Fault Tolerant Mechanism Design*

Ryan Porter† Amir Ronen‡ Yoav Shoham§

Moshe Tennenholtz¶

June 10, 2008

Abstract

We introduce the notion of *fault tolerant mechanism design*, which extends the standard game theoretic framework of mechanism design to allow for uncertainty about execution. Specifically, we define the problem of task allocation in which the private information of the agents is not only their costs of attempting the tasks but also their probabilities of failure. For several different instances of this setting we present both, positive results in the form of mechanisms that are incentive compatible, individually rational, and efficient, and negative results in the form of impossibility theorems.

Keywords Mechanism design, decentralized task allocation, game theory, uncertainty.

1 Introduction

In recent years, the point of interface between computer science and mechanism design, or MD for short, has been a site of great activity (e.g. [22, 16, 4, 19, 15, 2, 1, 8]). MD, a sub-area of game theory, is the science of crafting protocols for self-interested agents, and as such is a natural fodder for computer science in general and AI in particular. The uniqueness of the MD perspective is that it concentrates on protocols for non-cooperative agents. Indeed, traditional game theoretic work on MD focuses solely on the incentive aspects of the protocols.

*A preliminary version of this paper appeared in UAI 2002 [17]. This version contains proofs that were omitted from the conference version and several new theorems. In addition, the presentation has been considerably amended.

†Amazon.com

‡Faculty of Industrial Engineering & Management, Technion, Israel. The author was supported in part by grant number 53/03 − 10.5 from the Israel Science Foundation. Email: amirr@ie.technion.ac.il

§Department of Computer Science, Stanford University. The author was supported in part by NSF-ITR grant number: IIS-0205633-003. Email: shoham@stanford.edu

¶Faculty of Industrial Engineering & Management, Technion, Israel. The author was supported in part by grant number 53/03 − 10.5 from the Israel Science Foundation. Email: moshet@ie.technion.ac.il

A promising application of MD to AI is the problem of task allocation among self-interested agents (see, e.g., [18]). When only the execution costs are taken into account, the task allocation problem allows standard mechanism design solutions. However, this setting does not take into consideration the possibility that agents might fail to complete their assigned tasks. When this possibility is added to the framework, existing results cease to apply. The goal of this paper is to investigate robustness to failures in the game theoretic framework in which each agent is rational and self-motivated. Specifically, we consider the design of protocols for agents that not only have private cost functions, but also have privately-known probabilities of failure.

What criteria should such protocols meet? Traditional MD has a standard set of criteria for successful outcomes, namely social efficiency (maximizing the sum of the agents' utilities), individual rationality (positive utility for all participants), and incentive compatibility (incentives for agents to reveal their private information). Fault Tolerant Mechanism Design (FTMD) strives to satisfy these same goals; the key difference is that the agents have richer private information (namely probability of failure in addition to cost). As we will see, this extension presents novel challenges.

To demonstrate the difficulties encountered when facing such problems, consider even the simple case of a single task for which each agent has zero cost and a privately-known probability of success. A straw-man protocol is to ask each agent for its probability, choose the most reliable agent (according to the declarations) and pay it a fixed, positive amount if it succeeds, and zero otherwise. Of course, under this protocol, each agent has an incentive to declare a probability of success of one, in order to maximize its chances of receiving a payment, at no cost to itself.

Before moving to a formal definition of the problem, it is important to distinguish between different possible failure types. The focus of this work is on failures that occur when agents make a full effort to complete their assigned tasks, but may fail. A more nefarious situation would be one in which agents may also fail deliberately when it is rational to do so. While we do not formally consider this possibility, we will revisit it at the end of the paper to explain why many of our results hold in this case as well. Finally, one can consider the possibility of irrational agents whose actions are counter to their best interests. This is the most difficult type of failure to handle, because the presence of such agents can affect the strategy of rational agents, in addition to directly affecting the outcome. We leave this case to future work.

1.1 Our Contribution

In this paper we study progressively more complex task allocation problems. We start with the case of a single task. Even in this simple setup it is not possible to apply the standard solutions of mechanism design theory (Generalized Vickrey Auction (GVA)). Informally, the main reason for this is that the value of the center depends on the *actual* types of the agents and not just on the chosen allocation. We define a mechanism SingleTask with properties similar to those of GVA (or more precisely, to a slightly more general version of GVA that takes the center's value in account). The mechanism offers a contract to each agent in which the payment is contingent on whether the task is completed. A mechanism is called incentive compatible in dominant strategies (DSIC) if the agent always gets a contract that maximizes its expected utility when declaring its actual type. Similarly, a mechanism is called individually rational (IR) if a truthful agent always gets a contract that guarantees it a non-negative expected utility. (We stress that agents may end up losing due to their own failures.) A DSIC mechanism is called ex-post economically efficient (EE) if when the agents are truthful, the mechanism's allocation maximizes the expected total welfare, which equals the expected center's value minus the agents' costs. Finally, a DSIC mechanism is ex-post individually rational for the center (CR) if when the agents are truthful, the center's expected utility is non-negative, no matter what the actual vector of types is. We define a mechanism called MultipleTask that generalizes the previous mechanism to the case of multiple tasks and additive values for the center and show that it maintains all the above properties.

Theorem The MultipleTask mechanism satisfies DSIC, IR, CR, EE.

We then study more complicated settings in which it is impossible to satisfy all the above properties simultaneously. In most task allocation situations the value of the center is likely to be combinatorial and not additive (see example in Section 2.5). Let n denote the number of agents and t the number of tasks. We show the following impossibility theorem:

Theorem When V is combinatorial, no mechanism exists that satisfies DSIC, IR, CR, and EE for any $n \geq 2$ and $t \geq 2$.

Fortunately, when CR is relinquished, it is possible to satisfy the other properties.

Theorem The MultipleTask mechanism satisfies DSIC, IR, EE, even when V is combinatorial.

Next we study situations in which there are dependencies between tasks. This complicates the setup further because now the cost of an agent can depend on the actual types of other agents. We show the following impossibility results:

Theorem When dependencies exist between tasks, even when the center's valuation is non combinatorial, no mechanism exists that satisfies DSIC, IR, CR, and EE for any $n \geq 2$ and $t \geq 2$.

Theorem When dependencies exist between tasks and the center's valuation is combinatorial, no mechanism exists that satisfies DSIC, IR, and EE for any $n \geq 2$ and $t \geq 2$.

In light of the above theorems we relax our properties to hold only in an ex-post equilibrium. We then present a modification of our mechanism called Ex-Post-MultipleTask and show that it satisfies the equilibrium version of our properties.

Theorem Mechanism Ex-Post-MultipleTask satisfies ex-post IC, IR, and EE, even when dependencies exist between the tasks and the center's valuation is combinatorial.

The above mechanisms suffer from two major deficiencies, which seem unavoidable in our setup. First, the agents may end up with large losses. Second, even the expected center's value may be negative. Our final result shows that these drawbacks can be overcome when it is possible to verify the cost of the agents after the tasks are performed. Given any n positive constants $(\chi_1, \ldots, \chi_n)$, we define a mechanism called Ex-Post-CompensationAndBonus. This mechanism is a modification of the compensation and bonus mechanism introduced in [15], adjusted to handle the possibility of failures. Let u^* denote the optimal expected welfare. Then:

Theorem Under the verification assumption, Mechanism EX-POST-COMPENSATIONANDBONUS satisfies ex-post IC, IR, EE, and CR, even when dependencies between the tasks exist and the center's valuation is combinatorial. Moreover, for every $\epsilon > 0$ and a type vector θ, when the constants χ_i are small enough, the expected center's utility is at least $u^* \cdot (1 - \epsilon)$.

1.2 Related Work

The work presented in this paper integrates techniques of economic mechanism design (an introduction to MD can be found in [12, chapter 23]) with studies of fault tolerant problem solving in computer science and AI.

In particular, the technique used in our mechanism is similar to that of the Generalized Vickrey Auction (GVA) [21, 5, 10] in that it aligns the utility of the agents with the overall welfare. (More precisely, our mechanism resembles a generalized version of GVA that also takes the center's value into account.) This similarity is almost unavoidable, as this alignment is perhaps the only known general principle for solving mechanism design problems. However, because we allow for the possibility of failures, we will need to change the GVA in a significant way in order for our mechanisms to achieve this alignment.

Because we have added probabilities to our setting, our mechanisms may appear to be related to the Expected Externality Mechanism (or d'AGVA) [6], but there are key differences. In the setting of d'AGVA, the probabilistic component is the distribution from which the types of the agents are drawn, and this distribution is assumed to be common knowledge among the participants. The two key differences in our setting are that no such common knowledge assumption is made and that d'AGVA uses the Bayesian-Nash equilibrium as its solution concept.

A specific problem of task allocation with failures in the context of networking is studied in [9]. The model and the questions addressed in this work are very different from ours.

The design of protocols that are robust to failures has a long tradition in computer science (for a survey, see e.g. [11]). Work in this area, however, almost always assumes a set of agents that are by and large cooperative and adhere to a central protocol, except for some subset of malicious agents who may do anything to disrupt the protocol. In mechanism design settings, the participants fit neither of these classes, and all are instead modelled as being self-interested. A paper in the spirit of computer science that considers failures in mechanism design is [7]. This work assumes that agents know the types of all other rational agents and limits the failures that can occur by bounding the number of irrational agents. Under these assumptions, the paper characterizes the set of full Nash implementations.

The problem of procuring a path in a graph in which the edges are owned by self interested parties has received a lot of attention in recent years (e.g. [1, 8, 13, 15]). It is a private case of the task allocation problem we are studying. The works mentioned above did not discuss the possibility of failures.

In principle agent problems the agents are required to exert costly efforts in order to perform some joint action. There is some technical similarities between such problems and ours as, typically, the effort level of each agent affects the overall success probability of the joint action. Recently, principle agent problems that incorporate combinatorial aspects were studied by several researchers (e.g. [20, 3, 14]. The setup and focus of these papers are essentially different from ours. Our setting emphasizes the elicitation of private information with regard to the probability of success of task execution, a topic which to the best of our knowledge has not been treated in the principal-agent and mechanism design literature.

