An Incentive Compatible Mechanism for Distributed Resource Planning

Erhan Kutanoglu
S. David Wu
Lehigh University

Report No. 00T-011

An Incentive Compatible Mechanism for Distributed Resource Planning

Erhan Kutanoglu
Department of Industrial Engineering
University of Arkansas
4207 Bell Engineering Center, Fayetteville, AR 72701
erhank@engr.uark.edu
and

S. David Wu
Manufacturing Logistics Institute
Department of Industrial and Mfg. Systems Engineering
Lehigh University
200 W. Packer Ave., Bethlehem, PA, 18015
david.wu@lehigh.edu

Abstract

We study collaborative resource planning arises when resource managers need to coordinate their schedule with a group of internal or external customers. We design a "schedule selection game" where all participating agents state their scheduling preferences via a valuation scheme, and the mechanism selects the final schedule based on the collective input. A majority of distributed coordination approaches in the literature assume that agents have the good will to coorperate and they always provide private information truthfully. Our contention is that without proper incentives, agents may not reveal this information trufully and they may not behave in a way that is aligned with overall system efficiency. We examine the issue of incentive compatibility, and we propose a direct revelation mechanism that implements an optimal schedule under dominant strategies. The mechanism is a modification of the well known Vickrey-Groves-Clarke mechanism, and we analyze its properties including budget balanceness, and individual rationality. We then illustrate the mechanism using real-world data obtained from an electronics manufacturer.

1 Introduction

The advances of digital communicaitons and the web is having profound impact to decision making within and across firms. Decision making needs to be increasingly decentralized so as to support the heterogeneous interaction between functional groups across conventional structures and boundaries. Supply chain coordination requires not only intra-firm coordination across divisional/functional boundaries, but also inter-firm decisions between customers and suppliers. In this paper, we will focus on the coordination of resource allocation and scheduling decisions, although many of the same principle applies to decision coordination in general. We consider an environment where multiple agents have direct access to an electronic bulletin board (such as a web page) that posts alternative resouce allocation schedules. The agents vote for their choice of schedules based on preferences, and the system determines the "best" schedule based on a certain mechanism. Our aim is to characterize and design such a mechanism. We believe the mechanism must be robust and easy to implement in order to enhance web-based distributed decision making in other forms.

Our main focus in this analysis will be the design of an "incentive compatible" mechanism, an often overlooked subject not only in operations management but also in the distributed decisions literature. In any multi-agent environment, it is crucial to consdier the perspective of each agent in the decision making process and their economic incentives. If agents are not provided with proper incentives, they may not perform in a way that is aligned with the overall system goals. To ensure *incentive compatibility* we need a reward mechanism for agents that would align their individual (local) interests with the system-wide (global) objectives. To this end, we consider a *schedule selection game* as follows: the system proposes resource allocation for a particular period of time by posting a number of alternative resource schedules (e.g., specifying which *resources* will be allocated for what type of *jobs* over which particular *time slots*), agents competing for the resources have private information concerning the true benefits of each schedule to the job types they represent. Agents are to provide a valuation of each schedule and this information will be ultimately used to identify a final schedule. Since each schedule allocates resources differently, agents could have drastically different preferences over the schedules under consideration. But there is no gurantee that each agent's valuation will be truthful as some may benefit from reporting a misleading valuation. A primary issue is to design incentives that would elicit agents' true preferences so that the overall best schedule could be selected. We will describe a mechanism that identifies the optimal solutions regardless of the fact that individual preferences are known only by each individual, while various forms of manipulations or misrepresentations are possible.

Designing an incentive compatible procedure in the environment such as the above is known as a *mechanism design problem*, a subject of extensive study in the microeconomics and game theory literature [16]. We will focus on a specific subarea of this literature known as the *direct revelation mechanisms.* Similar problems are studied in many different contexts and applications. The origin of all these studies is the "optimal control problem" of a central authority who tries to control the agents in an organization. The problem was first posed and studied as an incentive problem by Groves in the early seventies [8, 9]. A reflection of this mechanism on a resource allocation problem in a decentralized organization can be found in [12]. These procedures have been referred to as Groves mechanisms or Groves incentive schemes. In a different context, Groves studies the problem of coordinating economic agents' decisions considering external effects [10]. This line of research goes back to Vickrey's seminal work on "counterspeculation" and design of second-price auction mechanisms [18]. Although there are many applications of Groves mechanisms in the economics literature (see, e.g., Brewer and Plott [3]), applications to modeling and optimization did not exist until more recently. A recent implementation of this sort considers an incentive-compatible budget allocation problem for site decontamination projects [5]. A general treatment of Groves mechanisms with a special focus on Articial Intelligence oriented Multi-agent systems can be found in [17].

Groves [11] proposes a related but different approach around the same time he developed Groves mechanisms. In this study, the presence of central control is no longer needed. He analyzes a collective "public choice" problem where agents try to implement a "socially efficient" (or optimal) public project. In this paper, we follow an approach closely related to this latter line of research where the mechanism involves collective decision making and self-coordination of job agents rather than a center-based control approach (we provide a brief discussion on this issue in Section 4). As common in all previous incentive compatibility studies, we will assume the existence of a transferable commodity which is usually represented by "money." We will consider self-interested agents who try to maximize their local (net) utility as a combined function of their original utility and the amount of reward (money) that the mechanism allocates. We interpret the original utility as "profits" that an agent makes when a certain schedule is implemented. Without loss of generality, this assumption simplifies the discussion of the mechanism since the original utility and the transfer have the same unit ("money" or dollars). In Section 4, we discuss the implications and possible alternatives of this assumption. In Section 2, we first outline a direct revelation mechanism for the schedule selection game, and then discuss various properties and relevant game theoretic results such as dominant strategy equilibrium. We then investigate related issues such as the generation of candidate schedules and the quality of the selected schedule through the mechanism. We give an example to illustrate a typical implementation of the approach in a real-life scheduling environment in Section 3. The last section provides conclusions and future research topics.

2 The Schedule Selection Game

We first define some notations used throughout the chapter: For a generic vector n-vector x, x_i denotes the ith element of the vector, i.e., $x = (x_1, \ldots, x_n)$, and x_{-i} denotes $n-1$-vector of the elements of x without x_i, i.e., $x_{-i} = (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$. We use this notion of "others" $(-i)$ to represent the other agents' actions/decisions with respect to the agent under consideration, agent i. Similarly, for a given set $X \equiv \cup X_i$, we will use X_{-i} to denote the set X without X_i. Thus, we can interpret $-i$ in the subscript as the original vector or set without the ith component, i.e., $-i \equiv \mathcal{N} \setminus \{i\}$.

