Simple sequencing problems with interdependent costs

Roland Hain [a], Manipushpak Mitra [b]

[a] *Studienstiftung des deutschen Volkes, Ahrstrasse 41, D-53175 Bonn, Germany*
[b] *Economic Research Unit, Indian Statistical Institute, 203, B.T. Road, Kolkata-700108, India*

Received 4 September 2002

Available online 10 December 2003

Abstract

In this paper we analyze simple sequencing problems under incomplete information and interdependent costs. We prove the necessity of concave cost function for implementability of such problems. Implementability means that one can achieve aggregate cost minimization in ex-post equilibrium. We also show that simple sequencing problems are implementable if and only if the mechanism is a 'generalized VCG mechanism.' We then consider first best implementability, that is implementability with budget balancing transfer. We prove that for implementable n agent simple sequencing problems, with polynomial cost function of order $(n-2)$ or less, one can achieve first best implementability. Finally, for the class of implementable simple sequencing problems with "sufficiently well behaved" cost function, this is the only first best class.

JEL classification: C44; C72; C78; D82

Keywords: Simple sequencing problems; Ex-post equilibrium; First best implementability

1. Introduction

In this paper we consider the problem of a planner who has to provide a facility to a finite set of agents. Alternatively, one can also consider the problem of a group of agents who wants to use a facility. An institute (like a college or a university) that has only one computer is one example.[1] Providing access of one runway facility to aeroplanes, for landing and takeoff, is another example. In these situations, each agent has one job to process using the facility. It takes different time periods for different agents to process their jobs. The facility can be used by only one agent at a time in a queue. Waiting is costly for all agents. In these sequencing situations, costs are interdependent since the cost of an agent depends not only on her own processing time but also on the processing time of all agents who precedes her in the queue. The planner's or group's objective is to select an efficient queue to minimize the aggregate cost, given that the number of agents, in need for the facility, is known. In these sequencing situations, it is quite natural to assume that the job processing time of the agents is private information. Moreover, if it is costly to monitor the agents' action or to verify the true job processing time, then there is an incentive problem. Agents, if asked, will announce their job processing time strategically. Given the incentive problem we ask the following question: Can the planner or group design a mechanism such that it is in the interest of the agents to reveal their true processing time? We refer to such a problem as *simple sequencing problem* (or SSP) *with interdependent costs*. We call this problem 'simple' because the form of the cost function is assumed to be known and identical for all agents.

Sequencing problems in the absence of interdependent cost were analyzed, among others, by Suijs (1996) and Mitra (2002). Discrete time sequencing problems (or queueing problems) with single facility and multiple facilities were analyzed by Mitra (2001) and Mitra (2003), respectively. In all these papers, the Vickrey–Clarke–Groves (VCG) mechanisms (due to Vickrey (1961), Clarke (1971) and Groves (1973)) uniquely solve the incentive problem in dominant strategies and guarantees efficiency of decision. In contrast to these papers, we assume that processing time is private information. As a result, an agent's utility from the facility depends directly on the processing time of other agents. Hence the VCG mechanism fails to solve the incentive problem. Given this impossibility, our first objective is to identify the class of mechanisms that implement an SSP in 'ex-post equilibrium.' Implementability in ex-post equilibrium means that one can find mechanisms that satisfy efficiency of decision and ex-post incentive compatible. A mechanism is ex-post incentive compatible if truth-telling is a best response to truth-telling by everyone else. The results in this direction are, to the most part, a straightforward extension of the existing literature. We first show that for implementability, it is necessary that the cost function is concave. We then show that an SSP is implementable if and only if the mechanism is a 'generalized VCG (or G-VCG) mechanism.'

Our *main objective* is to identify the subclass of first best implementable SSPs. An SSP is first best implementable if it is implementable with a transfer scheme that adds up to zero in all states. Thus, first best guarantees costless information extraction along with efficiency of decision. In this regard, our analysis is 'similar' to the analysis on first best with VCG mechanisms under different private values set up (see Green and Laffont (1979), Hurwicz and Walker (1990), Laffont and Maskin (1982), Liu and Tian (1999), Mitra (2001, 2002, 2003), Suijs (1996), Tian (1996) and Walker (1980)). We show that an implementable n agent SSP, with polynomial cost of order $(n - 2)$ or less, is first best implementable. Moreover, for SSPs with "sufficiently well behaved" cost function, this is the only first best implementable class.

It is important to note that the mechanism cannot be conditioned upon ex-post observable true processing costs of the agents since we are assuming that it is costly to monitor the agents' action or to verify the true job processing time. This can happen in reality. In the institute example, with one research facility (like a computer or a spectrometer or a telescope or a tunnel-microscope), the institute authorities might have to hire a person to monitor the actions or to verify the true processing cost of the agents. This cost can be large since, for certain facilities, the person hired for this purpose must have some knowledge specific to the facility. Moreover, if one can achieve first best implementability, by assuming that the mechanism cannot be conditioned upon ex-post observable information, then information extraction is costless. Clearly, first best implementability is 'superior' to the other option where the institute authority hires a person (by incurring a positive cost) either to monitor the actions or to verify the true processing cost. In general, for other sequencing examples with one facility first best implementability eliminates the need to either monitor actions of the agents or verify the true processing cost of the agents. This gives a strong justification to look for first best implementable simple sequencing problems.

The paper is organized in the following way. We conclude this section by relating our work to the existing literature. We then formalize SSPs in Section 2. In Section 3, we provide results on implementability of SSPs. In Section 4, we address the issue of first best implementability. We conclude our analysis in Section 5. All proofs are provided in Appendix A.

1.1. Related literature

Mechanism design problems with interdependent valuation have been analyzed in the context of auction (see Ausubel (1999), Dasgupta and Maskin (2000), Jéhiel and Moldovanu (2001) and Perry and Reny (2002)) and in the context of trading (see Fieseler et al. (2002) and Gresik (1991)). Bergemann and Välimäki (2002) address the general mechanism design problem with interdependent valuation by restricting signals to be one dimensional. Our analysis of implementability in ex-post equilibrium follows from the general results provided by Dasgupta and Maskin (2000), Jéhiel and Moldovanu (2001) and Bergemann and Välimäki (2002). However, we adopt the model of Bergemann and Välimäki (2002) for comparing our implementability results since it is most suited for this purpose.

In the partnership context, Fieseler et al. (2002) argue that, with their 'generalized Groves mechanism,' it is possible to apply expected externality payments (à la Arrow (1979) and d'Aspremont and Gérard-Varet (1979)) to achieve budget balancedness. They point out that, for the expected externality mechanism, truth-telling is a Bayesian but not an ex-post equilibrium. We impose more demanding conditions for first best implementability, namely ex-post incentive compatibility, efficiency of decision and budget balancedness. In the private value setup, Walker (1980) derived the necessary condition for budget balancedness of VCG transfers. We argue that Walker's condition is also necessary to balance a G-VCG transfer.

2. Simple sequencing problems

Let $\mathbf{N} \equiv \{1, 2, \ldots, n\}$ be the set of agents in need of the facility. Each agent $j \in \mathbf{N}$ takes $s_j \in (0, \bar{s}] \subseteq \mathbf{R}_{++}$ units of time to process her own job. Since the facility can be used by only one agent at a time, the agents will have to use the facility sequentially. By means of a permutation $\sigma = (\sigma_1, \ldots, \sigma_n)$ on $\mathbf{N}$, one can describe the position of each agent in the queue. Let $\mathbf{\Sigma}$ be the set of all possible permutations of $\mathbf{N}$. Therefore, a queue σ is a mapping from the set of agents $\mathbf{N}$ to $\mathbf{\Sigma}$. Let $\mathcal{P}_j(\sigma) = \{p \in \mathbf{N} - \{j\} \mid \sigma_p < \sigma_j\}$ be the predecessor set of j in σ and $\mathcal{P}_j^c(\sigma) = \{q \in \mathbf{N} - \{j\} \mid \sigma_j < \sigma_q\}$ be the successor set of j in σ. Let $F(S_j)$ measure the cost of agent $j \in \mathbf{N}$ if her job processing is complete at time point $S_j \in \mathbf{R}_{++}$. Therefore, the cost of an agent is a mapping $F : \mathbf{R}_{++} \to \mathbf{R}_+$. We assume that F is continuous and strictly increasing. Given a processing time vector $s = (s_1, \ldots, s_n)$ and a queue σ, the cost of agent $j \in \mathbf{N}$ is $F(S_j(\sigma; s))$, where $S_j(\sigma; s) = \sum_{p \in \mathcal{P}_j(\sigma)} s_p + s_j$. The utility of agent j, in state $s = (s_1, \ldots, s_n)$ and in the queue σ is $U_j(\sigma, t_j; s) = v_j - F(S_j(\sigma; s)) + t_j$ where v_j is the gross benefit, derived by agent j, from the facility and t_j is the transfer that she receives.

A queue $\sigma^* \in \mathbf{\Sigma}$, given s, is *efficient* if $\sigma^* \in \operatorname{argmin}_{\sigma \in \mathbf{\Sigma}} \sum_{j \in \mathbf{N}} F(S_j(\sigma; s))$. For a state $s = (s_1, \ldots, s_n)$, a queue σ^* is efficient if and only if for all pairs of agents $\{j, i\}$ such that $s_j < s_i$, the condition $\sigma_j^* < \sigma_i^*$ is satisfied. Note that there are states for which we can have more than one efficient queue. For example, let $n = 3$ and let $s = (s_1, s_2, s_3)$ be a state such that $s_3 < s_1 = s_2$. For s, $\sigma = (\sigma_1 = 2, \sigma_2 = 3, \sigma_3 = 1)$ and $\tilde{\sigma} = (\tilde{\sigma}_1 = 3, \tilde{\sigma}_2 = 2, \tilde{\sigma}_3 = 1)$ are both efficient. Thus, we have an efficiency correspondence. An *efficient rule* is a single valued selection from the efficiency correspondence. An efficient rule can always be selected from the efficiency correspondence by selecting an appropriate tie breaking rule. In this paper we will use the following tie breaking rule: if $s_i = s_j$ then $\sigma_i^* < \sigma_j^*$ if $i < j$.

