements for virtual agents. To use the model for interactive virtual agents, it needs to operate in real-time. For this reason, we focus on features that are readily available at the time head movements are generated. In addition, we plan to make the model portable to other systems by using features such as part of speech tags that are easily obtainable even when using different language tools.

In this paper, we present a data-driven, automated approach to generate speaker nonverbal behavior, which we demonstrate and evaluate by learning when head nods should occur. Specifically, the approach uses a machine-learning technique (i.e. learning a hidden Markov model [8]) to create a head nod model from annotated corpora of face-to-face human interaction, relying on the linguistic features of the surface text. Figure 1 illustrates the overview of the procedures to learn the model. Once the patterns of when people nod are learned, then it can be used to generate head nods for virtual agents by encoding a new sample with the factors used for learning and feeding it to the model to obtain the most likely head movement.

1. HEAD NOD PREDICTION MODEL

1.1 Gesture Corpus

For this work, we used the AMI Meeting Corpus [1]. It is a set of multi-modal meeting records, which includes 100 meeting hours. The corpus includes annotations of meeting context such as participant IDs and topic segmentations as well as annotations on each participant's transcript and movements. Annotations of each meeting are structured in an XML format and are cross-referenced through meeting IDs, participant IDs, and time reference. For this work, we used the recordings of 17 meetings, each consisted of three to four participants, which adds up to be around eight hours of meeting annotation.

1.2 Data Alignment and Feature Selection

Among all the annotations included in the corpus, we used the transcript of each speaker, the dialog acts of each utterance, and the type of head movements observed while the utterance was spoken. The head types annotated in the corpus are: nod, shake, nodshake, other, and none. We also obtained the part of speech tags and phrase boundaries (e.g. verb phrases and noun phrases) by sending the utterances through a natural language parser. In addition, we combined the features from our previous work in Nonverbal Behavior Generator (NVBG) [6], which is a rule-based system that analyzes the agent's cognitive processing and the syntactic and semantic structure of the surface text to generates nonverbal behaviors for virtual humans. We looked for keywords that trigger the rules associated with head nods in NVBG and called those keywords *key lexical entities*. From the 17 meeting recordings we used, we collected 10,000 sentences and wrote a script to cross-reference the corresponding annotation files and aligned the features on a word level.

When training hidden Markov models, we want to keep the number of features low by eliminating uncorrelated features when given a limited number of data samples. Therefore, we reduced the number of features by counting the frequency of head nods that occurred with each feature and selected a subset of them. Based on the results of the frequency counts, the final features selected for training are:

- **Part of Speech**: Conjunction, Proper Noun, Adverb, Interjection, Remainder
- **Dialog Act**: BackChannel, Inform, Suggest, Remainder
- **Sentence Start**: y, n
- **Noun Phrase Start**: y, n
- **Verb Phrase Start**: y, n
- **Key Lexical Entities**: y, n

1.3 Training Process

To learn the head nod model, hidden Markov models (HMM) were trained. For this work, the input is a sequence of feature combinations representing each word. The sequential property of this problem led us to use HMMs to predict head nods. After aligning each word of the utterances with the selected features as described above, trigrams of these words were formed as the data set. For each trigram, the head type was determined by the majority vote method; if more than two out of three words co-occurred with a nod,

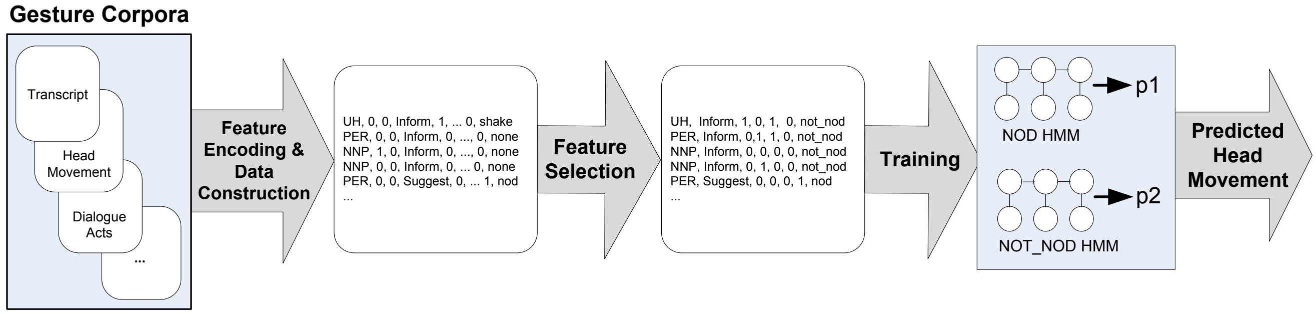

Figure 1: Overview of the head nod prediction framework. The information in the gesture corpus is encoded and aligned to construct the data set. The feature selection process chooses a subset of the features that are most correlated with head nods. Using these features, probabilistic sequential models are trained and utilized to predict whether or not a head nod should occur.

Measurement	Equation	Value
Accuracy	(tp+tn) / (tp+fp+tn+fn)	.8528
Precision	tp / (tp+fp)	.8249
Recall	tp / (tp+fn)	.8957
F-measure	2*precision*recall / (precision+recall)	.8588

Table 1: Measurements for the performance of the learned model.

the trigram was classified as a nod instance. To determine whether a trigram should be classified as a nod, we trained two HMMs, a 'NOD HMM' and a 'NOT_NOD HMM,' and fed the same trigram into both models and compared the results of each model.

To train a 'NOD HMM,' we collected all the positive instances of 'nod' trigrams from the entire set of trigrams. Then, we left out 20% of the 'nod' trigrams as a test set, which is used in the final evaluation step, and used the remaining 80% of the data for training. To train the 'NOD HMM,', we performed a standard 10-fold cross-over validation. Similarly, we repeated these steps to train the 'NOT_NOD HMM.' Finally, we ran the test set (20% of the entire data left out) through the 'NOD HMM' and 'NOT_NOD HMM' and classified each sample to have the head movement of whichever model produced a higher probability.

1.4 Results and Conclusion

To measure the performance of our learned model, we computed the accuracy, precision, recall, and F-measure of the learned model. Table 1 summarizes the results with the equations used for computing the measurements. The results show that the model can predict head nods with high precision, recall, and accuracy rate even without a rich markup of the surface text (i.e. only using the syntactic/semantic structure of the utterance and dialog act).

This work could be extended in several ways. Currently we are working on detecting the emotional state from each utterance and adding this into the feature set to investigate whether emotional data improves the learning. Further analysis of the linguistic structure may also be performed using additional language tools to extract features such as emphasis points and contrast points. We can also extend the work by learning the patterns of different head movements other than nods. Finally, we plan to conduct evaluations with human subjects to investigate if the head movements generated by the model are perceived to be natural.

Acknowledgments

This work was sponsored by the U.S. Army Research, Development, and Engineering Command (RDECOM), and the content does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.

2. REFERENCES

[1] J. Carletta. Unleashing the killer corpus: experiences in creating the multi-everything AMI meeting corpus. *Language Resources and Evaluation Journal*, 41(2):181–190, 2007.