The schedule selection game can be described as follows. There is a set of job agents $\mathcal{N}$, indexed by $i = 1, \ldots, N$, and a set of feasible shop schedules Y known to all agents. Each schedule is a complete and feasible job-shop schedule where each job receives a particular block of resource time for each of its operations. A job agent represents one and exactly one job in the schedule (the analysis of this to the case where each agent is responsible for more than one job is possible and left for future research). The impact (valuation) of any particular schedule to a job agent is private information only known to the agent. The job agents must collectively choose an *outcome* consisting of (1) a shop schedule y from the set of feasible schedules Y, and (2) *transfers* of a numeraire commodity ("money") from/to all agents, $\{\delta_i\}_{i\in\mathcal{N}}$. Here δ_i denotes the amount of money transferred to (if positive) or from (if negative) agent i. We assume that the preferences of each job agent over the outcomes are summarized by a utility function u_i. We also assume that an agent i only cares about the transfer δ_i and the profit π_i he receives from the schedule y:

$$u_i(y, \delta_i) = \pi_i(y) + \delta_i. \tag{1}$$

Note that the utility u_i of agent i does not depend on other agents' transfers, δ_{-i}. Thus, agent i prefers outcome (y, δ) with schedule y and transfer vector $\delta \equiv (\delta_1, \ldots, \delta_N)$ to another outcome (y', δ') if and only if $u_i(y, \delta_i) \geq u_i(y', \delta'_i)$. For a classical JSP with total weighted tardiness objective, the profit π_i of an agent i can be interpreted as negative weighted tardiness costs that is converted to monetary units. In the following discussion, we will use the generic term "profit" while keeping in mind that it can represent a variety of objectives including weighted tardiness. Simply stated, the schedule selection game is a decision procedure (a mechanism). Through the use of proper transfer functions (incentives) agents will always reveal their true preference while the collection of all agents' decisions correspond to the best solution possible for all.

2.1 The Mechanism Design Problem

We now discuss several important properties of the *schedule selection game* using a game theoretic framework. We assume that a job agent's preferences are known only by the agent himself. In a manufacturing context, the preferences may be motivated by delivery requirements or contractual agreement specific to the job, the performance evaluation a product manager is subject to, or any other constraints that may be considered "local." Any decision procedure that is responsive to individual preferences will require some type of communication or signaling of preference information. The decision mechanism uses the communicated information to choose the "best" schedule among the alternatives in Y.

We first define the schedule selection game as an N-person noncooperative game with incomplete information. A *mechanism* Γ is a collection $(\{M_i\}_{i \in \mathcal{N}}, h(\cdot))$ where M_i is a set of alternative messages μ_i for agent i, and $h(\cdot)$ is an *outcome function* or *allocation rule*. Function $h(\cdot)$ specifies for every joint message $\mu \equiv (\mu_1, \ldots, \mu_N)$, (1) a collective schedule y, and (2) a transfer profile δ, that is $h(\mu) = (y, \delta)$. In the following discussion, we separate the outcome function into two parts, i.e., $h(\mu) = (y(\mu), \delta(\mu))$, where $y(\cdot)$ and $\delta(\cdot) \equiv (\delta_i(\cdot), \ldots, \delta_N(\cdot))$ denote the *schedule selection function* and *transfer function*, respectively. Thus, we have a game composed of N players corresponding to the job agents. The strategy set of player i is the message space M_i of agent i, and the preferences of player i over joint strategies μ (messages) are defined by the utility function u_i and by the outcome function $h(\cdot)$, i.e., player i prefers μ over μ' if and only if $u_i(h(\mu)) \geq u_i(h(\mu'))$. This defines an N-person non-cooperative game of incomplete information since each player's preferences are known only to himself.

Typically, in a game theoretic framework, the goal of a mechanism is to choose a particular outcome using the outcome function $h(\cdot)$ for a particular realization of agents' utility functions, say $\{\pi_i\}_{i\in\mathcal{N}}$. From a system optimization point of view, a desirable result would be to choose an *optimal*, or *socially efficient*, schedule y^*, i.e., find a schedule y^* in Y such that $\sum_i \pi_i(y)$ is maximized (Note that here "optimal schedule" is equivalent to "socially efficient schedule" which is different when the auction mechanism design is considered where "optimality" and "efficiency" differ in their objectives).

We say that mechanism Γ with outcome function $h(\cdot) = (y(\cdot), \delta(\cdot))$ *implements* an optimal schedule y^* if there is an *equilibrium* joint message μ^* of the game induced by Γ whose outcome is y^*. Note that the implementation of a particular outcome depends on the equilibrium concept used.

Since the specific form of the outcome function is a mechanism design choice, we seek outcome functions that implement a particular schedule, more specifically an optimal one. From this point of view, we are particularly interested in *direct revelation mechanisms* which normally involve two stages: (1) In the first stage, each agent is asked to reveal his true profit from the schedules, i.e., each $\mu_i(y)$ is the value reported by i that supposedly represents the profit obtained by agent i if y is selected. (2) After all agents report their values, the outcome function $h = (y(\cdot), \delta(\cdot))$ chooses the maximizer of the the total reported profit, i.e.,

$$y^*(\mu) = \arg\max_{y\in Y} \sum_i \mu_i(y), \tag{2}$$

as if all agents have reported their true profits, and the payments are made according to transfer function $\delta(\mu)$.

Note that if all agents truthfully report their profits, then the schedule selection function (2) finds an optimal schedule y^* in Y. In fact, this requirement of having the agents communicate true profits is the essence of direct revelation mechanisms. However, because of the dispersed information and individual preferences over schedules, it may not be in any agent's best-interest to "tell the truth". For example, by reporting falsely his valuations of the schedules y, an agent may be able to secure a larger transfer than necessary to compensate him for any change in y that his false reporting causes.

Thus, without a proper *incentive scheme* that induces "truth telling" behavior, an agent may lie about his profits, with the intent to maximize his local utility, $u_i(y^*(\mu))$. Hence, the primary focus in designing direct revelation mechanisms is on constructing a proper incentive scheme. Such schemes are characterized by carefully designed *transfer functions*, $\delta(\cdot)$, which is a part of the outcome function. With such an incentive scheme, the agents voluntarily report their true profits. To formally study the aforementioned issue of *incentive compatibility*, i.e., inducing truth telling as an equilibrium behavior, we need to define a specific equilibrium for the game induced by the direct revelation mechanism. In the following, we summarize the concept of dominant strategy equilibrium for the mechanism and analyze its properties.

2.2 Dominant Strategy Equilibrium

Dominant strategy equilibrium states that each player plays his best strategy that is individually best for him regardless of the strategy chosen by any other players.

Definition 1 *(Dominant Strategy Equilibrium) For mechanism Γ with message space $\{M_i\}_{i\in\mathcal{N}}$ and outcome function $h(\cdot)$, a joint message $\mu^* = (\mu_1^*, \ldots, \mu_N^*)$ constitute a dominant strategy equilibrium if, for every i,*

$$u_i(h(\mu_i^*, \mu_{-i})) \geq u_i(h(\mu_i, \mu_{-i}))\ \forall\ \mu_i \in M_i,\ \forall\ \mu_{-i} \in \mathcal{M}_{-i}$$

where $\mu_{-i} \equiv (\mu_1, \ldots, \mu_{i-1}, \mu_{i+1}, \ldots, \mu_N)$ and $\mathcal{M}_{-i} \equiv \prod_{i'\neq i} M_{i'}$.

Thus, if a dominant strategy equilibrium exists for a mechanism, then each player would have a strong incentive to play his dominant strategy. For a direct revelation mechanism with proper incentives, this means that every job agent has an incentive to report his true valuation of the schedules no matter what the other agents report since doing so maximizes his own (true) utility, i.e., $\mu_i^*(y) = \pi_i(y)\ \forall y \in Y$.