2 The Basic Model

In this section we describe our basic model. It will be modified later for more complicated settings. Sections 2.1 – 2.4 introduce our basic setup, the class of mechanisms we consider and the notations that are related to them, describe the utilities of the participants and the goals that a mechanism must satisfy. Section 2.5 provides two examples of task allocation problems.

2.1 Participants

An FTMD problem consists of a set of tasks $\tau = \{1, \ldots, t\}$, a set $N = \{1, \ldots, n\}$ of self-interested agents to which the tasks can be assigned, and a center M who assigns tasks to agents and pays them for their work. The center and the agents will collectively be called the *participants*.

Prior to acting within the mechanism, each agent i privately observes its *type* $\theta_i \in \Theta_i$. A type of agent i contains, for each task j, the probability $p_{ij} \in [0, 1]$ of successfully completing task j, and the nonnegative cost $c_{ij} \in \Re^+$ of attempting the task. We will represent a type as $\theta_i = (p_i, c_i)$, where $p_i = (p_{i1}, \ldots, p_{it})$ and $c_i = (c_{i1}, \ldots, c_{it})$. Throughout the paper we assume the cost of attempting a task is independent of its success probability, that the total agent cost is the sum of the costs of its attempted tasks, and that all the success probabilities are independent.

Let $\theta = (\theta_1, \ldots, \theta_n)$ denote a profile of types, consisting of one for each agent. We will use $\theta_{-i} = (\theta_1, \ldots, \theta_{i-1}, \theta_{i+1}, \ldots, \theta_n)$ to denote the same profile without the type of agent i, and $\theta = (\theta_i, \theta_{-i})$ as an alternative way of writing the full profile.

For simplicity we assume that each task can be assigned only once. The center does not have to allocate all the tasks. For notational convenience we assume that all the non-allocated tasks are assigned to a dummy agent 0, which has zero costs and success probabilities for all tasks. The payment to this dummy agent is always zero.

2.2 Mechanisms

In general, the protocol for the interaction between the agents and the center could be arbitrarily complicated, to consist of conditional plans of action in a multi-round interaction. A much simpler class of mechanisms to consider is that of *direct mechanisms*. A direct mechanism $\Gamma = (\Theta_1, \ldots, \Theta_n, g(\cdot))$ is a mechanism in which each agent i, after observing its type, declares a type $\hat{\theta}$ to the center (we will later justify this restriction). The function $g : \Theta_1 \times \ldots \times \Theta_n \to O$ maps the declared types of the agents to an output $o \in O$, where an output $o = (f_1, \ldots, f_n, \hat{r}_1(\cdot), \ldots, \hat{r}_n(\cdot))$ specifies both the allocation and the payment function (contract) given to each agent. We now elaborate on our notation regarding allocation, task completion, and payment.

Each $f_i(\hat{\theta})$ records the set of tasks assigned to agent i when $\hat{\theta}$ is the vector of declared types. An agent i then incurs a cost $c_i(f_i(\hat{\theta})) = \sum_{j \in f_i(\hat{\theta})} c_{ij}$ to attempt this set. The whole allocation is denoted by f.

We let $\mu_i = (\mu_{i1}, \ldots, \mu_{it})$ denote the *actual* completion vector for agent i (i.e., $\mu_{ij} = 1$ if agent i completed task j, and $\mu_{ij} = 0$ if agent i either failed to complete or was not assigned task j). To aggregate the completion vectors across all agents, we will use μ for a vector of t terms. Each coordinate μ_j specifies whether task j was completed.

An allocation f and a vector of success probabilities p for the agents define a probability distribution over the possible completion vectors. We denote this probability by $\mu_f(p)$. Note that this distribution depends on the actual types of the agents but not on their costs.

Given the vector of agent declarations $\hat{\theta}$, the mechanism gives a contract $\hat{r}_i(.)$ to each agent i. The actual payment $\hat{r}_i(\mu)$ given to the agent is a function of the completed tasks by *all* agents. (Sometimes, it is possible to condition this payment on μ_i only.) We let $r_i(\hat{\theta}, \mu) =_{df} \hat{r}_i(\mu)$ denote a function that maps both, the vector of declarations and the actual completion vector, to the agent's payment.

Figure 1 summarizes the setting from the perspective of an individual agent i.

Notation. Let $\mathcal{D}$ be a distribution and X a random variable over $\mathcal{D}$. We let $\mathbf{E}_{\mathcal{D}}[X]$ denote the expected value of X taken over $\mathcal{D}$. In particular, given an allocation f and a vector of agents' completion probabilities p, we let $\mathbf{E}_{\mu_f(p)}[F(\mu)]$ denote the expected value of $F(\mu)$ where μ is distributed according to $\mu_f(p)$. We note that p is not necessarily the true completion vector.

2.3 Participant Utilities

Agent i's utility function, $u_i(g(\hat{\theta}), \mu, \theta_i) = \hat{r}_i(\mu) - c_i(f_i(\hat{\theta}))$, is the difference between its payment and the actual cost of attempting its assigned tasks. Such a utility function is called quasi-linear.

Sequence of Events for Each Agent

1. Privately observes its type θ_i
2. Declares a type $\hat{\theta}_i$ to the mechanism
3. Is allocated a set $f_i(\hat{\theta})$ of tasks to attempt and is given a contract $\hat{r}_i(.)$.
4. Attempts the tasks in $f_i(\hat{\theta})$
5. Receives a payment of $\hat{r}_i(\mu)$ based on the actual completion vector of all tasks.

Figure 1: Overview of the setting.

Since our setting is stochastic by nature, the definitions need to take into account the agent's attitude towards lotteries. We adopt the common assumption that the participants are risk neutral. We leave the relaxation of this assumption to future research. A profile of allocations $f = (f_1, \ldots, f_n)$ and a profile of true probabilities p together induce a probability distribution $\mu_f(p)$ over completion vectors. Hence, an agent's expected utility, in equilibrium and before any job is attempted, equals $\overline{u}_i = \overline{u}_i(\hat{\theta}) = \mathbf{E}_{\mu_f(p)}[u_i(g(\hat{\theta}), \mu, \theta_i)]$. Each agent i thus tries to maximize its expected utility $\overline{u}_i$. Note that this definition is only with regard to the interim outcome of the mechanism (the contracts) as the agent cannot even know whether its own attempts will succeed.

The function $V(\mu)$ defines the center's nonnegative valuation for each possible completion vector. We normalize the function so that $V(0, \ldots, 0) = 0$. For now, we assume that the center has a non-combinatorial valuation for a set of tasks. That is, for all $T \subseteq \tau$, $V(T) = \sum_{j \in T} V(\{j\})$.

The center's utility function is the difference between its value of the completed tasks and the sum of the payments it makes to the agents: $u_M(g(\hat{\theta}), \mu) = V(\mu) - \sum_i \hat{r}_i(\mu)$.

The total welfare W of the participants is equal to the value to the center of the completed tasks, minus the costs incurred by the agents. Given an allocation f and a vector of true types (p, c), the expected total welfare $\mathbf{E}_{\mu_{f(\theta)}(p)}[W(f(\theta), \mu)]$ thus equals $\mathbf{E}_{\mu_f(p)}[V(\mu) - \sum_i c_i(f_i)]$.

2.4 Mechanism Goals

Our aim in each setting is to construct mechanisms that satisfy the following four goals: incentive compatibility, individual rationality (for the agents), individual rationality for the center, and economic efficiency. We stress that since the agents only have probabilistic information about their success, our definitions are ex-ante, meaning that the agents get the best contracts when they declare their actual types and the mechanism chooses an allocation that maximizes the expected total welfare.

Definition 1 (Incentive Compatibility in Dominant Strategies) *A direct mechanism satisfies* dominant strategy incentive compatibility *(DSIC) if for every agent i, type θ_i, possible declaration θ'_i, and vector of declarations for the other agents $\hat{\theta}_{-i}$, it holds that:*

$$\mathbf{E}_{\mu_{f((\theta_i,\hat{\theta}_{-i}))}(p)}[u_i(g((\theta_i,\hat{\theta}_{-i})),\mu,\theta_i)] \geq \mathbf{E}_{\mu_{f((\theta'_i,\hat{\theta}_{-i}))}(p)}[u_i(g((\theta'_i,\hat{\theta}_{-i})),\mu,\theta_i)].$$

In other words, for every agent i, no matter what the types and declarations of the other agents are, the agent gets the best *contract* when it is truthful (i.e. a contract that maximizes the agent's expected utility over its own failure probabilities). Since, an agent's utility depends only on the declarations of the other agents but not on their actual types, these types are omitted from the definition above.

Remarks While restricting ourselves to direct mechanisms may seem limiting at first, the *Revelation Principle for Dominant Strategies* (see, e.g., [12]) tells us that we can make this restriction without loss of generality. Note that the optimality of being truthful does not rely on any belief that the agent may have about the types or the declarations of the other agents (as opposed to Bayesian-Nash mechanisms).

The next property guarantees that the expected utility of a truthful agent is always non-negative. This means that it is beneficial for the agents to participate in the mechanism.

Definition 2 (Individual Rationality) *A direct mechanism satisfies* individual rationality *(IR) if:*

$$\forall i,\theta,\hat{\theta}_{-i} : \mathbf{E}_{\mu_{f((\theta_i,\hat{\theta}_{-i}))}(p)}[u_i(g((\theta_i,\hat{\theta}_{-i})),\mu,\theta_i)] \geq 0.$$

In other words, when the agent is truthful, its *contract* guarantees it a non-negative expected utility. We stress that agents may end up with a negative utility and even with a negative payment. It is possible to fix this by adding to each agent an amount of money equaling the supremum of possible costs that the agent may have given the reports of the other agents, i.e. $\sup_{\theta_i} c_i(f_i((\theta_i,\hat{\theta}_{-i})))$. However, the addition of such a constant will imply unacceptably high payments that will destroy other properties of the mechanism.

Definition 3 (Center's Rationality) *A direct mechanism satisfies* ex-post individual rationality for the center *(CR) if it satisfies DSIC and if:*

$$\forall\theta : \mathbf{E}_{\mu_{f(\theta)}(p)}[u_M(g(\theta),\mu)] \geq 0.$$

In other words, when the agents are truthful, the allocation guarantees the center a non negative expected utility.

Our final goal is to maximize the expected welfare of the participants.