[2] U. Hadar, T. J. Steiner, E. C. Grant, and F. C. Rose. Kinematics of head movements accompanying speech during conversation. *Human Movement Science*, 2:35–46, 1983.

[3] U. Hadar, T. J. Steiner, and F. C. Rose. Head movement during listening turns in conversation. *Journal of Nonverbal Behavior*, 9(4):214–228, 1985.

[4] D. Heylen. Challenges ahead: Head movements and other social acts in conversations. In *AISB 2005, Social Presence Cues Symposium*, 2005.

[5] A. Kendon. Some uses of the head shake. *Gesture*, 2:147–182(36), 2002.

[6] J. Lee and S. Marsella. Nonverbal behavior generator for embodied conversational agents. In *In Proceedings of the 6th International Conference on Intelligent Virtual Agents, Marina del Rey, CA*, pages 243–255. Springer, 2006.

[7] E. Z. McClave. Linguistic functions of head movements in the context of speech. *Journal of Pragmatics*, 32:855–878(24), June 2000.

[8] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

Coalition Structure Generation Utilizing Compact Characteristic Function Representations

(Extended Abstract)

Naoki Ohta
Kyushu University, Graduate School of ISEE
744, Motooka, Nishi-ku
Fukuoka, Japan
ohta@agent.is.kyushu-u.ac.jp

ABSTRACT

Forming effective coalitions is a major research challenge in AI and multi-agent systems. Coalition structure generation (CSG), which involves partitioning a set of agents into coalitions so that social surplus is maximized, is a central research topic due to its computational complexity. In this paper, we present new methods for CSG utilizing recently developed compact representation schemes for characteristic functions. We characterize the complexity of CSG under these representation schemes. In this context, the complexity is driven more by the number of "synergy coalition groups" than by the number of agents. Furthermore, we develop mixed integer programming formulations and show that an off-the-shelf optimization package can solve these problems quite efficiently.

Categories and Subject Descriptors

I.2.11 [Distributed Artificial Intelligence]: Multiagent systems; J.4 [Social and Behavioral Sciences]: Economics

General Terms

Theory, Economics

Keywords

Coalitional Game Theory, Optimization Problem

1. INTRODUCTION

Coalition formation is an important capability in automated negotiation among self-interested agents. Coalition structure generation (CSG) involves partitioning a set of agents into coalitions so that social surplus is maximized. This problem has become a popular research topic in AI and multi-agent systems. The CSG problem is equivalent to a complete set partition problem [7], and various algorithms for solving the CSG problem have been developed. Sandholm et al. propose an anytime algorithm with worst-case guarantees [6]. However, the worst-case time complexity is $O(n^n)$, where n is the number of agents. On the other hand, Dynamic Programming (DP) based algorithms [7, 3] are guaranteed to find an optimal solution in $O(3^n)$. Arguably, the state-of-the-art algorithm is the IP (integer partition) algorithm [4]. This is an anytime algorithm, which divides the search space into partitions based on integer partition, and performs branch & bound search. Although the worst-case time complexity for obtaining an optimal solution is $O(n^n)$, in practice, IP is much faster than DP based algorithms. Furthermore, Rahwan et al. introduce an extension of the IP algorithm that utilizes DP for preprocessing [2].

As far as we are aware, all existing works on CSG assume that the characteristic function is represented implicitly, and we have oracle access to the function—that is, the value of a coalition (or a coalition structure as a whole) can be obtained using some procedure. This is because representing an arbitrary characteristic function explicitly requires $\Theta(2^n)$ numbers, which is prohibitive for large n. However, characteristic functions that appear in practice often display significant structure, and it is likely that such characteristic functions can be represented much more concisely. Indeed, recently, several new methods for representing characteristic functions have been developed [1]. These representation schemes capture characteristics of interactions among agents in a natural and concise manner, and can reduce the representation size significantly. Surprisingly, to our knowledge, these representation schemes have not yet been used for CSG; this is what we set out to do in this paper. We examine synergy coalition groups (SCGs) [1] which is one of these compact representation schemes. The optimal choice of a representation scheme depends on the application.

Quite interestingly, we find that there exists some common structure among these cases: in essence, the problem is to find a subset of "SCGs" that maximizes the sum of rule values under certain constraints. For each case, we show that solving the CSG problem is NP-hard, and the size of a problem instance is naturally measured by the number of "SCGs" rather than the number of agents. Also, we give a mixed integer programming (MIP) formulation that captures this structure. We show that an off-the-shelf optimization package (CPLEX) can solve the resulting MIP problem instances quite efficiently.

2. MODEL

Let $A = \{1, 2, \ldots, n\}$ be the set of agents. A characteristic function $v : 2^A \to \Re$ assigns a value to each set of agents (coalition) $S \subseteq A$. We assume that each coalition's value

Cite as: Coalition Structure Generation Utilizing Compact Characteristic Function Representations (Short Paper), Naoki Ohta, Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009), Decker, Sichman, Sierra and Castelfranchi (eds.), May, 10–15, 2009, Budapest, Hungary, pp. XXX-XXX.

is nonnegative. A coalition structure CS is a partition of A, i.e., $CS = \{S_1, S_2, \ldots\}$ satisfies the following conditions: $\forall i, j\ (i \neq j),\ S_i \cap S_j = \emptyset,\ \bigcup_{S_i \in CS} S_i = A$. The value of a coalition structure CS, denoted as $V(CS)$, is calculated as follows: $V(CS) = \sum_{S_i \in CS} v(S_i)$. An optimal coalition structure CS^* is a coalition structure that satisfies the following condition:$\forall CS, V(CS^*) \geq V(CS)$.

3. CSG USING SCG

Conitzer et al. introduce a concise representation of a characteristic function called a synergy coalition group (SCG) [1]. The main idea is to explicitly represent the value of a coalition only when there exists some positive synergy.

Definition 1 (SCG). An SCG consists of a set of pairs of the form: $(S, v(S))$. For any coalition S, the value of the characteristic function is $v(S) = \max\{\sum_{S_i \in p_S} v(S_i) : p_S$ is a partition of S, i.e., all the S_i are disjoint and $\bigcup_{S_i \in p_S} S_i = S$, and for all the S_i, $(S_i, v(S_i)) \in SCG\}$. To avoid senseless cases that have no feasible partitions, we require that $(\{a\}, 0) \in SCG$ whenever $\{a\}$ does not receive a value elsewhere in SCG.

Thus, if the value of a coalition S is not given explicitly in SCG, it is calculated from the possible partitions of S. Using this original definition, we can represent only super-additive characteristic functions. But, if the characteristic function is super-additive, solving CSG becomes trivial: the grand coalition (the coalition of all agents) is optimal. To allow for characteristic functions that are not super-additive, we add the following requirement on the partition p_S.

- $\forall p'_S \subseteq p_S$, where $|p'_S| \geq 2$, $(\bigcup_{S_i \in p'_S} S_i, v(\bigcup_{S_i \in p'_S} S_i))$ is not an element of SCG.