In many real life situations agents have private information about their own job processing time. If the processing time vector $s = (s_1, \ldots, s_n)$ is private information, then the problem is to design a mechanism that will elicit this information truthfully. Using the Revelation Principle, we concentrate on direct mechanisms where each agent reports her own processing time (or type) and based on this report, the planner (or group) decides on the queue and the transfer vector for the set of agents. Formally, a direct mechanism $\mathbf{M}$ is a pair $\langle \sigma, \mathbf{t} \rangle$, where $\sigma : (0, \bar{s}]^n \to \mathbf{\Sigma}$ and $\mathbf{t} \equiv (t_1, \ldots, t_n) : (0, \bar{s}]^n \to \mathbf{R}^n$. We represent an SSP (*with interdependent cost*) by $\Gamma = \langle \mathbf{N}, F, (0, \bar{s}] \rangle$, where $\mathbf{N}$ is the number of agents, F is the common cost function and $(0, \bar{s}]$ is the interval of job processing time. Under $\mathbf{M} = \langle \sigma, \mathbf{t} \rangle$, given an announcement $\hat{s} = (\hat{s}_1, \ldots, \hat{s}_n) \in (0, \bar{s}]^n$ in state $s = (s_1, \ldots, s_n) \in (0, \bar{s}]^n$, the utility of agent j is given by $U_j(\sigma(\hat{s}), t_j(\hat{s}); s) = v_j - F(S_j(\sigma(\hat{s}); s)) + t_j(\hat{s})$. Note that the efficient queue is determined on the basis of the announced processing cost of all agents and the cost that each agent incurs depends on the actual cost of her own predecessors in the queue as well as her own processing cost. We conclude this section by defining implementability and first best implementability in ex-post equilibrium.

Definition 1. For an SSP $\Gamma = \langle \mathbf{N}, F, (0, \bar{s}] \rangle$, the efficient queue $\sigma^* : (0, \bar{s}]^n \to \mathbf{\Sigma}$ is *implementable in ex-post equilibrium*, if there exists a mechanism $\mathbf{M} = \langle \sigma^*, \mathbf{t} \rangle$ such that, for all $j \in \mathbf{N}$, for all $(s_j, s'_j) \in (0, \bar{s}]^2$ and for all processing time vectors $s_{-j} \in (0, \bar{s}]^{n-1}$, $U_j(\sigma^*(s), t_j(s); s) \geqslant U_j(\sigma^*(s'_j, s_{-j}), t_j(s'_j, s_{-j}); s)$.

Definition 2. For an SSP $\Gamma = \langle \mathbf{N}, F, (0, \bar{s}] \rangle$, the efficient queue $\sigma^* : (0, \bar{s}]^n \to \mathbf{\Sigma}$ is *first best implementable* (*in ex-post equilibrium*) if there exists a mechanism $\mathbf{M} = \langle \sigma^*, \mathbf{t} \rangle$ that implements it with a budget balancing transfer, i.e. $\sum_{j \in \mathbf{N}} t_j(s) = 0$ for all $s \in (0, \bar{s}]$.

3. Implementability in ex-post equilibrium

We start this section by providing a necessary condition for implementability of an SSP. Before doing so, we need two more definitions. We define the *first order incremental loss of amount h at x* as $\Delta(h)F(x) = F(x+h) - F(x)$. Secondly, a cost function F is *concave* if for all $x, y \in (0, n\bar{s}]$ and for all $\lambda \in [0,1]$, $F(\lambda x + (1-\lambda)y) \geqslant \lambda F(x) + (1-\lambda)F(y)$. One obvious property of a concave F is that for all $\{s_j, s'_j, s_i\} \in (0, \bar{s}]^3$ with $s_j < s'_j$ and for all $x \in (0, (n-2)\bar{s}]$, $\Delta(s_i)F(x + s'_j) \leqslant \Delta(s_i)F(x + s_j)$. With these definitions at hand, we can state a necessary condition for implementability:

Proposition 1. *For an SSP Γ, σ^* is implementable in ex-post equilibrium **only if** the cost function F is concave.*

Concavity of the cost function is the equivalent of the well-known "single-crossing property" (see Ausubel (1999), Perry and Reny (2002)) in our framework. To elicit private information, each agent j has to be compensated for her *aggregate incremental loss* in σ^*. The aggregate incremental loss of agent j, in σ^*, is the difference between her actual cost in σ^* and her own job processing time (that is $\Delta(\sum_{p \in \mathcal{P}_j(\sigma)} s_p)F(s_j) = F(S_j(\sigma; s)) - F(s_j)$). Since, for implementability, it is necessary that the aggregate incremental loss must be *non-increasing*, we need concavity of F. Following Bergemann and Välimäki (2002) we now define the G-VCG mechanism. Let $C_{-j}(\sigma^*(s); s') = \sum_{i \neq j} F(S_i(\sigma^*(s); s'))$ be the aggregate cost of all but agent j in state s' and in the queue $\sigma^*(s)$.

Definition 3. For any Γ, a mechanism $\mathbf{M} = \langle \sigma^*, \mathbf{t} \rangle$ is said to be a *G-VCG mechanism* if, for all $j \in \mathbf{N}$ and for all announcement vectors $\hat{s}_{-j} \in (0, \bar{s}]^{n-1}$, the transfer scheme $t_j : (0, s]^n \to \mathbf{R}$ satisfies the following conditions:

(i) For announcements $(\hat{s}_j, \hat{s}'_j) \in (0, \bar{s}]^2$ such that $\sigma^*_j(\hat{s}) = \sigma^*_j(\hat{s}'_j, \hat{s}_{-j})$, $t_j(\hat{s}'_j, \hat{s}_{-j}) = t_j(\hat{s})$.
(ii) For announcements $(\hat{s}_j, \hat{s}'_j) \in (0, \bar{s}]^2$ with $\sigma^*_j(\hat{s}) = \sigma^*_j(\hat{s}'_j, \hat{s}_{-j}) - 1$,

$$t_j(\hat{s}'_j, \hat{s}_{-j}) - t_j(\hat{s}) = C_{-j}(\sigma^*(\hat{s}); \tilde{s}_j, \hat{s}_{-j}) - C_{-j}(\sigma^*(\hat{s}'_j, \hat{s}_{-j}); \tilde{s}_j, \hat{s}_{-j}) \tag{1}$$

where $(\tilde{s}_j, \hat{s}_{-j})$ is the state for which both $\sigma^*(\hat{s})$ and $\sigma^*_j(\hat{s}'_j, \hat{s}_{-j})$ are efficient, that is for the state $(\tilde{s}_j, \hat{s}_{-j})$, $\sum_{i \in \mathbf{N}} F(S_i(\sigma^*(\hat{s}); \tilde{s}_j, \hat{s}_{-j})) = \sum_{i \in \mathbf{N}} F(S_i(\sigma^*(\hat{s}'_j, \hat{s}_{-j}); \tilde{s}_j, \hat{s}_{-j}))$.

Proposition 2. *For any Γ, a mechanism* $\mathbf{M} = \langle \sigma^*, \mathbf{t} \rangle$ *is a G-VCG mechanism* if and only if *for all announced processing time vectors* $\hat{s} \in (0, \bar{s}]^n$ *and for all* $j \in \mathbf{N}$,

$$t_j(\hat{s}) = \begin{cases} \sum_{p \in \mathcal{P}_j(\sigma^*(\hat{s}))} V_p(\hat{s}) + h_j(\hat{s}_{-j}) & \text{if } \sigma_j^*(\hat{s}) \neq 1, \\ h_j(\hat{s}_{-j}) & \text{if } \sigma_j^*(\hat{s}) = 1, \end{cases} \tag{2}$$

where $V_p(\hat{s}) = \Delta(\hat{s}_p) F(S_p(\sigma^*(\hat{s}); \hat{s}))$ *and where* h_j *is an arbitrary function of* $\hat{s}_{-j}$.

The importance of G-VCG mechanism is captured in the next proposition.

