Tendering mechanism for stochastic dynamic multi-agent projects

Endre Csóka
Eötvös Loránd University, Budapest, Hungary

Abstract

The main purpose of this paper is to model and solve the situation when we want to get a multi-agent project made, and we can call for tender for the tasks. A general stochastic dynamic multi-agent project requires a high level of cooperation, such as making decisions depending on some earlier chance events of other agents, while the working processes of the players are private information. Nevertheless, we will show an efficient incentive-compatible and individually rational mechanism for such a general model; and for perfect competition, we will show another mechanism, which is coalitional incentive-compatible, too.

1 Introduction

We deal with the design of stochastic multi-agent projects, where the agents work on their own separate tasks but the utility of the project is an arbitrary function of the results of the tasks. Stochastic scheduling problems are important cases in point. For a simple example, when the project consists of some parallel tasks and the completion time of the full project is the only matter; because here, the result of a task consists only of the completion time and the utility is a function of the maximum of these completion times.

The workflows of most real-life tasks contain chance events and decision opportunities. With such tasks, efficiency requires the agents to make the best decisions considering all earlier chance events of all agents. Continuing with the example of the parallel tasks, it may happen that an agent can choose between a faster and a cheaper way to continue, and if some of the other tasks are late then he should choose the cheaper way, but if all are on time then he should choose the faster way.

The real difficulty arises from that the workflow, the chance events and these decisions are private information of the corresponding agent. Furthermore, for example, if an agent can choose between a faster and a slower way, which differ only in the cost and the probability distribution on the completion time, then no one else can ever verify which way the agent chose. So even though the agents can lie, we want to make them interested in revealing all their private information, and in making the globally optimal decisions.

In our simplified model, there is a player called the principal(♀) and she can contract with some of the other players called agents(♂) to work with. Each agent can work according to his stochastic decision tree, which contains decision points and chance points as internal nodes, and a result and a real cost at each leaf. Costless contractible communication is allowed throughout the process and contracts can be made between the principal and each agent that specify the payment between them depending on both the achieved result of the agent and the communication between them. At the end, the principal pays the money to each agent according to their contract, she gets the utility corresponding to the set of the achieved results of the agents, and each agent pays the cost of the achieved leaf of his tree.

1.1 Related works

A similar stochastic dynamic model was studied recently (and parallelly) by Athey and Segal (2004–2007)[1], and their team mechanism uses the same technique as this paper. However, there are essential differences between the two papers. While that paper deals with a fixed set of somewhat interdependent players, we have a central player (the principal) and a fixed set of competiting agents, and an agent takes part in the project only if he agrees with the principal. And because of the very different motivation, balancing the utilities of the agents seems to have no use here, while the perfect competition is out of question there.

The revelation principle was introduced by Gibbard (1973)[8] and extended to the broader solution concept of Bayesian equilibrium by Dasgupta, Hammond and Maskin (1979)[6], Holmstrom (1977)[10], and Myerson (1979)[15]. It tells us that any equilibrium of rational communication strategies for the agents can be simulated by an equivalent incentive-compatible direct-revelation mechanism, where a trustworthy mediator maximally centralizes communication and makes honesty and obedience rational equilibrium strategies for the agents. Accordingly, we will achieve our goal by a dynamic direct mechanism.

Ex post Nash equilibrium was discussed as "uniform incentive compatibility" by Holmstrom and Myerson (1983) and it is increasingly studied in game theory (see Kalai (2002)[12]) and is often used in mechanism design as a more robust solution concept (Cremer and McLean (1985)[4], Dasgupta and Maskin (2000)[5], Perry and Reny (2002)[17]). Our (main) solution concept will be the a kind of ex post Nash equilibrium, which is ex post with respect to the private information of other players, and ex ante with respect to the future chance events. In order to prove this equilibrium, we will use the technique introduced by Myerson (1986)[14].

1.2 Paper organisation

Section 1.3 shows a motivating example and the execution in the first price mechanism with truthful and obedient agents. Section 2 describes the formal model, which is the most difficult part of the paper. We **strongly suggest** reading Sections 2.2, 2.3, 2.4 with patient attention. If the Reader manages to clearly understand the model then the mechanisms in Section 3 and the proofs about the second price mechanism in Section 4 should be easy to understand. Section 5 is about the first price mechanism, and its interpretation will require some thought from the Reader. Section 6 compares the two mechanisms, and after the Conclusion (Section 7), there are some notable thoughts about the results.

1.3 Example for the process

We consider a project that is very risky, but may gain huge utility. It consists of two tasks, the principal should choose one application per task, and if both succeed in time then the principal gets a large sum of money, that is 60 here, but if either fails to do it then the success of the other task has no use.

The trees in Figure 1/a and 1/b are descriptions of stochastic decision trees in the following sense. The possible executions of a task are the paths from the root to a leaf, where the timing of the events is represented by their heights. The solid squares denote decision points, at which the agent can choose the branch to continue. The other internal nodes denote such chance points at which the branch to continue is chosen randomly with probability 1/2 for each branch. At each leaf, there is a tick denoting the success or a cross denoting the failure, and the number after this sign shows his total cost. (The numbers on the edges are only to show how these asked payments are calculated.)

Let us now consider two agents applying for the two different tasks. Assume that both are essentially truthful and obedient. In this case, each of them applies by reporting his own

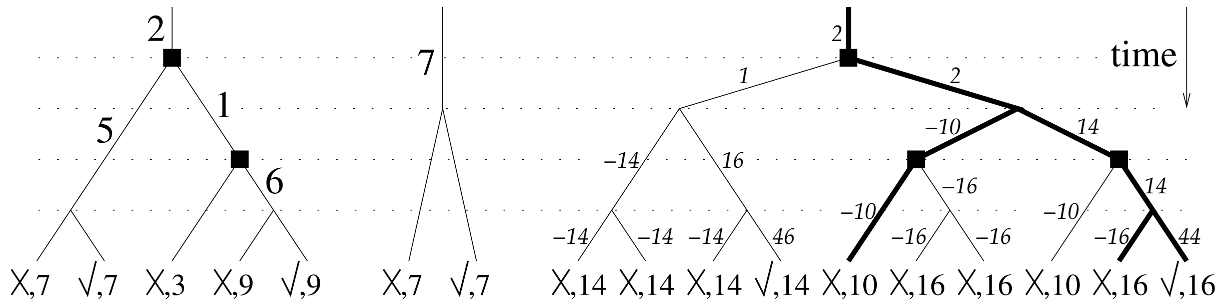

Figure 1/a. Figure 1/b. Figure 1/c.

decision tree but with a constant extra cost. Their applications are shown in Figure 1/a and 1/b. (So their decision trees are the same but with a constant less costs of the leaves.) Specifically, the first agent asks 2 units of money beyond his expenses. He either begins the task at the beginning with a cost of 5 and probability of the success of 1/2, or he makes preparations for a cost of 1, and at a later point in time, if the principal asks for it then he can try to complete the task for a cost of 6 with probability of the success of 1/2. In the other application, the cost plus the desired expected payoff of the other task is 7 and the probability of the success is 1/2.

The principal, after receiving all applications, considers each pair for the two different tasks, evaluates that how much payoff could she have with them, and chooses the pair producing her the most payoff. The evaluation of the applications shown by the first two trees is described in the third tree (Figure 1/c). This tree is a "product" of the two trees; namely, assuming that these two trees are the same as the two decision trees, the product describes the possible overall executions of both tasks. We can construct it by following the applications by time, and creating an appropriate branching point if it occurs in one of them, and then continue this on both branches. For example, the path to the 5th leaf from the left in the third tree describes the pair of the paths to the middle leaf of the first tree and the left leaf of the second tree. In details, the first agent chooses working by the second way, then the second agent fails to complete the task and then the first agent does not try to complete his task. At each leaf, there is a tick denoting the success of both tasks or a cross denoting the failure of either, and the number shows the basic total payment asked by the agents.