Assuming the schedule selection (2) of an outcome function, agent i, for every y, solves the utility maximization problem

$$\mu_i^*(y) = \arg\max_{\mu_i \in M_i} u_i(y^*(\mu_i, \mu_{-i}), \delta_i(\mu_i, \mu_{-i}))$$

or

$$\mu_i^*(y) = \arg\max_{\mu_i \in M_i} \pi_i(y^*(\mu_i, \mu_{-i})) + \delta_i(\mu_i, \mu_{-i})$$

for any $\mu_{-i} \in \mathcal{M}_{-i}$. Note that through messages the agent is able to affect both the final schedule selection and his transfer which in turn influence the total payoff to agent i. As mentioned above, it is the goal of the transfer function $\delta(\cdot)$ to provide incentives to the agents to reveal their true valuations regardless of what the other agents report.

Consider the case where transfers are all zero, i.e., $\delta(\mu) = 0 \ \forall \ \mu \in \mathcal{M}$. In this case, $u_i(y, \delta_i = 0) = \pi_i(y)$. Thus, without transfers, each job agent will try to maximize his original profit when he is asked to report his valuation of the schedules. As a self-interested agent, job agent i will not tell the truth by using the following reporting rule (assuming minimum profit of 0 from any schedule):

$$\mu_i^*(y, \delta_i = 0) = \begin{cases} L_i & \text{if } y = \arg\max_{y' in Y} \pi_i(y') \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

where L_i is a (large) number which may or may not be equal to true value $\pi_i(y)$. Hence, without transfer, the agents tend to inflate the profit for their local best schedule, and report minimum possible profit (0) for all others. Using these reported values, it is clear that the outcome function as defined in (2) and $\delta(\cdot) = 0$ cannot implement an optimal schedule. In fact, the following proposition is a special case of a result known as Gibbard-Satterthwaite Impossibility theorem [11, 16].

Proposition 1 *A direct revelation mechanism Γ with schedule selection function defined in (2) cannot implement an optimal schedule in dominant strategies if transfers are not allowed.*

In fact, a specific form of transfer function known as *Groves-Clarke incentive scheme* (Groves [9, 11], and Clarke [4]) is sufficient to guarantee the implementation of an optimal schedule in dominant strategies as outlined in the following proposition (Note how the net utility functions of the agents and the joint profits are aligned through schedule selection and transfer functions).

Proposition 2 *A direct revelation mechanism $\Gamma = (\{M_i\}_{i \in \mathcal{N}}, h = (y^*(\cdot), \delta^*(\cdot))$ with schedule selection function defined by (2) implements an optimal schedule in dominant strategies if, for all $i \in \mathcal{N}$,*

$$\delta_i^*(\mu) = \sum_{i' \neq i} \mu_{i'}(y^*(\mu)) - \sum_{i' \neq i} \mu_{i'}(y^*_{-i}(\mu_{-i})) \tag{4}$$

where $y^*_{-i}(\cdot)$ denotes the outcome that would be selected without agent i's messages, i.e.,

$$y^*_{-i}(\mu_{-i}) = \arg\max_{y \in Y} \sum_{i' \neq i} \mu_{i'}(y)$$

Proof: See Appendix A.

Note that any joint message μ with $\mu_i(y) = \pi(y) + L_i \; \forall i$ where L_i is now a constant does not change the selection of an optimal schedule. Hence, the above proposition holds even if the agents increase or decrease their respective profits by a constant amount. A direct revelation mechanism Γ with outcome function $h(\cdot) = (y^*(\cdot), \delta^*(\cdot))$ (see equations (2) and (4) respectively) is known as *Groves-Clarke mechanism* ([8, 11, 4]. Since the G-C mechanism can be viewed as a generalization of Vickrey's second price auction, it is sometimes called Vickrey-Groves-Clarke (VCG) mechanism. See, e.g., Krishna and Perry [13]). Here, the second term on the right hand side can be interpreted as a charge equal to the total profit that would be possible if agent i did not participate in the decision making process at all (or equaivalently, if he reported indifference, $\mu_i(y) = 0$ for all i).

Note that agent i's transfer δ_i in a Groves-Clarke mechanism is zero if his announcement does not change the schedule selection relative to what would be optimal for agents $\mathcal{N} \setminus \{i\}$, i.e., if $y^*_{-i}(\mu_{-i}) = y^*(\mu)$. It is negative if his participation causes the others to choose a different schedule, where the agent is called *pivotal*. In this case, he pays the difference he makes on the others' profits. In a way, the payment represents the *externality* that i exerts on the other $I-1$ agents by his (and his job's) presence in the process. Thus, in Groves-Clarke mechanism, all agents end up paying zero or some positive amount of money (another AI-oriented application of Groves-Clarke mechanism can be found in Sandholm [17]).

We showed that if the agents use a direct revelation mechanism with the outcome function defined by (2) and (4), they will have a strong incentive to report their true valuations of the schedules, be able to coordinate their individual preferences, and select an overall optimal schedule. However, we should note that there are many issues to cover for a successful implementation of such a direct revelation mechanism. We will analyze some of these in the following sections.

2.3 Participation and Asynchronous Decision Making

In the previous sections, we showed that each agent has an incentive to truthfully report his local profit when asked. However, we did not question if the agents have an incentive to *participate* in the "schedule selection" process in the first place. We implicitly assumed that all agents are required to participate in the game without other alternatives. If the agents are free to participate while full participation is necessary for overall system performance, then the *individual rationality* or *participation* constraints should be considered in the design stage so that the agents are provided with the incentives to participate in the process.

We first analyze if total participation is necessary for the ultimate selection of the optimal schedule (in Y). This is also an important issue if the scheduling game is to be implemented asychronously, say on an Internet environment. We first note that even if an agent chooses not to participate, the schedule selected without his participation will include his job. By design, a schedule selected with full or partial participation will be a complete and feasible schedule inclusive of all operations of all jobs. Since the agents' strategies are defined as messages, not as individual scheduling decisions, the only part missing from a nonparticipant is his messages. In this case, the agent does not pay or receive any transfer but is bound to accept the schedule that other agents collectively choose without his input. Thus, we treat the non-participation of an agent as the case where he reports zero profits for all schedules in Y, i.e., he is indifferent for all schedules.

This fact allows us to analyze a form of asynchronous decision making in the schedule selection game. For instance, the schedule selection function can be activated any time to find the current best schedule with potentially partial messages. Without waiting for all of the messages from all of the agents, the outcome function can select a schedule with respect to the currently reported values from currently participating agents. More importantly, once having submitted his message, there is no need for an agent to revise his message, regardless of which other agents will end up participating. This results in two properties potentially useful for asynchronous applications: (1) the agents do not have to report their profit functions simultaneously, and (2) the agents do not have to report their profits for all the schedules, i.e., they may report on a subset of the schedules.

Before starting a formal analysis, we recall that with a Groves-Clarke mechanism if *all* agents report their profits for *all* schedules, an optimal schedule in set Y will be selected. We consider this the *full participation case.* Consider an alternative, *partial participation case*, where a subset of agents, say $I \subset \mathcal{N}$, choose not to participate while the rest, denoted by $I' = \mathcal{N} \setminus I$, report for *all* of the schedules. In the next proposition, we show that if none of the agents in I is charged a positive amount of money in the full participation case, then their non-participation does not affect the schedule selected in the partial participation case.