Proposition 3. *For all Γ, with strictly increasing and concave cost functions, σ^* is implementable. Moreover, for a Γ, σ^* is implementable* if and only if *the mechanism is a G-VCG mechanism.*

We try to provide the reason behind the implementability property of the G-VCG mechanism. Let $p(j)$ be the immediate predecessor of agent j in the queue σ, that is $p(j) = \{i \in \mathcal{P}_j(\sigma) \mid \sigma_i = \sigma_j - 1\}$. We define the *incremental loss* of agent j, in state s and in queue σ, as

$$\mathcal{V}_j(\sigma; s) = \begin{cases} \Delta(s_{p(j)}) F\Big(\sum_{q \in \mathcal{P}_{p(j)}(\sigma)} s_q + s_j \Big) & \text{if } \sigma_j(s) \neq 1, \\ 0 & \text{otherwise.} \end{cases}$$

The incremental loss of agent j is the additional cost that j incurs due to the presence of her immediate predecessor $p(j)$ in the queue σ. Consider a state s and the efficient queue $\sigma^*(s)$. For the state s, the incremental loss of agent j is $\mathcal{V}_j(\sigma^*(s); s) = \Delta(s_{p(j)}) F(\sum_{q \in \mathcal{P}_{p(j)}(\sigma^*(s))} s_q + s_j)$ if $\sigma_j^*(s) \neq 1$ and $s_{p(j)} \leqslant s_j$ and $\mathcal{V}_j(\sigma^*(s); s) = 0$ otherwise. Let $\widehat{S}_j^{\sigma_j^*(s)} = \{x \in (0, \bar{s}] \mid \sigma_j^*(x, s_{-j}) = \sigma_j^*(s)\}$. Consider all possible processing time $s'_j \in \widehat{S}_j^{\sigma_j^*(s)}$ of agent j and define her *maximum possible incremental loss* as $\mathcal{V}_j^*(\sigma^*(s); s) = \max_{s'_j \in \widehat{S}_j^{\sigma_j^*(s)}} \mathcal{V}_j(\sigma^*(s); s'_j, s_{-j})$. Due to concavity of the cost function, we get

$$\mathcal{V}_j^*(\sigma^*(s); s) = \begin{cases} V_{p(j)}(s) & \text{if } \sigma_j^*(s) \neq 1, \\ 0 & \text{otherwise.} \end{cases}$$

The maximum possible incremental loss of an agent $j \in \mathbf{N}$, in queue position $\sigma_j^*(s) \neq 1$, is the first order difference of amount $s_{p(j)}$ at time point $S_{p(j)}(\sigma(s); s)$ and it is 0 if $\sigma_j^*(s) = 1$. Why is the maximum possible incremental loss important? Consider a state $s \in (0, \bar{s}]^n$, an efficient queue $\sigma^*(s)$ and an agent j with processing time s_j. Assume that all agents have reported truthfully. Simplifying the aggregate incremental loss of agent j we get

$\Delta(\sum_{p\in\mathcal{P}_j(\sigma^*(s))} s_p)F(s_j) = \sum_{p\in\mathcal{P}_j(\sigma^*(s))} \Delta(s_p)F(S_p(\sigma^*(s);s)+s_j)$.[2] Using concavity of the cost function F we obtain

$$\sum_{p\in\mathcal{P}_j(\sigma^*(s))} \Delta(s_p)F\big(S_p\big(\sigma^*(s);s\big)+s_j\big) \leqslant \sum_{p\in\mathcal{P}_j(\sigma^*(s))} \Delta(s_p)F\big(S_p\big(\sigma^*(s);s\big)\big) = \sum_{p\in\mathcal{P}_j(\sigma^*(s))} V_p(s).$$

Thus, in state s, the aggregate incremental loss of an agent j, in an efficient queue $\sigma^*(s)$, is no more than the sum of maximum possible incremental loss of her own and that of all her predecessors in the queue. The transfer of an agent j in a G-VCG mechanism is the maximum possible incremental loss of agent j and that of all her predecessors in the queue, up to a constant. This transfer is enough to compensate an agent for her aggregate incremental loss in the queue and it guarantees ex-post incentive compatibility.

4. First best implementability

In this section we consider the prospects of first best implementability. Given Proposition 3, achieving first best implies identifying implementable SSPs for which one can find balanced G-VCG transfers. Consider any G-VCG mechanism for an implementable Γ. Note that for each $s \in (0,\bar{s}]^n$, if we add up the G-VCG transfer (2) for all agents and set it to zero, we get

$$\mathbf{V}(s) + \sum_{i\in\mathbf{N}} h_i(s_{-i}) = 0 \tag{3}$$

where, in state s, $\mathbf{V}(s) = \sum_{j\in\mathbf{N}}(n-\sigma_j^*(s))V_j(s)$ is the *weighted aggregate maximum possible incremental loss*. The general implication of (3) follows from the Cubical Array Lemma due to Walker (1980). Before stating the lemma we provide some more notations. Consider two states $s=(s_1,\ldots,s_n)$ and $s'=(s'_1,\ldots,s'_n)$. We define an index set $P\subseteq\mathbf{N}$ and a state $s(P)$ by replacing those s_j in s by the corresponding s'_j from s' for which $j\in P$. Formally, for a set $P\subseteq\mathbf{N}$, $s(P)=(s_1(P),\ldots,s_n(P))$ is a state such that

$$s_j(P) = \begin{cases} s_j & \text{if } j\notin P,\\ s'_j & \text{if } j\in P. \end{cases}$$

Lemma 1. *For any Γ with concave cost function F, σ^* is first best implementable* only if *for all $\{s,s'\}\in(0,\bar{s}]^n\times(0,\bar{s}]^n$,*

$$\sum_{P\subseteq\mathbf{N}} (-1)^{|P|}\mathbf{V}\big(s(P)\big) = 0 \tag{4}$$

where $|P|$ denotes the cardinality of the set P.

[2] Observe that we can always write $\Delta(b+c)F(a) = F(a+b+c)-F(a) = F(a+b+c)-F(a+b)+F(a+b)-F(a) = \Delta(c)F(a+b)+\Delta(b)F(a)$. By applying this relation repeatedly in the appropriate order we get the required simplification.

Walker (1980) proved Lemma 1 for VCG mechanisms where it is necessary that the total surplus, in each state, is $(n-1)$ type separable. Like the total surplus of a VCG mechanism, the weighted aggregate maximum possible incremental loss for a G-VCG mechanism (in condition (3)) must also be $(n-1)$ type separable for first best. For both VCG and G-VCG mechanisms, the transfer of an agent has two components. The first component is a function of the announcements of all agents and the second component is a function of the announcements of all but one agent. It is because of this similarity that the Cubical Array Lemma (due to Walker (1980)) is also applicable for G-VCG mechanisms. Lemma 1 will be used in proving our main theorem. Before stating it, we provide a relevant definition.

Definition 4. A function f is *sufficiently well behaved* if it has a power series representation in its entire open domain, that is there exists y_0 in its open domain such that the function f has the form $f(y)=\sum_{l=0}^{\infty} c_l(y-y_0)^l$.[3]

Theorem 1. *Consider a $\Gamma=\langle \mathbf{N}, F, (0,\bar{s}]\rangle$ with concave cost function F and with $|\mathbf{N}|=n$ agents.*

(1) If *for Γ, F is a polynomial of order $(n-2)$ or less then σ^* is first best implementable.*
(2) *Given a Γ with a sufficiently well-behaved F, σ^* is first best implementable* only if *F is a polynomial of order $(n-2)$ or less.*

Before we outline the proof of Theorem 1 we have to present another lemma. We define the *second order cross-partial difference* at x of amounts (a_1,a_2) as $\Delta(a_1)\Delta(a_2)F(x)=\Delta(a_1)[F(x+a_2)-F(x)]=F(x+a_1+a_2)-F(x+a_2)-F(x+a_1)-F(x)$. In general the mth order cross-partial difference at x of amount $(a_1,\dots,a_m)$ is given by $[\prod_{i=1}^{m}\Delta(a_i)]F(x)$. Observe that for a linear function $F^1(x)=b_0+b_1x$, the second order cross partial difference of amounts (a_1,a_2) at some point y is zero, that is $\Delta(a_1)\Delta(a_2)F^1(y)=0$. Similarly, for a polynomial function F^2 of order two (that is for $F^2(x)=b_0+b_1x+b_2x^2$), it is easy to verify that $\Delta(a_1)\Delta(a_2)\Delta(a_3)F^2(y)=0$. The next lemma is a generalization of this idea.

Lemma 2. *If F is a polynomial function of order $m(=0,1,\dots)$, then for any set of numbers $\{a_1,\dots,a_{m+1},x\}$, $[\prod_{r=1}^{m+1}\Delta(a_r)]F(x)=0$.*

Idea of the proof of Theorem 1. To prove the first part, we first construct a generalized VCG mechanism for a Γ having concave and polynomial cost function F of order $(n-2)$ or less. Consider the G-VCG mechanism $\mathbf{M}^*=\langle\sigma^*,\mathbf{t}^*\rangle$ where for all $j\in\mathbf{N}$ and for all s_{-j}, $h_j^*(s_{-j})=-\sum_{i\neq j} g_{ij}(s_{-j})$ where $g_{ij}(\cdot)$ is a function of s_{-j} and is defined as

$$g_{ij}(s_{-j})=\sum_{r=1}^{\sigma_i^*(s_{-j})}(-1)^{\sigma_i^*(s_{-j})-r}\left\{\frac{(\sigma_i^*(s_{-j})-r)!(n-\sigma_i^*(s_{-j})-1)!}{(n-r-1)!}\right\}z_{ir}(s_{-j})$$

[3] A function f is said to be *well behaved* if it is infinitely differentiable in its open domain. A sufficiently well behaved function is well behaved but the converse is not true.

where $z_{ir}(\cdot)$ is also a function of s_{-j} and is defined as

$$z_{ir}(s_{-j}) = \sum_{\mathcal{P}_{i,r-1}(\sigma^*(s_{-j}))\subset\mathcal{P}_i(\sigma^*(s_{-j}))} \Delta(s_i)F\biggl(\sum_{q\in\mathcal{P}_{i,r-1}(\sigma^*(s_{-j}))} s_q + s_i\biggr)$$

where $\mathcal{P}_{i,\alpha}(\sigma^*(s_{-j}))$ is an α-element subset of $\mathcal{P}_i(\sigma^*(s_{-j}))$. We then show, using Lemma 2, that the G-VCG mechanism $\mathbf{M}^* = \langle\sigma^*, \mathbf{t}^*\rangle$ is budget balancing.