Task	Agent	Center's value	c_i	p_i
S_A	A_1	300	100	0.8
S_A	A_2	300	80	0.7
S_B	A_1	200	100	0.8
S_B	A_2	200	90	0.9

Figure 2: Outsourcing example.

Definition 4 (Economic Efficiency) *A direct mechanism satisfies* ex-post economic efficiency *(EE) if it satisfies DSIC and if:*

$$\forall \theta, f' : \mathbf{E}_{\mu_{f(\theta)}(p)}[W(f(\theta), \mu)] \geq \mathbf{E}_{\mu_{f'}(p)}[W(f', \mu)].$$

We let $f^*(\theta)$ denote an allocation that maximizes the expected welfare of the participants.

2.5 Examples of Task Allocation Problems

We now provide two examples of task allocation problems. The first one falls into our basic framework. The second example demonstrates additional aspects of task allocation problems. These will be addressed in Section 4.

2.5.1 Outsourcing of Independent Projects

Consider a company that would like to outsource two large independent projects S_A and S_B. The company has business relationships with two potential contractors, A_1 and A_2, which are able to perform these projects. The values of the company (which is the center here), and the costs and success probabilities of the agents (potential contractors) are described in Figure 2.

Suppose that the center allocates S_A to A_1 and S_B to A_2. This allocation and the actual types of the agents define the following distribution on the completion vectors: $\mathbf{Pr}[(1,1)] = 0.72, \mathbf{Pr}[(1,0)] = 0.08, \mathbf{Pr}[(0,1)] = 0.18, \mathbf{Pr}[(0,0)] = 0.02$. Thus, the expected center's value is $\mathbf{E}[V(\mu)] = 0.72 \cdot 500 + 0.08 \cdot 300 + 0.18 \cdot 200 + 0.02 \cdot 0 = 420$. The total agent cost equals 190 and thus the expected welfare equals 230. Suppose that agent A_1 receives the following contract (depending on its completion vector μ_1 only): $\hat{r}_1(\{1\}) = 200$ and $\hat{r}_1(\{0\}) = 0$. The agent's cost is 100 and thus its expected utility is $\overline{u}_1 = 0.8 \cdot 200 - 100 = 60$.

Why focus on incentive compatible mechanisms? In a mechanism design setup, the behavior of the participants is determined by the selfish considerations of the agents and is not controlled by the mechanism. This behavior has a crucial effect on the outcomes of the mechanism. Consider, for instance, the following protocol for the outsourcing problem: The agents first declare their types; the mechanism then computes the allocation that maximizes the expected welfare according to $\hat{\theta}$ and pays 200 per each completed task. In such a case an agent with a low cost for a task will declare a high reliability even if its actual reliability is small. This can severely damage the center's value. In general, the behavior of the agents in arbitrary protocols is highly unpredictable, depending on their beliefs, risk attitudes, computational and cognitive ability, etc.. Thus, the standard approach in mechanism design, which we adopt, is to focus on mechanisms that admit specific solution concepts that make them more predictable (incentive compatible mechanisms).

Why can GVA mechanisms not be applied to our settings? Only a handful of generic methods for the construction of incentive compatible mechanisms are known to date. Perhaps the most important of these constructions is the GVA method [5, 10, 21]. In a nutshell, GVA can be applied to the following setup: let X denote the set of possible outputs of the mechanism; each participant has a valuation function $v_i : X \to R$ depending only on the mechanism's output, and the goal is to maximize the total welfare. GVA is a direct mechanism. It chooses an output $x \in X$ that maximizes the total welfare according to the declaration vector $\hat{v}$. The payment of each agent i is defined as $\sum_{j \neq i} \hat{v}_i(x) + h_i(\hat{v}_{-i})$ where $h_i(.)$ is any function independent of i's declaration. (In particular, $h_i(.)$ is often defined as the optimal welfare that can be obtained without agent i.) Roughly speaking, the utility of an agent in GVA mechanisms is identified with the welfare measured by the declarations of the other agents and its actual valuation. It is possible to generalize GVA by adding an artificial dummy agent whose valuation represents the center's preferences. On the surface it looks as if GVA can be applied to our setup. Yet, there is one crucial difference that precludes this: the value of the center is dependent on the *actual* failure probabilities of the agents and not only on the chosen allocation. Thus, if the computation of the payment is conducted according to the GVA formula, the agents will have a clear incentive to report success probabilities of 1 and artificially raise the center's value used to compute their payments. Nevertheless, we can apply the principles behind GVA to obtain task allocation mechanisms with incentive properties that resemble those of GVA. An application of GVA to our setup is demonstrated in Section 3.2.

2.5.2 Path Procurement with Failures

We now introduce a more complicated example that does not fall into our basic framework but into extensions of it that will be defined later. Consider the following path procurement problem: Given are a directed graph G with two distinguished nodes s and t. Each edge e in the graph G is owned by a self interested *agent* $o(e)$; an agent may own more than one edge. The actual cost c_e of routing an object along edge e is *privately* known to its owner $o(e)$. The center has a value V for procuring a path from s to t. The utility of each agent i is the difference between its payment r_i and its actual cost, i.e., $r_i - \sum_{e \in P : o(e)=i} c_e$, where P denotes the chosen s-t path.

In our framework each edge is a task. The center's valuation equals V if the completion vector contains a path, and zero otherwise. Note that this valuation is combinatorial. Various versions of the above problem were studied extensively in recent years (e.g. [1, 8, 15]).

Now consider the natural possibility that agents may fail to route the object so it may get lost. This possibility adds many complications to our basic setup. In particular, a lost object will not be routed further. Thus, the agents that need to carry the object further will not be able to attempt their tasks, and therefore, their *costs* will be reduced. Consider, for example, the instance in Figure 3. Suppose that each edge is owned by a single agent. The lowest path is composed of two edges. Suppose that the mechanism chooses this path. If the first agent along the path completes its task e_2, the second agent will attempt e_3 and bear the cost. On the other hand, if the first agent fails, the second edge will not be attempted and the cost of its owner will be zero. In other words, the cost of agents may be dependent on the *actual* types of others. As we shall see, this has implications for the properties of the mechanisms that can be obtained. Finally, we note that while without failures, computing the optimal path can be done in polynomial time, the computation becomes much more difficult when failures are introduced.

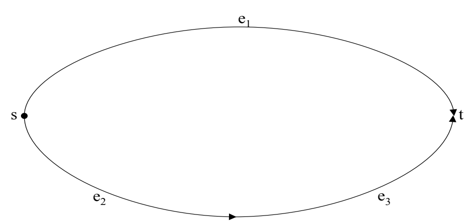

Figure 3: A path procurement instance

3 Single Task Setting

We will start with the special case of a single task, in order to show our basic technique for handling the possibility of failures. For expositional purposes, we will analyze two restricted settings (the first restricts the probabilities of success to one, and the second restricts the costs to zero), before presenting our mechanism for the full single-task setting.

Because there is only one task, we can simplify the notation. Let c_i and p_i denote c_{i1} and p_{i1}, respectively. Similarly, we let $V = V(\{1\})$ be the value that the center assigns to the completion of the task, and μ records the success or failure of the attempt to complete it.

3.1 Case 1: Only Costs

When we do not allow for failures (that is, $\forall i \; p_i = 1$), the goal of EE reduces to assigning the task to the lowest-cost agent. This simplified problem can be solved using the well-known second-price (or Vickrey) auction [21], with a reserve price of V. In this mechanism, the task is assigned to the agent with the lowest declared cost, and that agent is paid the second-lowest declared cost. If no agent's declared cost is below the reserve price of V, then the task is not allocated; and, if V lies between the lowest and second-lowest declared costs, then the agent is paid V.

3.2 Case 2: Only Failures

We now restrict the problem in a different way and assume that all the costs are zero ($\forall i\ c_i = 0$). In this case, the goal is to allocate the task to the most reliable agent.

Interestingly, we cannot use a straightforward application of the GVA for this case. Such a mechanism would ask each agent to declare its probability of success and then assign the task to the agent with the highest declared probability. It would set the payment function for all agents not assigned the task to 0, while the agent would be paid the amount by which its presence increases the (expected) welfare of the other agents and the center: $\hat{p}_{[1]}V - \hat{p}_{[2]}V$ (where $\hat{p}_{[1]}$ and $\hat{p}_{[2]}$ are the highest and second highest declared probabilities, respectively). To see that this mechanism uses the GVA payment formula, note that for all the agents except the one with the highest probability $\hat{p}_{[1]}$, the optimal welfare with and without the agent cancel each other out. For the winning agent, the welfare of the other agents equals the center's expected value $\hat{p}_{[1]}V$. Without the agent, the best expected welfare equals $\hat{p}_{[2]}V$, as all costs are 0. Thus, the agent's payment follows.

Clearly, such a mechanism is not incentive compatible, because the payment to the agent depends on its own declared type. Since there are no costs, it would in fact be a dominant strategy for each agent to declare its probability of success to be one.

To address this problem, we alter our payment rule so that it also depends on the outcome of the attempt, and not solely on the declared types, as it does in GVA. The key difference in our setting that forces this change is the fact that the true type of an agent now directly affects the outcome, whereas in a standard mechanism design setting the type of an agent only affects its preferences over outputs.

We accomplish our goals by replacing $\hat{p}_{[1]}$ with μ, which in the single task setting is simply an indicator variable that is 1 if the task was completed, and 0 otherwise. The payment rule for the agent is now $V \cdot \mu - \hat{p}_{[2]} \cdot V$. Just as in the previous restricted setting, this agent is the only one that has a positive expected utility for attempting the task, under this payment rule. Specifically, its expected utility is $V \cdot (p_i \cdot (1 - \hat{p}_{[2]}) - (1 - p_i) \cdot \hat{p}_{[2]})$, which is positive if and only if $p_i > \hat{p}_{[2]}$. Note that this mechanism is incentive compatible. Thus, truth-telling is always the best strategy for the agents regardless of the others' types and declarations.

3.3 Case 3: Costs and Failures

To address the full single-task setting, with both costs and failures, we combine the two mechanisms for the special cases. Mechanism SingleTask, defined below, assigns the task to the agent whose declared type maximizes the expected welfare. The agent's payment starts at a baseline of the negative of the expected welfare if this agent did not participate, and it is then paid an additional V if it successfully completes the task.