Proposition 3 *If the total transfer of the agents in set I is zero under the full participation Groves-Clarke mechanism, then the scheduling selection function in the partial participation* *case selects an optimal schedule y^* in Y and the total transfer is the same as that of the full participation case.*

Proof: See appendix B.

This shows that to make an optimal schedule selection, participation from the nonpivotal agents is not essential, i.e., full participation is not necessary for optimal schedule selection. However, if any of the pivotal agents choose not to participate then the schedule selected without their messages may be suboptimal, i.e., $y^*(\mu_{I'} = \pi_{I'}) \neq y^*(\mu = \pi)$ when non-participating agents I include pivotal agents. Hence, for the full "utilization" of the mechanism (to select the best schedule in Y), we need to guarantee participation from the pivotal agents. In general, as mentioned above, the mechanism designer needs to consider *individual rationality* or *participation* constraints to have (all) the agents participate the process. Thus, an efficient mechanism must guarantee that agents will not be worse off by participating in the mechanism. Since we cannot discriminate against agents based on the fact that they would have zero transfer in the full participation case, we cannot consider the participation constraints only for pivotal agents, i.e., the constraints must be considered for all the agents. One common assumption is that non-participants are not rewarded or charged any transfer.

Normally, these constraints take the form of

$$u_i(y, \delta_i) \geq IR_i(\cdot)$$

where IR_i is the "break-even" payoff of agent i which in general can be a function of his profit function, outcome function, etc. In its simplest form, IR_i can be 0, for example, if we assume that every agent's utility from not participating is 0 (In the scheduling context, this can be the case where the agent's job is left to the next planning horizon, where the profit from the schedule can be normalized to lowest possible profit 0). Another simple form is to have IR_i depend on a default schedule that would be chosen when there is no full participation (coordination). If we denote this by $\bar{y}$, then

$$IR_i(\bar{y}) = \pi_i(\bar{y}).$$

This means that, for every agent, the mechanism should guarantee a positive gain from participation with respect to the "uncooperative" solution. In a more general case, IR_i can depend on the outcome function, y^*. This might be the result of externalities exerted by participants on non-participants, which is definitely a case in our setting. Hence, if each agent is to consider a "worst case scenario" for IR_i, i.e., the profit from a schedule that can be selected by the others without his input, the participation constraints take the form

$$u_i(\cdot) \geq \pi_i(y'_{-i}(\mu_{-i}))$$

where $y'_{-i}(\mu_{-i})$ denotes the schedule selected without i's messages that would possibly minimize his profit.

There are different levels of implementation of the individual rationality constraints, such as *ex ante*, *interim,* and *ex post* individual rationality [16]. In general, satisfying all IR conditions at the same time may not be possible. We can show however, some interim and ex post participation constraints are readily satisfied by using the Groves-Clarke mechanism. For example, if we let $y'_{-i}(\mu_{-i})$ to be the schedule selected by the other agents in the dominant strategy equilibrium, i.e.,

$$y'_{-i}(\cdot) = y^*_{-i}(\cdot),$$

then we can show that ex post IR constraints are satisfied. To show this, we only need to prove that the payment by agent i in equilibrium does not exceed the decrease in his profit from the optimal schedule to the schedule selected by the others without him, i.e.,

$$-\delta^*_i(\pi) \geq \pi_i(y^*(\pi)) - \pi_i(y^*_{-i}(\pi_{-i})).$$

Satisfying the more demanding participation constraints is not obvious, and there has been some negative results reported in this regard [11]. This is magnified when the agents have a large negative impact on other agents' profits as in our case: Job agents compete for resources to finish early, and in general, a high profit for a job might mean a low profit for another. Thus, an effect of cooperation might be to choose a schedule that significantly reduces an agent's (before transfer) profits from what the agent could achieve when there were no cooperation. In this case, if the participation is voluntary, then the agent would not have an incentive to participate the mechanism. One possible solution of this difficulty is to constrain the mechanism such that the agents have positive benefits from cooperation in long-term expectation. This means that the agents will gain from participation in the long run over repeated runs of the mechanism. See [10] for an example. We will leave this subject for future study.

2.4 Budget Balanceness

An issue in transfer-based mechanisms is to have the total transfers satisfy the budget balanceness constraint. For example, one mechanism designer may need to have a nonnegative total transfer or nonpositive total transfer, or even more demanding, zero total transfer (no waste of money), i.e., fully balanced budget. In a centerless Groves-Clarke mechanism such as ours, one reasonable requirement might be that no outside source of financing is allowed, i.e.,

$$\sum_i \delta_i \leq 0, \tag{5}$$

This is easily satisfied by the Groves-Clarke transfer function outlined above. In this case, all agents are charged or taxed by some nonnegative amount, usually total transfer being strictly negative.

A more demanding requirement is that of having the total transfer zero. This requirement is called *balanced budget* and formally denoted by the constraint

$$\sum_i \delta_i = 0.$$

Note that, in this case, some δ_i will have to be positive, i.e., some agents will be paid a positive amount. Since the total transfer of money is zero, we can interpret this as having some of the agents pay to the other agents; no money enters or leaves the system (ideally, the *gainers* of the collective schedule selection should compensate the aggregate loss of *losers* of the coordination). However, it is shown that, in general, there is no direct revelation mechanism that both implements an optimal solution in dominant strategies and satisfies balanced budget constraint for every possible message profile [7]. Thus, the presence of private information means that the agents must either accept some waste of money ($\sum_i \delta_i \leq 0$) as in the Groves-Clarke transfers, or give up on finding an optimal schedule all the time. If Bayesian incentive compatibility (as opposed to dominant strategy incentive compatibility) is satisfactory for mechanism design, then it is possible to devise a transfer function that will induce incentive compatibility on the expected values. One such mechanism is first proposed by Arrow [1] and d'Aspremont and Gerard-Varet [6] (AGV mechanism that does not guarantee IR). This is recently refined by Krishna and Perry [13] to include individual rationality.

A rather simple way to overcome the budget balancing problem in dominant strategies is to introduce an extra agent whose preferences are known, or who has no preferences over the schedules. Let "agent 0" denote this extra agent. This agent's utility might be set as

$u_0 = t_0(\mu)$ which summarizes that he only cares the transfer but not the schedule selection (agent 0 can be interpreted as a central authority who tries to maximize his total transfer like a seller). In this case, if we set $\delta_0^*(\mu) = -\sum_{i \in \mathcal{N}} \delta_i(\mu)$, then the Groves mechanism both satisfies the balanced budget requirement and selects the optimal schedule. Agent 0 collects all the payments from the agents which guarantees a nonnegative profit for him. In fact, we can interpret this agent as "the center" who can design and impose a mechanism, and manage the schedule selection and transfers among the agents.

