Creating incentives to prevent intentional execution failures

Yingqian Zhang
Delft University of Technology
Delft, The Netherlands
Email: Yingqian.Zhang@tudelft.nl

Mathijs de Weerdt
Delft University of Technology
Delft, The Netherlands
Email: M.M.deWeerdt@tudelft.nl

***Abstract*—When information or control in a multiagent system is private to the agents, they may misreport this information or refuse to execute an agreed outcome, in order to change the resulting end state of such a system to their benefit. This may result in execution failures. When only information is private, mechanisms such as VCG use payments to create incentives for truthful behavior, and can then guarantee a non-negative utility for all agents. However, when control is also private, such existing mechanisms lose truthfulness and individual rationality: payments should depend on the actual outcome (not on the planned outcome) and some agents should be compensated. We give a more general version of the known negative result in the context of actions with dependencies, and we give a mechanism that can guarantee a nonnegative utility to the agents and is truthful in an ex-post Nash equilibrium.**

I. Introduction

A multiagent system often involves a set of self-interested agents, which may manipulate the system by mis-reporting their private information. Research into mechanism design is about creating incentives for such self-interested agents to report the correct information, i.e., to be truthful. Such a mechanism usually consists of *(i)* a social choice function, which selects a socially optimal outcome given the declaration of agents; and *(ii)* a set of payments, which decide for each agent how much it pays (or receives) to (or from) the mechanism. Traditionally, the agents start executing the outcome selected by the mechanism *after* they receive or pay the payments. In this way, however, in many settings, the outcome selected by a mechanism may fail to execute for two main reasons: (i) agents may fail to execute their actions due to external events, called *accidental failures*; or (ii) agents may have declared some capabilities while they are *not able to deliver on* these promises during execution, or they may simply *refuse to execute* (part of) their assigned actions. In this paper we study the latter type of *intentional (or deliberate) execution failures*.

Existing work has largely ignored execution failures, except for, for instance, a recent paper by Porter et al. [8]. The focus of their work is on accidental failures. They give a number of results on dealing with accidental failures for variants of the problem, such as single action settings, multiple actions, and actions with dependencies where an action can be attempted only if some other actions are successfully completed. They have shown that for this setting of dependencies among actions, no mechanism exists that is truthful (in dominant strategies), efficient, individually rational, and rational for the center (similar to being weakly budget balanced). They also explain why their mechanism in the case of dependencies does not work for intentional failures, but they have not found an alternative solution.

This exactly is the focus of the current paper: *creating incentives for agents to prevent deliberate execution failures when actions may be dependent upon each other*. It is important when designing a mechanism to prevent such intentional failures, especially when there are dependencies between the actions of different agents, such as in scheduling with precedences and multiagent planning [2]. In these settings, the deliberate failures of any agent may result in a failure of the plan (or schedule).

As shown in previous work [8], [9], standard mechanisms that do not take into account the actual execution of the computed outcome in situations with dependencies cannot prevent lying. A standard Vickrey-Clarke-Groves (VCG) mechanism [1], [3], [10] (one of the most widely used truthful mechanisms) works as follows: (i) The mechanism asks the agents to declare their types. (ii) The mechanism finds an optimal outcome, and computes for each agent its payment. (iii) The mechanism informs the agents of the outcome, and asks each agent to deliver (or receive) the payment. To see informally why the standard VCG fails, consider the following multiagent planning problem (MAP).

A multiagent planning problem is concerned with planning by and for a group of (self-interested) agents. Such a MAP contains a private, individual planning problem for each agent. A typical individual planning problem of agent i includes a set of operations (with some costs attached, and a pre- and post-condition) that i can perform, a set of goals (with reward values), and the current state of this agent. The solution of a MAP is a plan: a partially ordered sequence of actions that, when executed successfully, results in a set of achieved goals for some of the agents. The utility of a plan is defined as the difference between the total reward of the achieved goals and the total cost of the actions used. The mechanism design problem of MAP is, given all agents' declared private planning problems, to determine both the plan that has the highest utility, and the payments of all agents.