The reader can verify that imposing the restriction of either $p_i = \hat{p}_i = 1$ or $c_i = \hat{c}_i = 0$ on each agent i except for the "dummy" agent 0 (which always has $p_0 = 0$ and $c_0 = 0$) reduces this mechanism to the ones described above.

Mechanism 1 SINGLETASK

Let $j \leftarrow \arg\max_k(\hat{p}_k \cdot V - \hat{c}_k)$ {break ties in favor of smaller k}
$f_j(\hat{\theta}) = \{1\}$
$r_j(\hat{\theta}, \mu) = V \cdot \mu - max_{k \neq j}(\hat{p}_k \cdot V - \hat{c}_k)$
for all $i \neq j$ **do**
  $f_i(\hat{\theta}) = \emptyset$
  $r_i(\hat{\theta}, \mu) = 0$

We remind the reader that the dummy agent never gets any payment so the indices above refer only to the real agents. To exemplify the execution of Mechanism SINGLETASK, consider the types listed in Table 1. Let V be 210. If all declarations are truthful, the task is assigned to agent 3, resulting in an expected total welfare of $0.9 \cdot 210 - 60 = 129$. If agent 3 did not participate, the task would instead be assigned to agent 2, for an expected welfare of $210(1.0) - 100 = 110$. The payment that agent 3 receives is thus $210 - 110 = 100$ if it succeeds and -110 if it fails. Agent 3's own cost is 60, and thus its expected utility is $100(0.9) - 110(0.1) - 60 = 19$.

Agent	c_i	p_i
1	30	0.5
2	100	1.0
3	60	0.9

Table 1: Agent types used in an example for Mechanism SINGLETASK.

Theorem 3.1 *The* SINGLETASK *mechanism satisfies DSIC, IR, CR, EE.*

The proof of this theorem is omitted because it follows directly from Theorem 4.1.

4 Multiple Tasks

We now return to the original setting, consisting of t tasks for which the center has a non-combinatorial valuation. Because the setting disallows any interaction between tasks, we can construct a mechanism (MULTIPLETASK, formally specified below) that satisfies all of our goals by generalizing Mechanism SINGLETASK.

This mechanism allocates tasks to maximize the expected welfare according to the declared types. The payment rule for each agent is divided into two terms. The second term is an offset equal to the expected welfare if agent i did not participate.

Mechanism 2 MultipleTask

for all i **do**

$f_i(\hat{\theta}) = f_i^*(\hat{\theta})$

$\hat{r}_i(\mu_i) = \mathbf{E}_{\mu_{-i}{}_{f^*(\hat{\theta})}(\hat{p}_{-i})}[W_{-i}(f^*(\hat{\theta}), (\mu_i, \mu_{-i}))] - \mathbf{E}_{\mu_{-i}{}_{f^*_{-i}(\hat{\theta}_{-i})}(\hat{p}_{-i})}[W_{-i}(f^*_{-i}(\hat{\theta}_{-i}), \mu_{-i})]$

This term is independent of agent i. The first term is a function of agent i's completion vector μ_i. Given μ_i, the mechanism measures the expected welfare of all other participants $\mathbf{E}_{\mu_{-i}{}_{f^*(\hat{\theta})}(\hat{p}_{-i})}[W_{-i}(f^*(\hat{\theta}), (\mu_i, \mu_{-i}))]$ according to the *declarations* of the other agents. In this way, agent i's payment does not depend on the true types of the other agents, allowing us to achieve incentive compatibility. Note that for agents who are assigned no tasks, these two terms are identical, and thus they receive zero payment. Note also, that μ_i affects only the center's valuation but not the valuations of the other agents. The mechanism is equivalent to the SingleTask mechanism applied to each task separately. Nevertheless, the above formulation is more convenient to generalize.

Consider the outsourcing example of Section 2.5.1. Suppose that the agents are truth-telling. The optimal allocation allocates S_A to A_1 and S_B to A_2. We shall now compute the contract offered to Agent 1. Let us first compute the second term. Without A_1 both items will be allocated to A_2. This yields an expected value of $300 \cdot 0.7 + 200 \cdot 0.9 = 390$ to the center and thus the second term is $390 - 170 = 220$. Suppose that A_1 completes its task. The expected center's value is then $300 + 0.9 \cdot 200 = 480$. Thus, the expected welfare of the others in this case is $480 - 90 = 390$, and therefore, $\hat{r}(\{1\}) = 390 - 220 = 170$. When A_1 fails, the center's expected value drops to 180 and the expected welfare of the others equals 90. Thus, $\hat{r}(\{0\}) = -130$. Note that in this case the agent pays a fine to the mechanism, which may be undesirable. Currently, we do not know how to deal with this issue, or in general, how to minimize the risk that the agents face. The expected utility of the agent is $\overline{u}_{A_1} = (0.8 \cdot 170 - 0.2 \cdot 130) - 100 = 10$.

Theorem 4.1 *The* MultipleTask *mechanism satisfies DSIC, IR, CR, EE.*

Proof: We will prove each property separately.

1. *Individual Rationality (IR):*

   Consider an arbitrary agent i, a profile of true types θ, and a profile $\hat{\theta}_{-i}$ of declared types for all agents other than agent i. We will show that agent i's expected utility, if it truthfully declares its type, is always non-negative.

   Let $f^* = f((\theta_i, \hat{\theta}_{-i}))$. By the definition of the mechanism, the expected utility of the agent when it is truthful is:

   $$\overline{u}_i = \mathbf{E}_{\mu_i{}_{f^*((p_i, \hat{p}_{-i}))}}[\hat{r}_i(\mu_i) - c_i(f_i^*)].$$

   This utility is independent of the actual types of the other agents. Consider $\hat{r}_i(\mu_i)$. The second term of it, $h_i := \mathbf{E}_{\mu_{-i}{}_{f^*_{-i}(\hat{\theta}_{-i})}(\hat{p}_{-i})}[W_{-i}(f^*_{-i}(\hat{\theta}_{-i}), \mu_{-i})]$, is independent of agent i.

Since the agent is truthful, $c_i(f_i^*) = \hat{c}_i(f_i^*)$. Observe that the total welfare equals $W_{-i} + c_i(.)$. Thus,

$$\begin{aligned}
\overline{u}_i &= \mathbf{E}_{\mu_{i\,f^*}(p_i)}[\mathbf{E}_{\mu_{-i\,f^*}(\hat{p}_{-i})}[W_{-i}(f^*, (\mu_i, \mu_{-i})) - \hat{c}_i(f_i^*) - h_i]] \\
&= \mathbf{E}_{\mu_{i\,f^*}(p_i)}[\mathbf{E}_{\mu_{-i\,f^*}(\hat{p}_{-i})}[W(f^*, (\mu_i, \mu_{-i}))]] - h_i \\
&= \mathbf{E}_{\mu_{f^*}((p_i, \hat{p}_{-i}))}[W(f^*, \mu)] - h_i.
\end{aligned}$$

The first term of the above expression is exactly the expected welfare measured according to $(\theta_i, \hat{\theta}_{-i})$. $f_{-i}^*(\hat{\theta}_{-i})$ is a feasible allocation of the tasks among the agents. Since agent i does not get any tasks in it: $h_i = \mathbf{E}_{\mu_{f_{-i}^*(\hat{\theta}_{-i})}((p_i, \hat{p}_{-i}))}[W(f_{-i}^*(\hat{\theta}_{-i}), \mu)]$. Since f^* optimizes $\mathbf{E}_{\mu_{f^*}((p_i, \hat{p}_{-i}))}[W(f^*, \mu)]$, the individual rationality follows.

2. *Incentive Compatibility (DSIC):*

   Again consider an arbitrary triplet i, θ, $\hat{\theta}_{-i}$. Let $f^* = f(\theta_i, \hat{\theta}_{-i})$ denote the allocation when agent i is truthful. Let $\hat{\theta}_i$ be another declaration for the agent and let $f' = f((\hat{\theta}_i, \hat{\theta}_{-i}))$ denote the resulting allocation. Let $\overline{u}_i$ and $\overline{u}_i'$ denote the expected utilities of agent i in both cases. We need to show that $\overline{u}_i \geq \overline{u}_i'$. The same steps as in the individual rationality case imply that:

$$\begin{aligned}
\overline{u}_i &= \mathbf{E}_{\mu_{f^*}((p_i, \hat{p}_{-i}))}[W(f^*, \mu)] - h_i \\
\overline{u}_i' &= \mathbf{E}_{\mu_{f'}((p_i, \hat{p}_{-i}))}[W(f', \mu)] - h_i.
\end{aligned}$$

   (In both cases W is measured according to the actual cost of agent i.) Since f^* optimizes $\mathbf{E}_{\mu_{f^*}((p_i, \hat{p}_{-i}))}[W(f^*, \mu)]$, the incentive compatibility follows.

3. *Individual Rationality for the Center (CR):* We will actually prove the stronger claim that the center's utility is always non-negative for all true types θ, regardless of the declared type $\hat{\theta}$ and output μ of the attempts.

   Because the center's valuation is non-combinatorial (additive), its utility can be described as a simple sum: $u_M(g(\hat{\theta}), \mu) = \sum_i (V(\mu_i) - r_i(\hat{\theta}_i, \mu))$.