3 An Industry Case and Illustrative Example

3.1 Background

In this section, we use an industry case study to illustrate how the schedule selection game outlined in the previous section can be implemented in a real world environment. We base our example on real scenarios and data that we collected from an electronics assembly plant. The plant produces electronic components for automotive manufacturers worldwide. These components include electronic engine control (EEC) modules, anti-lock brake systems (ABS) control, speed control unit and others. A variety of products are produced in this facility. There are *product managers* each responsible for the production of a family of similar products. By nature of the automotive supply chain, each manager has his/her own product-specific constraints and preferences driven by contractual agreements with each customer and supplier, delivery arrangements, demand characteristics, quality and technological specifications. The product managers must compete for the same set of resources while trying to maximize his/her own interests and preferences at the moment. The shared resources are a collection of automated production lines called the consolidated surface mount device (SMD). Approximately 80% of all electronic products in the plant go through this area, where a pick and place process using a series of SMD machines and testing stations form the main operations. The focus has been on maximizing the efficiency of the capital intensive SMD area while aligning plant-wide objectives with product level requirements so as to maintain responsiveness to customers.

Ideally, each agent in our distributed environment would correspond to a different product manager who is a decision making unit in the actual system. However, due to the way scheduling data is currently managed, we collect primarily order data from the EEC product family, where a single product manager and several schedulers are responsible for the production. While the case example does not capture the real competition among different product managers, it does simulate the distinctly different interests of customers and their products. We assume that each job agent represents an order or a lot with a determined number of specific type from the EEC family. In constructing the case, it quickly became apparent that designing an "agent decision structure" is a non-trivial task of its own if there is a choice among alternative structures. Issues such as how to designate the right level of responsibilities to the right agent (e.g., how many products an agent should represent, in what level of details, etc.) maybe critical to the overall performance of the overall mechanism. Here, we assume a particular design which is consistent with our initial view where each job agent is responsible for exactly one job/order. An implicit assumption is that a production manager responsible for multiple products would have access to multiple job (software) agents' utilities. But the mechanism does not consider the potential dependency among these jobs. This particular design can be extended to a more complex agent design structure where each agent is responsible for a multiple of orders for a product type. However, the product manager may use a more complex utility function to sort out the priority trade-off among his/her products. This does not change the basic structure of the schedule selection game.

As the EEC product manager clearly pointed out to us, the main bottleneck in the production environment is the SMD resources and operations. This allows us to assume that a reasonable lead time with low variation can be approximated for downstream, non-bottleneck operations. Therefore, we may view scheduling as a single-stage problem where each job consists of a single operation with its due-date calculated by subtracting the downstream lead time from the original order due-date. Although the problem is single-stage, an SMD operation can be performed by any machine in a specific subset of the SMD lines. There are currently around 10 SMD machines on the line and one or more of these (up to 5) can alternatively process a specific SMD operation. However, for a given service mount operation these alternative machines are not identical in processing speed, yield, and quality.

To illustrate the significance of private information and local constraints a job agent maintains, the issues surrounding order due-dates provide good examples. Order due-dates are at best moving targets subject to frequent revisions from the customer (via electronic schedule releases, or informal off-line communications). Worse, due-dates are often coupled with business issues such as pricing, potential discounts, consolidation of freights due to transportation or container restrictions, future order status, long-term good will, and inventory restrictions. In other words, behind a simple due-date performance measure there is a significant amount of private information only known to the product manager in charge. As one of the product managers stated, the managers may spend more than half of his/her time dealing with these customer related issues. This does not even include other product specific issues related to component suppliers, transportation providers, or the policies and reward systems specific to the facility, all of which influence a product manager's decisions.

3.2 A Numerical Example

In the following, we construct a numerical example to illustrate some of the finer points of the schedule selection game. The numbers used in the example are from real data. We make the following implementation decisions:

- We assume that a primary concern of each job agent is to meet the job due-date.
- We assume that lot sizing and routing decisions are made before scheduling decisions. This assumption leads to two important simplifications: (1) setup times can be added to the processing times of the operations, and (2) the problem can be decomposed into a series of single machine problems since the machine assignments are made through routing decisions. We use the lot-sizing and routing decisions made by the scheduling software that has been implemented in the plant during the test period. Note, however, any algorithm can be used instead. Hence, we can think of the scheduling process as an integral part of a larger planning/routing/scheduling decision structure where the higher-level decisions are an input for our scheduling process.
- At any one time there may be thousands of jobs in the system database. To limit ourselves to a reasonable number of jobs, we constructed a "candidate job set" which have due dates smaller than a certain threshold value. In this way, we eliminated jobs that would not be scheduled in the current planning horizon even if they were considered. In the following numerical example, we excluded the jobs (i.e., did not place them in the candidate job set) with due dates later than 10 days from current date.
- We assume hourly time buckets for our discrete time model of scheduling problem. Due to this, we needed to consolidate really short jobs to form "super" jobs that have sizes

comparable to our time bucket. We use the same planning horizon (5 days, 120 hours) that is employed in the plant's current practice.

Formally, each job agent i, and his job, can be represented by a 4-tuple (a_i, m_i, p_i, d_i), where a_i is the release time (or earliest start time) of job i, m_i is the SMD machine that is assigned, p_i is the processing time requirement on machine m_i including setup time, and d_i is the derived job due date. To construct the candidate schedules in set Y, we used the Lagrangean-based auction-theoretic algorithm developed in our earlier studies [14, 15]. Thus, we have a set of feasible schedules from which we form the schedule selection game.

We view the implementation of a direct revelation mechanism, or more specifically a Groves-Clarke mechanism, as follows: All candidate schedules in set Y are posted on a public electronic bulletin board. The job agents (or possibly software agents that represent human agents) then provide their own valuation for each schedule using their own criteria based on private information. Since each agent's performance depends only on its job's allocation in a schedule and its transfer, an agent need not to see the allocations of other jobs. From this point of view, this procedure is asynchronous. When all job agents provide the necessary valuation, the schedule with the maximum total reported utility is selected as an outcome of the mechanism and the transfers are calculated. To simulate this process, we use the negative value of weighted tardiness for all utility measures, or equivalently, we minimize weighted tardiness as "disutility".

To further illustrate this step, we assume that there is a "utility" evaluation table to be filled by the job agents. Each job agent has access to his part of each schedule and fills a value for each schedule in a space reserved for it. See Table 1 for an example of such a table. In this example, we have 33 jobs to be scheduled on a single SMD machine. Note that similar tables can be constructed for other jobs and resources since they are independent in our current setting. In the more complex job shop scheduling problem, we would still have a single table for relating the jobs to each schedule. To have a manageable size example, we have created 8 schedules. Each job agent in this example fill the tardiness values for each of these 8 schedules. The table shows the (truly) reported tardiness values ($\mu_i^*(y)$) for each job under each schedule.