The second part of the theorem is proved in two steps. The first step will be to construct a pair of states and then apply condition (4) in Lemma 1 to get a general necessary condition. Consider any Γ for which σ^* is implementable. Consider two states s and s', both belonging to $(0, \bar{s}]^n$, such that $s = (s_1 = x, s_2 = 2x, \ldots, s_n = nx)$ and $s' = (s'_1 = nx, s'_2 = x, \ldots, s' = (n-1)x)$. Applying Lemma 1 we get the *general necessary condition* for first best implementability which is given by

$$\Delta^{n-1}(x)F\bigl(w_1(n)x\bigr) = \Delta^{n-1}(x)F\bigl(w_2(n)x\bigr) + \Delta^{n-1}(x)F\bigl(w_3(n)x\bigr) \tag{5}$$

where

$$w_1(n) = \frac{(n-1)n}{2}, \qquad w_2(n) = \frac{(n-1)(n+2)}{2}, \quad \text{and} \quad w_3(n) = \frac{n(n+1)}{2}.$$

The final step is to apply the fact that the cost function is sufficiently well behaved and derive the result using this general necessary condition.

Observe that for SSPs with two agents, first best implementability is impossible.[4] With three agents, all SSPs with linear cost function are first best implementable. For four agents, consider the class of SSPs $\Gamma^* = \langle \mathbf{N} = \{1, 2, 3, 4\}, F^*, (0, \bar{s}]\rangle$ where $F^*(x) = a_1x + a_2x^2$ for all $x \in (0, n\bar{s}]$ and only one of the following two conditions holds:

(1) $a_1 > 0$ and $a_2 = 0$, and
(2) $a_1 > 0$, $a_2 < 0$, $\bar{s} < \infty$, and $a_1 \geqslant -2na_2\bar{s}$.

It is easy to verify that this class is first best implementable. One can similarly obtain the class of first best implementable SSPs with more than four agents. Thus, there exists non-linear cost functions for which an implementable SSP is also first best implementable.

Remark 1. The requirement of polynomial cost of order $(n-2)$ for first best implementability of an SSP is 'similar' to the conditions obtained in Mitra (2001) for a queueing problem. This is because of the necessity of condition (4) for budget balancedness of both VCG and G-VCG mechanisms. In the sequencing problems considered in Mitra (2002), due to the presence of individual specific cost parameter, first best was possible only with linear cost. Since this individual specific cost parameter is absent for SSPs, first best can be achieved for some non-linear cost functions as well.

Why is a polynomial cost of order $(n-2)$ important for first best implementability? To answer this question we need to interpret condition (4) (i.e., $\sum_{P\subseteq\mathbf{N}}(-1)^{|P|}\mathbf{V}(s(P)) = 0$).

---

[4] This is obvious from condition (5). Note that sufficient well-behavedness of F is not required for this impossibility.

It states that the weighted total group externality must add up to zero while moving from any state $s = (s_1, \ldots, s_n)$ to any other state $s' = (s'_1, \ldots, s'_n)$ where the weights are (-1) for groups (P) with odd number of members and are 1 for groups (P) with even number of members. Polynomial cost of order $(n-2)$ satisfies a particular type of negative weighted group externality that results from condition (4). Consider two true states $s = (s_1, s_2 = a, \ldots, s_n = a) \in (0, \bar{s}]^n$ and $\hat{s} = (s_1, \hat{s}_2 = b, \ldots, \hat{s}_n = b) \in (0, \bar{s}]^n$ where $b < s_1 < a$. In state s, $\sigma_1^*(s) = 1$ and the cost of agent 1 is $F(s_1)$ and in state $\hat{s}$, $\sigma_1^*(\hat{s}) = n$ and her cost is $F((n-1)b + s_1)$. Starting from the state s, consider a state $s(P)$ where actual processing time of any $P \subseteq (\mathbf{N} - \{1\})$ agents changes from a to b. While moving from state s to state $s(P)$, the queue position of agent 1 changes from $\sigma_1^*(s) = 1$ to $\sigma_1^*(s(P)) = |P| + 1$ and hence her cost changes from $F(s_1)$ to $F(|P|b + s_1)$.[5] This increase in agent 1's cost is due to the negative externality imposed by agents of the set P. Since we can select a group of size $|P|$ from the set $\mathbf{N} - \{1\}$ in $\binom{n-1}{|P|}$ ways, $\binom{n-1}{|P|}F(|P|b + s_1)$ is the total cost that can result for agent 1 if we consider negative externality, imposed on her, by all possible groups from the set $\mathbf{N} - \{1\}$ of size $|P|$. Therefore, $\sum_{P \subseteq \mathbf{N}-\{1\}} (-1)^{|P|} \binom{n-1}{|P|} F(|P|b + s_1)$ is *the weighted aggregate negative group externality*, that can be imposed on agent 1 by all possible groups of different sizes (from the set $\mathbf{N} - \{1\}$), while moving from state s to $\hat{s}$. Here the weights are 1 if the group size is even and are -1 if the group size is odd. If the cost function is a polynomial of order $(n-2)$ then this weighted negative group externality is zero, that is $\sum_{P \subseteq \mathbf{N}-\{1\}} (-1)^{|P|} \binom{n-1}{|P|} F(|P|b + s_1) = \Delta^{n-1}(b)F(s_1) = [\prod_{i \neq 1} \Delta(\hat{s}_i)]F(s_1) = 0$.[6] Observe that this group externality condition guarantees that the general necessary condition (given by condition (12)) is satisfied. Using Lemma 2, it is easy to verify that, in general, for all $j \in \mathbf{N}$ and for all pairs of states $s' = (s'_j, s'_{-j})$ and $\hat{s}' = (s'_j, \hat{s}'_{-j})$, such that $\sigma_j^*(s') = 1$ and $\sigma_j^*(\hat{s}') = n$, we get $[\prod_{i \neq j} \Delta(\hat{s}'_i)]F(s'_j) = 0$ if the cost function is a polynomial of order $(n-2)$. Thus, *a polynomial of order $(n-2)$ guarantees that the weighted aggregate negative group externality that can be imposed on any agent j, by all other agents and with all possible groups, while moving from a state where agent j is first in the queue to a state where she is last in the queue, must add up to zero*. It is this group externality condition that guarantees first best for an SSP.

5. Conclusions

In this paper we have analyzed SSPs with interdependent costs. In the implementability context, we show the necessity of concave cost function. Given that the cost function is concave, we show that the class of G-VCG mechanism uniquely implements an SSP. We then address the question of first best implementability of an implementable SSP. This is tantamount to finding conditions on the cost function under which one can get balanced G-VCG transfers. In our main theorem we show the important role played by polynomial costs of order $(n-2)$ in first best implementing an SSP. We then provide a group externality

[5] Note that $s(P) = s$ if $P = \phi$ and $s(P) = \hat{s}$ if $P = \mathbf{N} - \{1\}$.

[6] Note that these weights (that is, 1 if group size is even and -1 if the group size is odd) is due to condition (4) in Lemma 1.

argument that captures the importance of polynomial costs of order $(n-2)$ for balancing a G-VCG transfer.

One important observation of this paper is that the Cubical Array Lemma, which is necessary for first best implementability of VCG mechanisms, is also necessary for achieving the same with G-VCG mechanisms in any implementable SSP. Even for the general class of decision problems, that was analyzed in Bergemann and Välimäki (2002), the G-VCG mechanism satisfies first best implementability only if it satisfies the Cubical Array Lemma.[7]

We admit that the necessity of concave cost functions for implementability and polynomial cost functions of order $(n-2)$ for first best implementability with sufficiently well behaved cost functions is quite restrictive as it rules out many interesting cost functions like exponential functions. However, this simple problem leaves many questions unanswered. How will the results differ if the cost functions are different for different agents? What happens if one considers discounted costs? What happens if there are multiple facilities? These questions can be taken up for future research.

Acknowledgments

The authors are grateful to Debashish Goswami, Eric Maskin, Georg Nöldeke, Arunava Sen and one anonymous referee for their invaluable advice and suggestions. Manipushpak Mitra thanks the seminar participants at the Indian Statistical Institute, University of Bonn, University of Warwick and University of Essex, and the seminar participants at the VIIth Spring Meeting of Young Economists 2002 and at the Sixth International Society for Social Choice and Welfare Meeting 2002. Financial support from the Deutsche Forschungsgemeinschaft Graduiertenkolleg 629 at the University of Bonn is also gratefully acknowledged. The authors are solely responsible for the remaining errors.