   We now show that all terms in this sum are non-negative. Consider an arbitrary agent i. Due to the payment definition:

$$\begin{aligned}
& V(\mu_i) - r_i(\hat{\theta}_i, \mu) \\
= \;& V(\mu_i) - \mathbf{E}_{\mu_{-i\,f^*(\hat{\theta})}(\hat{p}_{-i}))}[W_{-i}(f^*(\hat{\theta}), (\mu_i, \mu_{-i}))] + \mathbf{E}_{\mu_{-i\,f_{-i}^*(\hat{\theta}_{-i})}((\hat{p}_{-i}))}[W_{-i}(f_{-i}^*(\hat{\theta}_{-i}), \mu_{-i})].
\end{aligned}$$

   Consider the second term of the above expression. The only influence μ_i has on W_{-i} is to affect $V(\mu_i)$. Let $\tilde{W}_{-i}$ denote the expected welfare of the other agents when agent i fails in all its tasks. Note that due to the additivity of the center's valuation $W_{-i} = V(\mu_i) + \tilde{W}_{-i}$. Thus we get:

$$\begin{aligned}
& V(\mu_i) - r_i(\hat{\theta}_i, \mu) \\
= \;& V(\mu_i) - (V(\mu_i) + \mathbf{E}_{\mu_{-i\,f^*(\hat{\theta})}(\hat{p}_{-i})}[\tilde{W}_{-i}(f^*(\hat{\theta}), \mu_{-i})]) + \mathbf{E}_{\mu_{-i\,f_{-i}^*(\hat{\theta}_{-i})}(\hat{p}_{-i})}[W_{-i}(f_{-i}^*(\hat{\theta}_{-i}), \mu_{-i})] \\
= \;& \mathbf{E}_{\mu_{-i\,f_{-i}^*(\hat{\theta}_{-i})}(\hat{p}_{-i})}[W_{-i}(f_{-i}^*(\hat{\theta}_{-i}), \mu_{-i})] - \mathbf{E}_{\mu_{-i\,f^*(\hat{\theta})}(\hat{p}_{-i})}[\tilde{W}_{-i}(f^*(\hat{\theta}), \mu_{-i})].
\end{aligned}$$

In the last line, both terms compute the expected welfare ignoring agent i's contribution to the center's value and its cost. The second term is equal to the expected welfare of an allocation that gives all the tasks in $f^*(\hat{\theta})_i$ to the dummy agent. This is also a feasible allocation of the tasks to all agents but i. Thus, the optimality of $f^*(\hat{\theta}_{-i})$ implies that

$$\mathbf{E}_{\mu_{-i}{}_{f^*_{-i}(\hat{\theta}_{-i})}(\hat{p}_{-i})}[W_{-i}(f^*_{-i}(\hat{\theta}_{-i}), \mu_{-i})] - \mathbf{E}_{\mu_{-i}{}_{f^*(\hat{\theta})}(\hat{p}_{-i})}[\tilde{W}_{-i}(f^*(\hat{\theta}), \mu_{-i})] \geq 0.$$

Since this argument holds for every agent i, the center's utility is always non-negative.

4. *Economic Efficiency (EE):* Immediate from the choice of the allocation $f(\cdot)$.

□

4.1 Combinatorial Valuation

So far we assumed that the center's valuation is simply the sum of the values it assigns to each completed task. This assumption is unrealistic in most settings. A natural generalization of our basic setting is to allow the center's valuation $V(\cdot)$ to be any non decreasing function of the accomplished tasks. Unfortunately, in this setting, it is impossible to satisfy all of our goals simultaneously. Before we show this let us note that this result is not surprising. In principle, budget balance and efficiency do not mix well. In particular, it is known that even without failures, the payment of any path procurement mechanism which must always procure a path can be much higher than the actual cost of the winning path [8]. It is not difficult to show that this result implies the impossibility to satisfy all our goals. Yet, our setup is more general and the problem occurs already in very simple instances. Thus, for completeness, we formulate the theorem and the proof.

Theorem 4.2 *When V is combinatorial, no mechanism exists that satisfies DSIC, IR, CR, and EE for any $n \geq 2$ and $t \geq 2$.*

Proof: The basic intuition is as follows. Consider the case of two tasks, each of which can only be completed by one of the agents. The center only has a positive value (call it x) for the completion of both tasks. Since both agents add a value of x to the system, they can each extract a payment arbitrarily close to x from the center under an incentive compatible mechanism. This causes the center to pay $2x$ although it will gain only x from the completion of the tasks.

The formal proof is by induction. We first show that no mechanism exists that satisfies DSIC, IR, CR, and EE for the base case of $n = t = 2$. The inductive step then shows that, for any $n, t \geq 2$, incrementing either n or t does not alter this impossibility result.

Base Case: Assume by contradiction that there exists a mechanism Γ_1 that satisfies the four properties above for $n = t = 2$. The four types that we will use in this proof, θ_1, θ'_1, θ_2, and θ'_2, are defined in Table 2. The center only has a positive

θ_1 :	$p_{11} = 1$	$c_{11} = 0$	$p_{12} = 0$	$c_{12} = 0$
θ'_1 :	$p'_{11} = 1$	$c'_{11} = 2$	$p'_{12} = 0$	$c'_{12} = 0$
θ_2 :	$p_{21} = 0$	$c_{21} = 0$	$p_{22} = 1$	$c_{12} = 0$
θ'_2 :	$p'_{21} = 0$	$c'_{21} = 0$	$p'_{22} = 1$	$c'_{22} = 2$

Table 2: Agent types for proof of Theorem 4.2.

value when both tasks are completed. Specifically, $V(\oslash) = V(\{t_1\}) = V(\{t_2\}) = 0$, and $V(\{t_1, t_2\}) = 3$.

We will use three possible instances in order to derive properties that must hold for Γ_1. In each instance, the true and declared type of agent 2 is θ_2.

Instance 1: Let the true and declared type of agent 1 be θ_1. By EE, task 1 is assigned to agent 1, and task 2 is assigned to agent 2. Formally, $f_1(\theta_1, \theta_2) = \{1\}$ and $f_2(\theta_1, \theta_2) = \{2\}$.

The expected utility for agent 1 is simply its payment when both tasks are completed, because both agents always complete their assigned task, and a_1 has no costs.

$$\mathbf{E}_{\mu_{f((\theta_1,\theta_2))}((p_1,p_2))}[u_1(g(\theta_1, \theta_2), \mu, \theta_1)] = r_1((\theta_1, \theta_2), (1, 1)).$$

Instance 2: Now, let agent 1's true and declared type be θ'_1.

By EE, the task allocation would not change from the previous instance. Both tasks would still be completed, and agent 1's expected utility would be:

$$\mathbf{E}_{\mu_{f((\theta'_1,\theta_2))}((p'_1,p_2))}[u_1(g(\theta'_1, \theta_2), \mu, \theta'_1)] = r_1((\theta'_1, \theta_2), (1, 1)) - 2.$$

By IR, it must be the case that $r_1((\theta'_1, \theta_2), (1, 1)) \geq 2$.

Instance 3: In this instance, let agent 1's true type be θ_1 and let its declared type be θ'_1. Its expected utility would be the same as in instance 2, except that agent 1 now has zero cost.

$$\mathbf{E}_{\mu_{f((\theta'_1,\theta_2))}((p'_1,p_2))}[u_1(g(\theta'_1, \theta_2), \mu, \theta_1)] = r_1((\theta'_1, \theta_2), (1, 1)).$$

Now, return to the original instance 1, in which agent 1's true and declared type is θ_1. In order for agent 1 not to have an incentive to declare θ'_1, it must be the case that agent 1 is paid at least 2 in this instance.

$r_1((\theta_1, \theta_2), (1, 1)) \geq r_1((\theta'_1, \theta_2), (1, 1)) \geq 2$.

Moving on to agent 2, due to the symmetry of the types, the same argument implies that $r_2(\theta, (1, 1)) \geq 2$. However, we now see that the center has to pay too much to the agents.

$$\mathbf{E}_{\mu_{f(\theta)}(p)}[u_M(g(\theta), \mu)] = V(\{t_1, t_2\}) - r_1(\theta, (1, 1)) - r_2(\theta, (1, 1)) \leq 3 - 2 - 2 = -1.$$

Since this violates CR, we have reached a contradiction, and proved the base case.

Inductive Step: We now prove the inductive step, which consists of two parts: incrementing n and incrementing t. In each case, the inductive hypothesis is that no mechanism satisfies DSIC, IR, CR, and EE for $n = x$ and $t = y$, where $x, y \geq 2$.

Part 1: For the first case, we must show that no mechanism exists that satisfies DSIC, IR, CR, and EE for $n = x+1$ and $t = y$, which we will prove by contradiction. Assume that such a mechanism Γ_1 does exist.

Consider the subset of instances where $n = x + 1$ and $t = y$ such that there exists an "extra" agent who has a cost of 1 and success probability 0 for every task. Because of EE, Γ_1 can never assign the task to the extra agent. Because of IR, Γ_1 can never receive a positive payment from the extra agent. Since the only effect that the extra agent can have on the mechanism is to receive a payment from the center, we can construct a mechanism that satisfies DSIC, IR, CR, and EE for all instances where $n = x$ and $t = y$ as follows: add the extra agent to the profile of declared types, execute Γ_1, and ignore the payment function and assignment for the extra agent. The existence of such a mechanism contradicts the inductive hypothesis.

Part 2: For the second case, we need to show that no mechanism can satisfy DSIC, IR, CR, and EE for $n = x$ and $t = y + 1$. We use a similar proof by contradiction, starting from the assumption that such a mechanism does exist.

Consider the subset of instances where $n = x$ and $t = y + 1$ such that there exists an "extra" task t_e that is not involved in any dependencies and for which the center receives no value from its completion. Since an assignment rule that never assigns the extra task to an agent will never prevent the mechanism from satisfying the four goals, the existence of a mechanism that satisfies these goals implies the existence of a mechanism Γ_1 that satisfies the goals and never assigns the extra task.

We can then reach a contradiction using a similar construction: create a mechanism for $n = x$ and $t = y$ that adds an extra task and then execute Γ_1. Since such a mechanism will satisfy DSIC, IR, CR, and EE for $n = x$ and $t = y$, we have again contradicted the inductive hypothesis, and the proof is complete. □

When CR is given up, it is possible to attain the other goals.

Theorem 4.3 *The* MultipleTask *mechanism satisfies DSIC, IR, EE, even when V is combinatorial.*

We omit the proof of this theorem, due to its similarity to the proof of Theorem 4.1. Intuitively, the potential for a combinatorial V does not change the fact that the mechanism aligns the utility of the agents with the welfare of the entire system. Moreover, the utility of an agent is still independent of the *true* types of the other agents.