Using these reported values, the Groves-Clarke mechanism selects a schedule using the minimization version of the schedule selection function $y^*(\mu^*)$ (our goal is to minimize total disutility or total weighted tardiness). Thus, the selected schedule (y^*) is the last one

Table 1: An example table for the Groves-Clarke mechanism for 8 schedules of 33 jobs

	Schedules ($y \in Y$)										
Job (i)	1	2	3	4	5	6	7	8	y^*_{-i}	$-\delta^*_i$	$\pi_i - \delta^*_i$
1	36	4	4	4	4	0	0	0	1	20	20
2	7	6	11	9	9	0	0	0	8	0	0
3	4	11	0	0	0	21	0	4	8	0	4
4	0	14	0	0	0	0	15	0	7	9	9
5	0	0	0	0	0	0	0	22	8	0	22
6	29	0	0	8	8	0	0	0	1	13	13
7	8	19	24	10	10	0	0	0	8	0	0
8	0	16	0	0	65	0	17	0	7	11	11
9	16	0	5	12	12	0	0	0	1	0	0
10	0	2	7	14	14	0	0	0	8	0	0
11	0	25	30	14	18	0	0	0	8	0	0
12	0	27	32	16	15	0	0	0	8	0	0
13	0	6	11	16	16	0	0	0	8	0	0
14	0	6	11	16	16	0	0	23	8	0	23
15	0	14	0	0	0	21	15	0	7	9	9
16	0	7	12	17	17	0	0	24	8	0	24
17	26	21	1	21	0	63	68	67	8	0	67
18	0	32	37	17	16	0	0	0	8	0	0
19	29	37	8	8	8	0	0	0	1	13	13
20	0	36	38	18	17	0	0	0	8	0	0
21	11	39	41	13	13	0	0	0	8	0	0
22	0	17	19	20	20	20	0	0	8	0	0
23	0	0	0	0	0	0	5	5	8	0	5
24	0	18	20	21	21	21	0	24	8	0	24
25	0	0	0	0	0	0	4	4	8	0	4
26	0	18	0	0	67	0	19	0	7	13	13
27	0	20	0	0	69	0	21	0	7	15	15
28	14	48	10	16	16	0	0	0	8	0	0
29	0	21	1	21	70	0	0	0	8	0	0
30	1	25	25	26	26	26	0	0	6	5	5
31	8	29	29	30	30	30	27	0	7	21	21
32	0	52	52	19	18	0	0	0	8	0	0
33	12	0	61	11	54	4	0	12	8	0	12
Total	201	570	489	377	649	206	191	185		129	314

(schedule 8) in the table with a total tardiness of 185. The mechanism next calculates the transfers from agents. Recall that, to calculate Clarke transfer for agent i, we need to find what the other agents would select without i's messages, $y^*_{-i}(\mu_{-i})$. The column titled y^*_{-i} lists the schedules that would be selected if the corresponding agent did not report any values. Note that if an agent does not affect the others' selection (i.e., $y^*_{-i} = y^* = 8$) then his transfer is zero, otherwise he pays the increase that he makes on others' total tardiness. Consider, for example, job number 6. Without job 6's reported values, the others would select schedule 1 with total tardiness of 172 ($= \min_{y \in Y} \sum_{i' \neq 6} \mu_{i'}(y^*_{-6}(\mu_{-6}))$). The difference between the tardiness of schedule 8 and that of 1 can be seen due to the involvement of job 6. Thus, the agent for job 6 pays the difference: $185 - 172 = 13$. Note that even after paying this amount, the agent's total cost (tardiness + payment) with the selection of schedule 8 is 13 ($= 0 + 13$) which is still lower than what would be if the other agents collectively selected schedule 1 ($13 < 29$) without his participation. We can show that this is true for every job agent. The agents with negative transfers (1, 4, 6, 8, 15, 19, 26, 27, 30, and 31) are charged 129 units in total, i.e., they pay this amount to collectively select schedule 8 instead of another alternative. The last column in the table lists the agents' total costs after transfer. Finally, note that, although each agent individually has a positive benefit with respect to what the others would choose, it may be worse off with respect to another schedule. For example, agent 6 pays 13 (with total cost of 13) but there are in fact schedules with his tardiness values of 8 or even 0 (assuming these are without transfers, recall individual rationality conditions). In fact, some agents (including 6) are worse off even when their total costs are compared to the mean tardiness for which a schedule is randomly (with equal probability across schedules) selected from set Y.

4 Discussion

We now discuss several interesting issues related to incentive compatible schedule selection in light of the illustrative example.

4.1 Generation of Schedules

A direct revelation mechanism for distributed scheduling involves many inter-related issues in order for it to be implemented successfully: First of all, we need to address the question of *how* the schedules collected in set Y are generated. Obviously schedules in this set affect the quality of the final schedule selected in a direct revelation mechanism. It is also important since it affects the computational implication of the overall mechanism.

We first assume that each schedule in Y represents a feasible time slot allocation from the shared resources to the job agents, i.e., in each schedule, non-overlapping contiguous blocks of time slots from each machine are assigned to the job agents for their respective operation on that machine. We may interpret the set of the candidate schedules as a collection of Gantt charts. Thus, we can identify a schedule y in set Y by using job - machine - time slot assignment indicators such as $X_{ikt}(y) = 1$ if job i is assigned to time slot t on machine k in schedule y (or equivalently a starting time and a completion time for each operation). In this representation, a job agent does not need to know the full schedule (related to the others) to compare the schedules; knowing his own assignment is enough to compute his net utility. A job agent can compute the value of a schedule (profit that it can make or tardiness costs) by using the part of the schedule that is related to his job. If we collect all the time slots assigned to agent i in $X_i(y) \equiv \{X_{ikt}(y)\}_{k=1,\ldots,M,t=1,\ldots,T}$, then the job agent can compute his profit by using X_i:

$$\pi_i(y) = \pi_i(X_i(y))$$

We know that the number of schedules to be presented to the agents are limited by computational resources, such as memory and computer time. One possibility is to consider all feasible schedules that make up the original scheduling problem among the job agents. For example, if the problem is a classical job shop scheduling problem (JSP) then there are exponential number of schedules (all semi-active schedules for regular performance measures such as weighted tardiness) from which we can choose our candidate set of schedules. An important issue is then which schedules for the original scheduling problem should be put in the set of "candidate schedules" Y. An interesting but somewhat trivial result regarding the quality of the final schedule collectively selected is that if the set of candidate schedules Y consists of an optimal schedule of the original scheduling problem (say JSP where all feasible schedules are considered), then this schedule will definitely be selected through the mechanism, i.e., if Y is guaranteed to have a certain performance guarantee on the total profit, then the schedule selected using the schedule selection function has the same performance guarantee. This is summarized in the following proposition.

Proposition 4 *The Groves-Clarke mechanism is guaranteed to choose a global optimal* *schedule of the original problem and achieve the optimal joint profit if the set Y consists of an optimal schedule, i.e.,* $\sum_i \pi_i(y^*) = v(JSP)$, *where* $v(JSP)$ *denotes the value of a (global) optimal schedule.*

Then the remaining question is how do we select a subset of good schedules to be placed in set Y? What if we are interested in including the optimal schedule in this set? We list several alternatives for the construction of the candidate schedule set as follows:

- *Historical information*: In most practical environments, schedule selection is not a one time process, but an exercise to be repeated time and again. If this is the case, an important source of information for the generation of quality schedules is the performance of those selected in the past. Schedules that have been implemented in the past, with their realized performance measures, may help identify promising schedules for the current selection process. Of course the usefulness of the historic information depends on the variability of problem instances over time. Suppose that every week a similar set of jobs (similar routing, processing, and release and shipment requirements) compete for shared machines. It is likely that the past performance might reveal patterns in the schedules that can be used to generate candidate schedules.