Appendix A. Proofs

Proof of Proposition 1. We first consider two states that differ by the type of agent $j \in \mathbf{N}$. We then apply the implementability conditions to get the result. We consider any five numbers (a, s_j, s_i, s'_j, b) all belonging to $(0, \bar{s}]$ such that $a \leqslant s_j < s_i < s'_j \leqslant b$. Using these numbers we construct the states $s = (s_j, s_{-j})$ and $s' = (s'_j, s_{-j})$, where $s_p = a$ for all $p \in P' \subseteq \mathbf{N} - \{j, i\}$ and $s_q = b$ for all $q \in S' = \mathbf{N} - P' - \{j, i\}$. From the construction and from the efficiency criterion, it follows that $\sigma^*_j(s) = |P'| + 1 < \sigma^*_i(s) = |P'| + 2$ and $\sigma^*_j(s') = |P'| + 2 > \sigma^*_i(s') = |P'| + 1$. Therefore, we are considering two states $s = (s_j, s_{-j})$ and $s' = (s'_j, s_{-j})$ such that agent i is the immediate successor of agent j in state

[7] This can easily be seen from the definition of the class of G-VCG mechanisms given by conditions (14) and (17) in Bergemann and Välimäki (2002).

s and is the immediate predecessor of agent j in state s'. Applying the implementability condition in states $s = (s_j, s_{-j})$ and $s' = (s'_j, s_{-j})$, for agent j, we get

$$U_j\big(\sigma^*(s), t_j(s); s\big) \geqslant U_j\big(\sigma^*(s'), t_j(s'); s\big) \quad \text{and}$$
$$U_j\big(\sigma^*(s'), t_j(s'); s'\big) \geqslant U_j\big(\sigma^*(s), t_j(s); s'\big).$$

Simplifying these two conditions we get that the difference $t_j(s'_j, s_{-j}) - t_j(s)$ must lie in $[\Delta(s_i)F(a|P'| + s'_j), \Delta(s_i)F(a|P'| + s_j)]$ where $a|P'| = \sum_{p \in \mathcal{P}_j(\sigma^*(s))} s_p$. Hence, it is necessary that $\Delta(s_i)F(a|P'| + s'_j) \leqslant \Delta(s_i)F(a|P'| + s_j)$. Observe that this inequality must be satisfied for all possible selection of five numbers (a, s_j, s_i, s'_j, b) all belonging to $(0, \bar{s}]$ such that $a \leqslant s_j < s_i < s'_j \leqslant b$ and for all possible selection of $P' \subseteq \mathbf{N} - \{i, j\}$ satisfying $P' \cup S' = \mathbf{N} - \{i, j\}$ and $P' \cap S' = \phi$. This implies concavity of F since $s_j < s'_j$. □

Proof of Proposition 2. To prove the necessity part of the proposition we derive the explicit form of the transfer satisfying conditions (i) and (ii) in the definition of the G-VCG transfer. We fix the announcement of all agents except agent j at $\hat{s}_{-j}$. Let $\hat{s}_j$ be the announcement of agent j such that she gets the first queue position, that is $\sigma^*_j(\hat{s}) = 1$. Using condition (i) in the definition of the G-VCG transfer, we fix $t_j(\hat{s}) = h_j(\hat{s}_{-j})$ for $\sigma^*_j(\hat{s}) = 1$ where $h_j(\hat{s}_{-j})$ is an arbitrary function of $\hat{s}_{-j}$. Note that from condition (i) it also follows that for all $s'_j \in S_j^{\sigma^*_j(\hat{s})}$, the transfer of agent j must remain unchanged at $h_j(\hat{s}_{-j})$. Now consider two states $\bar{s} = (s_j, \hat{s}_{-j})$ and $\bar{s}' = (s'_j, \hat{s}_{-j})$ such that $\sigma^*_j(\bar{s}) = \sigma^*_j(\bar{s}') - 1$ and $s_j \neq s'_j$. From efficiency of decision it follows that there exists an agent p such that $s_j \leqslant \hat{s}_p \leqslant s'_j$. From efficiency it also follows that at $\tilde{s}_j = \hat{s}_p$,

$$\sum_{i \in \mathbf{N}} F\big(S_i\big(\sigma^*(\bar{s}); \tilde{s}_j, \hat{s}_{-j}\big)\big) = \sum_{i \in \mathbf{N}} F\big(S_i\big(\sigma^*(\bar{s}'); \tilde{s}_j, \hat{s}_{-j}\big)\big).$$

Using these observations and simplifying (1) in the definition of the G-VCG mechanism, we get

$$t_j(\bar{s}') - t_j(\bar{s}) = \Delta(\hat{s}_p)F\big(S_p\big(\sigma^*(\bar{s}); \bar{s}\big)\big)\big(= V_p(\bar{s})\big). \tag{6}$$

Solving (6) recursively, by using $t_j(\hat{s}) = h_j(\hat{s}_{-j})$ for $\sigma^*_j(\hat{s}) = 1$, we get the transfer given by condition (2). The sufficiency part of the proposition is now obvious. □

Proof of Proposition 3. Observe that to prove Proposition 3, it is enough to prove the second statement in the proposition, by making use of the fact that the cost function is strictly increasing and concave. We start by proving the *necessity* of the proposition. Consider an SSP $\Gamma = \langle \mathbf{N}, F, (0, \bar{s}] \rangle$. Let $\mathbf{M} = \langle \sigma^*, \mathbf{t} \rangle$ be the mechanism that implements Γ. We assume (without loss of generality) that the implementable transfer is of the form

$$t_j(s) = \sum_{p \in \mathcal{P}_j(\sigma^*(s))} \Delta(s_p)F\big(S_p\big(\sigma^*(s); s\big)\big) + h_j(s).$$

To prove the necessity part of the proposition, we prove that for all $j \in \mathbf{N}$ and for all *true* $s_{-j} \in (0, \bar{s}]^{n-1}$, $h_j(s_j, s_{-j}) = h_j(s'_j, s_{-j})$ for all s_j and s'_j in $(0, \bar{s}]$.

Step 1. Consider first the case where s_j and s'_j are such that $\sigma^*(s_j, s_{-j}) = \sigma^*(s'_j, s_{-j})$. From the implementability requirement for agent $j \in \mathbf{N}$ in states $s = (s_j, s_{-j})$ and $s' = (s'_j, s_{-j})$, it follows that

$$U_j\big(\sigma^*(s), t_j(s); s\big) \geqslant U_j\big(\sigma^*(s'), t_j(s'); s\big) \quad \text{and}$$
$$U_j\big(\sigma^*(s'), t_j(s'); s'\big) \geqslant U_j\big(\sigma^*(s), t_j(s); s'\big).$$

Simplifying the inequalities using $\sigma^*(s_j, s_{-j}) = \sigma^*(s'_j, s_{-j})$, $\mathcal{P}_j(\sigma^*(s_j, s_{-j})) = \mathcal{P}_j(\sigma^*(s'_j, s_{-j}))$, and the general transfer $t_j(\cdot)$, specified above, we get $0 \leqslant h_j(s_j, s_{-j}) - h_j(s'_j, s_{-j}) \leqslant 0$. Therefore, if s_j and s'_j are such that $\sigma^*(s_j, s_{-j}) = \sigma^*(s'_j, s_{-j})$, then $h_j(s_j, s_{-j}) = h_j(s'_j, s_{-j})$.

Step 2. Now consider the case where s_j and s'_j are such that $\sigma^*(s_j, s_{-j}) \neq \sigma^*(s'_j, s_{-j})$, $|\sigma^*_j(s_j, s_{-j}) - \sigma^*_j(s'_j, s_{-j})| = 1$ and hence $||\mathcal{P}_j(\sigma^*(s_j, s_{-j}))| - |\mathcal{P}_j(\sigma^*(s'_j, s_{-j}))|| = 1$. We have two possible subcases

(i) $\mathcal{P}_j(\sigma^*(s'_j, s_{-j})) - \mathcal{P}_j(\sigma^*(s_j, s_{-j})) = \{q'\}$ where $s_j \leqslant s_{q'} \leqslant s'_j$ (with at least one strict inequality), and
(ii) $\mathcal{P}_j(\sigma^*(s_j, s_{-j})) - \mathcal{P}_j(\sigma^*(s'_j, s_{-j})) = \{p'\}$ where $s'_j \leqslant s_{p'} \leqslant s_j$ (with at least one strict inequality).

We first consider subcase (i). Applying the implementability requirement for agent $j \in \mathbf{N}$ and simplifying it, using the conditions in subcase (i), we get $h_j(s'_j, s_{-j}) - h_j(s_j, s_{-j}) \in [A(s'_j), A(s_j)]$ where the function

$$A(x) = \Delta(s_{q'})F\bigg(\sum_{p\in\mathcal{P}_j(\sigma^*(s_j,s_{-j}))} s_p + x\bigg) - \Delta(s_{q'})F\bigg(\sum_{p\in\mathcal{P}_j(\sigma^*(s_j,s_{-j}))} s_p + s_{q'}\bigg).$$

Note that $A(x)$ is continuous and non-increasing in $x \in [s_j, s'_j]$ due to concavity of F. Moreover, $A(s'_j) \leqslant 0$, $A(s_j) \geqslant 0$ and $A(s_{q'}) = 0$. For all $\bar{s}_j \in [s_j, s_{q'})$, $h_j(\bar{s}_j, s_{-j}) = h_j(s_j, s_{-j})$ because of $\sigma^*(s_j, s_{-j}) = \sigma^*(\bar{s}_j, s_{-j})$ and Step 1. Similarly, for all $\tilde{s}_j \in (s_{q'}, s'_j]$, $h_j(\tilde{s}_j, s_{-j}) = h_j(s'_j, s_{-j})$ since $\sigma^*(s'_j, s_{-j}) = \sigma^*(\tilde{s}_j, s_{-j})$. Choosing $\bar{\varepsilon} = \min_{x\in\{s_j, s_{j'}\}} |s_{q'} - x|$, we have shown that $\delta(\varepsilon) = h_j(s_{q'} - \varepsilon, s_{-j}) - h_j(s_{q'} + \varepsilon, s_{-j}) = \text{const}\ \forall\varepsilon \in (0, \bar{\varepsilon})$. By continuity of $A(x)$ and since $\delta(\varepsilon) \in [A(s_{q'} + \varepsilon), A(s_{q'} - \varepsilon)]$, we get the result that $\delta(\varepsilon) = 0\ \forall\varepsilon \in (0, \bar{\varepsilon})$. Thus, $h_j(s'_j, s_{-j}) = h_j(s_j, s_{-j})$. Subcase (ii) is analogous to subcase (i).

Steps 3...*n*. For $\dot{s}_j$ and $\dot{s}'_j$ such that $|\sigma^*_j(\dot{s}_j, s_{-j}) - \sigma^*_j(\dot{s}'_j, s_{-j})| = k \in \{2, \ldots, n-1\}$ we apply the argument $|\sigma^*_j(s_j, s_{-j}) - \sigma^*_j(s'_j, s_{-j})| = 1$ inductively to get the result.