In order to demonstrate the mechanism consider a situation in which the center wants to procure a pair of tasks A and B. There are two agents A_1 and A_2 that are each capable of performing each of the jobs. A_1 has zero cost and probability 1 for performing task A but has a high cost for performing task B. On the other hand, A_2 has zero cost and probability 1 for performing task B but a high cost for performing A. Suppose that the center's value is 3 if both tasks are completed and 0 otherwise. Consider an application of the MultipleTask mechanism to this instance. Assume that both agents are truthful. The mechanism will allocate task A to A_1 and B to A_2. Let us consider $\hat{r}_{A_1}(\{1\})$. Without the agent, it is optimal to allocate both tasks to the dummy agent and obtain zero welfare. Hence, $\mathbf{E}[W_{-1}] = 0$. Since the probability that A_2 will succeed in B is 1 and its cost is 0, $\mathbf{E}_{\mu_{-1}{}_{f^*(\hat{\theta})}(\hat{p}_{-1})}[W_{-1}(f^*(\hat{\theta}), (\{1\}, \mu_{-1}))] = 3$. Thus, $\hat{r}_{A_1}(\{1\}) = 3$. The case of A_2 is similar. Since both agents will succeed, the overall payment will be 6 causing the center a loss of 3.

4.2 Dependencies among Tasks

We now consider the natural possibility of dependencies among tasks. We will study both the case of additive and combinatorial center valuations. Consider, for example, the path procurement instance described in Figure 3. Consider the lower path. The agent that owns the second edge of the path can attempt to route the object only if it will be successfully routed by the first edge of the path. Thus, when this path is chosen, the cost of the second agent is dependent on the actual type of the first one.

We say that a task j is *dependent* on a set S of tasks if j cannot be attempted unless all tasks in S are successfully finished. We assume that there are no dependency cycles. The tasks must be executed according to a topological order of the underlying dependency graph. If a task cannot be attempted, the agent assigned to that task does *not* incur the costs of attempting it.

Definition 5 (Task allocation mechanism with dependencies) *A* task allocation mechanism with dependencies *is a direct mechanism composed of the following stages:*

Decision *Given the declaration vector $\hat{\theta}$, the mechanism computes the allocation $f(\hat{\theta})$ and the payment functions $\hat{r}_i(\mu)$ of all agents.*

Work *The agents attempt their tasks according to some arbitrary topological order that is computed before the decision stage. The cost of each agent i is the sum of the costs of all its attempted tasks. If a task j was not attempted, $\mu_j = 0$. The work stage is over when there are no more allocated tasks that can be attempted.*

Payment *Each agent i receives a payment of $\hat{r}_i(\mu)$ from the mechanism.*

Remarks It is possible to consider more complicated mechanisms that may retry or reallocate tasks. The positive results in this section will hold for these cases but the decision stage will be even more complex from a computational point of view. It is also natural to consider mechanisms in which the center may force agents to attempt dummy tasks that will cause them artificial costs even when they cannot attempt their actual tasks. We leave these to future research. Note that, given an allocation f and a vector of types Θ, the distribution over the completion vectors

$\mu_f(p)$ is no longer independent and may be non trivial to compute. We note that in this setup, it is assumed that the dependencies among the tasks are known to the center.

Unfortunately, the possibility of dependencies also makes it impossible to simultaneously satisfy DSIC, IR, CR, and EE.

Theorem 4.4 *When dependencies exist between tasks, even when the center's valuation is non combinatorial, no mechanism exists that satisfies DSIC, IR, CR, and EE for any* $n \geq 2$ *and* $t \geq 2$*.*

Proof: We will prove by contradiction that no mechanism can satisfy DSIC, IR, CR, and EE for the case of $n = t = 2$. By an inductive argument similar to the one used in the proof of Theorem 4.2, the result holds for all $n, t \geq 2$.

Assume that there exists a mechanism Γ_1 that satisfies DSIC, IR, CR, and EE for $n = t = 2$. We will use three possible instances in order to derive properties that must hold for Γ_1, but lead to a contradiction. The constants in these instances are that task 2 is dependent on task 1 and that the center has a value of 5 for task 2 being completed, but no value for the completion of task 1 in isolation. The five types that we will use, θ_1, θ_1', θ_1'', θ_2, and θ_2', are defined in Table 3.

$\theta_1:$	$p_{11} = 1$	$c_{11} = 2$	$p_{12} = 1$	$c_{12} = 1$
$\theta_1':$	$p_{11}' = 1$	$c_{11}' = 2$	$p_{12}' = 0$	$c_{12}' = 0$
$\theta_1'':$	$p_{11}'' = 1$	$c_{11}'' = 0$	$p_{12}'' = 1$	$c_{12}'' = 4$
$\theta_2:$	$p_{21} = 0$	$c_{21} = 1$	$p_{22} = 0$	$c_{12} = 0$
$\theta_2':$	$p_{21}' = 1$	$c_{21}' = 1$	$p_{22}' = 0$	$c_{22}' = 0$

Table 3: Agent types for proof of Theorem 4.4.

Instance 1: The true types are θ_1 and θ_2, and the declared types are θ_1 and θ_2'. To satisfy EE in the case in which θ_1 and θ_2' are instead the true types, task 1 is assigned to agent 2, and task 2 to agent 1. That is, $f_1(\theta_1, \theta_2') = \{2\}$ and $f_2(\theta_1, \theta_2') = \{1\}$. Since agent 2's true type is θ_2, it will fail to complete task 1, preventing task 2 from being attempted. Thus, $\mu = (0, 0)$ with probability 1. The expected utility for agent 1 is then:

$\mathbf{E}_{\mu_{f(\theta_1,\theta_2')}((p_1,p_2))}[u_1(g(\theta_1, \theta_2'), \mu, \theta_1)] = r_1((\theta_1, \theta_2'), (0, 0))$.

Instance 2: The true types are θ_1' and θ_2, and the declared types are θ_1, and θ_2'. Thus, the only difference from instance 1 is agent 1's true type, which is insignificant because agent 1 never gets to attempt a task. Thus, we have a similar expected utility function:

$\mathbf{E}_{\mu_{f((\theta_1,\theta_2'))}((p_1',p_2))}[u_1(g(\theta_1, \theta_2'), \mu, \theta_1')] = r_1((\theta_1, \theta_2'), (0, 0))$.

Instance 3: The true types are θ_1' and θ_2, and the declared types are θ_1', and θ_2'. Agent 1's declared type has been changed to also be θ_1'. Both tasks will be allocated to the dummy agent: $f_1(\theta_1', \theta_2') = f_2(\theta_1', \theta_2') = \emptyset$. Therefore, $\mu = (0, 0)$ still holds with probability 1, and we get the following equations for the expected utility of the two agents:

$\mathbf{E}_{\mu_{f((\theta'_1,\theta'_2))}((p'_1,p'_2))}[u_1(g(\theta_1,\theta'_2),\mu,\theta'_1)] = r_1((\theta'_1,\theta'_2),(0,0))$

$\mathbf{E}_{\mu_{f((\theta'_1,\theta'_2))}((p'_1,p'_2))}[u_2(g(\theta_1,\theta'_2),\mu,\theta'_2)] = r_2((\theta'_1,\theta'_2),(0,0))$.

If $r_2((\theta'_1,\theta'_2),(0,0)) < 0$, then IR would be violated if θ'_2 were indeed the true type of agent 2. Since the center thus cannot receive a positive payment from agent 2, and it never gains any utility from the completed tasks, the CR condition requires that $r_1((\theta'_1,\theta'_2),(0,0)) \leq 0$. Thus, agent 1's utility cannot be positive: $\mathbf{E}_{\mu_{f((\theta'_1,\theta'_2))}((p'_1,p'_2))}[u_1(g(\theta_1,\theta'_2),\mu,\theta'_1)] \leq 0$.

Notice that if agent 1 lied in this instance and declared its type to be θ_1, then we are in instance 2. So, to preserve DSIC, agent 1 must not have an incentive to make this false declaration. That is, it must be the case that: $\mathbf{E}_{\mu_{f((\theta_1,\theta'_2))}((p'_1,p_2))}[u_1(g(\theta_1,\theta'_2),\mu,\theta'_1)] = r_1((\theta_1,\theta'_2),(0,0)) \leq \mathbf{E}_{\mu_{f((\theta'_1,\theta'_2))}((p'_1,p'_2))}[u_1(g(\theta_1,\theta'_2),\mu,\theta'_1)] \leq 0$.

Instance 1: Now we return to the first instance. Having shown that $r_1((\theta_1,\theta'_2),(0,0)) \leq 0$, we know that when agent 1 declares truthfully in this instance, its expected utility will be non-positive.

We will now show that agent 1 must have a positive expected utility if it falsely declares θ''_1. In this case, both tasks are assigned to agent 1. That is, $f_1(\theta''_1,\theta'_2) = (1,1)$. We know that $r_1((\theta''_1,\theta'_2),(1,1)) \geq 4$ by IR for agent 1, because if θ''_1 were agent 1's true type, then both tasks would be completed and agent 1 would incur a cost of 4.

We now know that if agent 1 falsely declares θ''_1 in instance 1, then:

$\mathbf{E}_{\mu_{f((\theta''_1,\theta'_2))}((p_1,p_2))}[u_1(g(\theta''_1,\theta'_2),\mu,\theta_1)] = r_1((\theta''_1,\theta'_2),(1,1)) - (c_{11}+c_{12}) \geq 4-3 = 1$.

Thus, agent 1 has an incentive to falsely declare θ''_1 in instance 1, violating DSIC. Thus, we have reached a contradiction and completed the proof for the case of $n,t=2$. As stated before, this argument extends to all $n,t \geq 2$. □

Intuitively, there are two main problems that are caused by dependencies – the first is that they add a combinatorial nature to the center's valuation and the second is that we cannot avoid making an agent's payment depend on the true types of the other agents, because the tasks it attempts depend on the success or failure of the other agents. Next, we show that in the general case, even without CR, one cannot expect incentive compatible mechanisms.