State means the state of the project at a point in time; it can be represented by a point in the graph of the third tree. We define the values of all states from the bottom to the top, indicated by italics in Figure 1. Let the value of an endstate be 60 if both tasks succeed and 0 otherwise, minus the total cost. Values of states in the same edge are the same. The value before a decision point is the maximum of the values after. The value before each chance point is the average of the values after.

After the values are calculated, the project would be executed by the following way. At each decision point, the principal asks the corresponding agent to choose the branch with the highest value. At each chance point, the corresponding agent tells the chance event, and the principal pays him the signed difference between the values of the states after and before. At the end, the agents have to deliver the results and get the costs at the leaf.

For the first agent, this means that the principal asks him – a bit surprisingly – to work in the second way, that is to make only preparations, and then the principal either asks him to do nothing and he gets 3, or she asks him to try to complete the task, and he gets 9 and ± 30 for the risk, so he gets 39 if he succeeds, and he pays 21 if he fails. For the second agent this means that he gets 7 and ± 12 for the risk, so if he succeeds then he gets 19, but if he fails then he pays 5. (Note that these extremely high risks are more the peculiarity of this simple and risky example than of this process.)

This way, the payoff of the principal is surely equal to the value of the starting state because at the end, she gets the value of the endstate, and whenever the value of the current state changes, she pays this difference.

Of course, we will show that the players are interested in similar truthful and obedient strategies, and in this case the process works efficiently.

In the formal model, the costs will not belong to the tree, and for the main results, we will use not the constant extra costs but a kind of second price compensations.

2 The basic model

2.1 General notions

For any symbol x, the definition of x_i will also define x as the vector of all meaningful x_i, and x_S as the vector of x_i for all $i \in S$. If x is a vector of elements and R is a vector of sets then $x \in R$ means $\forall i : x_i \in R_i$. x_{-j} means the vector of all x_i except x_j, and (x_{-j}, y) means the vector x by exchanging x_j to y.

The concept of **actions** of a player refers to what he effectively does, and when he does them. The concept of **information** of a player refers to what he knows and believes at a particular point in time. (And he always knows the current time.) The **strategy set** $\boldsymbol{Str_y}$ of a player y consists of all functions that assign a feasible action to each possible information.[1] Therefore, a **game** can be defined by the set of the players, their action sets, information histories and payoffs. A **subgame** means the game from a point in time, given the earlier actions and random events. Formally, **state** is a synonym of subgame, but grammatically, state will refer mainly to the starting point in time; for example, we say that a state T_1 is *later* than another state T_2 if T_1 is a subgame of T_2. The dependence of any function or relation f on the game G is denoted by the form f^G, but if G is the default game then we may omit this argument.

For the sake of lucidity and gender neutrality, we use feminine or masculine pronouns depending on the gender (♀ or ♂) assigned to the player.

2.2 Interpretation

There is a principal(♀) who wants to get a particular project completed by some agents(♂). The principal can be considered as the designer of the mechanism. We consider her as a player but with a fixed and publicly known strategy, so we do not handle with her in the equilibrium concept.

At the beginning, she can negotiate with the agents about the payment system, namely, how the payments depend on their achievements as well as the information they send and receive during the work. Then the agreed agents can start to work.

Each agent can affect his own work, for example by using more or less workers; or being more hurry for some more fatigue, which is equivalent to some more cost. Furthermore, each agent gets feedbacks from his work, such as about unexpected failures, or simply faster or slower progresses. The dynamics of them is described by his decision tree, and each agent knows his own tree.

The players cannot completely observe the executions of the other tasks; for example, it can happen that no one else can be sure that the completion time of a task of an agent was achieved by a fast and expensive work but with worse luck or by a slow and cheap work with better luck. However, some other aspects which directly affect the utility of the overall outcome, such

[1] In fact, the strategy set is such a subset of all such functions by which each strategy profile with the non-deterministic and stochastic variables uniquely determine the execution of the game. This is only an analytical requirement, which we do not resolve here but we simply assume that there exists such a strategy set containing the later mentioned strategies.

as the completion time, are required to be contractible. This requirement is handled by the concept of **result**.

The result can be interpreted as the contractible aspects of the execution of one task, and the requirement is that these aspects of all players must determine the utility of the principal.

To put it the other way around, let us call some possible executions of a particular task *equivalent* if the utility is always the same with all executions, no matter what the executions of the others are. Then the results are the equivalence classes of the executions, and we require them to be contractible.

For the sake of generality, we allow the principal to also have a decision tree and we allow the agents to benefit from different results. Thus, the type of each player consists of a decision tree and a utility function.

For the example (Section 1.3), these concepts are the following.

	decision tree	result	utility function
principal	trivial (tree with no branch)	trivial (constant)	if both tasks succeed then 60, otherwise 0
agent 1	1st tree in Figure 1	success or failure of task 1	−(cost of his achieved leaf)
agent 2	2nd tree in Figure 1	success or failure of task 2	−(cost of his achieved leaf)

For each player, we assume that he/she has some basic information, such as his own workflow, but we do not assume anything about how much he/she knows about the other things. By the idea of Myerson[14], we handle this by the following way. We define the model as essentially of perfect information, but we are looking for such a Nash equilibrium in which every player uses only his basic information. Because such equilibrium implies that no matter how much information the players really have beyond their basic information, this strategy profile is surely Nash equilibrium.

2.3 Model

Our model is the following game, denoted by G.

Players. There are a player $\boldsymbol{C}$ called the **principal**(♀) and some players $\boldsymbol{1, 2, ...n}$ called **agents**(♂). Let $\boldsymbol{Ag} = \{1, 2, ...n\}$ and $\boldsymbol{Pl} = Ag \cup \{C\}$. Usually, i and j will refer to an agent.

Each player y has a **decision tree** $\boldsymbol{d_y}$, which is a rooted branching tree structure consisting of the following terms. The internal nodes of the tree are of two kinds: the **chance points** and the **decision points**; and the third kind of nodes are the leaves. To each chance point there is assigned a probability distribution on its children. There is a positive real point in time assigned to every internal node, which is later than the time of its parent (if exists). Each leaf l has a **result** $\boldsymbol{r(l)}$ assigned. For the sake of simplicity, we assume that any two nodes of two different trees have different points in time. We denote the set of leaves in d_y by $\boldsymbol{L_y}$.

Executing d_y means the following process.

- We start from the root.
- If the current node is a decision point then, at its point in time, the agent moves to an arbitrary child of it.
- If the current node is a chance point then, at its point in time, we move to a random child chosen with the assigned probabilities. We call these choices as chance events.
- If the current node is a leaf then the process ends. We denote this leaf by $\boldsymbol{l_y}$ and the result of this leaf by $\boldsymbol{r_y}$, and r denotes the (multi)set of all r_y.

The action sets of the players are the following.

- Each agent can send time-stamped instant messages to the principal at any time, and vica versa, but only a finite number in the aggregate.
- At time -1, each agent decides whether he takes part in the game.
- At time 0, the principal chooses a subset Acc of the participating agents. We call them accepted agents.
- After time 0, each chosen agent as well as the principal executes his/her decision tree.
- At the end, the principal determines a transfer vector $t : Acc \to \mathbb{R} \cup \{-\infty\}$.

The information of a player at a point in time consists of the following.