- *Preprocessing*: We can view the schedule selection game as a stage in a distributed scheduling procedure preceeded by a preprocessing or screening stage. In this view, a set of high-quality schedules are collected in set Y in the preprocessing stage. The two methods outlined in our previous works [14, 15] will be useful to isolate good quality candidate schedules. These are in fact what we used to generate the candidate schedules in the example above. The feasibility restoration routine applied to a Lagrangean schedule as outlined in [14] will produce a number of schedules which can be placed in set Y. In [15], we showed that one of a small number of precedence (or ranking) constraints must hold in an optimal solution. A preprocessing stage will involve fixing one out of these alternative precedence constraints among the jobs and generating different schedules each of which satisfies an alternative ranking. Since, we know that this preprocessing does not eliminate the optimal schedule, the schedule selection mechanism will be able to select the optimal schedule for the original problem. The only prohibitive element of this type of preprocessing is the number of the candidate schedules allowed, that is, the size of Y. This may be controlled to a certain degree by effectively excluding some of the rankings from consideration although a fully optimal schedule cannot be guaranteed in this case.

- Another rather different alternative is to have the agents propose full feasible schedules that are supposed to improve a default schedule that will be implemented in case any better schedule is not proposed for a certain time period. One such implementation for train scheduling can be found in [2] where the agents are provided with the incentives, i.e., rebates from the global performance improvement, to propose high-quality schedules. Note that, in this case, although the generation of schedules are carried out along with the online selection of a schedule, the private information assumption is rather destroyed since each agent is assumed to have full information to propose a fully feasible schedule.

We should point out that these methods are not mutually exclusive, i.e., they can be combined in a single process to make the candidate schedule set smaller and more effective in finding an optimal schedule.

4.2 Aligning Original Performance and Transfers

As we have outlined above, an issue is to align the "normal" performance measure (e.g., weighted tardiness) with the unit of transfers. This is a prerequisite for a sensible "utility" definition of the agents, i.e., after transfer (net) performance, $u_i(y) = \pi_i(y) + \delta_i$. The agents should take into account the effects of transfers as well as the initial performance. In the mechanism design literature, both the initial performance and the transfers are interpreted and expressed in monetary terms, which is what we did in our discussion. However, when the two are not in the same units, we need a normalizing function to convert the initial performance such that the overall utility makes sense for the agents. For example, one can use a function that calculates actual tardiness costs including any late charges or express shipments costs. Thus, the "profit" of agent i is defined as

$$\pi_i(y) = f(T_i(y)),$$

where T_i is tardiness of job i in schedule y and f is a conversion function.

Furthermore, the transfers should be explicitly processed through actual transactions so that the transfers can affect overall utility of the agents. Considering that actual monetary transactions can be difficult to implement in a intra-plant competition, we may imitate the environment by distributing some type of "currency" to the agents and allow them to spend for better scheduling selections. Another alternative is to align the incentives of the agents with their overall performance using an "evaluation measure". In this way, the compensation to an agent will directly depend on the the evaluation measure, which is the net utility of the agent after transfers. Note that in the latter interpretation, we assume the existence of a controller or a "center" that rewards or punishes the agents depending on their realized evaluation measures.

4.3 Coordinated Scheduling with a Central Authority

We have presented the mechanism in the previous sections as a collective decision making process without a center. However, it may be modified to accommodate a central authority. For example, if we assume that (1) the center is responsible for generating the schedules in set Y, (2) selects a mechanism with a message space and an outcome function that chooses a schedule and arranges the transfers, (3) collects money from the agents with negative δ_i and pays to the agents with positive δ_i, then there are only technical differences which are not critical for our analysis.

However, if we consider a central agency who makes decisions that may influence the agents' decisions and who has authority over the agents' actions, the issues differ significantly. The center here may allocate resources (e.g., time windows on machines assigned to job agents) that limit the decisions left to the local agents, and he/she may choose to reward or punish the agents depending on their actual performance (note that in the extreme case, the center's decisions may induce exactly one solution in the agent's feasible space). The main difficulty for an "authoritarian center" while making these decisions is that he/she may not have perfect information about local constraints and preferences so that sound decisions may not be possible, or alternatively, it may be essential to elicit private information from the agents in order to maximize the center's utility. Since the overall performance is the result of decisions from both the center and the agents, the nature of the coordination and the balance of the authority present a significantly different design problem.

We note the following differences as compared to the centerless design:

- The main issue here is to elicit private information from the local agents so that (1) the center makes the best allocation of shared resources, and then (2) the agents make best local decisions that are aligned with the global objectives.

- The center might have different incentive schemes that are compatible with the overall objective. One common tool is an evaluation measure for the agents that has been used in other Groves mechanisms (see, e.g., [12]). When the compensation to an agent is tied to the evaluation measure of the agent, the incentive compatibility is assured. As expected, the form of the evaluation measure function is similar to the net utility function defined in previous sections.

- For this mechanism to be implemented, we need to assume that (1) the center can enforce resource assignments but he has no control on the local decisions of the agents, and (2) the center can measure the final performance of the agents at the end of the execution of a schedule (realized values of profits) and reward or punish them according to these realized performances. Hence, the center is assumed to have a role of both a decision maker and a *controller*. Thus, the mechanism is not only a direct revelation mechanism from a game theoretic point of view, but also a *control mechanism* from an organization control view [12].

- Possible steps for such mechanism may be as follows: (1) The agents are asked to report *maximum profit* that they can achieve for possible time window assignments. (2) The center makes time window assignment decisions based on the reported values by the agents. (3) The agents make the local decisions under the center's time window restrictions for their respective jobs, and they use their assigned time windows to process their jobs. (4) At the end of schedule execution, each agent's performance is measured, and rewards are calculated and distributed to the agents.

A more formal analysis of the mechanism design problem with a central agency is left for future study.

5 Conclusions

In this study, we present a simple and robust scheduling model that would enhance a distributed and possibly web-based decisions making process for resource scheduling problems. Although, we primarily focus on intra-plant scheduling problem by considering agents who share resources, an extension to the inter-plant general resource allocation problem is possible An analysis of this in the near future will most likely lead us to new and interesting results, especially in the context of "collaborative" and web-based supply chain management and coordination of multi-company activities. We primarily address the issue of incentive compatibility in an environment that we called the "schedule selection game." We showed that despite the difficulties involved in balancing the budget and ensuring individual rationality, an incentive compatible mechanism can be devised for distributed scheduling decision structure. We also discuss additional challenges and issues posed by the scheduling game, such as schedule generation, and transfer units. We also explore the application context of this scheduling game using an industry case and we illustrate the mechanism using a numerical example drawn from a real-life setting. Although we touched on topics such as alternative schedule generation, asynchronous scheduling, and other agent design structures than single-agent-single-job, in-depth coverage of these and similar topics are left for future study.

References

[1] K. J. Arrow. The property rights doctrine and demand revelation under incomplete information. In M. Boskin, editor, *Economics and Human Welfare*, pages 23–39. Academic Press, New York, NY, 1979.

[2] P. J. Brewer. Decentralized computation procurement and computational robustness in a smart market. Technical report, Department of Economics, Georgia State University, Atlanta, GA, 1997.

[3] P. J. Brewer and C. R. Plott. A binary conflict ascending price (BICAP) mechanism for the decentralized allocation of the right to use railroad tracks. *International Journal of Industrial Organization*, 14:857–886, 1996.

[4] E. Clarke. Multipart pricing of public goods. *Public Choice*, 11:17–33, 1971.