Theorem 4.5 *When dependencies exist between tasks and the center's valuation is combinatorial no mechanism exists that satisfies DSIC, IR, and EE for any $n \geq 2$ and $t \geq 2$.*

Proof:(Sketch) The proof is similar to the proof of Theorem 4.4 and thus we only sketch it. Consider the following setup. There are three tasks and two agents. The center's valuation is $V(\{1,2\}) = V(3) = 3$. Task 2 is dependent on task 1. An illustration to this setup is when the center needs to procure a path in a graph that contains two disjoined paths. One path has two edges $(1,2)$ and one has only one edge 3. Suppose that agent 1 is the only one capable of performing task 1 and agent 2 is the only agent capable of performing tasks 2 and 3. Consider the case that agent 1 declares a zero cost and success probability 1. Suppose that the actual success probabilities of agent 2 are 1 in both, task 2 and task 3. It is known [15] that the payments that are offered to an agent are dependent only on the chosen allocation and the declarations of the other agents. In other words, the mechanism must offer prices to agent 2 for each of the two task sets $\{1,2\}$ and $\{3\}$, which are independent of agent 2's declaration. The mechanism must then choose the allocation that gives agent 2 a maximal utility according to its declaration (see [15]). Let $u^2_{\{1,2\}}(F)$ denote the expected utility of agent 2 when the chosen allocation is $\{\{1\},\{2\}\}$ and agent 1 fails, $u^2_{\{1,2\}}(S)$ its expected utility when agent 1 succeeds, and $u^2_{\{3\}}$ its expected utility when the allocation $\{\phi,\{3\}\}$ is chosen (suppressing the costs of agent 2). It is always possible to set the costs of agent 2 such that $u^2_{\{1,2\}}(F) \neq u^2_{\{1,2\}}(S)$. Suppose that $u^2_{\{1,2\}}(F) < u^2_{\{1,2\}}(S)$. It is also possible to set the costs of agent 2 such that $u^2_{\{1,2\}}(F) < u^2_{\{3\}} < u^2_{\{1,2\}}(S)$. But then the allocation that agent 2 prefers depends on the **actual** failure probability of agent 1. In other words, if agent 1's success probability is 0 yet it reports 1, it is better for agent 2 to report a cost that will cause the mechanism to choose the allocation $\{\phi,\{3\}\}$. Similarly, if agent 1 reports a small probability on task 1 but its actual success probability is high, it is better for agent 2 to report a high cost for task 3 which will cause the mechanism to choose the allocation $\{\{1\},\{2\}\}$. This contradicts DSIC. The reasoning when $u^2_{\{1,2\}}(F) > u^2_{\{1,2\}}(S)$ is similar. □

In light of Theorem 4.5 we now abandon the notion of implementation in dominant strategies. Fortunately, we can still provide equilibrium versions of our properties. We say that a vector of strategies is an equilibrium if no agent can get a better contract by deviating to another strategy regardless of the actual *types* of the other agents. Our notion is weaker than a dominant strategy equilibrium, but stronger than a Bayesian-Nash equilibrium, because the equilibrium does not depend on any beliefs that the agents may have about the types of the other agents. We now define the equilibrium versions of our goals.

Definition 6 (Ex-post Incentive Compatibility) *A direct mechanism satisfies* ex-post incentive compatibility *(ICE) if:*

$$\forall i,\theta,\theta_i',\ \mathbf{E}_{\mu_{f((\theta_i,\theta_{-i}))}(p)}[u_i(g(\theta_i,\theta_{-i}),\mu,\theta_i)] \geq \mathbf{E}_{\mu_{f((\theta_i',\theta_{-i}))}(p)}[u_i(g(\theta_i',\theta_{-i}),\mu,\theta_i)].$$

In other words, in equilibrium, each agent gets the best contract by being truthful.

Definition 7 (Ex-post Individual Rationality) *A direct mechanism satisfies* ex-post individual rationality *(IRE) if:*

$$\forall i,\theta,\ \mathbf{E}_{\mu_{f((\theta_i,\theta_{-i}))}(p)}[u_i(g(\theta_i,\theta_{-i}),\mu,\theta_i)] \geq 0.$$

In other words, in equilibrium, the expected utility of a truthful agent is always non-negative. We stress that agents may end up losing, for example, due to their own failures.

Definition 8 (Ex-post Economic Efficiency) *A direct mechanism satisfies* ex-post economic efficiency *(EEE) if it satisfies ex-post IC and if:*

$$\forall \theta, f'(\cdot),\ \mathbf{E}_{\mu_{f(\theta)}(p)}[W(f(\theta), \mu)] \geq \mathbf{E}_{\mu_{f'(\theta)}(p)}[W(f'(\theta), \mu)].$$

We now define the Mechanism EX-POST-MULTIPLETASK, which differs from Mechanism MULTIPLETASK only in that the first term of the payment rule uses the *actual* completion vector, instead of the distribution induced by the declarations of the other agents.

Mechanism 3 EX-POST-MULTIPLETASK

for all i **do**

$f_i(\hat{\theta}) = f_i^*(\hat{\theta})$

$r_i(\hat{\theta}, \mu) = W_{-i}(f^*(\hat{\theta}), \mu) - \mathbf{E}_{\mu_{-i}{}_{f^*_{-i}(\hat{\theta}_{-i})}(\hat{p}_{-i})}[W_{-i}(f^*_{-i}(\hat{\theta}_{-i}), \mu_{-i})]$

While the first term of the payment is calculated according to the actual completion vector μ, the second term depends only on the reported types of the other agents.

Theorem 4.6 *The* EX-POST-MULTIPLETASK *mechanism satisfies ex-post IC, IR, and EE, even when dependencies exist between the tasks, and the center's valuation is combinatorial.*

Proof:(Sketch) The proof is similar to the proof of Theorem 4.1 and thus we only sketch it. From the definition of the setup, given an allocation, the cost of each agent j is determined by the task completion vector, as this vector determines which tasks the agent attempted. We denote the agent's cost by $c_j(\mu)$. The expected cost $\overline{c}_j(\mu)$ of the agent is thus determined by the actual types of the agent and the allocation.

Ex-post Individual Rationality (IRE): Assume that all agents are truthful. We will show that the expected utility of each agent i is non negative. Let $f^* = f(\theta)$ be the resulting allocation. Recall that f^* maximizes the expected welfare. For every completion vector μ, the agent's utility is given by:

$$u_i = -c_i(\mu) + W_{-i}(f^*, \mu) - \mathbf{E}_{\mu_{-i}{}_{f^*_{-i}(\theta_{-i})}(p_{-i})}[W_{-i}(f^*_{-i}(\theta_{-i}), \mu_{-i})].$$

The last term is independent of agent i. We will denote it by h_i. Given the completion vector μ, $W(f^*, \mu) = -c_i(\mu) + W_{-i}(f^*, \mu)$. Thus, by the linearity of expectation, the expected utility of the agent is given by:

$$\begin{aligned} \overline{u}_i &= \mathbf{E}_{\mu_{f^*}(p)}[-c_i(\mu) + W_{-i}(f^*, \mu)] - h_i \\ &= \mathbf{E}_{\mu_{f^*}(p)}[W(f^*, \mu)] - h_i. \end{aligned}$$

As in the proof of Theorem 4.1, the optimality of f^* implies that the second term is bounded by the first, i.e. that $\overline{u}_i \geq 0$.

Ex-post Incentive Compatibility (ICE): The equilibrium expected utilities of the agents are calculated as in the IR case. Apart from that, the proof is identical to the IC part of the proof of Theorem 4.1.

Ex-post Economic Efficiency (EEE): Immediate from the choice of f^*. □

Remark In contrast to the previous setups we do not know how to generalize this result to the case of possible deliberate failures. The reason is that failures may reduce the set of tasks that the agent will be able to attempt, and thus by failing, the agent can reduce its own cost.

Example Examine the path procurement example described in Figure 3. Consider the Ex-Post-MultipleTask mechanism in which the set of possible outputs is the set of all $s - t$ paths (i.e., the mechanism either acquires one of the paths or no path at all). Suppose that the upper edge e_1 has a success probability of 0.1, and each of the lower edges e_2 and e_3 have probability 0.5. Assume that all costs are 1 and the center's value from a path completion is 20. Assume that agent 1 owns e_1, e_2 and agent 2 owns e_3.

Suppose that the agents are truthful and consider agent 2. The mechanism will choose the the lower path $\{e_2, e_3\}$. The cost of agent 2 now depends on agent 1. If agent 1 will not complete its task, the agent's cost will be 0. Without the agent the expected welfare is $h_2 = 0.2 \cdot 20 - 1 = 1$. Consequently, the contract that will be offered to agent 2 will be:

$$\hat{r}_2(\{0,1,1\}) = (20 - 1) - 1 = 18 \text{ (In this case } u_2 = 17)$$
$$\hat{r}_2(\{0,1,0\}) = (0 - 1) - 1 = -2 \text{ (In this case } u_2 = -3)$$
$$\hat{r}_2(\{0,0,0\}) = (0 - 1) - 1 = -2 \text{ (In this case } u_2 = -2).$$

All other combinations are impossible. The expected utility of the agent will be 2.5. Note that the agent may lose even when it does not attempt its tasks. Now suppose that the actual probability of e_2 is 0 but agent 1 falsely reports it as 0.5. In this case agent 2 will always lose. Thus, it is better for it to report a probability 0 and cause the mechanism to choose the upper path. This lie would increase the agent's utility to 0.

5 Cost Verification

The Ex-Post-MultipleTask mechanism presented in Section 4.2 has two drawbacks, which seem unavoidable in our basic setup. First, the mechanism satisfies IR only in expectation (and in equilibrium). Therefore, participating agents may end up with large losses. This phenomenon may sabotage business relationships between the center and the agents, lead to law suits, etc. The second drawback is that the total payment might be very high, resulting in a negative expected utility to the center. In other words, the mechanism does not satisfy CR.

Previous work [15] has stressed the importance of ex post verification. That work considered a representative task-scheduling problem and showed that when the center can verify the costs of the agents *after* the work has been done, the set of implementable allocation functions increases dramatically. [15] did not consider the possibility of failures. We shall show that both drawbacks mentioned above can be overcome when cost verification is feasible.

Verification assumption The center can pay the agents after the tasks are performed. At this time the center knows the *actual cost* c_i of each agent i.

Below we define Mechanism Ex-Post-CompensationAndBonus . This is a variant of the compensation and bonus mechanism presented in [15]. The mechanism works as follows. Given the agents' declarations, the mechanism allocates the tasks optimally. After the work stage is over, the mechanism knows the actual cost of each agent. The mechanism compensates each agent according to its actual cost. It then gives each agent a bonus proportional to the actual welfare.