Example 1. Let there be five agents a, b, c, d, e and let $SCG = \{(\{a\}, 0), (\{b\}, 0), (\{c\}, 1), (\{d\}, 2), (\{a, b\}, 3), (\{a, b, c\}, 3)\}$. In this case, $v(\{a, b, c, d\}) = v(\{a, b, c\}) + v(\{d\}) = 5$. We cannot use $v(\{a, b\}) + v(\{c\}) + v(\{d\}) = 6$, because $\{a, b\} \cup \{c\} = \{a, b, c\}$ appears in SCG.

The (modified) SCG can represent any characteristic function, including characteristic functions that are non-super-additive, or even non-monotone. This is because in the worst case, we can explicitly give the value of every coalition. Due to the additional condition, only these explicit values can then be used to calculate the characteristic function.

We show that when searching for CS^*, we need to consider only the coalitions that are explicitly described in SCG.

Theorem 1. There exists a coalition structure CS for which $V(CS) = V(CS^*)$ and $\forall S \in CS, (S, v(S)) \in SCG$.

We omit the proofs in this report.

Due to Theorem 1, finding CS^* is equivalent to a weighted set packing problem—equivalently, to the winner-determination problem in combinatorial auctions [5], where each agent is an item and each coalition described in SCG is a bid.

Theorem 2. When the characteristic function is represented as an SCG, finding an optimal coalition structure is NP-hard. Moreover, unless $\mathcal{NP} = \mathcal{ZPP}$, there exists no polynomial-time $O(|SCG|^{1-\epsilon})$ approximation algorithm for any $\epsilon > 0$.

Definition 2 (MIP formulation of CSG for SCG). The problem of finding CS^* can be modeled as follows.

$$\begin{aligned} max & \sum_{(S,v(S)) \in SCG} v(S) \cdot x(S) \\ s.t. & \ \forall a \in A, \sum_{S \ni a} x(S) = 1, \\ & x(S) \in \{0, 1\} \end{aligned}$$

$x(S)$ is 1 if S is included in CS^*, 0 otherwise.

In this formulation (which corresponds to a standard winner determination formulation), the number of binary variables is equal to $|SCG|$, and the number of constraints is equal to the number of agents.

Our methods can solve a problem with 100 agents and 100 $SCGs$ in less than 10 millisecond.

4. CONCLUSION

We showed that coalition structure generation can scale up significantly when the characteristic function is represented using recently developed a compact representation scheme which is called SCGs . For this case, we proved that the problem is NP-hard and inapproximable, and developed MIP formulations. Experimental results illustrated that while the state-of-the-art algorithm, which does not make use of compact representation, requires around 90 minutes to solve a problem with 27 agents, our methods can solve a problem with 100 agents and 100 $SCGs$ in less than 10 millisecond. Future work includes developing anytime/approximation algorithms that utilize these representation schemes.

5. REFERENCES

[1] V. Conitzer and T. Sandholm. Complexity of constructing solutions in the core based on synergies among coalitions. Artificial Intelligence, 170(6):607–619, 2006.

[2] T. Rahwan and N. R. Jennings. Coalition structure generation: dynamic programming meets anytime optimisation. In Proc. of the 23rd Conf. on Artificial Intelligence (AAAI), pages 156–161, 2008.

[3] T. Rahwan and N. R. Jennings. An improved dynamic programming algorithm for coalition structure generation. In Proc. of the 7th Int. joint Conf. on Autonomous Agents and Multi-agent Systems (AAMAS), pages 1417–1420, 2008.

[4] T. Rahwan, S. D. Ramchurn, V. D. Dang, A. Giovannucci, and N. R. Jennings. Anytime optimal coalition structure generation. In Proc. of the 22nd Conf. on Artificial Intelligence (AAAI), pages 1184–1190, 2007.

[5] T. Sandholm. Algorithm for optimal winner determination in combinatorial auctions. Artificial Intelligence, 135(1-2):1–54, 2002.

[6] T. Sandholm, K. Larson, M. Andersson, O. Shehory, and F. Tohmé. Coalition structure generation with worst case guarantees. Artificial Intelligence, 111(1-2):209–238, 1999.

[7] D. Y. Yeh. A dynamic programming approach to the complete set partitioning problem. BIT Numerical Mathematics, 26(4):467–474, 1986.

Reputation-Based Decisions for Cognitive Agents (Thesis Abstract)

Isaac Pinyol
Artificial Intelligence Research Institute
Campus UAB, Bellaterra
Barcelona, Spain
ipinyol@iiia.csic.es

1. INTRODUCTION

Computational trust and reputation models have been recognized as key to design and implement multiagent systems [6]. These models manage and aggregate the information needed by agents to efficiently select partners in uncertain situations. In open multiagent systems agents have unknown intentions and thus, some kind of *interaction control* is necessary to ensure a well-fare society. Several approaches can be taken for this endeavor, each of them providing certain level of control. At the security level, the use of cryptography and digital signatures ensures privacy, integrity and authenticity of messages. At the organization level it is possible to define protocols and norms that agents must follow to interact, for instance, by defining electronic institutions. Finally, at the social level, reputation and trust models endow agents with a powerful social control artifact that permits them to *evaluate* potential partners considering certain criteria before the interaction is produced.

In recent years several reputation and trust models have been developed [10]. Most of them following game theoretical approaches prepared to deal with relatively simple environments. However, if we want to undertake problems found in socially complex virtual societies, more sophisticated trust and reputation systems based on solid cognitive theories are needed. One such cognitive theory is defined in [3].

This theory focuses on the impact that social evaluations have in the mental state of the agents and not on the computation of such evaluations. In this sense, the theory proposes that agents evaluate the performances of other agents according to certain criteria, and that these evaluations (social evaluations) can be only believed by the agents, only communicated by the agents or both believed and communicated. When a social evaluation is believed by a group of agents the theory refers to it as *image*. On the contrary, when a social evaluation circulates in the society it is refereed as *reputation*.

From this generic overview, the theory then develops a more individualistic vision. From a single agent, it describes a typology of possible decisions that autonomous agents can make involving social evaluations:

- *Epistemic decisions* cover the decisions about updating and generating social evaluations.
- *Pragmatic-strategic decisions* are decisions of how to behave with potential partners using social evaluations information, and thus, how agents use them to reason.
- *Memetic decisions* refer to the decisions of how and when to spread social evaluations.

Traditionally, the field of computational trust and reputation systems has been focused on developing and formalizing models as providers of social evaluations: on epistemic decisions. However, little attention has been paid to pragmatic-strategic and memetic decisions. This doctoral thesis embraces then these two *types* of decisions.

Currently, agents' decisions of how to use reputation information and how and when to spread it have been designed *ad-hoc* lacking any systematic or formal procedure. We claim that due to the cognitive nature of social evaluations, when facing complex societies pragmatic-strategic and memetic decisions can be as important as epistemic decisions. From this perspective, for a cognitive agent, the way a social evaluation is build can have the same importance as the final evaluation.