- the decision trees of all players,
- the **utility functions** of all players $\boldsymbol{u_y} : L_y \times \{\text{sets of results}\} \to \mathbb{R}$,
- the earlier actions of all players,
- the earlier chance events of all players,
- his/her own chance event at the current time (if exists),
- at the end, the already achieved results.

The payoff $\boldsymbol{p}$ of the agents are defined as follows.

$$p(i) = \begin{cases} u_i(l_i, r) + t_i & \text{if } i \in Acc \\ 0 & \text{if } i \in Ag - Acc, \end{cases} \tag{1}$$

and the payoff of the principal is

$$p(C) = u_C(l_C, r_{Acc}) - \sum_{i \in Acc} t_i \tag{2}$$

We call the pair $\boldsymbol{\theta_y} = (d_y, u_y)$ the **type** of y, and we denote the game given the types of the players by $G(\theta)$. Most expressions will be functions of the types as nondeterministic parameters, but mostly we will not denote this dependence. We denote the subgame after the choice of Acc by $\boldsymbol{G_0(\theta_{Acc}) = G_{Acc}}$.

2.4 Goal

Denote the **strategy** of a player y by $\boldsymbol{s_y}$. For each player y, $p(y)$ is a function of the strategies of all players, and of the chance events. $p(y)$ restricted to a domain implied by a condition con is denoted by $p(y)[con]$, for example, $p(i)[i \in Acc] = u_i(l_i, r) + t_i$. Instead of $p(y)[s = str]$, we will simply use $p(y)[str]$. We will denote this dependence only in cases of necessity.

The **preference** of an agent i, denoted by $\boldsymbol{\preceq_i}$, is a partial ordering on the strategy profiles of all players. Let $\mathbb{E}$ denote the expected value, which means (by default) the marginalisation over the chance events. Keep in mind that $\mathbb{E}(p(i))$ is still a function of the (types and) strategies of the players. The default preference of each agent is to achieve higher expected payoff. Namely, $s \succeq_i s'$ iff $\mathbb{E}(p(i)[s]) \geq \mathbb{E}(p(i)[s'])$ (where both sides are functions of the types).

The **basic information** of each player means the part of his/her information containing, at a point in time, the following.

- His/her own type,
- all messages he/she received strictly earlier,
- for the principal, the agents' decisions about whether they stay in the game, provided that time > -1
- for an agent, whether the principal allowed him to work, provided that time > 0
- his/her chance events up to and including the current time,
- his/her earlier actions,
- for the principal, the set of results r.

Simple strategy of a player means such a strategy that assigns the action to his/her basic information.

We call a strategy profile s of the players consisting of simple strategies an **information-invariant Nash equilibrium**, if each $i \in Ag$ and $s'_i \in Str_i$ satisfies $s \succeq_i (s_{-i}, s'_i)$. With default preferences, this means $\mathbb{E}(p(i)[s]) \geq \mathbb{E}(p(i)[(s_{-i}, s'_i)])$.

Let $\boldsymbol{P} = Acc \cup \{C\}$ called **participants**. We call the sum of the payoffs of all players the **payoff of the system**, and we denote it by $\boldsymbol{p(Pl)}$. Clearly,

$$p(Pl) \overset{(1)+(2)}{=} \sum_{y \in P} u_y(l_y, r). \tag{3}$$

Let the **value of a game** H be

$$\boldsymbol{v(H)} = \max_{s \in Str^H} \mathbb{E}(p^H(Pl)[s]). \tag{4}$$

Clearly,

$$\mathbb{E}(p(Pl)) \leq v(G). \tag{5}$$

Our main goal is to find a strategy profile s such that,

- $\mathbb{E}(p(Pl)[s]) = v(G)$
- s is an information-invariant Nash equilibrium.

2.5 Interpretation of dependent action sets

In many cases like in scheduling problems, some actions of agents may be restricted by some actions of other agents. For example, one cannot start building the roof of a house before someone completes the walls. But the model deals only with independent working processes. To resolve this problem, we ignore all restrictions, hereby extending the action sets of the restricted agents, but we define $u_C = -\infty$ at the extended overall executions. The point of this is that if a strategy profile satisfying the restrictions is an equilibrium in the extended model then it must be an equilibrium in the original model, as well. But to do that, as u_C is defined not on overall executions but on sets of results, the restricting and the restricted events are always required to be contractible; that is, whether something "impossible" happened must be a function of the results. This means for the example that each result of the agent building the walls must contain the completion time m_c, each result of the agent building the roof must contain the starting time m_s, and if $m_s < m_c$ then $u_C = -\infty$.

To sum up, this model can be applied also in such cases when one's possible decisions can be restricted by some contractible events in other tasks.

3 Mechanisms

Mechanism means a simple strategy of the principal. Her actions consist of

- her choice of Acc;
- her communication;
- the execution of her tree,
- her choice of t.

3.1 First price mechanism

The principal assumes the agents to use the communication protocol below. (For example, she defines such an equivalent protocol that cannot be violated.)

The principal requires the agents to send her a type (which are not necessarily their own types) at time -1. We call this message of i as his **application** $\boldsymbol{\theta_i^*} = (d_i^*, u_i^*)$. Let us use the notion $\theta_C^* = \theta_C^*$, $d_C^* = d_C$, $u_C^* = u_C$. Then the principal considers the strategy profile $\boldsymbol{s^*}$ of $G(\theta^*)$ which maximizes $\mathbb{E}(p^{G(\theta^*)}(Pl))$, and

- she accepts the agents who would be chosen agents by s^*;
- she makes the decisions in her own tree corresponding to s_i^* and considers her own chance event in $G(\theta^*)$ to be the same as her chance event in G.
- whenever an agent i reaches a chance point in $G(\theta^*)$, she requires him to send her a chance event, and she considers the event in this message to be the chance event in $G(\theta^*)$;
- whenever an agent i is about to reach a decision point in $G(\theta^*)$, she sends him the decision corresponding to s_i^*, and she considers this to be his decision in $G(\theta^*)$;
- At the end, the principal achieves an endstate in $G(\theta^*)$, which corresponds to one leaf $\boldsymbol{l_y^*}$ and result $\boldsymbol{r_y^*}$ per participant y. If for an agent i, $r_i \neq r^*(l_i^*)$ then $t_i = -\infty$; otherwise, denote the set of the chance events during the execution of d_i^* in $G(\theta^*)$ by $\boldsymbol{Ch_i}$, and for each $ch \in Ch_i$, denote the states just before and after ch by T_-^{ch} and T_+^{ch}. Let

  $$\boldsymbol{\delta(ch)} = v(T_+^{ch}) - v(T_-^{ch}). \tag{6}$$

  Then the principal chooses

  $$t_i = -u_i^*(l_i^*, r) + \sum_{ch \in Ch_i} \delta(ch). \tag{7}$$

3.2 Second price mechanism

We define the **value of a set** S **of applications** by $\boldsymbol{v(\theta')} = v(G(\theta'))$. The **value of an application** θ_i^* means

$$\boldsymbol{v_i^+ = v_i^+(\theta^*) = v^+(\theta_i^*)} = v(\theta^*) - v(\theta_{-i}^*). \tag{8}$$

(Formally, the notion $v^+(\theta_i^*)$ is a bit confusing, but it will be useful.)

The second price mechanism is the same as the first price mechanism but the principal pays v_i^+ more to each agent i, namely,

$$t_i = v_i^+(\theta^*) - u_i^*(l_i^*, r) + \sum_{ch \in Ch_i} \delta(ch).$$

3.3 Further definitions and notions

From now on, the mechanism will be fixed, the default mechanism will be the first price mechanism, and we will no longer denote the dependence of the strategy of the principal. In details, s will refer to s_{Ag} and p will always be restricted to the first price mechanism. We denote the payoffs restricted to the second price mechanism by $\boldsymbol{p_2}$. Furthermore, we say that a mechanism s_C information-invariantly Nash implements a goal if there exists a strategy profile $s = s_{Ag}$ such that (s, s_C) is an information-invariant Nash equilibrium and the goal will be achieved using this strategy profile.

