Available online at www.sciencedirect.com
SCIENCE DIRECT®
ELSEVIER
Decision Support Systems 39 (2005) 197–217
Decision Support Systems
www.elsevier.com/locate/dsw

Mechanism design for software agents with complete information[☆]

Thomas C. O'Connell[a,*,1], Richard E. Stearns[b]

[a] Department of Mathematics and Computer Science, Skidmore College, Saratoga Springs, NY 12866,USA
[b] Department of Computer Science, University at Albany, SUNY, Albany, NY 12222,USA

Available online 29 November 2003

Abstract

We investigate the mechanism design problem when the agents and the mechanism have computational restrictions. In particular, we examine how results in the mechanism design literature are affected when the social choice rule requires the mechanism to solve a computationally difficult optimization problem. Both dominant strategy and Nash implementation are considered for a multiagent version of the maximum satisfiability problem. We show that the best a mechanism can guarantee is that at least half of the maximum number of simultaneously satisfiable agents will be satisfied by the outcome. Our analysis highlights some of the difficulties that arise in applying results from mechanism design to computational problems. In particular, our results show that using approximation in multiagent settings can be much less successful than in traditional computational settings because of the game theoretic guarantees required of the outcomes.

Keywords: Algorithmic mechanism design; Approximation algorithms; Game theory; Implementation theory

1. Introduction

With the advent of Internet computing and electronic commerce, there has been increasing interest in computational systems, referred to as *multiagent systems*, that involve the interaction of many different computer programs. These programs, or *software agents*, may be written by different people or companies with different goals in mind. In other words, the programs can be viewed as self-interested. Not surprisingly, the design and analysis of multiagent systems involves the tools of game theory and mechanism design (see Refs. [12,18,33,35] for examples). Developing a clear understanding of the computational issues involved in mechanism design should facilitate its use in multiagent system design. Therefore, in this paper, we consider how placing computational limitations on the agents and the mechanism affects classic results in the mechanism design literature. In particular, we investigate the effect of restricting the agents and the mechanism to polynomial time computation. (Section 3 provides details on exactly how this is done.) To focus our investigation, we consider a particular problem which we call *Multiagent MAXSAT* and restrict ourselves to complete information envi-

☆ See Ref. [26] for a preliminary version of this paper. Material from that paper is used with the permission of Kluwer Academic Publishers.

* Corresponding author.
E-mail address: oconnellt@acm.org (T.C. O'Connell).

[1] Supported in part by National Science Foundation Grant CCR-97-34936.

0167-9236/$ - see front matter © 2003 Elsevier B.V. All rights reserved.
doi:10.1016/j.dss.2003.10.007

ronments. In Multiagent MAXSAT, each agent's preferences over the set of possible outcomes can be described by a disjunction over negated and unnegated Boolean variables.

For example, consider a warehouse inhabited by several robots that have different and possibly conflicting goals. (This example is based on an example from Ref. [33].) Each robot is concerned only with satisfying its own goal and does not care whether any of the other robots satisfy their goals. Rather than spending time negotiating with one another when a conflict arises, the robots rely on an outside arbitrator to resolve the conflict quickly and equitably. The arbitrator is referred to as a *mechanism*. The mechanism's only goal is to have the outcome of its decisions satisfy some measure of social desirability called a *social choice rule*. If the mechanism is successful, it is said to *implement* the social choice rule. In this case, let the mechanism's goal be to satisfy as many of the robots as possible. In other words, from the point of view of the mechanism, any outcome that satisfies the maximum number of simultaneously satisfiable robots is a good outcome and any other outcome is a bad outcome. Suppose in this warehouse there are n blocks B_1 through B_n and one table. We can describe the state of the world using Boolean variables. Let x_i = true represent B_i being on the table and let x_i = false represent B_i being on the floor for $i = 1,\ldots,n$. If the robots' goals are restricted to those that can be represented by a disjunction over the Boolean variables and their negations, this is an instance of a multiagent MAXSAT problem.

The problem of assigning truth values to a set of variables so that the number of satisfied disjunctions is maximized is known to be a computationally difficult problem (see Ref. [6]). According to the widely held belief of computer scientists and logicians, namely that P≠ NP, it would be impossible for the mechanism to maximize the number of satisfied agents in every instance if the agents and the mechanism are limited to polynomial time computation. Therefore, any polynomial time mechanism must settle for outcomes that are approximately optimal. (Readers unfamiliar with the P vs. NP question should refer to the Appendix A for an explanation.)

The main results of this paper are as follows:

(1) The revelation principle states that, if there is a mechanism that implements a social choice rule, then there is a truthful revelation mechanism that implements the social choice rule, i.e., there is a mechanism that asks the agents to declare their preferences and for which truthful declaration is an equilibrium strategy (see Section 4). The revelation principle allows the discussion to be restricted to social choice rules that are implementable by truthful revelation mechanisms. We show that the revelation principle applies when the mechanism and the agents are restricted to polynomial time but *does not* apply when the mechanism is restricted and the agents are not. This implies that, in the latter situation, we cannot restrict our attention to truthfully implementable social choice rules.

(2) We provide a mechanism with a non-dictatorial outcome function that implements MAXSAT in dominant strategies. The mechanism runs in polynomial time but the agents require non-polynomial time to compute their dominant strategies. (Throughout this paper, we assume that P≠ NP.) This result is of interest because a classic theorem known as the Gibbard–Satterthwaite theorem states that in many situations, dominant strategy implementation with non-dictatorial outcome functions is impossible. Gibbard–Sattherwaite does not apply to Multiagent MAXSAT.

(3) We provide a mechanism such that all dominant strategy equilibrium outcomes satisfy at least half of the agents. In this case, the mechanism and the agents use only polynomial time.

(4) We provide a polynomial time mechanism that guarantees that each Nash equilibrium outcome satisfies at least half of the agents. This mechanism is in many ways superior to the mechanism we developed for dominant strategy implementation.

(5) We show that in the case of strong implementation in dominant strategy, Nash, undominated Nash or subgame perfect equilibrium, it is impossible to guarantee that the equilibrium outcomes will satisfy more than half of the maximum number of simultaneously satisfiable agents. In contrast, there are approximation algorithms for the non-multiagent version of MAXSAT that guarantee that 3/4 of the maximum number of simultaneously satisfiable agents will be satisfied (see Refs. [1,8,40]). This result suggests that we will be