Under this scenario the thesis analyzes the integration of a particular cognitive reputation model, Repage [11], into a cognitive agent architecture, *Belief, Desire Intention* (BDI).

Taking Repage as a paradigmatic example of cognitive reputation model, the integration allows us on the one hand, to properly formalize the logical reasoning process of a cognitive agent where reputation information is implicitly taken into account. Therefore, we provides a formal framework that directly faces pragmatic-strategic decisions. On the other hand, the logical reasoning process can be seen as a way to build arguments over agents' attitudes [7], and these arguments can be used in negotiation processes, persuasion, information exchange or for simply explanatory purposes. Thereby, each action, intention, desire and belief of an agent can be justified by building an argument that can include of course reputation information. Thus,we are able to offer a formal framework in which memetic decisions are formalized in the context of argumentation frameworks.

2. OBJECTIVES AND DEVELOPMENT

In this section we detail the main objectives of the thesis.

2.1 Integration of Repage in a BDI Architecture

As we mentioned, Repage[11] is a computational reputation system based on a cognitive theory or reputation [3], and whose main characteristic is the distinction between image (what agents believe) and reputation (what agents said) in terms of social evaluations. Taking this model as example, we specify a BDI agent architecture as a multicontext system where Repage information is completely integrated. The reasoning process of the agent shows how pragmatic-strategic reputation-based decisions are taken in a formal and systematic way. The research done in this specific task is described below:

- **Definition of probabilistic dynamic belief logic** Since Repage uses probabilities and actions when describing social evaluations, we defined a belief logic capable to capture all the information that Repage provides [9].
- **Specification of the BDI agent:** To capture in a formal dimension the reasoning process of an agent, we specify a BDI architecture where its belief based is described using the formalism stated above. This required to use also graded desires and intentions for a correct integration. The underlying ideas for specifying this model were taken from the work by Casali and colleagues [2].
- **Description of study cases:** One of the most important task is to describes scenarios to enhance the relevance and potential of the model, demonstrating the advantages of paying attention to the integration models in the reasoning process. We provide simple scenarios where this necessity is proved.

2.2 Argumentation on Social Evaluations

Focusing on memetic decisions, we take advantage of the BDI+Repage defined above to build a generic argumentation framework where reputation information is also present in the arguments. As we mentioned in the introduction, argumentation can be used in different interaction processes, like negotiation protocols or even simple information exchange, to give more strength to the communicated information. In this point our work include:

- **Definition of a formal argumentation framework:** As we mentioned, we defined our BDI+Repage as a multicontext system [5]. Some work have shown how multicontext systems can be used to build argumentation frameworks[7]. Then, taking this approach we defined also Repage as a multicontext system to define an argumentation framework where each agent's attitude can be justified also with information from Repage. This implies for instance that certain intention can be supported by desires and beliefs, and that these beliefs can be also justified by the information coming from Repage.
- **Study Cases:** After the generic framework is defined, we apply it to concrete negotiation or information exchange protocols.

2.3 Implementation and Simulations

The theoretical work is complemented with implementation and simulation results to show the performances of our models facing concrete scenarios. Thus, our work incorporates empirical results to show how the theoretical aspects can be instantiated with current platforms. We focus on the following aspects:

- **Prototypical Implementation:** The BDI+Repage model is implemented using JASON [1], a multiagent platform that offers the advantages of logic programming together with functionalities to define multiagent scenarios. Of course, a direct implementation of our theoretical models is not feasible, due the computational complexity. However, with appropriated simplifications and assumptions instantiations are more than possible, and even capable to provide massive simulation results.
- **Verification using Simulation:** Using a BDI+Repage implementation we put the model to work by defining multiagent environments where cognitive agents have to deal with bad/good reputation information in competitive markets, following some previous work ([8],[4]).

3. REFERENCES

[1] R. H. Bordini, J. F. Hübner, and M. Wooldridge. *Programming Multi-Agent Systems in AgentSpeak Using Jason*. John Wiley and Sons, Ltd, 2007.

[2] A. Casali, L. Godo, and C. Sierra. Graded models for bdi agents. In J. Leite and P. Torroni, editors, *CLIMA V, Lisboa, Portugal*, pages 18—33, 2004.

[3] R. Conte and M. Paolucci. *Reputation in artificial societies: Social beliefs for social order*. Kluwer Academic Publishers, 2002.

[4] A. di Salvatore, I. Pinyol, M. Paolucci, and J. Sabater. Grounding reputation experiments. a replication of a simple market with image exchange. In *Proceedings of the M2M'07, Marseille, France*, pages 32—45, 2007.

[5] F. Giunchiglia and L. Serafini. Multilanguage hierarchical logic (or: How we can do without modal logics). *Journal of AI*, 65:29—70, 1994.

[6] M. Luck, P. McBurney, O. Shehory, and S. Willmott. *Agent Technology: Computing as Interaction (A Roadmap for Agent Based Computing)*. AgentLink, 2005.

[7] S. Parsons, C. Sierra, and N. Jennings. Agents that reason and negotiate by arguing. *Journal of Logic and Computation*, 8(3):261–292, 1998.

[8] I. Pinyol, M. Paolucci, J. Sabater-Mir, and R. Conte. Beyond accuracy. reputation for partner selection with lies and retaliation. In *Proceedings of the MABS'07. Hawaii, USA.*, volume 5003 of *LNCS*, pages 128–140. Springer, 2007.

[9] I. Pinyol, J. Sabater-Mir, and P. Dellunde. Probabilistic dynamic belief logic for image and reputation. In *Proc. of the CCIA'08, Empuries, Spain.*, 2008.

[10] J. Sabater and C. Sierra. Review on computational trust and reputation models. *Artif. Intel. Rev.*, 24(1):33–60, 2005.

[11] J. Sabater-Mir, M. Paolucci, and R. Conte. Repage: Reputation and image among limited autonomous partners. *JASSS*, 9(2), 2006.

Scaling Multiagent Markov Decision Processes (Extended Abstract)

Scott Proper
Oregon State University
Corvallis, OR 97331-3202, USA
proper@eecs.oregonstate.edu

1. THREE CURSES OF DIMENSIONALITY

Markov Decision Processes (MDPs) have proved to be useful and general models of optimal decision-making in uncertain domains. However, approaches to solving MDP's using reinforcement learning that depend on storing the optimal value function and action models as tables do not scale to large state-spaces. Three computational obstacles prevent the use of standard approaches when dealing with problems with many variables. First, the state space (and time required for convergence) grows exponentially in the number of variables. This makes computing the value function impractical or impossible in terms of both memory and time. Second, the space of possible actions is exponential in the number of agents, so even one-step look-ahead search is computationally expensive. Lastly, exact computation of the expected value of the next state is slow, as the number of possible future states is exponential in the number of variables. These three obstacles are referred to as the three "curses of dimensionality".

Much prior work exists on the topic of scaling reinforcement learning to large state spaces. Many state abstraction and function approximation techniques exist. These techniques are a result of the desire to reduce the number of parameters used to represent the value function, and thus reduce memory requirements and time to converge. In addition to such techniques, methods to incorporate prior knowledge can be successful in speeding up convergence.