From now on, we assume that all agents use such strategy by which $r^*(l_i^*) \neq r_i$ cannot happen. (It is only a technical assumption, see Section 9.1.)

We call the state of $G(\theta^*)$ induced by the messages the principal sent and received until a point in time as the **presumed** state. We will use this word in analogous cases, too.

The mechanism can also be defined in subgames of G by making the further actions corresponding to the optimal strategy profile in the presumed subgame.

We define the **cost price strategy** $\boldsymbol{cp_i}$ of an agent i as follows. He takes part in the game, he sends his type as his application to the principal, and then, if the principal chooses him to work then he always makes the decision which she asks him for, and at each chance point, he sends her the chance event that occurred.

4 Efficiency and interests in the second price mechanism

Lemma 1. *Denote the state just before and all possible states just after an action of a player by T and $T_1, T_2, ...T_k$. Then*

$$v(T) = \max_i v(T_i) \tag{9}$$

Proof.

$$v(T) \overset{(4)}{=} \max_{s' \in Str^T} \mathbb{E}(p^T(Pl)[s']) = \max_i \max_{s' \in Str^{T_i}} \mathbb{E}(p^{T_i}(Pl)[s']) \overset{(4)}{=} \max_i v(T_i)$$ □

Lemma 2. *Denote the state just before and all possible states just after a chance point of an agent by T and $T_1, T_2, ...T_k$. Let $w_1, w_2, ...w_k$ denote the corresponding probabilities. Then $v(T) = \sum w_i \cdot v(T_i)$, or equivalently,*

$$\sum w_i(v(T_i) - v(T)) = 0. \tag{10}$$

Proof.

$$v(T) \overset{(4)}{=} \max_{s' \in Str^T} \mathbb{E}(p^T(Pl)[s']) = \sum w_i \cdot \max_{s' \in Str^{T_i}} \mathbb{E}(p^{T_i}(Pl)[s']) \overset{(4)}{=} \sum w_i \cdot v(T_i)$$ □

Lemma 3. *If the chance points of a player y correspond to his/her chance points in $G(\theta^*)$ with the same probabilities then*

$$\mathbb{E}(\sum_{ch \in Ch_y} \delta(ch)) = 0. \tag{11}$$

Proof. Using the notions in Lemma 2, $\mathbb{E}(\delta(ch)) \overset{(6)}{=} \sum w_i(v(T_i) - v(T)) \overset{(10)}{=} 0$, so $\sum \delta(ch)$, summing on all past chance events ch of y, is a martingale, so the expected values of the sums at the end (left side) and at the beginning (right side) are the same. □

Lemma 4.

$$\mathbb{E}(p(C)) = v(\theta^*) \tag{12}$$

Proof. Denote the actual presumed state by T, the achieved presumed endstate by N and denote the sequence of the presumed chance events of y until T by $Ch_y(T)$, and let $Ch(T) = \bigcup Ch_y(T)$. Then $v(T) - \sum_{ch \in Ch(T)} \delta(ch)$ is invariant during the game because of (6) and (9), so

$$v(\theta^*) = v(G(\theta^*)) = v(G(\theta^*)) - \sum_{ch \in Ch(G(\theta^*))} \delta(ch) = v(N) - \sum_{ch \in Ch(N)} \delta(ch) \overset{(4)(3)}{=} \mathbb{E}(\sum_{y \in P} u_y^*(l_y^*, r)$$

$$-\sum_{y \in P} \sum_{ch \in Ch_y} \delta(ch)) \overset{(7)}{=} \mathbb{E}(u_C^*(l_C^*, r) - \sum_{ch \in Ch_C} \delta(ch) - \sum_{i \in Acc} t_i) \overset{(11)}{=} \mathbb{E}(u_C(l_C, r) - \sum_{i \in Acc} t_i) \overset{(2)}{=} \mathbb{E}(p(C)).$$

□

Lemma 5.

$$\mathbb{E}(p(i)[s_i = cp_i]) = 0 \tag{13}$$

Proof. If $i \notin Acc$ then $p(i) \overset{(1)}{=} 0$, and if $i \in Acc$ then

$$\mathbb{E}(p(i)) \overset{(1)}{=} \mathbb{E}(u_i(l_i, r) + t_i) \overset{(7)}{=} \mathbb{E}(u_i(l_i, r) - u_i^*(l_i^*, r) + \sum_{ch \in Ch_i} \delta(ch)) \overset{(11)}{=} \mathbb{E}(u_i(l_i, r) - u_i^*(l_i^*, r)) = 0 \quad \square$$

Because of (5), the following theorem shows the efficiency of the cost price strategy profile, in accord with the revelation principle.

Theorem 6.

$$\mathbb{E}(p(Pl)[cp]) = v(G). \tag{14}$$

Proof 1. If $s = cp$ then $\theta^* = \theta$ and

$$\mathbb{E}(p(Pl)) = \mathbb{E}(p(C)) + \sum_{i \in P} \mathbb{E}(p(i)) \overset{(12)(13)}{=} v(G(\theta^*)) = v(G(\theta)) = v(G). \qquad \square$$

Proof 2. The presumed execution maximizes the presumed $\mathbb{E}(p(Pl))$ in the presumed game $G(\theta^*)$. If $s = cp$ then $G(\theta) = G(\theta^*)$ and the actual and the presumed states will always be the same throughout the game. Consequently,

$$\mathbb{E}(p^G(Pl)[cp]) = \mathbb{E}(p^{G(\theta^*)}(Pl)[s^*]) = \max_{s' \in Str^{G(\theta^*)}} \mathbb{E}(p^{G(\theta^*)}(Pl)[s']) \overset{(4)}{=} v(G(\theta^*)) = v(G). \qquad \square$$

Theorem 7. *In the second price mechanism, the cost price strategy profile is an information-invariant Nash equilibrium.*

Proof. The outline of the proof is as follows. We consider an arbitrary agent i and assume that all other agents use cost price strategy. Then we will prove that $\mathbb{E}(p_2(i)) \leq v(G) - v(\theta_{-i})$, and equality holds if i also uses cost price strategy. As the right side is independent of the strategy of i, this will prove the inequality.

Consider a mixed mechanism which is second price for i and first price for the other agents, namely

- $t_i = v_i^+ - u_i^*(l_i^*, r) + \sum_{ch \in Ch_i} \delta(ch)$,
- $t_j = -u_j^*(l_j^*, r) + \sum_{ch \in Ch_j} \delta(ch)$, if $j \neq i$.

