Of Mechanism Design and Multiagent Planning[1]

Roman van der Krogt [a] Mathijs de Weerdt [b] Yingqian Zhang [b]

[a] *Cork Constraint Computation Centre, University College Cork, Ireland*
[b] *Delft University of Technology, Delft, The Netherlands*

Many interesting applications of AI planning feature an environment with multiple agents. Often these agents represent companies or other autonomous entities which may have (partially) conflicting preferences. Such self-interested agents may be tempted to lie about their costs or the actions they can do in order to obtain an outcome that is more rewarding for them. We therefore study the multiagent planning problem from a mechanism design perspective, showing how to incentivise agents to be truthful. Below we first introduce our model of multiagent planning problems for self-interested agents, then we analyse where known results in mechanism design fail to deal with multiagent planning, and we propose a solution to this problem.

Formally, a *multiagent planning problem* $\theta \in \Theta$ for a set of agents is a tuple $\theta = (\theta_1, \ldots, \theta_n)$ of private planning problems $\theta_i \in \Theta_i$ for these agents. Agent i's planning problem θ_i consists of *(i)* a set of ground atomic formulae; *(ii)* a set of actions this agent may carry out; *(iii)* a cost function c_i that assigns a cost to each operator; *(iv)* that part of the (common) initial state the agent is aware of; *(v)* a set of goals G_i; and *(vi)* a reward function $r_i : G_i \to \mathbb{R}^+$, assigning a reward to each of the goals. The goals of different agents *can* be mutually exclusive. The solution to a multiagent planning problem is a plan π, which is a partially ordered sequence of actions. The space of all plans is denoted by Π. The *utility* of plan π is defined as: $U(\pi, \theta) = c(\pi, \theta) + r(\pi, \theta)$, where $c(\pi, \theta)$ is the cost of executing the plan and $r(\pi, \theta)$ denotes the *revenue* of π that is given by the reward functions for the goals that have been attained. An optimal planner returns the plan which has the highest utility.

Each agent i has preferences over the possible plans defined by its *valuation* $v_i(\pi, \theta) = c_i(\pi, \theta) + r_i(\pi, \theta)$. In this paper, we consider a mechanism design problem where the declaration of the type of all agents is the input, and a plan $\pi \in \Pi$ is the output of the mechanism. A mechanism using an optimal planning algorithm will choose the best plan in Π, which maximises the *social welfare* $v(\pi, \theta)$ that is the total valuation of the agents. The social welfare can be maximised only if the agents report their types truthfully. In order to achieve this, *payments* are introduced to penalise some agents and possibly reimburse some others based on their contribution to the social welfare. With payments, the *utility* of the agent i on the outcome π is defined by: $u_i(\pi, \theta) = v_i(\pi, \theta) - p_i(\theta)$. This utility is what rational agents aim to maximise.

We consider a mechanism to be a tuple $(f, p_1, \ldots, p_n)$ where $f : \Theta_1 \times \cdots \times \Theta_n \to \Pi$ is a planning function, and $p_1, \ldots, p_n$ are payment functions which specify for each agent the mount it pays. The goal of mechanism design for MAP is thus to find a mechanism $(f, p_1, \ldots p_n)$ such that $f(\theta)$ returns the plan which maximises the social welfare. We say a mechanism is *truthful* iff no agent can achieve a higher utility by lying about its type.

(Deposit-)VCG Mechanisms for MAP

When agents declare their type, they can lie in three different ways: *(i)* about the *value* of a plan, i.e. the costs, the rewards, and the goals; *(ii) under-reporting* the available actions; and *(iii) over-reporting* non-existing actions or states. We investigate how to design the truthful mechanisms to prevent such lying types for MAP. So-called Vickrey-Clarke-Groves (VCG) mechanisms are very successful in satisfying this property [1].