Example 1. *As a simple MAP example, let there be two*

agents. Agent 1 has a goal which is to complete task t_1. Completing t_1 requires actions a_1 and a_2 for which we also have a precedence relation $a_1 \prec a_2$ (i.e., a_1 has to be executed before a_2). Agent 1 itself is able to perform only action a_1, with cost $c_1(a_1) = 8$. The reward of achieving t_1 is 10. Agent 2 does not have any goal, but can execute action a_2 with cost $c_2(a_2) = 1$. The optimal plan ω for this planning problem is to execute a_1 and then a_2 such that the goal t_1 can be obtained. The utility of this plan is $10 - 8 - 1 = 1$.

Using the VCG payment [5], agent 2 receives from the mechanism the payment of 2. Therefore, agent 2's utility after execution of the plan is 1 (as its action a_2 costs 1); however, if agent 2 deliberately fails to execute action a_2, it will achieve a higher utility of 2. Agent 2 thus has an incentive not to execute this action.

A. Our contributions

The above example shows the problem of using standard mechanisms in the presence of intentional failures. In order to prevent intentional failures, the mechanism has to compute the payments based on the *actual outcome*, instead of *planned outcome*. This can be implemented by a two-stage mechanism [8] (see Section II).

However, even if we drop the conditions that the mechanism should be individually rational and rational for the center, *no mechanism exists that is truthful and efficient, if there exist any intentional failures*. So, intentional failure can be very harmful, as it destroys the truthfulness of the mechanisms. This strong negative result is presented in Section III.

We then present a mechanism that is *truthful in an ex-post Nash equilibrium* in Section IV. Such a mechanism prevents the agents from lying by ensuring that mis-reporting or deliberately failing to execute will not result in a higher utility. In addition, the mechanism is *strongly individually rational*, which means that a truthful agent will never end up with a negative utility when participating in the mechanism. Thus, our mechanism not only prevents mis-reporting and deliberate failures (in ex-post Nash), but also creates an incentive for agents to participate in the mechanism. In addition, we discuss under what conditions the design of an efficient and truthful mechanism (in dominant strategy) is possible.

II. Preliminaries

We now introduce some general concepts in mechanism design (notation loosely based on e.g. [5]), and a general direct mechanism in the context of execution failures, similar to [8].

Let the *type*, i.e., the private information, of each agent $i \in \{1, \ldots, n\}$ be denoted by θ_i. Let Θ_i be the allowable subset of types for agent i, and let $\Theta = \Theta_1 \times \cdots \times \Theta_n$. A *type profile* θ is a vector $(\theta_1, \ldots, \theta_n) \in \Theta$ associating each agent with a type. We use $\theta_{-i} = (\theta_1, \ldots, \theta_{i-1}, \theta_{i+1}, \ldots, \theta_n)$ to denote the type profile without the type of agent i. Furthermore, we assume an agent i's intention of executing its action is also part of this private information.

In this paper, like many others, we focus on so-called *direct-revelation mechanisms* where we expect the agents to reveal all private information immediately to the mechanism. Such a direct-revelation setting is especially interesting because all negative results immediately transfer to any other setting (by the revelation principle). Positive results indicate that there may be similar mechanisms in other, in-direct, mechanisms. Since execution intentions are part of an agent's type, in a direct-revelation mechanism, this is also part of an agent's declaration. For example, in Example 1, the declaration of agent 2 is $\hat{\theta}_2$: a_2 with cost 1, where agent 2 lies about its intention to execute action a_2. The true type of agent 2 is $\theta_2 = \{\emptyset\}$. An agent's private information thus consists of two parts, namely, the *intention to execute* (i.e., a_2) and the associated *value* (i.e., 1).

Given a type profile of the revealed types, a direct-revelation mechanism selects an *outcome* from the set of possible outcomes (denoted by Ω), and possibly a payment for each agent. Formally, a direct-revelation mechanism is defined by a function $g : \Theta \to \Omega$, which outputs an outcome given types Θ, and for each agent i, a payment function $p_i : \Omega \times \Theta \to \mathbb{R}$.