Clearly, the payoff and the preference of i and of the system are the same here as in the second price mechanism. Let p_* denote the payoffs here. Whether i is accepted or not,

$$\mathbb{E}(p_*(C)) = \mathbb{E}(p(C)) - v_i^+ \overset{(12)(8)}{=} v(\theta^*) - (v(\theta^*) - v(\theta^*_{-i})) = v(\theta^*_{-i}) = v(\theta_{-i}) \tag{15}$$

Using that for all $j \neq i$, $\mathbb{E}(p_*(j)) = \mathbb{E}(p(j)) \overset{(13)}{=} 0$,

$$\mathbb{E}(p_2(i)) = \mathbb{E}(p_*(i)) = \mathbb{E}(p(Pl)) - \mathbb{E}(p_*(C)) - \sum_{j \in Ag-\{i\}} \mathbb{E}(p(j)) \overset{(5)(15)(13)}{\leq} v(G) - v(\theta_{-i}),$$

and (14) shows that equality holds if i also uses cost price strategy. □

Summarizing Theorems 6 and 7,

Corollary 8. *The second price mechanism information-invariantly Nash implements $v(G) = \mathbb{E}(p(Pl))$.*

5 Efficiency and interests in the first price mechanism

For an application $\theta' = (d', u')$, let $\theta' + x = (d', u' + x)$, and for a strategy s' with application θ', let $s' + x$ mean the same strategy as s' but with application $\theta + x$. We call a strategy $cp_i + x_i$ and an application $d_i + x$ as **fair strategy** and **fair application** with **profit** x_i. Notice that if $s_i = cp_i + x_i$ then

$$u_i^*(l_i^*, r) = u_i(l_i, r) + x_i \tag{16}$$

A cost price strategy but with applying with an arbitrary u_i^* is called a **fair-like strategy**. Clearly, the cost price strategy is fair, and all fair strategies are fair-like strategies, as well.

Lemma 9.

$$\mathbb{E}(p(i)[i \in Acc; s_i = cp_i - x]) = x \tag{17}$$

Proof.

$$\mathbb{E}(p(i)) \overset{(1)}{=} \mathbb{E}(u_i(l_i, r) + t_i) \overset{(7)}{=} \mathbb{E}(u_i(l_i, r) - u_i^*(l_i^*, r) + \sum_{ch \in Ch_i} \delta(ch)) \overset{(11)}{=} \mathbb{E}(u_i(l_i, r) - u_i^*(l_i^*, r)) \overset{(16)}{=} x \quad \square$$

Theorem 10. *In G_{Acc}, if $\forall i \in Acc : s_i = cp_i - x_i$ then*

$$\mathbb{E}(p(Pl)) = v(G_{Acc}), \tag{18}$$

Proof 1. Notice that Lemma 4 works in G_{Acc} instead of G, too, namely, $p^{G_{Acc}}(C) = v(G_0(\theta^*_{Acc}))$. Notice also that the only difference between $G_{Acc} = G_0(\theta_{Acc})$ and $G_0(\theta^*_{Acc})$ is that each agent i gets x_i less payoff in the letter case. Hence,

$$p^{G_{Acc}}(C) = v(G_0(\theta^*_{Acc})) = v(G_{Acc}) - \sum_{i \in Acc} x_i, \tag{19}$$

thus in G_{Acc},

$$\mathbb{E}(p(Pl)) = \mathbb{E}(p(C)) + \sum_{i \in Acc} \mathbb{E}(p(i)) \overset{(19)(17)}{=} (v(G_{Acc}) - \sum_{i \in Acc} x_i) + \sum_{i \in Acc} x_i = v(G_{Acc}). \quad \square$$

Proof 2. The presumed execution maximizes the presumed $\mathbb{E}(p(Pl))$ in the presumed subgame $G_0(\theta^*_{Acc})$. If $\forall i \in Acc : s_i = cp_i - x_i$ then on one hand, the only difference between G_{Acc} and $G_0(\theta^*_{Acc})$ is that each agent i gets x_i less payoff in the letter case; and on the other hand, this remains the only difference between the actual and the presumed states throughout the game. Consequently,

$$\mathbb{E}(p^{G_{Acc}}(Pl)[(cp_i - x_i)_{i\in Acc}]) = \mathbb{E}(p^{G_0(\theta^*_{Acc})}(Pl)[s^*]) + \sum_{i\in Acc} x_i =$$

$$\max_{s'\in Str^{G_0(\theta^*_{Acc})}} \mathbb{E}(p^{G_0(\theta^*_{Acc})}(Pl)[s']) + \sum_{i\in Acc} x_i = \max_{s'\in Str^{G_{Acc}}} \mathbb{E}(p^{G_{Acc}}(Pl)[s']) \stackrel{(4)}{=} v(G_{Acc}). \qquad \square$$

Let the **signed value of an application** θ^*_i be

$$\boldsymbol{v_i^{\pm} = v_i^{\pm}(\theta_i^*) = v^{\pm}(\theta_i^*)} = \max_{i\in S\subset Ag} v(G_0(\theta^*_S)) - v(\theta^*_{-i}) \tag{20}$$

Lemma 1 shows that the principal chooses the set $Acc \subset Ag$ for which $v(G_0(\theta^*_{Acc}))$ is maximal. Hence,

$$v(\theta^*) = v(G(\theta^*)) \stackrel{(1)}{=} \max_{S\subset Ag} v(G_0(\theta^*_S)) = \max(\max_{S\subset Ag-\{i\}} v(G_0(\theta^*_S)), \max_{i\in S\subset Ag} v(G_0(\theta^*_S)))$$

$$\stackrel{(1)}{=} \max(v(\theta^*_{-i}), \max_{i\in S\subset Ag} v(G_0(\theta^*_S)))$$

and the second term is the greater one iff $i \in Acc$. Hence,

$$v_i^+ \stackrel{(8)}{=} v(\theta^*) - v(\theta^*_{-i}) = \max(v(\theta^*_{-i}), \max_{i\in S\subset Ag} v(G_0(\theta^*_S))) - v(\theta^*_{-i})$$

$$\stackrel{(20)}{=} \max(0, v_i^{\pm}) \tag{21}$$

and $v_i^{\pm} > 0$ iff $i \in Acc$.

For any $S \ni i$, $v(G_0((\theta^*_{S-\{i\}}, \theta^*_i - x))) = v(G_0(\theta^*_S)) - x$, so

$$v^{\pm}(\theta_i - x) = v^{\pm}(\theta_i) - x.$$

Thus, for a particular agent, there is exactly one fair application with a given signed value.

Theorem 11. *For an arbitrary agent i and value x, if every other agent use fair-like strategies then the fair strategy of i with signed value x gains him the highest expected payoff among all those strategies that use an application of this signed value.*

Proof. Given the application of an arbitrary agent j with fair-like strategy, what d_j is makes no difference on the payoffs of the other players, nor on their basic information. Thus, given the strategies of the all players but i, the payoff of i is independent of d_{-i}. Thus, we only need to and will prove the statement in the game $G' = G((\theta^*_{-i}, \theta_i))$ with $s_{-i} = cp_{-i}$.

If $x < 0$ then i will be rejected, so $p(i) = 0$. Otherwise,

$$\mathbb{E}(p(i)) = \mathbb{E}(p(Pl)) - \mathbb{E}(p(C)) - \sum_{j\in Ag-\{i\}} \mathbb{E}(p(j)) \stackrel{(5)(12)(13)}{\leq} v(G') - v(\theta^*) \stackrel{(8)(21)}{=} v(G') - v(\theta^*_{-i}) - x$$

and Theorem 6 shows that equality holds if $s_i = cp_i$. $\square$

First we describe our results more intuitively, then we present the formal results in the subsections.

Let the *strategy form* of a strategy s' mean $Form(s') = \{s' + x | x \in \mathbb{R}\}$. Let $Form(cp_i) = \{cp_i + x | x \in \mathbb{R}\}$ called the fair strategy form. Let $p[s'] = \mathbb{E}(p(j)[s_j = s'])$.

With a strategy s'_i with application θ'_i, if both applications θ'_i and $\theta'_i - x$ were accepted then $p[s'_i - x] = p[s'_i] + x$ (provided that the strategies of the other agents are independent of i's choice of x) and $v(\theta'_i - x) = v(\theta'_i) - x$, so $p[s'_i] + v(\theta'_i)$ depends only on $Form(s'_i)$. We call this sum as the *value of the strategy form* $v(Form(s'_i))$.