Previous work has shown that every VCG mechanism is truthful [1]. Indeed, we show that *the VCG mechanism for MAP with an optimal algorithm prevents lying about values and under-reporting, or a combination of both.* Using an optimal planning algorithm, VCG mechanisms work that well, because *(i)* the agents' utility and thus their incentives are aligned with the social welfare, and moreover *(ii)* the goal of the

[1] The full paper appeared in the Proc. of the 18th European Conference on Artificial Intelligence (ECAI'08), pp. 423-427, IOS Press.

algorithm is also to maximise the social welfare. Surprisingly, for the third type of lying, i.e. over-reporting, an agent *can* gain from reporting *more* than it has at its disposal. *The VCG mechanism with an optimal algorithm for MAP cannot prevent over-reporting.* Intuitively, the reason for this is that the outcome of MAP is a global, distributed plan that only achieves its value upon *successful* execution. This gives the agents additional ways to cheat which aren't prevented by the VCG mechanism: their penalties are based on what they *promise* to do; not on what they *actually* achieve. An agent may for example be rewarded for actions that it claims it has and that help other agents to achieve their goals, but which it cannot actually execute. If those actions are included in the generated plan π, the utility of π is not representing the social welfare. So even an optimal planner cannot guarantee to output a "best" plan which maximises the social welfare.

In order to avoid over-reporting, we introduce the *deposit-VCG* mechanism: first, the mechanism asks the agents to declare their types θ_i, then it asks each agent to pay the amount $r(G)$ (the total award of the goals in G) as a deposit. The mechanism then finds a plan π using an optimal algorithm f, taking into account only the agents who paid the deposit. After each agent i pays p_i according to the VCG formula, the mechanism informs the agents of the plan π, and each agent i executes its part. If any local plan fails due to the agent i's declaration, agent i will not get its deposit back. All other agents are returned their deposits. Since the separate deposit stage does not enlarge the strategy space of the agents, it is straightforward to see that if the agents are truthful under the VCG mechanism, they will not be better off by lying under the deposit-VCG mechanism. Consequently, deposit-VCG is truthful with respect to lying about values and under-reporting. Moreover, it also prevents over-reporting. Thus, *the deposit-VCG mechanism with an optimal algorithm is truthful for MAP*.

(Deposit-)VCG-based Approximations for MAP

The (deposit-)VCG mechanism requires that f makes optimal decisions. Except for some specific domains, this is intractable, as planning in general is PSPACE-complete. Hence, it is desirable to develop a *truthful*, *polynomial-time* mechanism which can produce reasonable results. We will call a mechanism *deposit-VCG-based*, if f is a sub-optimal algorithm and $p(\cdot)$ is calculated according to the deposit-VCG mechanism. Unfortunately, deposit-VCG-based mechanisms are generally not truthful. The reason is that VCG payments align the agent's utility with the value of the system's solution. Therefore by lying, an agent may "help" a non-optimal mechanism to achieve a better solution, and thus make more profit for itself.

It has been shown in [2] that a mechanism is truthful if the algorithm f is *maximal in its range* (MIR). Informally speaking, a planning algorithm f is MIR if it optimises the social welfare by selecting the best plan out of an on *forehand determined* set of allowable plans. Obviously, optimal planning algorithms are MIR. In general, non-optimal planning algorithms are not. However, for a number of planning domains approximations are known that can be used to create MIR mechanisms.

In the full version of the paper, we give one such example in the Blocks World (BW) domain. Although *optimal* planning for BW is NP-hard, we propose an MIR algorithm f_{bw} based on the work of [3]. We show that: *(i)* if the set of goals does not contain any conflicts, then (deposit-)VCG-based mechanism using f_{bw} is truthful; *(ii)* if, however, the goals have conflicts, and the social welfare depends on which goals are satisfied, then by limiting the number of goals to be attained by K, we can impose a polynomial bound on the mechanism's time complexity. Thus, a truthful (deposit-)VCG-based mechanism using f_{bw} can be achieved. More generally, *given a polynomial-time algorithm* $f_d : \Theta \rightarrow \Pi$ *for a planning domain* d *that is MIR on problems without conflicting goals, and an upper-bound* K *on the number of goals that is considered, an algorithm* f_d^K *exists that is MIR and polynomial in the input size.*

For our future work, we are interested in studying how other (approximation) algorithms for planning can be used to construct efficient and truthful mechanisms, focusing especially on variants of existing distributed MAP algorithms.

References

[1] N. Nisan. Introduction to mechanism design (for computer scientists). In *Algorithmic Game Theory*, pages 209–242. Cambridge University Press, 2007.

[2] N. Nisan and A. Ronen. Computationally feasible VCG mechanisms. *Journal of AI Research*, 29:19–47, 2007.

[3] J. Slaney and S. Thiébaux. Blocks world revisited. *Artificial Intelligence*, 125(1-2):119–153, 2001.