In case a selected outcome (or *planned outcome*) is not reached, because one or more agents fail to execute the tasks assigned to them, the payments should be dependent on the *actual outcome* (or *executed outcome*) instead [8], or at least on the declared valuations after execution [4]. The actual outcome is defined by a function $e : \Omega \times \Theta \to \Omega$ that given the *true* types of all agents and an outcome selected by a mechanism, returns the outcome that is really achieved. Since whether or not an agent will fail during execution is also private information to this agent, if no agent is lying, then obviously for any outcome ω selected by a mechanism, it should hold that $e(\omega, \theta) = \omega$.

In general, such a direct two-stage mechanism $(g, p_1, \ldots, p_n)$ works as follows [4].

Definition 1. *In a* two-stage mechanism

1) *The mechanism asks the agents to declare their types $\theta \in \Theta$, finds an outcome using function g, and informs each agent of its part of execution in the outcome $g(\theta)$.*
2) *The agents start to execute their assigned tasks (if they intend to), and after the outcome is executed, each agent i pays according to the executed outcome $e(g(\theta), \theta)$, i.e., agent i pays: $p_i(e(g(\theta), \theta), \theta)$.*

The preferences of an agent i with type θ_i are given by a utility function relying mainly on a valuation function $v_i(\omega, \theta)$, which assigns a real value to an outcome $\omega \in \Omega$. In this paper, we thus use $v_i(e(\omega, \theta), \theta)$ to denote the actual valuation of agent i on a planned outcome ω. Each (rational) agent tries to maximize its *utility*, which is defined as: $u_i(\omega, \theta) = v_i(e(\omega, \theta), \theta) - p_i(e(\omega, \theta), \theta)$.

The main aim of mechanism design is to construct mechanisms that select outcomes for which the sum of the valuations of all agents is as high as possible. These mechanisms are called *efficient*. A major step in being able to obtain such efficient outcomes is to give incentives to agents to reveal their true type. We say a mechanism is *truthful in dominant strategies*, if no agent can be worse off by revealing its true type (i.e., truth-telling), no matter what the other agents do. A

slightly weaker property is that of *truthfulness in an ex-post Nash equilibrium* which says that no agent can be worse off by revealing its true type if all the other agents also reveal their true type. A well-known result in this respect is that the VCG-mechanism is both efficient and truthful in dominant strategies, as well as individually rational [5].

Individual rationality is another property we would like a mechanism to have. A mechanism is usually called *individually rational* (or IR) if agents never receive negative utility in (the dominant strategy or Nash) equilibrium. In this paper, we would like agents to also have a nonnegative utility guaranteed in case other agents are not truthful, specifically since part of the execution may then fail. We call mechanisms for which this holds *strongly individually rational*. This is an important property since it provides an incentive for truthful agents to participate in a multiagent application. In earlier work the same concept has been referred to as a *participation constraint* [6].

Definition 2. *A mechanism is* strongly individually rational *(sIR) if for every agent i, for every type profile $\hat{\theta} = (\theta_i, \hat{\theta}_{-i})$ where agent i is following the equilibrium strategy (i.e., i is truthful), $\hat{\theta}_{-i}$ denotes all other agents' types except i, the utility of agent i is non-negative, i.e., $u_i(e(g(\hat{\theta}), \theta), \theta) \geq 0$.*

In this paper we focus on a setting where agents have rewards or costs for certain tasks or actions, and these actions may depend upon the execution of actions by other agents. Such dependencies are represented by the operator $\prec$, e.g., $a_1 \prec a_2$ means that action a_2 is dependent upon the successful execution of action a_1. The valuation of each agent is then defined by the sum of the costs (or rewards) of all actions (or tasks) that the agent executes. Such a setting is common in multiagent applications, such as scheduling, task allocation and multiagent planning. We show that in this setting, there is no efficient two-stage mechanism that is truthful in dominant strategies. Then we give a mechanism that is truthful, efficient, and individually rational in an ex-post Nash equilibrium.

We omit all formal proofs in this paper due to the page limit.

