

# Negotiation Algorithms for Large Agreement Spaces

Dave de Jonge

Artificial Intelligence Research Institute IIIA-CSIC  
 Bellaterra, Catalonia, Spain  
 davedejonge@iia.csic.es

## 1 Introduction

In multiagent systems (MAS) the outcome of the actions of an agent usually depends on the actions of other agents, which may have conflicting goals. Moreover, since the other agents may be unknown and may not be benevolent, an agent generally cannot assume the other agents would be willing to help without getting anything in return. If each agent would simply take those actions that are individually best, the result will often be sub-optimal for each of them, like in the well known prisoner's dilemma. Therefore, agents in a MAS need to negotiate on what actions each will take.

Maximizing a utility function for a set of independent agents is also the goal of Distributed Constraint Optimization Problems (DCOP), but these problems are fundamentally different from negotiation problems, since DCOPs assume there is one global function to be optimized and the agents co-operate in order to find a good solution. In cases where the agents have their own private utility function, are selfish, and are unknown, one cannot apply a DCOP algorithm.

One could suggest to apply an optimization algorithm to find the solution for which the the sum of the individual utility functions is maximized. However, since agents are selfish and autonomous, there is no guarantee that all agents will agree with such a solution, as a strict subset of the agents may come to a solution in which each agent achieves higher utility than it would obtain from the socially optimal solution. Moreover, agents may disagree on what is the most fair solution.

Most existing negotiation algorithms assume simple scenarios where the utility of each agent is modeled as a linear function that is easy to calculate. They mostly work only for bilateral negotiations [Lai *et al.*, 2008] and assume the range of options is small, or they require the presence of a trusted, impartial mediator [Klein *et al.*, 2003]. They assume that for any given utility value it is always possible to find a proposal that indeed yields this utility value [Faratin *et al.*, 2000]. These assumptions are rarely satisfied in real-world situations. In this thesis we focus on complex problems and address a number of realistic assumptions that make the application of existing negotiation algorithms unfeasible:

- The space of solutions is huge.
- Utility is non-linear and difficult to calculate.
- The environment is only partially observable.
- The environment changes due to actions of others.

- Decisions have to be made within a limited time frame.
- Solutions may involve many agents, possibly human.

Many difficult problems belong to this class, e.g: school timetable construction [Willemsen, 2002], route scheduling for package delivery companies [MJC2, 2011], automated stock trading, or the online purchase of holiday packages.

We introduce a new family of negotiation algorithms named Negotiation Based Branch & Bound(NB<sup>3</sup>) that apply a Branch & Bound search tree to search for solutions that can be proposed to the other agents. We test it on two test cases: the Negotiating Salesmen Problem and the strategic board game Diplomacy. The first is a new, artificial problem that adds negotiation to the well known Traveling Salesman Problem. The second is an existing strategy game, normally played by humans, in which negotiation skills are essential.

## 2 The NB<sup>3</sup> Algorithm

In order to solve the problem of negotiating in complex environments, we propose to use a Branch & Bound search algorithm to explore the space of possible agreements, and find those that may be good enough to propose to other agents. An agent in the MAS may run this algorithm to negotiate with the other agents. Since the MAS is assumed to be heterogeneous, the algorithm does not assume anything about the algorithms that the other agents are running. They may also be running NB<sup>3</sup>, or any other negotiation algorithm, or they may even be humans.

We assume that each agent has a set of possible actions it can execute, and a joint plan is defined as a set of actions of several agents. The search space is the space of all such joint plans. The execution of a joint plan yields a certain amount of utility for each agent. Therefore, each agent has its own utility function over the space of joint plans and each agent aims to maximize its own utility function. The higher the utility a certain plan yields to an agent  $\alpha$ , the more likely it is that  $\alpha$  will agree with the execution of that plan.

An agent that runs the NB<sup>3</sup> algorithm explores the space of joint plans by splitting a search tree using a best-first search strategy. Each node in the tree represents a subset of the space of joint plans, and each arc between two nodes represents a possible action of any of the agents. The branch from the root to a given node  $n$  represents the partial joint plan consisting of all actions associated with the arcs in the branch.

For each node, the agent stores an upper bound and a lower bound for the utility for each agent in the MAS. These bounds are calculated based on a model the agent has about the preferences of the other agents. If for a given node the lower bounds for each agent participating in the plan are high enough, our agent will propose the plan to the other agents. If for a given node, for any of the agents participating in the plan the upper bound is too low, then this node can be pruned as it is unlikely that that agent will accept the proposal. The question whether a node is good enough to be proposed or not is decided by a time based concession strategy. The closer to the deadline of the negotiations, the more the agent is willing to concede to others.

