Models and Algorithms for Ad Hoc Coordination in Multiagent Systems

Stefano V. Albrecht
School of Informatics
University of Edinburgh
s.v.albrecht@sms.ed.ac.uk

Abstract

The goal of this thesis is to develop new models and algorithms for the ad hoc coordination problem. Therein, the problem is to design an autonomous agent which is able to achieve optimal flexibility and efficiency in a multiagent system in which the behaviour of the other agents is not a priori known. This problem is motivated by applications such as human-machine interaction, robot search and rescue, and financial markets, which often provide no mechanisms for prior behavioural coordination (e.g. communication protocols, shared world models, etc.). This short paper gives a brief account of the current state of the thesis and future work.

1 Introduction

A multiagent system consists of two or more decision makers, called agents, who choose actions in a common environment to achieve certain goals. Many multiagent systems are developed with a homogeneous setting in mind, which means that all agents use the same decision algorithm and are a priori aware of this fact. However, there are important applications for which this assumption may not be adequate, including human-machine interaction, robot search and rescue, and financial markets. In these applications, there are often no mechanisms for prior coordination of the agent’s behaviours, e.g. communication protocols, central task allocation, and shared world models. Therefore, it is important that an agent be able to effectively coordinate its actions without knowing a priori how the other agents behave.

This problem is hard since the agents may exhibit a large variety of behaviours. General-purpose algorithms for multiagent systems are often impracticable, either because they take too long to produce effective policies or because they rely on prior coordination of behaviours (Albrecht and Ramamoorthy 2012). The goal of this thesis is to address this problem by developing new models and algorithms for multiagent systems in which the behaviour of other agents is not a priori known. Specifically, we consider the ad hoc coordination problem (Albrecht and Ramamoorthy 2013), in which the goal is to design an autonomous agent, called the ad hoc agent, which is able to achieve optimal flexibility and efficiency in a multiagent system that admits no prior coordination between the ad hoc agent and the other agents.

2 Related Work

Recent works that address ad hoc coordination include (Bowling and McCracken 2005; Dias et al. 2006; Stone et al. 2010). However, the assumptions made by the solutions proposed therein imply that they only address certain aspects of the larger problem. For example, in (Bowling and McCracken 2005; Dias et al. 2006) it is assumed that all agents follow complex pre-specified plans which define roles and synchronised action sequences for each role, and in (Stone and Kraus 2010; Stone, Kaminka, and Rosenschein 2010; Agmon and Stone 2012) it is assumed that the other agents’ behaviours are a priori known and fixed (i.e. they do not learn), and that all agents, including the ad hoc agent, have common payoffs. Furthermore, the problem descriptions in these works are of a procedural nature, associated with the specific tasks considered therein. Thus, there is a need for a formal model of the ad hoc coordination problem.

A related problem is known in game theory as the incomplete information game, in which each player has some private information which is relevant to its decision making. What relates this problem to ad hoc coordination is the fact that no player knows the private information and, therefore, behaviour of any other player. Harsanyi (Harsanyi 1967) introduced Bayesian games in which the private information of a player is abstractly represented by its type, admitting a solution in the form of the Bayesian Nash equilibrium. However, while the notion of private information is useful to describe ad hoc coordination, the learning processes studied therein (e.g. (Kalai and Lehrer 1993; Dekel, Fudenberg, and Levine 2004)) are not directly applicable, since the focus has traditionally been on equilibrium attainment but not on efficiency.

3 Contributions

As a first step towards understanding the ad hoc coordination problem, I compared five multiagent learning algorithms in a comprehensive set of ad hoc coordination problems, using criteria such as convergence rate, social welfare and fairness, and equilibrium attainment (Albrecht and Ramamoorthy 2012). The compared algorithms, while representing major approaches in the field (e.g. opponent modelling, policy hill-climbing, regret-minimisation) were originally developed in a heterogeneous setting. The contribution of this work is to show how these algorithms perform in ad hoc coordination.
proponents with possibly heterogeneous settings, which is valuable information for the design of ad hoc agents. The results show that there is, in fact, no clear winner amongst the compared algorithms, each being superior with respect to some criteria but inferior with respect to others.

As was discussed in Section 2, there is currently no formal model of the ad hoc coordination problem. In this thesis, I propose to conceptualise this problem formally using a game-theoretic model, called the stochastic Bayesian game, in which the behaviour of a player is determined by its private information, or type (Albrecht and Ramamoorthy 2013). Based on this model, I derive a solution, called Harsanyi-Bellman Ad Hoc Coordination (HBA), which utilises a set of user-defined types in a planning procedure to find optimal actions in the sense of Bayesian Nash equilibrium and Bellman optimal control. I show how HBA can be implemented as a reinforcement learning procedure and provide empirical evaluations in a multiagent logistics domain and a human-machine experiment, showing that HBA is more flexible and efficient than several alternative algorithms.

The idea behind HBA is to reduce the complexity of the problem by assuming that there is a latent set of policies for each agent (their type spaces) and a latent distribution over these policies, and that an expert can provide informed guesses as to what these policies might be (the user-defined type spaces). Two important theoretical questions in this regard are under what conditions HBA is guaranteed to learn the latent type distribution, and how closely the user-defined type spaces must approximate the true type spaces in order for HBA to effectively solve its task. I address both of these questions in (Albrecht and Ramamoorthy 2014), by analysing the convergence conditions of various posterior formulations and by proving several termination guarantees for HBA. I also propose a novel characterisation of type space optimality which is based on the notion of probabilistic bisimulation.

4 Future Work

HBA computes a Bayesian posterior distribution over the user-defined types based on the history of observed states and actions. Part of this posterior is a prior distribution in which the user specifies initial beliefs as to the likelihood of the various types. The question I am currently investigating is if HBA can automatically derive prior beliefs from the user-defined type spaces to further maximise its efficiency. The plan is to analyse several alternative formulations for the prior, both empirically and theoretically, and to conduct a human-machine experiment to investigate how humans may assign prior beliefs over types. This work is done in collaboration with Dr. Jacob Crandall at the Masdar Institute of Science and Technology, and I expect it to be completed in mid-2014 (presumably before the doctoral consortium).

There are two more open questions which I would like to address in my thesis research:

Automatic type generation: Currently, HBA assumes that an expert can provide manually specified types for the problem at hand. However, this can be a cumbersome task in complex domains. I would like to investigate how HBA could generate useful types from the problem description, so that the burden of having to manually specify types can be alleviated, or perhaps eliminated altogether.

Hierarchical type specification: As we apply HBA to increasingly complex problems, it becomes apparent that the types, too, become increasingly complex and difficult to specify. I believe this complexity could be reduced by a hierarchical specification of types, in which types consists of smaller sub-types which are easier to specify.

Based on past experience, I expect each of these projects to take between 4 and 6 months. However, at the time of writing the present paper, it is unclear when and in what order I will start to work on them. This depends primarily on when I can complete my current work (on prior beliefs) and whether new, perhaps more urgent questions arise from it.

References