We now prove the *sufficiency* part of the proposition. Let s_{-j} be the true processing time of all agents except j. We define the benefit of agent $j \in \mathbf{N}$, when she reports s'_j, given her true type s_j as $B(s'_j, s_j)$ which is given by $B(s'_j, s_j) = U_j(\sigma^*(s'), t_j(s'); s) - U_j(\sigma^*(s), t_j(s); s)$. Here $s = (s_j, s_{-j})$ and $s' = (s'_j, s_{-j})$. To prove the proposition we will prove that for all $s'_j \in (0, \bar{s}]$ and for all $s_j \in (0, \bar{s}]$, $B(s'_j, s_j) \leqslant 0$. There are two possible subcases:

(a) $\mathcal{P}_j(\sigma^*(s')) \subset \mathcal{P}_j(\sigma^*(s))$, and
(b) $\mathcal{P}_j(\sigma^*(s)) \subseteq \mathcal{P}_j(\sigma^*(s'))$.

For subcase (a), define $\overline{P_j} = \mathcal{P}_j(\sigma^*(s)) - \mathcal{P}_j(\sigma^*(s'))$. Here we get $\sigma_j^*(s) > \sigma_j^*(s')$ and the benefit of agent j is

$$B(s'_j, s_j) = \Delta\biggl(\sum_{q\in\overline{P_j}} s_q\biggr) F\biggl(\sum_{p\in\mathcal{P}_j(\sigma^*(s'))} s_p + s_j\biggr) - \sum_{q\in\overline{P_j}} \Delta(s_q) F\biggl(\sum_{r\in\mathcal{P}_q(\sigma^*(s))} s_r + s_q\biggr).$$

By repeatedly applying the relation $\Delta(h_1+h_2)F(x) = \Delta(h_1)F(x) + \Delta(h_2)F(x+h_1)$ we get

$$\Delta\left(\sum_{i=1}^n h_i\right) F(x) = \sum_{i=1}^n \Delta(h_i) F\left(x + \sum_{j=1}^i h_j\right).$$

Applying this relation on the first term of $B(s'_j, s_j)$ we get

$$\Delta\biggl(\sum_{q\in\overline{P_j}} s_q\biggr) F\biggl(\sum_{p\in\mathcal{P}_j(\sigma^*(s'))} s_p + s_j\biggr) = \sum_{q\in\overline{P_j}} \Delta(s_q) F\biggl(\sum_{r\in\mathcal{P}_q(\sigma^*(s))} s_r + s_j\biggr).$$

Thus, the benefit of agent j's from misreporting is given by

$$B(s'_j, s_j) = \sum_{q\in\overline{P_j}} \Delta(s_q)\biggl[F\biggl(\sum_{r\in\mathcal{P}_q(\sigma^*(s))} s_r + s_j\biggr) - F\biggl(\sum_{r\in\mathcal{P}_q(\sigma^*(s))} s_r + s_q\biggr)\biggr].$$

It is obvious that $B(s'_j, s_j) \leqslant 0$ since $s_q \leqslant s_j$ for all $q \in \overline{P_j}$ and F is concave. The proof of subcase (b) is analogous and thus omitted. □

Proof of Lemma 2. For notational simplicity let F^k represent any polynomial function of order $k(=0, 1, \ldots)$. Therefore, $F^k(x) = \sum_{i=0}^k b_i x^i$ for all x in the domain of F^k. It is quite easy to see that for any polynomial function F^k of order k we get the following:

(1) For all $\{\alpha, x\} \in \mathbf{R}^2$, $\Delta(\alpha)F^k(x) = F^k(x+\alpha) - F^k(x) = \widetilde{F}^{k-1}(x)$ where $\widetilde{F}^{k-1}$ is a polynomial function of order $(k-1)$.

We now apply an induction argument to derive the result. For $m = 0$, $F^0(x) = b_0$ and $\Delta(a_1)F^0(x) = b_0 - b_0 = 0$. Thus, Lemma 2 holds for $m = 0$. We assume that Lemma 2 holds for $m = m_0$, that is, for all $\{a_1, \ldots, a_{m_0+1}, x\}$, $[\prod_{r=1}^{m_0+1} \Delta(a_r)]F^{m_0}(x) = 0$ where

F^{m_0} is any polynomial function of order m_0. We will now have to show that Lemma 2 also holds for $m = m_0 + 1$. Observe that

$$\left[\prod_{r=1}^{m_0+2} \Delta(a_r)\right] F^{m_0+1}(x) = \left[\prod_{r=1}^{m_0+1} \Delta(a_r)\right] \Delta(a_{m_0+2}) F^{m_0+1}(x).$$

From (1) we get $\Delta(a_{m_0+2})F^{m_0+1}(x) = \widetilde{F}^{m_0}(x)$. Thus,

$$\left[\prod_{r=1}^{m_0+2} \Delta(a_r)\right] F^{m_0+1}(x) = \left[\prod_{r=1}^{m_0+1} \Delta(a_r)\right] \widetilde{F}^{m_0}(x).$$

Since the lemma is true for $m = m_0$ we get $[\prod_{r=1}^{m_0+1} \Delta(a_r)]\widetilde{F}^{m_0}(x) = 0$. Thus, $[\prod_{r=1}^{m_0+2} \Delta(a_r)]F^{m_0+1}(x) = 0$. □

Proof of Theorem 1. We first prove the first part of the theorem. To do that we construct a particular G-VCG mechanism for an SSP with polynomial cost function of order $(n-2)$ and show that the transfers add up to zero for all possible processing time vectors. For an implementable SSP with a polynomial cost of order $(n-2)$ or less, consider the G-VCG mechanism $\mathbf{M}^* = \langle \sigma^*, \mathbf{t}^* \rangle$ where for all $j \in \mathbf{N}$ and for all s_{-j},

$$h_j^*(s_{-j}) = -\sum_{i \neq j} g_{ij}(s_{-j}). \tag{7}$$

Here $g_{ij}(\cdot)$ is a function defined as

$$g_{ij}(s_{-j}) = \sum_{r=1}^{\sigma_i^*(s_{-j})} (-1)^{\sigma_i^*(s_{-j})-r} \left\{ \frac{(\sigma_i^*(s_{-j}) - r)!(n - \sigma_i^*(s_{-j}) - 1)!}{(n-r-1)!} \right\} z_{ir}(s_{-j}) \tag{8}$$

where

$$z_{ir}(s_{-j}) = \sum_{\mathcal{P}_{i,r-1}(\sigma^*(s_{-j})) \subset \mathcal{P}_i(\sigma^*(s_{-j}))} \Delta(s_i) F\left(\sum_{q \in \mathcal{P}_{i,r-1}(\sigma^*(s_{-j}))} s_q + s_i \right)$$

and $\mathcal{P}_{i,\alpha}(\sigma^*(s_{-j}))$ is an α-element subset of $\mathcal{P}_i(\sigma^*(s_{-j}))$.

Step 1. We first prove $\sum_{j \neq i} g_{ij}(s_{-j}) = (n - \sigma_i^*(s))\Delta(s_i)F(S_i(\sigma^*(s); s))$ for all $\sigma_i^*(s) \neq n$. Since the sum

$$\sum_{j \neq i} g_{ij}(s_{-j}) = \sum_{\substack{j \neq i \\ j \notin \mathcal{P}_i(\sigma^*(s_{-j}))}} g_{ij}(s_{-j}) + \sum_{j \in \mathcal{P}_i(\sigma^*(s_{-j}))} g_{ij}(s_{-j}),$$

we simplify each of these two sums in separate steps. We first consider the sum

$$\sum_{\substack{j \neq i \\ j \notin \mathcal{P}_i(\sigma^*(s_{-j}))}} g_{ij}(s_{-j}).$$

Observe that from the efficient rule we get $\sigma_i^*(s_{-j}) = \sigma_i^*(s)$, for all agents $j \notin \{\mathcal{P}_i(\sigma^*(s)) \cup \{i\}\}$. Also observe that each set $\mathcal{P}_{i,r-1}(\sigma^*(s))$ occurs $(n - \sigma_i^*(s))$ times in the sum

$$\sum_{\substack{j \neq i \\ j \notin \mathcal{P}_i(\sigma^*(s_{-j}))}} g_{ij}(s_{-j}).$$

Using these two observations, we get

$$\sum_{\substack{j \neq i \\ j \notin \mathcal{P}_i(\sigma^*(s_{-j}))}} g_{ij}(s_{-j}) = \sum_{r=1}^{\sigma_i^*(s)} (-1)^{\sigma_i^*(s)-r} \left(\frac{(\sigma_i^*(s) - r)!(n - \sigma_i^*(s))!}{(n-r-1)!} \right) L(r,s) \tag{9}$$

where

$$L(r,s) = \sum_{\mathcal{P}_{i,r-1}(\sigma^*(s)) \subseteq \mathcal{P}_i(\sigma^*(s))} \Delta(s_i) F\left(\sum_{q \in \mathcal{P}_{i,r-1}(\sigma^*(s))} s_q + s_i \right)$$

and $\mathcal{P}_{i,\alpha}(\sigma^*(s))$ is an α-element subset of $\mathcal{P}_i(\sigma^*(s))$. We now consider the other sum $\sum_{j \in \mathcal{P}_i(\sigma^*(s_{-j}))} g_{ij}(s_{-j})$. Observe first that from efficiency condition we get $\sigma_i^*(s_{-j}) = \sigma_i^*(s) - 1$ for all $j \in \mathcal{P}_i(\sigma^*(s))$. Secondly, observe that each set $\mathcal{P}_{i,r-1}(\sigma^*(s))$ appears $(\sigma_i^*(s) - r)$ times in $\sum_{j \in \mathcal{P}_i(\sigma^*(s_{-j}))} g_{ij}(s_{-j})$. Using these two observations we get

$$\sum_{j \in \mathcal{P}_i(\sigma^*(s_{-j}))} g_{ij}(s_{-j}) = \sum_{r=1}^{\sigma_i^*(s)-1} (-1)^{\sigma_i^*(s)-r-1} \left(\frac{(\sigma_i^*(s) - r)!(n - \sigma_i^*(s))!}{(n-r-1)!} \right) L(r,s). \tag{10}$$

By adding the sums given by (9) and (10) and then simplifying, using $(-1)^{\sigma_i^*(s)-r} + (-1)^{\sigma_i^*(s)-r-1} = 0$, we get

$$\sum_{j \neq i} g_{ij}(s_{-j}) = \big(n - \sigma_i^*(s)\big) \Delta(s_i) F\bigg(\sum_{j \in P_i(\sigma^*(s))} s_j + s_i \bigg). \tag{11}$$

Therefore, from condition (11) we get $\sum_{j \neq i} g_{ij}(s_{-j}) = (n - \sigma_i^*(s)) \Delta(s_i) F(S_i(\sigma^*(s); s))$ for all $i \in \mathbf{N}$ such that $\sigma_i^*(s) \neq n$.