In [4] I addressed the three curses of dimensionality, providing solutions to each. To solve the problem of exploding state space, I introduced a kind of function approximation called "tabular linear functions". To solve action space explosion, I used a hill climbing technique over the action search space. To solve the problem of computing the expected value of the next state, I introduced ASH-learning, which is a model-based average reward algorithm that uses afterstates to reduce the number of future states it is necessary to examine.

2. ASSIGNMENT-BASED DECOMPOSITION

A common approach to dealing with issues of scaling is to take advantage of domain-specific structure. Consider the setting of cooperative multiagent reinforcement learning, where the agents are trying to cooperate to maximize a global reward signal. The structure of such multiagent domains can be taken advantage of by decomposing the states and actions.

In my thesis I propose a new technique for dealing with scaling issues; in particular, I consider the problem of coordinating multiple agents that share a common reward function through a centralized controller. Many domains can be decomposed into a set of weakly coupled agents, where each agent needs to know only limited information about the others. This allows significant scaling by limiting the amount of global information and facilitates local decision-making. I demonstrate how to implement these techniques using a variety of common value iteration-based reinforcement learning techniques, including model-free Q-learning and model-based methods.

Rather than addressing separate solutions to each of the three curses of dimensionality, I propose a single technique for decomposing certain reinforcement learning problems such that all the curses of dimensionality are addressed. In my thesis, I consider a problem of multiple agents and multiple tasks, where the agents are to be assigned to tasks in an optimal fashion. I call these problems multiagent assignment MDPs. Given an assignment, the agents might work almost independently of each other. However, the assignment can potentially change opportunistically. I also show that the optimal value function even in the simplest of such assignment tasks is not expressible as a coordination graph. The difficulty is enforcing conditions such as assigning at most two agents to each task to get a reward.

I present a new assignment-based decomposition [5] approach where the action-selection step is split into two levels. At the top level agents are assigned to tasks and at the lower level the tasks are performed by the teams with minimal dynamic coordination. This is similar to the hierarchical multiagent reinforcement learning of [3], except that I learn a value function only at the lower level and use search to optimize the higher level. My approach thus scales much better since it is not necessary to store an exponentially large value function at the top level.

I will also show how assignment-based decomposition may be expanded and scaled to solve difficult problems, with many agents and tasks. Fast search methods (such as those based on hill climbing or bipartite matching algorithms) are useful here as the space of possible assignments grows very large as the number of agents and tasks increases. In addition, I will show how using transfer learning and generalization techniques will allow a policy learned on only a few agents or tasks may scale to many agents and tasks.

3. COORDINATION GRAPHS

When decomposing the states and actions of cooperative agents, the issue of coordination of agent actions presents itself. Recent work using coordination graphs between agents has been shown to be successful here [1, 2]. The nodes of the graph represent agents and the arcs between them represent potential interactions between them. The long-term value of a joint action over all agents is expressed as a sum of all the interaction terms, where each such term is based on the actions and states of two agents. Bayesian network inference algorithms such as variable elimination and belief propagation have been adapted to finding the best joint action that maximizes the total reward.

Unfortunately, in many domains, coordination graphs are not static but change dynamically based on the states and actions of the agents. The approaches based on coordination graphs are adapted to dynamic state-based coordination [1, 2]. For example, in the approach of [2], a set of rules dictate which agent should coordinate with whom, and the value of a state is based on the current coordination graph.

I will demonstrate a technique for combining coordination graphs and assignment-based decomposition by adding a context-sensitive coordination graph at the lower level of the assignment-based decomposition. Doing this allows us some advantages over using either technique alone through separation of concerns. First, consideration of details such as collision avoidance can be delegated to lower levels, freeing the top level to focus on assignment decisions. Second, the coordination graph at the lower level can take advantage of knowing the assignment when making coordination decisions. Third, since the lower level value functions are used in making the higher level assignment decisions, collision information is indirectly percolated to the assignment level.

4. RELATIONAL TEMPLATES

In [4] I introduced a new description of a function approximation method called "Tabular Linear Functions" (TLFs). TLFs are a means of combining tables and linear functions in such a way as to preserve some of the best qualities of both. I will take this reseach further, describing how to expand and apply TLFs to a relational setting to create a function approximation method I call "Relational Templates". The use of relational templates greatly expands the kinds of domains that TLFs may be applied to.

I will also show how the use of relational templates facilitates transfer learning and the ability to generalize across multiple domains. Relational templates make be easily re-used across different (similar) domains. Also, parameters learned on one domain may often be transferred or generalized to multiple similar domains. I will show how to combine relational templates with assignment-based decomposition to easily scale a complex multiagent domain from few to many agents and tasks.

5. BIPARTITE SEARCH

Assignment-based decomposition solves many of the three curses of dimensionality, but introduces a new curse of it's own: how to scale the assignment search problem as the number of agents increases? With many agents and tasks, there are correspondingly many possible assignments. In [5], I describe three simple methods for search: exhaustive search, sequential greedy assignment, and swap-based hillclimbing. All of these methods have trade off solution speed and solution quality. I will introduce a new, more sophisticated approximate search technique for solving the assignment search problem: iterated bipartite assignment search. This search algorithm quickly provides a high-quality approximation of the true optimal assignment, allowing assignment-based decomposition to scale to much larger numbers of agents and tasks.

6. PRELIMINARY RESULTS

I have implemented assignment-based decomposition successfully on many domains, including product delivery domains, multiagent predator-prey domains, and real time strategy (RTS) game simulations. For this latter domain, I implemented a simple RTS game simulation on a 10x10 gridworld. Agents vary in number from 3-12 archers or infantry, and may face off against up to 4 enemy "tasks", either towers, knights, or ballista. These enemy units are more powerful than friendly units, and thus agents must coordinate in teams of up to three in order to destroy the enemy. Units are described by attributes such as location, hit points, damage, range, and mobility. I used a total reward version of ASH-learning [4] and assignment-based decomposition to solve this domain. Rewards were either +1 for a kill, -1 for a death, and -.1 per time step. As may be seen on this preliminary result in Figure 1, assignment-based decomposition greatly outperforms "flat" ASH-learning. Not only that, flat ASH-learning requires seven times as much CPU time to complete a single run.

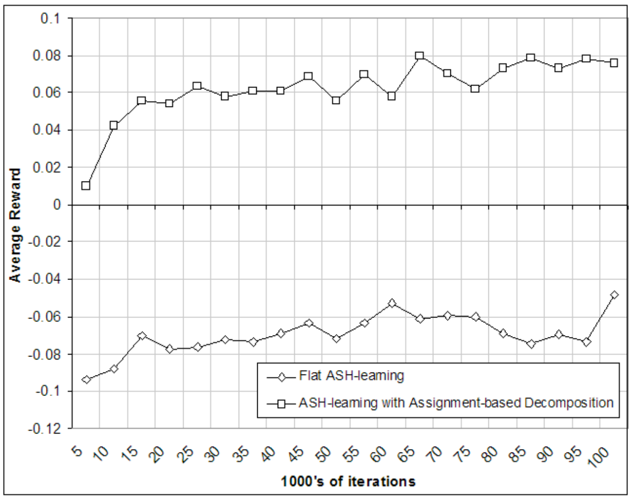

Figure 1: Comparison of flat vs. assignment-based decomposition in 6 agent vs. 2 task RTS domain.