III. The harm of intentional failures

In this section, we show that intentional failures can be very harmful: even if we are able to catch the agent that caused an execution failure by monitoring its execution outcome, there exists no efficient mechanism that is truthful in dominant strategies. To arrive at this result, we use an idea that can be attributed to the generalized Vickrey auction [5]. We show that a payment can depend on the outcome or on the types of other agents, but not its own declared type in order to create the right incentives.

Proposition 1. *Given a two-stage mechanism, a type profile of all other agents except i, and two possible declarations of agent i resulting in the same outcome, the payments of agent i must be the same for the mechanism to be truthful.*

Intuitively, when the other agents' declarations are the same and the actual outcome is the same, agent i should receive the same payment from the mechanism, regardless of its declaration. Otherwise, i can increase its utility by misreporting its type.

Given this property of the payment function, we arrive at the main, negative result of this paper.

Theorem 1. *If intentional execution failures are possible, no efficient mechanism is truthful in dominant strategies, even if a two-stage mechanism is used.*

The proof of this result uses a MAP domain with precedence relations among the actions agents need to execute. For this domain, we show that even if every intentional execution failure of agents can be detected by monitoring, no mechanism can be both efficient and truthful since there exist no payments that always guarantee the agents are not better off by lying.

IV. Dealing with intentional failures

We have seen that the existence of intentional failures destroys truthfulness in dominant strategies. We therefore aim at a solution that is truthful in an ex-post Nash equilibrium. Also we want to guarantee for each agent a non-negative utility (i.e., strong individual rationality). Before doing so, we first show why VCG payments fail to make the mechanism strongly individually rational. Then we present our main positive result: a payment that gives us a strongly individually rational mechanism that is truthful in an ex-post Nash equilibrium. After that, in Section IV-C, we discuss under which condition this mechanism is also truthful in dominant strategies.

A. VCG payments are not sIR

We now define the VCG payment [5].

Definition 3. *(VCG payment) Given the executed outcome $\omega' = e(\omega, \theta)$ (which may be different from the planned outcome $\omega = g(\theta_i, \theta_{-i})$), the* VCG payment *of each agent i is defined by: $p_i(\omega', \theta) = h_i(\theta_{-i}) - \sum_{j \neq i} v_j(\omega', \theta_j)$, where $h_i(\theta_{-i})$ can be any non-negative function that is not dependent on agent i's declaration θ_i.*

It is well known that mechanisms with such a VCG payment are individually rational [7]. However, we now show that when agents can refuse to execute part of their assigned tasks, the mechanism is no longer strongly individually rational, because other agents may end up with a negative utility.

Proposition 2. *A two-stage mechanism with the VCG payment is not strongly individually rational in the presence of intentional failures.*

The proof of Proposition 2 uses a VCG payment that is not strongly individually rational when there is an execution failure. The reason is that the utility of the (truthful) agents decreased due to this failure. We propose in the following section a mechanism that punishes the agents who caused failures and compensates those who were disadvantaged.

B. A mechanism that is sIR and truthful in ex-post Nash

We define an alternative payment when intentional failures occur. We keep the VCG payment in the extended payment structure in order to deal with situations where there are no intentional failures.

Definition 4. *(Mechanism with compensation)*
Given the executed outcome $\omega' = e(\omega, \hat{\theta})$, a mechanism with compensation *works as a two-stage mechanism (Definition 1) and has the following payment functions.*

- *If agent i's failure is detected, its payment is at least $\max_{\theta_i^* \in \Theta_i} \{v_i(\omega', \theta_i^*)\}$, and the other agents' $j \neq i$ payments are given by $v_j(\omega', \hat{\theta}_j) - h'_j(\hat{\theta}_{-j})$, where $\hat{\theta}_j$ is the declaration of j, and $h'_j(\hat{\theta}_{-j})$ is any non-negative function not dependent on $\hat{\theta}_i$.*
- *If no execution failure is detected, we use the VCG payment of Definition 3.*

Lemma 1. *The mechanism with compensation is truthful in an ex-post Nash equilibrium.*

Informally, when all other agents are reporting their true types but agent i decides to fail its execution, this agent will be detected by execution monitoring, and therefore its payment will be greater than its maximal possible gain, and then its utility will be negative. If agent i diverts from truthful behavior in any other way, there is no execution failure. Consequently, the standard VCG payment applies, so the utility of agent i will not be higher than when it reports its true type [5].