Step 2. Now we consider $\sum_{j \neq i} g_{ij}(s_{-j})$ for agent i with $\sigma_i^*(s) = n$ and show that it is equal to zero. For any $j \neq i$ we get $\sigma_i^*(s_{-j}) = n - 1$ since $\sigma_i^*(s) = n$. Moreover, for any such $j \neq i$, $g_{ij}(s_{-j}) = \sum_{r=1}^{n-1} (-1)^{n-1-r} z_{ir}(s_{-j})$. Since the term $z_{ir}(s_{-j})$ is given by $z_{ir}(s_{-j}) = \sum_{\mathcal{P}_{i,r-1}(\sigma^*(s_{-j})) \subset P_i(\sigma^*(s_{-j}))} \Delta(s_i) F(\sum_{q \in \mathcal{P}_{ir}(\sigma^*(s_{-j}))} s_q + s_i)$, we get $g_{ij}(s_{-j}) = [\prod_{l \neq j} \Delta(s_l)] F(s_i)$. This step means that the term $g_{ij}(s_{-j})$ is equal to the $(n-1)$th order cross-partial difference of amount $\{s_l\}_{l \neq j}$ at s_i. Since F is a polynomial of order $(n-2)$, from Lemma 2 we get $g_{ij}(s_{-j}) = 0$. Therefore, for an agent i such that $\sigma_i^*(s) = n$, $\sum_{j \neq i} g_{ij}(s_{-j}) = 0$. Thus, we get $\sum_{j \neq i} g_{ij}(s_{-j}) = (n - \sigma_i^*(s)) \Delta(s_i) F(S_i(\sigma^*(s); s))$ for all $i \in \mathbf{N}$. Finally, we consider the sum $\sum_{j \in \mathbf{N}} h_j^*(s_{-j})$ and show that it is equal to $-\mathbf{V}(s)$. Since $\sum_{j \in \mathbf{N}} h_j^*(s_{-j}) = -\sum_{i \neq j} g_{ij}(s_{-j})$, we get

$$\sum_{j\in\mathbf{N}} h_j^*(s_{-j}) = -\sum_{j\in\mathbf{N}}\sum_{i\neq j} g_{ij}(s_{-j}) = -\sum_{i\in\mathbf{N}}\sum_{j\neq i} g_{ij}(s_{-j}), \quad \text{or}$$

$$\sum_{j\in\mathbf{N}} h_j^*(s_{-j}) = -\sum_{i\in\mathbf{N}}\big(n-\sigma_i^*(s)\big)\Delta(s_i)F\big(S_i\big(\sigma^*(s);s\big)\big), \quad \text{or}$$

$$\sum_{j\in\mathbf{N}} h_j^*(s_{-j}) = -\mathbf{V}(s).$$

The last step guarantees condition (3) for the G-VCG mechanism $\mathbf{M}^* = \langle\sigma^*, \mathbf{t}^*\rangle$.

We now prove the second part of the theorem. The first step will be to construct a pair of states and then apply condition (4) in Lemma 1 to get a general necessary condition. The final step will be to apply the fact that the cost function is sufficiently well behaved and derive the result using this general necessary condition.

Step 1. Consider any implementable SSP Γ. Consider two states s and s', both belonging to $(0,\bar{s}]^n$, such that $s = (s_1 = x, s_2 = 2x, \ldots, s_n = nx)$ and $s' = (s_1' = nx, s_2' = x, \ldots, s' = (n-1)x)$. For this pair $\{s, s'\}$, we consider the sum $\sum_{P\subseteq\mathbf{N}}(-1)^{|P|}\mathbf{V}(s(P))$. The construction of the pair $\{s, s'\}$ is such that $\sum_{P\subseteq\mathbf{N}}(-1)^{|P|}\mathbf{V}(s(P))$ is independent of all the virtual marginal surplus terms with weights $(n-\sigma_j^*(s(P))) \in \{2,3,\ldots,n-1\}$. Hence, $\sum_{P\subseteq\mathbf{N}}(-1)^{|P|}\mathbf{V}(s(P))$ includes all virtual marginal surplus terms with weights $(n-\sigma_j^*(s(P))) = 1$ for all $P\subseteq\mathbf{N}$. By collecting all these terms and simplifying it we get $\sum_{P\subseteq\mathbf{N}}(-1)^{|P|}\mathbf{V}(s(P)) = \sum_{k=0}^{n-2}(-1)^k\binom{n-2}{k}\{\Delta(nx-x)F(\alpha(k)x-x) - \Delta(nx)F(\alpha(k)x)\}$ where $\alpha(k) = (n-1)(n+2)/2 - k$. Simplifying this condition using the relation $\Delta(\alpha x)F(\beta x) = \Delta((\alpha-1)x)F((\beta+1)x) + \Delta(x)F(\beta x)$ recursively and then by substituting $\sum_{P\subseteq\mathbf{N}}(-1)^{|P|}\mathbf{V}(s(P)) = 0$ from condition (4) in Lemma 1, we get

$$\Delta^{n-1}(x)F\big(w_1(n)x\big) = \Delta^{n-1}(x)F\big(w_2(n)x\big) + \Delta^{n-1}(x)F\big(w_3(n)x\big) \tag{12}$$

where $w_1(n) = (n-1)n/2$, $w_2(n) = (n-1)(n+2)/2$, and $w_3(n) = n(n+1)/2$. Condition (12) is a *general necessary condition* for first best implementability of any implementable SSP.

Step 2. Using the restriction that the cost function F is sufficiently well behaved, we first try to simplify a term of the form $\Delta^{n-1}(x)F(wx)$. The reason for doing this is that all terms in the general necessary condition (12) are of this form. Observe that

$$\Delta^{n-1}(x)F(wx) = \sum_{k=1}^{n}(-1)^{k-1}\binom{n-1}{k-1}F\big((w+k-1)x\big)$$

where

$$\begin{aligned} F\big((w+k-1)x\big) &= \sum_{l=0}^{\infty} c_l\big((w+k-1)x - y_0\big)^l \\ &= \sum_{l=0}^{n-2} c_l\big((w+k-1)x - y_0\big)^l + \sum_{l=n-2}^{\infty} c_l\big((w+k-1)x - y_0\big)^l. \end{aligned}$$

Therefore, we have re-written $\Delta^{n-1}(x)F(wx)$ as the sum of two polynomials. The first one is a polynomial of order $(n-2)$, that is

$$\sum_{k=1}^{n}(-1)^{k-1}\binom{n-1}{k-1}\left\{\sum_{l=0}^{n-2}c_l\big((w+k-1)x-y_0\big)^l\right\}$$

and the second sum is a polynomial with all higher-order terms, that is

$$\sum_{k=1}^{n}(-1)^{k-1}\binom{n-1}{k-1}\left\{\sum_{l=n-1}^{\infty}c_l\big((w+k-1)x-y_0\big)^l\right\}.$$

We first consider the former and show that it is equal to zero. By substituting $d(wx)=wx-y_0$ and by writing $((w+k-1)x-y_0)^l$ as $(d(wx)+(k-1)x)^l$ and then taking it's binomial expansion, we get

$$\sum_{k=1}^{n}(-1)^{k-1}\binom{n-1}{k-1}\left\{\sum_{l=0}^{n-2}c_l\big((w+k-1)x-y_0\big)^l\right\}$$
$$=\sum_{l=0}^{n-2}c_l\sum_{m=0}^{l}\binom{l}{m}\big(d(wx)\big)^{l-m}x^m\gamma(m)$$

where $\gamma(m)=\sum_{k=1}^{n}(-1)^{k-1}\binom{n-1}{k-1}(k-1)^m$. From Euler's identity we know that $\gamma(m)=0$ for all integers $m\in\{1,\dots,n-2\}$.[8] Therefore, *the first polynomial of order $(n-2)$, that is $\sum_{k=1}^{n}(-1)^{k-1}\binom{n-1}{k-1}\{\sum_{l=0}^{n-2}c_l((w+k-1)x-y_0)^l\}=0$ for any set of real numbers $\{c_0,\dots,c_{n-2}\}$.*

Thus, $\Delta^{n-1}(x)F(wx)$ is equal to the other polynomial with all higher-order terms, i.e.

$$\Delta^{n-1}(x)F(wx)=\sum_{k=1}^{n}(-1)^{k-1}\binom{n-1}{k-1}\left\{\sum_{l=n-1}^{\infty}c_l\big((w+k-1)x-y_0\big)^l\right\}.$$

By writing $\alpha(w,m)=\sum_{k=1}^{n}(-1)^{k-1}\binom{n-1}{k-1}(w+k-1)^m$ after taking the binomial expansion of the term $((w+k-1)x-y_0)^l$ in the sum and then simplifying it we get the following expression:

$$\sum_{k=1}^{n}(-1)^{k-1}\binom{n-1}{k-1}\left\{\sum_{l=n-1}^{\infty}c_l\big((w+k-1)x-y_0\big)^l\right\}$$
$$=\sum_{l=n-1}^{\infty}c_l\sum_{m=0}^{l}\binom{l}{m}(-y_0)^{l-m}\alpha(w,m)x^m.$$

We now try to evaluate the value of $\alpha(w,m)$. By taking the binomial expansion of $(w+(k-1))^m$ we get $\alpha(w,m)=\sum_{m_0=0}^{m}\binom{m}{m_0}w^{m-m_0}\gamma(m_0)$. From Euler's identity we know that $\gamma(m_0)=0$ for all $m_0\leqslant n-2$. Hence, $\alpha(w,m)=\sum_{m_0=n-1}^{m}\binom{m}{m_0}w^{m-m_0}\gamma(m_0)$.