7. REFERENCES

[1] C. Guestrin, S. Venkataraman, and D. Koller. Context specific multiagent coordination and planning with factored MDPs. In *AAAI '02: Proceedings of the 8th National Conference on Artificial Intelligence*, pages 253–259, Edmonton, Canada, July 2002.

[2] J. R. Kok and N. A. Vlassis. Sparse Cooperative Q-learning. In R. Greiner and D. Schuurmans, editors, *ICML '04: Proceedings of the 21st International Conference on Machine Learning*, pages 481–488, Banff, Canada, July 2004. ACM.

[3] R. Makar, S. Mahadevan, and M. Ghavamzadeh. Hierarchical multi-agent reinforcement learning. In *Proceedings of the 5th International Conference on Autonomous Agents*, pages 246–253, Montreal, Canada, 2001. ACM Press.

[4] S. Proper and P. Tadepalli. Scaling model-based average-reward reinforcement learning for product delivery. In J. Fürnkranz, T. Scheffer, and M. Spiliopoulou, editors, *ECML '06: Proceedings of the 17th European Conference on Machine Learning*, volume 4212 of *Lecture Notes in Computer Science*, pages 735–742. Springer, 2006.

[5] S. Proper and P. Tadepalli. Solving multiagent assignment markov decision processes. In *AAMAS '09: Proceedings of the 8th International Joint Conference on Autonomous Agents and Multiagent Systems (to be published)*, 2009.

Commitment Abstraction for Efficient Planning and Coordination in Stochastic Multi-agent Systems

(Extended Abstract of Doctoral Thesis)

Stefan J. Witwicki
Computer Science and Engineering
University of Michigan
Ann Arbor, MI 48109
witwicki@umich.edu

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent Systems*

General Terms

Algorithms, Design, Performance, Theory

Keywords

Multiagent Systems, Planning Under Uncertainty, Coordination, Commitments, Decentralized Markov Decision Processes, Agent Models, Policy Abstraction

1. INTRODUCTION

This thesis is motivated by the vast complexity of cooperative stochastic multi-agent planning, where agents can affect the transitions and rewards of one another, but in order to coordinate their interactions effectively, must account for the uncertainty in these actions. To combat this complexity, I exploit interaction structure in *weakly-coupled* problems to compute coordinated agent policies. I contend that when the degree of inter-agent dependence is sufficiently limited, the multi-agent problem can be solved more efficiently if it is broken up into (partially) decoupled subproblems: formulation of individual agent policies and coordination of abstract interactions. In support of this thesis, I develop an approach by which individual agents plan with local behavioral models that incorporate only those portions of negotiated nonlocal behavior that are needed for effective coordination.

2. PROBLEM DESCRIPTION

Figure 1 illustrates a multi-agent planning problem represented in the TAEMS language (as described in [1]). The objective is to plan policies for two autonomous vehicle agents that coordinate their execution of hierarchical tasks with uncertain durations so as to maximize expected quality within mission deadlines. We can model this example as a Decentralized Markov Decision Process (DEC-MDP) as discussed by Becker, Zilberstein, and Lesser [1]. With the characteristics that follow, I outline a class of weakly-coupled DEC-MDPs that is the focus of thesis.

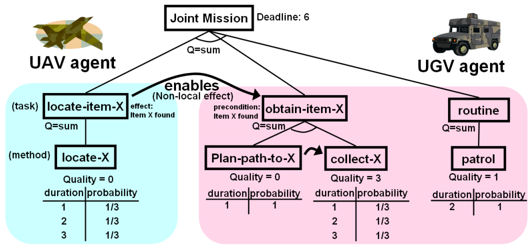

Figure 1: Autonomous Vehicle Example Problem

Temporal Grounding. Agents perform activities with well-defined (but often probabilistically uncertain) durations. The goals of the system are temporally constrained with strict deadlines. This is all modeled by a *finite-horizon* DEC-MDP for which *time* is a necessary state feature.

Decentralized Awareness. The agents do not have complete views of the world. Instead, each is only aware of a subset of information related to its individual activities. Technically, this corresponds to a *factored*, *locally fully observable* DEC-MDP where each agent's *local state* is composed of features related to the execution of its own tasks.

Structured Interactions. Agents have a limited influence on the outcomes of each others' activities. In particular, one agent may affect the local state transitions of another (sequentially, but not concurrently) through the *event-driven* manipulation of shared state features. In Figure 1, structured interaction occurs when the UAV *locates item X*, thereby *enabling* the UGV to successfully *obtain X*. I further assume an agent's non-local dependencies to be substantially less abundant than those within its local transitions.

Limited Planning Time. In many domains, it is important to plan coordinated behavior as soon as possible so as not to delay mission execution. Here, quickly-planned effective agent policies may be preferable to optimal policies that take longer to compute (or for large problems, are simply intractable). Thus, a solution to this class of problems is a method of policy computation that can (1) produce effective, coordinated (approximately-optimal) policies for problems both large and small, and (2) depending on problem difficulty, allow for trade-offs to be made between computation time and expected quality of computed behavior.

Previous work has solved related problems in restricted contexts [1, 4, 5], but no planning method (to date) constitutes a full solution to the class that I have outlined.

3. SOLUTION APPROACH

I propose an approach for coordinating interdependent agent activities through behavioral promises: *commitments*. A commitment encodes an agent's intention (and ability) to interact with other agents (by altering their local transitions). Because there is uncertainty in the system dynamics, a commitment also encodes time and probability information corresponding to when and with what likelihood other agents can expect the interaction to occur. Through negotiation of commitment values, the agents can plan their interactions and coordinate their individual behaviors around these planned interactions. The text that follows describes the components of my approach, cites results to date, and discusses planned research steps that I will take in completing my dissertation.

Commitment Modeling

Weakly-coupled problems involve highly-independent agents that interact with one another only in a limited capacity. Instead of considering all nonlocal dynamics, why not abstract only that which is relevant for planning an agent's limited interactions? Commitment models provide effective, compact approximations of external behavior. Although commitments have been studied in various classical planning domains (by Durfee and Lesser [2], for example), my problems call for the application of commitment theory [3] to Markov Decision Processes. I conjecture that planning with compact commitment-augmented local MDPs will allow weakly-coupled agents to coordinate complex, uncertain behavior efficiently. To test this theory, I have developed a method of augmenting MDPs with commitment models for *enablements* (as are present in Figure 1) [6], and plan to extend my models to represent other structured interactions.