[5] C. J. Corbett, F. J. C. Debets, and L. N. Van Wassenhove. Decentralization of responsbility for site decontamination projects: A budget allocation approach. *European Journal of Operational Research*, 86:103–119, 1995.

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

[7] J. Green and J. J. Laffont. Characterization of strongly individually incentive compatible mechanisms for the revelation of preferences for public goods. *Econometrica*, 45:427–438, 1977.

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

[9] T. Groves. Incentive compatible control of decentralized organizations. In Y. C. Ho and S. K. Mitter, editors, *Directions in Large-Scale Systems: Many-person Optimization and Decentralized Control*, pages 149–185. Plenum Press, New York, NY, 1976.

[10] T. Groves. Information, incentives and the internalization of production externalities. In S. A. Lin, editor, *Theory and Measurement of Economic Externalities*, pages 65–83. Academic Press, New York, NY, 1976.

[11] T. Groves. Efficient collective choice when compensation is possible. *Review of Economic Studies*, 46:227–241, 1979.

[12] T. Groves and M. Loeb. Incentives in a divisionalized firm. *Management Science*, 25(3):221–231, 1979.

[13] V. Krishna and M. Perry. Efficient mechanism design. Technical report, Department of Economics, Pennsylvania State University, 1998.

[14] E. Kutanoglu and S. D. Wu. On combinatorial auction and Lagrangean relaxation for distributed resource scheduling. *IIE Transactions*, 31(9):813–826, 1999.

[15] E. Kutanoglu and S. D. Wu. Proactive equilibrium for distributed job shop scheduling. Technical Report 99T, Department of Industrial and Manufacturing Systems Engineering, Lehigh University, 1999.

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

[17] T. Sandholm. Distributed rational decision making. In G. Weiss, editor, *Multiagent Systems: A Modern Introduction to Distributed Artificial Intelligence*, pages 11–68. MIT Press, 1987.

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

A Proof of Proposition 2

The proof is a modified version of the argument used in [11]. To prove the proposition, it is sufficient to show that the agents reveal their true profits in a dominant strategy equilibrium, i.e., for every $i \in \mathcal{N}$, $\mu_i^*(y) = \pi_i(y) \ \forall \ y \in Y$, where μ^* are dominant strategy equilibrium messages. We first denote the schedule that would be selected if $\mu_i(y) = \pi_i(y) \ \forall \ y \in Y$ as $y^*(\mu_i = \pi_i, \mu_{-i})$ when the others' joint message is shown by μ_{-i}. From definition of schedule selection function, we recall that $y^*(\mu_i = \pi_i, \mu_{-i})$ maximizes $\pi_i(y) + \sum_{i' \neq i} \mu_{i'}(y)$ over all schedules $y \in Y$. Then,

$$\pi_i(y^*(\mu_i = \pi_i, \mu_{-i})) + \sum_{i' \neq i} \mu_{i'}(y^*(\mu_i = \pi_i, \mu_{-i})) \geq \pi_i(y) + \sum_{i' \neq i} \mu_{i'}(y) \ \forall \ y \in Y. \tag{6}$$

The inequality (6) holds for any y in Y including $y^*(\mu_i, \mu_{-i}) \in Y$ for any $\mu_i \in M_i$. Thus,

$$\pi_i(y^*(\mu_i = \pi_i, \mu_{-i})) + \sum_{i' \neq i} \mu_{i'}(y^*(\mu_i = \pi_i, \mu_{-i})) \geq$$
$$\pi_i(y^*(\mu_i, \mu_{-i})) + \sum_{i' \neq i} \mu_{i'}(y^*(\mu_i, \mu_{-i})) \;\forall\, \mu_i \in M_i. \tag{7}$$

To make the discussion simpler let us define

$$R_i(\mu_{-i}) = \sum_{i' \neq i} \mu_{i'}(y^*_{-i}(\mu_{-i})).$$

Note that the left hand side of inequality (7) is the utility of i from $y^*(\mu_i = \pi_i, \mu_{-i})$ plus the term $R_i(\mu_{-i})$:

$$\pi_i(y^*(\mu_i = \pi_i, \mu_{-i})) + \sum_{i' \neq i} \mu_{i'}(y^*(\mu_i = \pi_i, \mu_{-i})) = u_i(y^*(\mu_i = \pi_i, \mu_{-i}), \delta^*) + R_i(\mu_{-i})$$

Similarly, for the right hand side of the inequality,

$$\pi_i(y^*(\mu_i, \mu_{-i})) + \sum_{i' \neq i} \mu_{i'}(y^*(\mu_i, \mu_{-i})) = u_i(y^*(\mu_i, \mu_{-i}), \delta^*) + R_i(\mu_{-i})$$

Then,

$$u_i(y^*(\mu_i = \pi_i, \mu_{-i}), \delta^*) + R_i(\mu_{-i}) \geq u_i(y^*(\mu_i, \mu_{-i}), \delta^*) + R_i(\mu_{-i})$$

or, since R_i is independent of μ_i,

$$u_i(y^*(\mu_i = \pi_i, \mu_{-i}), \delta^*) \geq u_i(y^*(\mu_i, \mu_{-i}), \delta^*) \;\forall\, \mu_i \in M_i.$$

This holds for every i and for every μ_{-i} which proves the result. $\diamondsuit$

B Proof of Proposition 3

If the total transfer of the agents in I is zero, i.e., $\sum_{i \in I} \delta^*_i = 0$, then under Groves-Clarke mechanism, $\delta^*_i = 0 \;\forall\, i \in I$. It is trivial to show that the proposition holds if there is only one agent in I since this corresponds to the case of a non-pivotal agent whose participation affects neither the selection of an optimal schedule nor the total transfer. For more general cases where $|I| > 1$, denote the set of participating (and truthfully reporting) agents by I'. Let $y^*(\mu_{I'} = \pi_{I'})$ denote the schedule that is selected in this partial participation case. Since any agent $i \in I$ is non-pivotal under the full participation case, for every $i \in I$, we have

$$\sum_{i' \neq i} \pi_{i'}(y^*(\mu = \pi)) = \sum_{i' \neq i} \pi_{i'}(y^*(\mu_{-i} = \pi_{-i}))$$

or more precisely, $y^*(\mu = \pi) = y^*(\mu_{-i} = \pi_{-i}) \;\forall\, i \in I$. Moreover, for every $i \in I$,

$$\sum_{i' \neq i} \pi_{i'}(y^*(\mu = \pi)) = \sum_{i' \neq i} \pi_{i'}(y^*(\mu_{-i} = \pi_{-i})) \geq \sum_{i' \neq i} \pi_{i'}(y) \;\forall y \;\in Y,$$

or

$$\sum_{i' \in I'} \pi_{i'}(y^*(\mu_{I'} = \pi_{I'})) \geq \sum_{i' \in I'} \pi_{i'}(y) \; \forall y \; \in Y$$

This means that each agent individually does not affect the selection of an optimal schedule, i.e., without agent i, $i \in I$, the others in I' choose $y^*(\mu = \pi)$, and this in turn leads to the agents in I' collectively selecting $y^*(\mu_{I'} = \pi_{I'}) = y^*(\mu = \pi)$. Because the set I is overall nonpivotal, their total transfer is still 0, and the overall total transfer does not change. $\diamondsuit$