We can now also show that this mechanism with compensation is strongly individually rational, i.e., a truthful agent will not receive negative utility, due to the payment scheme defined in Definition 4.

Lemma 2. *Given the declaration of the agents $\hat{\theta}$ and the planned outcome $\omega = g(\hat{\theta})$, the mechanism with compensation is strongly individually rational.*

These two lemmas immediately imply the following theorem.

Theorem 2. *The mechanism with compensation defined in Definition 4 is truthful and strongly individually rational in an ex-post Nash equilibrium.*

We now discuss when it is possible to develop a truthful mechanism in a dominant strategy.

C. Full verification

We have shown a strong impossibility result in Theorem 1 that even if all agents with an intention to fail their executions can be detected, no efficient mechanism is truthful in dominant strategies. However, it is possible to obtain a truthful mechanism in a dominant strategy equilibrium. But in order to achieve this, it is not sufficient to only monitor the execution outcomes, but *all* agents have to be verified during the execution stage. Here, a full verification means that the mechanism is capable of knowing the true type of every agent.

In some settings, such verification is possible. For instance, in the task scheduling domain considered by Nisan and Ronen [6], agents' declarations are the execution time for certain jobs. Assuming that agents will not delay such executions *on purpose*, we can verify the types of all agents regarding the scheduled jobs during the execution of the schedule. Thus, we know the true types of all agents. As a result, any lying agent will be detected, and then be punished by requiring a payment which is greater than its maximal possible gain (Definition 4). Hence, it is in every agent's interest to report its private information truthfully, no matter whether other agents lie or not. In this case, the compensation mechanism can be seen as a generalized version of the result on task scheduling with verification [6]. More specifically, if all planned actions can be verified, the mechanism with compensation (Definition 4) is truthful and strongly individually rational in a dominant strategy equilibrium.

V. Conclusion and Discussion

When agents are autonomous and self-interested, but still depend upon each other, agents can profit from intentionally failing to execute their part of the agreement. This mechanism design problem with intentional execution failures is the topic of this paper. We showed that there is no efficient mechanism for this problem that is truthful in dominant strategies. However, we were able to come up with an efficient two-stage mechanism that is not only truthful in an ex-post Nash equilibrium, but also strongly individually rational. Our mechanism is slightly more general than the standard VCG mechanism.

These two results are completely new, focusing on intentional failures, where earlier work by Porter et al. [8] deals with accidental failures in this setting. A natural question to ask is whether mechanisms exist that can deal with both accidental and intentional failures. Since it is impossible to distinguish an intentional failure from an accidental failure, there is no direct combination of the two methods, leaving open a challenging area of future work.

References

[1] E. H. Clarke. Multipart pricing of public goods. *Public Choice*, 11(1), 1971.

[2] M. Ghallab, D. Nau, and P. Traverso. *Automated Planning, theory and practice*. Morgan Kaufmann Publishers, 2004.

[3] T. Groves. Incentives in teams. *Econometrica*, 41(4):617–31, 1973.

[4] C. Mezzetti. Mechanism design with interdependent valuations: Efficiency. *Econometrica*, 72(5):1617–1626, 09 2004.

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

[6] N. Nisan and A. Ronen. Algorithmic mechanism design. *Games and Economic Behavior*, 35(1–2):166–196, 2001.

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

[8] R. Porter, A. Ronen, Y. Shoham, and M. Tennenholtz. Fault tolerant mechanism design. *Artif. Intell.*, 172(15):1783–1799, 2008.

[9] R. van der Krogt, M. de Weerdt, and Y. Zhang. Of mechanism design and multiagent planning. In M. Ghallab, C. D. Spyropoulos, N. Fakotakis, and N. Avouris, editors, *Proceedings of the 18th European Conference on Artificial Intelligence (ECAI-08)*, pages 423–427. IOS Press, 2008.

[10] W. Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *The Journal of Finance*, 16(1):8–37, 1961.