Commitment Enforcement

Agents can compute policies by applying standard MDP solution techniques to their commitment-augmented local models. But because they are modeling behavioral expectations, these local policies need to satisfy the commitments that agents have promised. I have developed a method of commitment enforcement that, unlike previous work that injects artificial rewards and penalties to bias agents' actions, constrains policies directly to probabilistically adhere to committed interactions [6]. My linear programming approach automatically determines whether or not a given commitment selection is feasible and, if it is, computes optimal local behavior with respect to the commitment selection.

Commitment Negotiation

My commitment infrastructure transforms the problem of computing coordinated behavior into a search over the space of possible commitments. In fact, I have proven that, for an interesting subset of those problems, there exist commitments that (when enforced) yield globally-optimal joint policies [8]. For difficult problems, searching the commitment space exhaustively will be intractable. But I have developed an effective approximate algorithm that iteratively selects a set of commitments, builds local policies that enforce those commitments, estimates global quality, and repeats until greedily converging [7]. I have also demonstrated the scalability of my approach, and made analytical arguments [8] about the advantages that it has over existing methods, but I plan to verify these arguments with further empirical comparisons to demonstrate its robustness compared to other approaches (e.g. [1]).

Flexible Temporal Abstraction

In addition to approximating behavior, commitments provide a natural abstraction of the timing of uncertain interactions. A complete commitment model of the UAV agent (from Figure 1) would model every possible time (1,2, and 3) that *locate X* could occur. Consider instead representing this interaction with only a single time (time 3, for example) and the probability of *locate X* finishing *by* that time. As I have shown [7], these temporally-abstract commitment models maintain compactness as we scale to problems with increased complexity. Commitments of this sort are capable of encoding a single interaction time, all possible times, or any number of times in between [8], allowing for a flexibility of approximation that I am in the process of evaluating.

Policies Over Commitments

The last component of my approach is motivated by the fact that there may be dependencies between interactions that my present commitment models do not consider. For example, if there is a chain of enablement interactions (whereby Agent 1 enables Agent 2, allowing Agent 2 to enable Agent 3, etc.), I may be able to take advantage of this dependency structure by explicitly accounting for changes in expected behavior. If Agent 1 fails to enable Agent 2, Agent 3 should change its expectation of getting enabled by Agent 2. I envision an extension to my approach that allows such changes to be incorporated into a meta-level policy over commitments.

4. CONTRIBUTIONS

I expect that my completed dissertation will contribute:

- a principled framework for nonlocal abstraction in MDPs,
- an arsenal of LP-based policy formulation techniques for constraining agent behavior,
- a scalable, efficient, flexibly-approximate solution methodology for a relevant class of DEC-MDP problem, and
- a novel system of dynamic commitments.

5. REFERENCES

[1] R. Becker, S. Zilberstein, and V. Lesser. Decentralized Markov decision processes with event-driven interactions. *AAMAS*, pages 302–309, 2004.

[2] E. Durfee and V. Lesser. Partial global planning: A coordination framework for distributed hypothesis formation. *IEEE Trans. SMC*, 21(5):1167–1183, 1991.

[3] N. Jennings. Commitments and conventions: The foundation of coordination in multi-agent systems. *The Knowledge Engineering Review*, 8(3):223–250, 1993.

[4] J. Marecki and M. Tambe. On opportunistic techniques for solving decentralized Markov decision processes with temporal constraints. *AAMAS*, pages 830–837, 2007.

[5] R. Nair, P. Varakantham, M. Tambe, and M. Yokoo. Networked distributed POMDPs: A synthesis of DCOP and POMDPs. *AAAI*, pages 133–139, 2005.

[6] S. Witwicki and E. Durfee. Commitment-driven distributed joint policy search. *AAMAS*, 480–487, 2007.

[7] S. Witwicki and E. Durfee. Commitment-based service coordination. *Int. Journal of AOSE*, 3(1):59–87, 2009. [http://www.umich.edu/~witwicki/pub/IJAOSE.pdf].

[8] S. Witwicki and E. Durfee. Flexible approximation of structured interactions in decentralized Markov decision processes (extended abstract). *AAMAS*, 2009.

Towards Scaling Up Search Algorithms for Solving Distributed Constraint Optimization Problems

(Extended Abstract)

William Yeoh
Computer Science
University of Southern California
Los Angeles, CA 90089
wyeoh@usc.edu

My thesis will demonstrate that distributed constraint optimization (DCOP) search algorithms can be scaled up (= applied to larger problems) by applying the knowledge gained from centralized search algorithms.

1. INTRODUCTION

Agent-coordination problems can be modeled as distributed constraint optimization (DCOP) problems [10, 12, 20]. A DCOP problem consists of a set of agents, each responsible for taking on (= assigning itself) a value from their finite domains. The agents coordinate their value assignments subject to a set of constraints. Two agents are said to be constrained if they share a constraint. Each constraint has an associated cost which depends on the values taken on by the constrained agents. A complete solution is an assignment of values to all agents. The cost of a complete solution is the sum of the constraint costs of all constraints resulting from the given value assignments. Solving a DCOP problem optimally means to find a complete solution such that the sum of all constraint costs is minimized. Finding such a cost-minimal solution is NP-hard [10].

This model is rapidly becoming popular for formulating and solving agent-coordination problems [6, 7, 5]. As a result, DCOP algorithms that use search techniques such as ADOPT (Asynchronous Distributed Constraint Optimization) [10] have been developed.

2. CONTRIBUTIONS

Since solving DCOP problems is NP-hard, my research concentrates on finding intelligent ways to scale up DCOP search algorithms such that they can be used in larger applications. DCOP search algorithms can be viewed as distributed versions of centralized search algorithms with assumptions that are specific to DCOP problems. For example, the solution space (= space of all possible solutions) of DCOP problems is bounded by the number of agents in the problem. Therefore, some of the knowledge gained by researchers investigating centralized search algorithms might apply to DCOP search algorithms as well.

To avoid reinventing the wheel, my thesis will center around scaling up DCOP search algorithms by applying the knowledge gained from centralized search algorithms. I made a design choice to reuse the framework of ADOPT, which is one of the pioneering DCOP search algorithms, as the starting platform for the work in my dissertation. The motivation for this decision is that ADOPT has been extended very widely [9, 1, 11, 2, 14]. In particular, my contributions lie along two axes: (1) Memory availability of agents and (2) Requirement of solution optimality.

For problems where the agents have a *minimal amount of memory* and the *cost-minimal solution is required*, I introduced a new DCOP search algorithm called Branch-and-Bound ADOPT (BnB-ADOPT) in [16], that speeds up ADOPT by one order of magnitude for sufficiently large DCOP problems. BnB-ADOPT is a memory-bounded asynchronous DCOP search algorithm that uses the message passing and communication framework of ADOPT but changes the search strategy of ADOPT from best-first search to depth-first branch-and-bound search. Experimental results show that BnB-ADOPT is faster than ADOPT for sufficiently large DCOP problems because the available heuristics for these problems are often uninformed. The key contribution of this work is the identification and verification of depth-first branch-and-bound search instead of best-first search as the preferred search strategy for DCOP problems, which is consistent with findings for centralized search algorithms [19].