The heuristic that determines which node to split depends on the proposals that other agents have made so far. The more another agent has conceded, the more the algorithm will be inclined to search for proposals involving that agent. In this way, we see that search and negotiation have become intertwined: not only do the proposals that are made depend on the search, but also the search is influenced by the proposals that have been made. This is an entirely novel concept in the field of automated negotiations.

Another unusual approach we take is that of the ‘*Free Communication Protocol*’. Most negotiation algorithms assume the agents obey a strict protocol like the Alternating Offers Protocol. We think however that it is unrealistic to assume that other, unknown, agents will obey the same protocol (especially if they are humans). Therefore we assume that any agent can make any offer at any time, so after making a proposal an agent does not have to wait for a counter proposal and can immediately continue making more proposals. Moreover, when an agent receives a proposal, it may accept it, reject it, or simply ignore it (i.e. not reply at all). If an agent makes a proposal but does not directly get a reaction, it is up to the agent itself to decide how long to wait for a reaction.

### 3 Negotiating Salesmen Problem

We have defined a new problem to be used as a test case for testing negotiation algorithms. We call this problem the *Negotiating Salesmen Problem* (NSP). It is a variant of the Traveling Salesman Problem (TSP), but with the main difference that there is more than one salesman, each of which is only interested in minimizing its own, individual path length.

The idea is that there is a set of cities, and each one of them has to be visited once by at least one of the salesmen. Initially, each city is assigned to one salesman that has to visit it. The salesmen are however allowed to exchange their cities, so that they may be able to decrease the distances they have to cover. For example: if a city  $v$  is assigned to agent  $\alpha$ , but  $\alpha$  prefers to visit another city  $v'$ , which is assigned to agent  $\beta$ , then  $\alpha$  may propose to  $\beta$  to exchange  $v$  for  $v'$ . If  $\beta$  however also prefers to have  $v'$  over  $v$  it will not accept this deal. And if no other agent wants to accept  $v$  either, then  $\alpha$  is obliged to travel along city  $v$ .

This domain is interesting because of its high complexity: to calculate the value of one single proposal, one needs to solve a TSP for every agent involved in the proposal. The number of possible proposals is very large, so one may need

to evaluate a lot of proposals before finding one that is good enough to propose or accept.

### 4 Diplomacy

Diplomacy is a popular board game that has existed since the 1950's and is nowadays mostly played over the Internet. It is played by seven players, that each represent one of the ‘Great Powers’ of Europe at the beginning of the 20th century. Each player owns a set of armies distributed over a map of Europe and the goal is to conquer at least half of Europe. Although players are essentially selfish, they need to form coalitions in order to advance in the game. Therefore, negotiation is an essential skill in this game.

This game is very interesting for our research, since it offers a very large space of possible actions and agreements, and the outcome of the actions of an individual player highly depends on the actions of the other players. Like in many other games there is no explicitly known utility function that determines the value of a given board configuration. Therefore, one needs to apply a heuristic approach. Moreover, the game board constitutes a complex asymmetrical graph (contrary to games like chess where the board is a square grid), so we cannot simplify the game by exploiting symmetries. Many attempts have been made to create automated agents that play Diplomacy, but most of them are not able to negotiate at all, and if they do, their negotiation skills are poor.

### 5 Progress

Experiments with the Negotiating Salesmen Problem have shown good results that indicate the NB<sup>3</sup> algorithm is indeed able to negotiate successfully under hard conditions. We are currently implementing a version of the algorithm for Diplomacy.

### 6 Acknowledgments

Supported by the Agreement Technologies CONSOLIDER project, and CHIST-ERA project ACE, and EU project 318770, PRAISE

### References

- [Faratin *et al.*, 2000] Peyman Faratin, Carles Sierra, and Nicholas R. Jennings. Using similarity criteria to make negotiation trade-offs. pages 119–126, 2000.
- [Klein *et al.*, 2003] M. Klein, P. Faratin, H. Sayama, and Y. Bar-Yam. Protocols for negotiating complex contracts. *Intelligent Systems, IEEE*, 18(6):32 – 38, nov-dec 2003.
- [Lai *et al.*, 2008] Guoming Lai, Katia Sycara, and Cuihong Li. A decentralized model for automated multi-attribute negotiations with incomplete information and general utility functions. *Multiagent Grid Syst.*, 4:45–65, January 2008.
- [MJC2, 2011] MJC2. [http://www.mjc2.com/transport\\_logistics\\_management.htm](http://www.mjc2.com/transport_logistics_management.htm), 2011.
- [Willemen, 2002] RJ Willemen. *School timetable construction : algorithms and complexity*. T.U. Eindhoven, 2002.