In each strategy form there exists a single strategy s'_i, using application θ'_i, for which $\theta'_i + x$ would be accepted if $x > 0$ and rejected if $x < 0$. Then $p[s'_i + x] = \{0 \text{ if } x < 0; \text{ and } v(Form(s'_i)) - x \text{ if } x > 0\}$.

$p[s_i] \le v(Form(s_i))$, so the strategy form with the highest value gains him the highest potential for his expected payoff. Another question is that expectedly how much he could exploit this potential, in the Bayesian meaning.

Theorem 11 implies that the fair strategy form has the highest value. Moreover, $p[cp_i - x] = \{x \text{ if } v(cp_i) > x; \text{ and } 0 \text{ otherwise}\}$, which is a quite efficient way for the exploitation of this potential. Of course, this only shows that the fair strategy with appropriate profit is *about* the best for an agent.

5.1 Interest in cost price strategy with perfect competition

The following definition of the perfect competition will roughly describe the limit of the cases when the maximum expected payoff that an agent is able to achieve tends to 0. Keep in mind that the principal accepts the applications with the positive (or nonnegative) signed values, and notice that the weaker (the partial ordering of) the preference the stronger the equilibrium.

We define the **perfect competition** in G by the following preferences of the agents. Each agent i prefers $\mathbb{E}(p(i))$ to be nonnegative, and of those, he prefers the greater $v_i^{\pm}$. Formally,

$$(s \preceq_i s') \Leftrightarrow \forall \theta : ((\mathbb{E}(p(i)[\theta, s]) < 0) \text{ or } (v^{\pm}(\theta_i^*)[\theta, s] < v^{\pm}(\theta_i^*)[\theta, s']))$$

Theorem 12. *In first price mechanism with perfect competition, cost price strategy of all agents is an information-invariant Nash equilibrium.*

Proof. Assume that every agent except i uses cost price strategy. What we have to prove is that i is interested in using cost price strategy, as well. If $p(i) \ge 0$ then

$$v_i^{\pm} \overset{(21)}{=} v_i^{+} \overset{(8)}{=} v(\theta^*) - v(\theta_{-i}^*) \overset{(12)}{=} p(C) - v(\theta_{-i}^*) =$$

$$\mathbb{E}(p(Pl)) - \sum_{j \in Ag} p(j) - v(\theta_{-i}^*) \overset{(13)}{=} \mathbb{E}(p(Pl)) - p(i) - v(\theta_{-i}^*) \overset{(5)}{\le} v(G) - v(\theta_{-i}^*),$$

and (14) shows that equality holds if i also uses cost price strategy. So this inequality gives an upper bound for his preference, which can always be achieved by cost price strategy. ☐

Summarizing Theorems 6 and 12,

Corollary 13. *The first price mechanism information-invariantly Nash implements* $\mathbb{E}(p(Pl)) = v(G)$ *with perfect competition.*

5.2 First price mechanism with imperfect competition

In this subsection, we consider θ not nondeterministic but stochastic variable, and we assume the information of each agent to be between his basic information and his information in the original formal model.[2] Let $\mathbb{P}_i$ and $\mathbb{E}_i$ denote the probability and expectation (with respect not only to the chance events) given the information of i.

Let us fix an agent i and s_{-i}. Consider a strategy s' and denote by θ'_i (as a stochastic variable) the application he would send using s'_i. Let (value) $\boldsymbol{V} = v^{\pm}(\theta_i)$, (difference) $\boldsymbol{D} = V - v^{\pm}(\theta'_i) = v(\theta_i) - v(\theta'_i)$ and $\boldsymbol{e} = \mathbb{E}_i(D|D < V)$.

In practice, V and D are almost independent and both have "natural" distributions, so $\mathbb{P}_i(e < V) = \mathbb{P}(\mathbb{E}_i(D|D < V) < V)$ is usually not smaller than $\mathbb{P}_i(D < V)$. This observation shows the importance of the following theorem.

Theorem 14. *If the other agents use fair strategy and $\mathbb{P}_i(\mathbb{E}_i(D|D < V) < V) \geq \mathbb{P}_i(D < V)$ holds then i prefers using a fair strategy, as well.*

Proof. Let $g(\theta'_i) = \mathbb{E}_i(p(i)[\theta'_i])$ and let x be the number by which $g(cp_i + x)$ is the largest possible. So we want to prove that $g(cp_i + x) \geq g(\theta'_i)$ for all s'_i.

Theoretically, let us allow for i submitting the fair application with the signed value $v^{\pm}(\theta'_i)$, that is, submitting $\theta_i + v^{\pm}(\theta_i) - v^{\pm}(\theta'_i)$, for an arbitrary θ'_i chosen by i. Normally, it is an invalid action, because i does not know the values of his possible applications at this time, but we allow it him here, and we denote this application by $fair(\theta'_i)$, and we denote the fair strategy but with application $fair(\theta'_i)$ by $\boldsymbol{fs(\theta'_i)}$.

θ^*_i is accepted iff $v^{\pm}(\theta*_i) > 0$, or equivalently, $D < V$. By the equation

$$g(s'_i) = \mathbb{P}_i(i \in Acc) \cdot \mathbb{E}_i(p(i)|s'_i, i \in Acc),$$

we get $g(fs_e) = \mathbb{P}_i(e < V)e$, and $g(fs(apl_i)) = \mathbb{P}_i(D < V)e$, whence we can simply get that

$$g(fs_x) - g(s_i) = (\mathbb{P}_i(e < V) - \mathbb{P}_i(D < V))e + (g(fs(apl_i)) - g(s_i)) + (g(fs_x) - g(fs_e)).$$

- If $e \leq 0$ then $g(s_i) \leq 0 = g(fs_0) \leq g(fs_x)$ so s_i cannot be a better strategy. Thus we assume that $e > 0$. In this case, $(\mathbb{P}_i(e < V) - \mathbb{P}_i(D < V))e \geq 0$. If s_i is fair then D is constant, so both sides are the same.
- Proposition 10 implies that $g(fs(apl_i)) - g(s'_i) \geq 0$.
- $g(fs_x) - g(fs_e) \geq 0$ by the definition of x.

To sum up, $g(fs_x) - g(s_i) \geq 0$, which proves the equilibrium. □

6 Advantages of the first price mechanism over the second price mechanism

6.1 Coalitions

In this section, we consider the case when we have disjoint sets of agents, called coalitions, and each agent in a coalition prefers the higher total expected payoff of all in his coalition.

[2] A bit more precisely, the players get information signals during the game, and there is a previously fixed common distribution of these signals depending on the types, the earlier actions and chance events, and this distribution satisfy some conditions which we do not detail here.

First price mechanism. Submitting more applications by the same agent is equivalent to a joint application, consisting of a tree starting with a decision point and continuing with all "products" (see Figure 1) of different subsets of the original applications, and (essentially) the sum of the utility functions. Thus, if a coalition played as one agent, called consortium, with their combined decision tree, total utility function and their overall information, then this consortium would be able to simulate the case when they played as different agents in a coalition. So each coalition would not come off badly by forming a consortium, by which we get the original game with this new set of agents.

As a consequence, if the competition remains perfect with this new set of players then the mechanism remains efficient.

Second price mechanism. Consider the case when two agents can submit such applications that are useless without each other. Assume that their cost price applications would be accepted. If either of them decreased the cost of his application by x then the surplus value of both applications would increase by x, so totally they get $2x$ compensation, which means x more total payoff for them. Using this trick, these players can get as much payoff as they want.

6.2 Reliance in the principal

The mechanism can be modified as follows. The players must use the communication protocol defined at the mechanism, but right before each chance point of each agent, the principal sends $\delta(ch)$ for each possible chance event ch (with $\sum w(ch)\delta(ch) = 0$).