Mechanism 4 Ex-Post-CompensationAndBonus

The allocation $f^*(\hat{\theta})$ maximizes the expected welfare $E[W]$. Given strictly positive constants $(\chi_1, \ldots, \chi_n)$ the payments are defined as follows.
for all i **do**
  The *bonus* of agent i is defined as $b_i(\hat{\theta}, \mu) = \chi_i \cdot W(f^*(\hat{\theta}), \mu)$
  The *payment* to agent i is $r_i(\hat{\theta}, \mu) = c_i + b_i(\hat{\theta}, \mu)$

Consider any ex-post IR mechanism. Since every agent must be paid at least its expected cost, an obvious bound u^* on the center's expected utility is thus $\mathbf{E}_{\mu_{f^*(\theta)}(p)}[W(f^*(\theta), \mu)] = \mathbf{E}_{\mu_{f^*(\theta)}(p)}[V(\mu) - \sum_i c_i]$.The center can always decide not to allocate any task, obtaining a utility of zero. Thus, $u^* \geq 0$.

Theorem 5.1 *Under the verification assumption, the* Ex-Post-CompensationAndBonus *mechanism satisfies ex-post IC, IR, EE, and CR, even when dependencies among the tasks exist and the center's valuation is combinatorial. Moreover, for every $\epsilon > 0$ and a type vector θ, when the constants χ_i are small enough, the expected center's utility is at least $u^* \cdot (1 - \epsilon)$.*

Proof: We start with ex-post IC. Consider agent i and assume that the other agents are truthful. The cost of the agent is compensated for by the mechanism. As a result, its utility equals its bonus.

When the chosen allocation is f^*, the expected bonus of the agent equals $\chi_i \cdot \mathbf{E}[W(f^*, \mu)]$ (from the linearity of expectation). When the agent is truthful the mechanism computes the allocation according to the actual type vector. In this case, the chosen allocation is exactly the one that maximizes $\mathbf{E}[W(., \mu)]$, and henceforth, the agent's own expected utility.

Ex-post IR is satisfied since when the agents are truthful, the expected welfare of $\mathbf{E}[W(f^*, \mu)]$ is clearly non negative. Ex-post EE is satisfied by the choice of the allocation.

If the optimal expected center's utility is zero, then so is the expected bonus and hence the expected center's utility of the mechanism. Otherwise, $u^* > 0$. The center's utility thus equals $\mathbf{E}_{\mu_{f(\theta)}(p)}[V(\mu) - \sum_i (c_i + \chi_i \cdot W(f^*(\theta), \mu))]$. By linearity of expectation, and the optimality of the allocation, this equals $u^* \cdot (1 - \sum_i \chi_i)$. By choosing $\sum_i \chi_i \leq \epsilon$ we get the desired bound on the expected utility of the center. □

Remarks It is possible to generalize mechanism EX-POST-COMPENSATIONANDBONUS by adding to the payment $r_i(.)$ a function $h_i()$ that is independent of agent i's declaration and actions. In particular, if we let W_{-i} denote the optimal expected welfare which can be obtained without agent i and define the bonus as $\chi_i \cdot (W(f^*(\theta), \mu)) - W_{-i})$ we get that agents that do not attempt any task are paid zero. Note that agents may still end up with a negative utility but small χ_i factors restrain their potential losses.

6 Conclusion

In this paper we studied task allocation problems in which agents may fail to complete their assigned tasks. For the settings we considered (single task, multiple tasks with combinatorial properties, and multiple tasks with dependencies), we provided either a mechanism that satisfies our goals or an impossibility result, along with a possibility result for a weaker set of goals. Interestingly, the possibility of failures, forced us to abandon the strong concept of dominant strategies.

It is worth pointing out that many of our results hold when the set of possible failures is expanded to include rational, intentional failures. Such failures occur when agents try to increase their utilities by not attempting assigned tasks (and thus not incurring the corresponding cost). Intuitively, our positive results continue to hold because the payment rule aligns an agent's utility with the welfare of the system. If failing to attempt some subset of the assigned tasks increased the overall welfare, these tasks would not be assigned to any agent. Obviously, all impossibility results would still hold when we expand the set of possible actions for the agents. Our positive results can also be extended to various settings in which the set of the possible decisions is more complex. Among the possible extensions are the reattempting of tasks after failure, sequential allocation of tasks, and task duplications. Such extensions can lead to a significantly higher utility for the center and lower risks for the agents, but can also complicate the computation of our mechanisms even further.

Many interesting directions stem from this work. The computation of our allocation and payment rules presents non-trivial algorithmic problems. The payment properties for the center may be further investigated, especially in settings where CR must be sacrificed to satisfy our other goals. Similarly, it is possible that the CR and EE properties can be approximated. This work did not investigate this possibility.

We believe that the most important future work will be to consider a wider range of possible failures, and to discover new mechanisms to overcome them. In particular, we would like to explore the case in which agents may fail maliciously or irrationally. For this case, even developing a reasonable model of the setting provides a major challenge. This is because it is not clear how to model the way that the strategic considerations of rational agents are affected by the presence of irrational or malicious agents.

Acknowledgement We thank the anonymous reviewers for many insightful comments. We thank Inbal Ronen for her comments on an earlier draft of this paper.

References

[1] Aaron Archer and Éva Tardos. Frugal path mechanisms. *Proceedings of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 991–999, 2002.

[2] B. Awerbuch, Y. Azar, and A. Meyerson. Reducing truth-telling online mechanisms to online optimization. In *Proceedings of the 35th Symposium on the Theory of Computing*, 2003.

[3] Moshe Babaioff, Michal Feldman, and Noam Nisan. Combinatorial agency. In *Proceedings of the 7th ACM Conference on Electronic Commerce*, pages 18 – 28, New York, NY, USA, 2006. ACM Press.

[4] C. Boutilier, Y. Shoham, and M.P. Wellman. Special issue on economic principles of multi-agent systems. *Artificial Intelligence*, 94, 1997.

[5] E. Clarke. Multipart pricing of public goods. *Public Choice*, 18:19–33, 1971.

[6] C. d'Aspremont and L.A. Gerard-Varet. Incentives and incomplete information. *Journal of Public Economics*, 11(1):25–45, 1979.

[7] Kfir Eliaz. Fault tolerant implementation. *Review of Economic Studies*, 69:589–610, 2002.

[8] Edith Elkind, Amit Sahai, and Ken Steiglitz. Frugality in path auctions. *Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2004)*, pages 701–709, 2004.

[9] Michal Feldman, John Chuang, Ion Stoica, and Scott Shenker. Hidden-action in multi-hop routing. In *EC '05: Proceedings of the 6th ACM Conference on Electronic Commerce*, pages 117–126, New York, NY, USA, 2005. ACM Press.

[10] T. Groves. Incentives in teams. *Econometrica*, 41:617–631, 1973.

[11] Nathan Linial. Game theoretic aspects of computing. In *Handbook of Game Theory*, volume 2, pages 1339–1395. Elsevier Science Publishers B.V, 1994.

[12] A. Mas-Colell, M.D. Whinston, and J.R. Green. *Microeconomic Theory*. Oxford University Press, 1995.

[13] Milena Mihail, Christos Papadimitriou, and Amin Saberi. On certain connectivity properties of the internet topology. In *Proceedings of the 44rd Annual IEEE Symposium on the Foundations of Computer Science (FOCS 2001)*. IEEE, 2003.

[14] Nolan Miller, Paul Resnick, and Richard Zeckhauser. Eliciting honest feedback: The peer prediction method. *Management Science*, 51(9):1359–1373, September 2005.

[15] N. Nisan and A. Ronen. Algorithmic mechanism design. *Games and Economic Behavior*, 35:166–196, 2001.

[16] D. C. Parkes and L. H. Ungar. Iterative combinatorial auctions: Theory and practice. In *AAAI,IAAI*, pages 74–81, 2000.

[17] R. Porter, A. Ronen, Y. Shoham, and M. Tennenholtz. Mechanism design with execution uncertainty. In *Proceedings of UAI-02*, 2002.

[18] Jeffrey Rosenschein and Gilad Zlotkin. *Rules of Encounter*. MIT Press, 1994.

[19] Y. Shoham and M. Tennenholtz. Fair imposition. In *IJCAI*, pages 1083–1088, 2001.

[20] Rann Smorodinsky and Moshe Tennenholtz. Overcoming free riding in multi-party computations - the anonymous case. *Games and Economic Behavior*, 2005.

[21] W. Vickrey. Counterspeculations, auctions, and competitive sealed tenders. *Journal of Finance*, 16:15–27, 1961.

[22] W.E. Walsh and M.P. Wellman. A market protocol for decentralized task allocation: Extended version. In *The Proceedings of the Third International Conference on Multi-Agent Systems (ICMAS-98)*, 1998.

A Notes on the computation of our mechanism

FTMD problems give rise to non-trivial computational problems. All the mechanisms in this paper require the computation of optimal allocations and the agents' payments. This section briefly comments on these computations.

A.1 Computing optimal allocations

In the single task setup, the task simply needs to be allocated to the agent i that maximizes $p_iV - c_i$. This can be computed in $O(n)$ time. When there are t tasks and the center's valuation is additive, the computation can be done for each task separately. A computation time of $O(n \cdot t)$ thus suffices for the additive case. The combinatorial setup is by far more difficult. In the most general case, even describing the center's value requires exponential space. Thus, it may be more interesting to consider specific problems in which the structure of the center's value facilitates some compact representation. Unfortunately, in many such cases, computing the optimal allocation is at least as hard as computing some joint probabilities, which is often PSPACE hard. One such example is a variant of our path procurement example (Section 2.5.2) in which the mechanism is allowed to procure any subset of the edges of the graph. A principle-agent problem with a similar underlying allocation problem was shown to be PSPACE hard in [3]. Their argument can be adopted to our setup as well.

An intriguing question is whether it is possible to approximate the optimal allocation and construct payment functions that yield incentive compatible mechanisms. Thus far, except for cases in which the agent types are one dimensional, this approach has rarely succeeded. We leave this to future research.

A.2 Computing the payments

The payments in the single task or the additive cases can clearly be computed in polynomial time.

Consider the combinatorial setup. In all the mechanisms described in this paper, the payments are calculated according to either an actual or expected welfare (given the allocation, costs, and the declared failure probabilities). While the actual welfare can easily be computed in polynomial time, the computation of an expected welfare is likely to be hard. Fortunately, given the allocation and costs, the welfare is simply a bounded stochastic variable and can therefore be approximated by standard sampling of the failure probabilities.