For problems where the agents have *more than the minimal amount of memory*, I introduced new caching schemes called MaxPriority, MaxEffort and MaxUtility in [18], that are tailored to DCOP search algorithms including ADOPT and BnB-ADOPT, and thus speed up both algorithms further. These caching schemes make use of the lower and upper bounds maintained by agents in ADOPT and BnB-ADOPT, as well as the knowledge of which search strategy is employed by ADOPT and BnB-ADOPT. Our experimental results show that the MaxEffort and MaxUtility schemes perform better than the other schemes for ADOPT, and the MaxPriority scheme is generally no worse than the other schemes for BnB-ADOPT. The speedup from caching for ADOPT is significantly larger than that for BnB-ADOPT since ADOPT needs to re-acquire information that was purged due to memory limitations. The key contribution of this work is the investigation of the different caching schemes and the identification of preferred schemes for the different algorithms. In general, these schemes should apply to other DCOP search algorithms as well since they also maintain lower and upper bounds on the solution quality.

For problems where the *cost-minimal solution is not required*, I introduced two approximation mechanisms for ADOPT and BnB-ADOPT that trade off solution quality for faster computation time in [17]. The approximation mechanisms, namely the Relative Error Mechanism and the Weighted Heuristics Mechanism, provide relative error bounds (i.e. a percentage off the minimal cost). These mechanisms complement existing mechanisms that only allow absolute error bounds (i.e. an absolute off the minimal cost). Additionally, experimental results show that the Weighted Heuristics Mechanism dominates the other mechanism. The key contribution of this work is the introduction of the Weighted Heuristics Mechanism, which should also apply to other DCOP search algorithms that use heuristics to guide their search. This mechanism was motivated by Weighted A* [13], an approximation algorithm based on the centralized search algorithm A* [3].

I conducted my experiments in three problem types that are commonly used to evaluate DCOP algorithms. The three problem types are the problem of coloring graphs, the problem of allocating targets to sensor networks and the problem of scheduling meetings. I measured the runtime of the algorithms using two commonly used metrics, namely time slices called cycles [10] and non-concurrent constraint checks [8].

3. FUTURE WORK

So far, my contributions only apply to static problems (i.e. problems that do not change over time). To complete my thesis, I plan to extend my work to dynamic problems by developing new DCOP search algorithms that operate efficiently in these environments. Specifically, I have two objectives in mind: (1) algorithms that find cost-minimal solutions of dynamic DCOP problems; and (2) algorithms that find error-bounded solutions of DCOP problems that are most similar to the solutions of the problems before they changed (due to changes in the environment). I plan to measure the similarity of two solutions by the number of agents that take on the same value in both solutions.

To achieve the first objective, I plan to develop DCOP search algorithms that perform a new search every time the DCOP problem changes but reuse information from the previous searches. Therefore, these algorithms should be faster than those that run each search from scratch. This plan is motivated by incremental centralized search algorithms [15].

To achieve the second objective, I plan to develop a DCOP search algorithm that employs limited discrepancy search [4]. Limited discrepancy search searches for solutions in the order of increasing numbers of discrepancies, i.e. numbers of agents that take on values different from their previous values, and is thus ideally suited for finding the most similar error-bounded solution.

Acknowledgments

This research has been partly supported by an NSF award to Sven Koenig under contract IIS-0413196. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of the sponsoring organizations, agencies, companies or the U.S. government.

4. REFERENCES

[1] E. Bowring, M. Tambe, and M. Yokoo. Multiply-constrained distributed constraint optimization. In *Proceedings of AAMAS*, pages 1413–1420, 2006.
[2] J. Davin and P. Modi. Hierarchical variable ordering for multiagent agreement problems. In *Proceedings of AAMAS*, pages 1433–1435, 2006.
[3] P. Hart, N. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, SSC4(2):100–107, 1968.
[4] W. Harvey and M. Ginsberg. Limited discrepancy search. In *Proceedings of IJCAI*, pages 607–615, 1995.
[5] R. Junges and A. Bazzan. Evaluating the performance of DCOP algorithms in a real world, dynamic problem. In *Proceedings of AAMAS*, pages 599–606, 2008.
[6] V. Lesser, C. Ortiz, and M. Tambe, editors. *Distributed Sensor Networks: A Multiagent Perspective*. Kluwer, 2003.
[7] R. Maheswaran, M. Tambe, E. Bowring, J. Pearce, and P. Varakantham. Taking DCOP to the real world: Efficient complete solutions for distributed event scheduling. In *Proceedings of AAMAS*, pages 310–317, 2004.
[8] A. Meisels, E. Kaplansky, I. Razgon, and R. Zivan. Comparing performance of distributed constraints processing algorithms. In *Proceedings of the Distributed Constraint Reasoning Workshop*, pages 86–93, 2002.
[9] P. Modi and S. Ali. Distributed constraint reasoning under unreliable communication. In W. Zhang and V. Sorge, editors, *Frontiers in Artificial Intelligence and Applications*, volume 112, pages 141–150. IOS Press, 2004.
[10] P. Modi, W.-M. Shen, M. Tambe, and M. Yokoo. ADOPT: Asynchronous distributed constraint optimization with quality guarantees. *Artificial Intelligence*, 161(1-2):149–180, 2005.
[11] F. Pecora, P. Modi, and P. Scerri. Reasoning about and dynamically posting n-ary constraints in ADOPT. In *Proceedings of the Distributed Constraint Reasoning Workshop*, pages 57–71, 2006.
[12] A. Petcu and B. Faltings. A scalable method for multiagent constraint optimization. In *Proceedings of IJCAI*, pages 1413–1420, 2005.
[13] I. Pohl. First results on the effect of error in heuristic search. *Machine Intelligence*, 5:219–236, 1970.
[14] M. Silaghi and M. Yokoo. Dynamic DFS tree in ADOPT-ing. In *Proceedings of AAAI*, pages 763–769, 2007.
[15] X. Sun and S. Koenig. The fringe-saving A* search algorithm – a feasibility study. In *Proceedings of IJCAI*, pages 2391–2397, 2007.
[16] W. Yeoh, A. Felner, and S. Koenig. BnB-ADOPT: An asynchronous branch-and-bound DCOP algorithm. In *Proceedings of AAMAS*, pages 591–598, 2008.
[17] W. Yeoh, S. Koenig, and X. Sun. Trading off solution cost for smaller runtime in DCOP search algorithms (short paper). In *Proceedings of AAMAS*, pages 1445–1448, 2008.
[18] W. Yeoh, P. Varakantham, and S. Koenig. Caching schemes for DCOP search algorithms. In *Proceedings of AAMAS*, 2009.
[19] W. Zhang and R. Korf. Performance of linear-space search algorithms. *Artificial Intelligence*, 79(2):241–292, 1995.
[20] W. Zhang, G. Wang, Z. Xing, and L. Wittenberg. Distributed stochastic search and distributed breakout: Properties, comparison and applications to constraint optimization problems in sensor networks. *Artificial Intelligence*, 161(1-2):55–87, 2005.