First price mechanism. This makes t_i to be a function of r_i and the communication between i and C, and Lemmata 9 and 5 remain true independently of the strategy of the principal.

Theorem 15. *In the modified model, if all agents use fair-type strategy then the principal is interested in choosing her first price mechanism.*

Proof. With the corresponding modifications on the decision trees, she can consider the agents to use cost price strategies. Then

$$\mathbb{E}(p(C)) = \mathbb{E}(p^{G_{Acc}}(C)) \overset{(13)}{=} \mathbb{E}(p^{G_{Acc}}(Pl)) \overset{(5)}{\le} v(G_{Acc}),$$

and Theorem 6 shows that equality holds if the principal uses her declared strategy. □

Second price mechanism. The main problem is how to verify that the principal did not determine any of the v_i^+ lower. If the principal seals a d_i^* and u_i^* right before receiving the applications, and at latest at the end, she makes them and all applications public and verifiable (except the applications that was neither best or "second best"), and sends each $\delta(ch)$ right before each chance point, then the agents can rely on her, but it does not seem to be able to make reliance without making much less information public.

7 Conclusion

In the second price mechanism, the cost price strategy profile is an information-invariant Nash equilibrium, and this way the expected payoff of the system is the largest possible, including the possibilities of making one's decisions depending on the earlier chance events of others.

In the first price mechanism, with some assumption (or approximation), there is a Nash equilibrium consisting of the same strategies but with asking a constant more money – his expected payoff in the case of acceptance – in each application.

The *disadvantage* of the first price mechanism over the second price one is that it requires a competition on each task, and it is fully efficient only with perfect competition.

The *advantages* of the first price mechanisms are

- forming cartels of agents does not worsen the efficiency as long as it does not decrease the competition, while in the second price mechanism, if two agents can submit such applications that are useless without each other then they may get as much payoff as they want;
- the agents need (almost) no reliance in the principal, even if they never get to know the applications of the others.

8 Further theoretical observations and special cases

8.1 An alternative approach of the mechanisms

As an alternative approach, we define the first price mechanism with the following differences. We define the contract as a function that determines the payment between two players depending on the results of all players and the communication between the two players, and an application is an offer for contract. At the beginning, each agent can send an application to the principal, and then she chooses which offers to accept. She uses the strategy of all her possible strategies by which her minimum possible expected payoff is the largest possible, where expectation is with respect only to her own chance events. So she is absolutely mistrustful meaning that she always expects to the worst possible joint behaviour of all agents.

The alternative second price mechanism is the same as the first price one but at the end, the principal pays v_i^+ more money to each agent i beyond the payment according to their contract.

With this approach, a fair application is the offer for using the same communication protocol as the protocol defined by the original fair application in the original mechanism, but he requires the principal to send the assignments as in Section 6.2, and the agent i asks for $-u_i^*(l_i^*, r) + \sum\limits_{ch \in Ch_i} \delta^*(ch)$, where δ^* denotes the assignment of the principal.

This approach requires slightly different proofs, but of course, our main results remain true. From now on, we will use this approach. We note that each application in the original model can be matched with an equivalent application in this model, but this is not true in the other direction.

8.2 Simplifications and the case with no parallel tasks

The messages the principal sends depend only on the earlier messages she got. Thus, if an agent i is sure that the principal receives no message from anyone else in the time interval $I = [a, b]$ then, without decreasing the value of his application, he can ask the principal (in the application) to send to him at a that what messages she would send to him during I depending on the messages she would have received from him before. Similarly, if i is sure that the principal does not send any message to anyone else during I then he can send all his messages only until b, and during I, and the principal should send to him that which messages she would send depending on the messages she would have got before.

As a consequence, consider the following project. It consists of two tasks, and the second task can only be started after the first one accomplished. The result of each agent for the first task (called first agent) consists of his completion time C_1. The result of each second agent consists of his starting time S_2 and the time C_2 he completes; and his decision tree starts with doing nothing until an optional point in time S_2, and then he can start his work. The utility function is of the form $u'(C_2)$ for some decreasing function $u' : \{\text{time}\} \to \{\mathbb{R}\}$ if $C_1 \geq S_2$, and $-\infty$ otherwise. In this case, the principal always communicates only with the agent who is just working at the time. So using the above observation, we can make simplified applications of the following form with the same values as of the fair applications.

In short, the first agents tell that for how much money they would complete the first task depending on the penalty. Then the applied penalty for the chosen second agent is the loss form the delayed completion, and the penalty for the first agent is how much more the second agent asks if he can start later. The principal chooses the pair by which she gains the most payoff.

In detail, the form of the application of the first agents is "We ask $h(C_1) - g_1(h)$ money for any $h : \{\text{time}\} \to \{\mathbb{R}\}$ chosen by the principal at the beginning", and for the second agents this is "We ask $u'(C_2) - g_2(S_2)$ money if we can start our work at S_2 and we complete it at C_2". $h(C_1)$ and $u'(C_2)$ describe the penalties here. In the simplified fair applications, g_1 and g_2 are chosen in such a way that make their expected payoff independent of the arguments, if the agents use their best strategies afterwards.

If all applications are so then the principal chooses a pair for which $g_1(g_2)$ is the greatest. Then she chooses $h = g_2$ for the first agent, and this way the principal gets $f(C_2) - (g_2(C_2) - g_1(g_2)) - (f(C_2) - g_2(S_2)) = g_1(g_2)$ payoff.

If a first agent has no choice in his decision tree, that is, his completion time C_1 is simply a probabilistic variable, then he should choose $g_1(h) = \mathbb{E}(h(C_1)) - c$, where c is his costs plus his profit.

8.3 Controlling and controlled players

For an example, consider a task of building a unit of railroad. An agent i can make this task for a cost of 100, but with 1% probability of failure, which would cause a huge loss $10,000$. Another agent j could inspect and in the case of failure, correct the work of i under the following conditions. The inspection costs 1. If the task was correct then he does nothing else. If not, he detects and correct the failure with 99% probability for a further cost 100, but he does not detect, so he does nothing with 1% probability. If both of them use cost price strategy and they are the accepted agents for the task then the mechanism works in the following way.

At the end, i gets 101.99 but pays 199 (totally he pays 97.01) compensation if he fails. j gets 1 if he correctly finds the task to be correct, he gets 200 if the task was wrong but he corrects it, but he pays 9800 if he misses correcting it.

Of course, with fair applications each of them gets his profit more payment. It can be checked that the expected payoff of each agent is his profit independently of the behaviour of the others, and the payoff of the principal is fixed.

8.4 Offer for cooperation

Consider the case when a player a simply wants to make an offer for cooperation with another player c, which c will either accept or reject. This situation is a special case of the first price mechanism, where a is the only agent and c is the principal, and the offer of a is his application.

9 Observations for application

9.1 Modifications during the process

In practice, the decision trees can be extremely difficult, thus submitting the precise fair applications is not expectable. Therefore, they can present it only in a simplified, approximating way. Generally, such inaccuracies do not significantly worsen the optimality; nevertheless, this loss can be much more reduced by the following observation.

Assume that someone, whose application has been accepted, can refine his decision tree during the process. It would be beneficial to allow him to carry out such modifications. The question is: on what conditions?

The answer is for us to allow him to modify the subtree of his application from the presumed state, if he pays the difference between the values of the presumed states with the original and the new applications. Or equivalently, the costs of the leaves of the subtree automatically decrease by this difference. It is easy to see that whether and how to modify is the same question as which application to submit among the applications with a given value. Consequently, Theorem 11 shows that exchanging to the agent's true fair application is in his interest.

Equivalently, each possible modification can be handled so as a chance point in the original application with 1 and 0 probabilities for continuing with the original and the modified application, respectively. Because at such chance point, the principal assigns 0 to the branch of not modifying, and to the modification, she assigns the difference between the values of the states after and before.

It may happen that in the beginning it is too costly for some agent to explore the many improbable branches of his decision tree, especially if he does not yet know whether his application will be accepted; but later however, it would be worth exploring better the ones that became probable. This kind of in-process modifications is what we want to make possible. We show that the interest of each agent in better scheduling of these modifications is about the same as the interest of the system in it.

The expected payoff of an agent with an accepted fair application is fixed and for a nearly fair agent, the small modifications of the other applications have negligible effect. As the modifications of each agent have no influence on the payoff of the principal and only this negligible influence on the expected payoff of other agents, the change of the expected payoff of the system is essentially the same as the change of the expected payoff of this agent. This confirms the above statement.

On the other hand, it is clear that if the utility function alters somewhat then everything can be rescheduled according to the new goals. Moreover, the principal is interested in describing her utility function in the same schedule as which is the best according to the preference of the system.

9.2 Risk-averse agents

Assume that an agent i has a strictly monotone **worth** funcion $h_i : \mathbb{R} \to \mathbb{R}$ and he prefers the higher $\mathbb{E}(h_i(p(i)))$.

We define an application **reasonable** as the same as the fair application with the only difference that the agent asks for

$$h_i^{-1}\big(h_i(-u_i^*(l_i^*, r)) + \sum_{ch \in Ch_i} d^*(ch)\big),$$

where d^* denotes the value assigned previously to the chance event by the principal. By a reasonable application, in the case of acceptance, the expected worth of the utility of the agent is independent of the choices of the principal. If all applications are reasonable then the payoff of the principal remains fixed. If the agent is risk-neutral then the reasonable application is fair. These are some reasons why reasonable applications work "quite good". We do not state that it is optimal in any sense, but a reasonable application may be better than a fair application in the risk-averse case.

We note that the evaluation of reasonable applications can be much more difficult than of fair applications, but if for each agent i, $h_i(x) = a_i - b_i \cdot e^{-\lambda x}$ then a similar recursion works as in (1) and (2).

9.3 Necessity of being informed about the own process

We assumed that none of the chosen players knew better anything about any chance event of any other chosen agent. We show here an example that fails this requirement and it makes the mechanism wrong. Consider two agents a and b that will surely be accepted. Assume that a believes the probability of an unfavourable event in his work to be 50 %, but another one, called B knows that the probability is 60%, he knows the estimation of a and he also knows that at a particular decision point of him, he will be asked for the decision corresponding to this chance event. It can be checked that if the application of a is fair then if b increases the asked payment in his application of the more probable case by an amount of money and decreases it in the other case by the same amount then the value of his application remains the same but this way, he bets $1:1$ with b on an event of 60% probability.

In order to limit such losses, a could rightfully say that larger bet can only be increased on worse conditions. Submitting reasonable application with concave worth function makes something similar, which is another reason to use this.

9.4 Agents with limited funds

This subsection is only a suggestion for the cases with such agents, and it is not optimal in any sense.

Our mechanism requires each agent to be able to pay so much money as the maximum possible damage he could have caused. But in many cases, there may be a plenty of agents who cannot satisfy this requirement. However, accepting such agent a may be a good decision, if a is reliable to some degree.

To solve this problem, a should find someone who has enough funds, and who takes the responsibility, for example for an appropriate fee. If the agent is reliable to some degree then he should be able to find such insurer player b. (It can be even the principal, but considered as another player.) This method may also be used when a has enough funds, but he is very risk-averse.

Here, a and b work similarly as a controlled and a controlling parties in Section 8.3. The difference is that b does not work here, and he knows the probability distribution of the result of a not from his own decision tree but from his knowledge about the reliability of a. Furthermore, the role of b here can be combined with his role in Section 8.3.

9.5 Communication

Under non-ideal circumstances, the messages the players send should be secret. However, the model requires the communication to be contractible. Both requirements can simply be satisfied using a cryptographic communication protocol.

9.6 Applications in genetic programming

This second price mechanism may also be useful in genetic programming, when we want to find an efficient algorithm for such a problem that can be distributed into slightly dependent subproblems, while the subprograms for these subproblems should cooperate, and their overall achievement is what we can evaluate. In this case we should experiment such subprograms simultaneously that also submits applications, and we simulate here a competitive market with a principal that uses our second price mechanism.

If the form of all possible cost price applications for each subproblem is simple enough then we may need to experiment less parameters in each subproblems than in the original problem, and that is why this method converges faster.

10 Acknowledgements

Special thanks to **Miklós Pintér** and thanks to Mihály Bárász, Katalin Friedl and many other people.

References

[1] *S Athey, I Segal: An Efficient Dynamic Mechanism*
http://ideas.repec.org/p/cla/levrem/122247000000001134.html

[2] *G Bergantinos, E Sánchez: How to distribute costs associated with a delayed project.*
Annals of Operations Research, Volume 109, Numbers 1-4 (January, 2002) pp. 159-174

[3] *Brânzei, Ferrari, Fragnelli, Tijs: Two Approaches to the Problem of Sharing Delay Costs in Joint Projects*
Annals of Operations Research, Volume 109, Numbers 1-4 (January, 2002) pp. 359-374

[4] *J Cremer, R McLean: Optimal selling strategies under uncertainty for a discriminating monopolist when demands are interdependent*
Econometrica, Volume 53 (1985), pp. 345-361

[5] *P Dasgupta, E Maskin: Efficient auctions*
The Quarterly Journal of Economics, Volume 115 (2000), pp. 341-388

[6] *P Dasgupta, P Hammond, E Maskin: The implementation of social choice rules: some general results on incentive compatibility*
Review of Economic Studies, Volume 46 (1979), pp. 185-216

[7] *A Estévez-Fernández, P Borm, H Hamers: Project games*
International Journal of Game Theory, Volume 36, Number 2 (October, 2007), pp. 149-176

[8] *A Gibbard: Manipulation of voting schemes: a general result*
Econometrica, Volume 41 (1973), pp. 587-601

[9] *T Groves: Incentives in teams*
Econometrica, Volume 41, No. 4 (Jul., 1973), pp. 617-631

[10] *B Holmstrom: On incentives and control in organizations*
Ph.D. thesis, Stanford University (1977)

[11] *S Ieong, A M-C So, M Sundararajan: Stochastic Mechanism Design*
Proceedings of the 3rd WINE (2007), LNCS 4858, pp. 269-280

[12] *E Kalai: Large robust games*
Econometrica, Volume 72 (2004), pp. 1631-1666

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

[14] *R Myerson: Multistage Games with Communication*
Econometrica, Volume 54 (1986), pp. 323-358

[15] *R Myerson: Incentive-compatibility and the bargaining problem*
Econometrica, Volume 47 (1979), pp. 61-73

[16] *A Papakonstantinou, A Rogers, E Gerding, N Jennings: A Truthful Two-Stage Mechanism for Eliciting Probabilistic Estimates with Unknown Costs*
In Proceedings of the 18th ECAI (2008), Patras, Greece

[17] *M Perry, P Reny: An ex post efficient auction*
Econometrica, Volume 70 (2002), pp. 1199-1212

[18] *M H Rothkopf, T J Teisberg, E P Kahn: Why Are Vickrey Auctions Rare?*
The Journal of Political Economy, Volume 98, No. 1 (Feb., 1990), pp. 94-109

[19] *W Vickrey: Counterspeculation, Auctions, and Competitive Sealed Tenders*
The Journal of Finance, Volume 16, No. 1. (Mar., 1961), pp. 8-37