[8] Euler's identity: $\sum_{q=0}^{t}(-1)^q\binom{t}{q}q^r=0$ for all $0\leqslant r<t$.

Now we calculate the value of the term $\gamma(m_0) = \sum_{k=1}^{n}(-1)^{k-1}\binom{n-1}{k-1}(k-1)^{m_0}$ for $m_0 \geqslant n-1$. Expanding $(k-1)^{m_0}$, using Stirling number of the second kind, we get $\gamma(m_0) = (-1)^{n-1}(n-1)!S(m_0, n-1)$.[9] Hence, we have obtained $\alpha(w, m) = (-1)^{n-1}(n-1)!G(w, m)$ where $G(w, m) = \sum_{m_0=n-1}^{m}\binom{m}{m_0}w^{m-m_0}S(m_0, n-1)$. Therefore,

$$\Delta^{n-1}(x)F(wx) = (-1)^{n-1}(n-1)!\sum_{l=n-1}^{\infty}c_l\sum_{m=n-1}^{l}\binom{l}{m}(-y_0)^{l-m}G(w, m)x^m. \quad (13)$$

By substituting (13) into (12) and simplifying, using $(-1)^{n-1}(n-1)! \neq 0$, we get

$$\sum_{l=n-1}^{\infty}c_l\sum_{m=n-1}^{l}\binom{l}{m}(-y_0)^{l-m}\beta(m)x^m = 0 \quad (14)$$

where the term $\beta(m)$ is

$$\begin{aligned}\beta(m) &= G(w_2, m) + G(w_3, m) - G(w_1, m)\\ &= \sum_{m_0=n-1}^{m}\binom{m}{m_0}\left(w_2^{m-m_0} + w_3^{m-m_0} - w_1^{m-m_0}\right)S(m_0, n-1).\end{aligned}$$

Note that $\beta(m) > 0$ since $0 < w_1 < w_2 < w_3$, $m - m_0 \geqslant 0$ for all $m_0 = n-1, \ldots, m$ and since $S(m_0, n-1) \geqslant 1$ for all integers $m_0 \geqslant n-1$. Therefore, using these results we get $\sum_{l=n-1}^{\infty}c_l\sum_{m=n-1}^{l}\binom{l}{m}(-y_0)^{l-m}\beta(m)x^m = \sum_{r=n-1}^{\infty}A_r x^r = 0$ where each coefficient $A_r = \sum_{l=r}^{\infty}c_l\binom{l}{r}(-y_0)^{l-r}\beta(r)$. The equation $\sum_{r=n-1}^{\infty}A_r x^r = 0$ implies that $A_r = 0$ for all $r = n-1, n, \ldots, \infty$. Therefore, using $\beta(r) > 0$, we get $B_r\ (= A_r/\beta(r)) = \sum_{l=r}^{\infty}c_l\binom{l}{r}(-y_0)^{l-r} = 0$ for all $r = n-1, n, \ldots, \infty$. Using the identity $\binom{l}{r} + \binom{l}{r+1} = \binom{l+1}{r+1}$ and simplifying $D_r = B_r + (-y_0)B_{r+1} (= 0)$ we get $D_r = \sum_{l=r}^{\infty}c_l\binom{l+1}{r+1}(-y_0)^{l-r} = 0$ for all $r = n-1, n, \ldots, \infty$. Since $\binom{l+1}{r+1} = \frac{l+1}{r+1}\binom{l}{r}$, $r+1 \neq 0$ and $B_r = 0$, we get $\sum_{l=r}^{\infty}lc_l\binom{l}{r}(-y_0)^{l-r} = 0$ for all $r = n-1, n, \ldots, \infty$. Similarly, by considering $D'_r = D_r + (-y_0)D_{r+1} = 0$ and using $\sum_{l=r}^{\infty}lc_l\binom{l}{r}(-y_0)^{l-r} = 0$ and $B_r = 0$ for all $r = n-1, n, \ldots, \infty$, we get $\sum_{l=r}^{\infty}l^2c_l\binom{l}{r}(-y_0)^{l-r} = 0$ for all $r = n-1, n, \ldots, \infty$. By continuing

[9] A Stirling number of the second kind $S(m_0, q)$, is defined as the coefficient of $[x]_q = x(x-1)\ldots(x-q+1)$ in the expansion of x^{m_0}, that is, $x^{m_0} = \sum_{q=0}^{m_0}S(m_0, q)[x]_q$ for every real number x and, more importantly, for every natural number m_0. Stirling number of the second kind are such that $S(m_0, 1) = S(m_0, m_0) = 1$. Moreover, these numbers are unimodal, i.e. they satisfy one of the following formulae:

(1) $1 = S(m_0, 1) < S(m_0, 2) < \cdots < S(m_0, M(m_0)) > S(m_0, M(m_0) - 1) > \cdots > S(m_0, m_0) = 1$ or
(2) $1 = S(m_0, 1) < S(m_0, 2) < \cdots < S(m_0, M(m_0) + 1) = S(m_0, M(m_0)) > \cdots > S(m_0, m_0) = 1$,

and $M(m_0 + 1) = M(m_0)$ or $M(m_0 + 1) = M(m_0) + 1$ where $M(m_0) = \max\{q \mid S(m_0, q)$ is maximum; $1 \leqslant q \leqslant m_0\}$. For a better understanding see Tomescu (1985).

this way recursively, we get, for any $p = 0, 1, \ldots, \infty$, $\sum_{l=r}^{\infty} l^p c_l \binom{l}{r}(-y_0)^{l-r} = 0$ for all $r = n-1, n, \ldots, \infty$. Thus, given any $p = 0, 1, \ldots, \infty$, we also get

$$\sum_{l=r}^{\infty}(l-h)^p c_l \binom{l}{r}(-y_0)^{l-r} = 0 \tag{15}$$

for all $r = n-1, n, \ldots, \infty$ and for any h. Using Stirling number of the first kind, consider

$$E_r = \sum_{l=r}^{\infty} c_l \binom{l}{r}\left\{\sum_{p=0}^{l-r} s(l-r, p)(l-r)^p\right\}(-y_0)^{l-r},$$

for all $r = n-1, n, \ldots, \infty$.[10] From condition (15) it follows that $E_r = 0$ for all r, since E_r can be written as $E_r = \sum_{p=0}^{l-r} s(l-r, p)\{\sum_{l=r}^{\infty} c_l \binom{l}{r}(l-r)^p(-y_0)^{l-r}\}$ and the second sum is zero. Simplifying the sum in the original expression of E_r we get $E_r = \frac{1}{r!}\sum_{l=r}^{\infty} l! c_l(-y_0)^{l-r} = 0$ for all $r = n-1, n, \ldots, \infty$ since by applying the properties of Stirling number of the first kind we know that $\sum_{p=0}^{l-r} s(l-r, p)(l-r)^p = (l-r)!$. Thus, we get $T_r = \sum_{l=r}^{\infty} l! c_l(-y_0)^{l-r} = 0$ for all $r = n-1, n, \ldots, \infty$. Observe that $T_r = r! c_r + (-y_0)T_{r+1} = r! c_r$ since $T_{r+1} = 0$. Moreover, since $T_r = 0$ and $r! > 0$, we get $c_r = 0$ for all $r = n-1, n, \ldots, \infty$. Hence, the general necessary condition (12) holds, for a cost function F of the form $F(y) = \sum_{l=0}^{\infty} c_l(y-y_0)^l$, for any selection of $\{c_0, \ldots, c_{n-2}\}$ and only if $c_l = 0$ for all $l = n-1, n, \ldots, \infty$. □

References

Arrow, K., 1979. The property rights doctrine and demand revelation under incomplete information. In: Economics and Human Welfare. Academic Press.

Ausubel, L.M., 1999. A generalized Vickrey auction. Mimeo. Department of Economics, University of Maryland. September.

Bergemann, D., Välimäki, J., 2002. Information acquisition and efficient mechanism design. Econometrica 70, 1007–1033.

Clarke, E.H., 1971. Multi-part pricing of public goods. Public Choice 11, 17–33.

Dasgupta, P., Maskin, E., 2000. Efficient auctions. Quart. J. Econ. 115, 341–388.

d'Aspremont, C., Gérard-Varet, L.A., 1979. Incentives and incomplete information. J. Public Econ. 11, 25–45.

Fieseler, K., Kittsteiner, T., Moldovanu, B., 2002. Partnerships, lemons and efficient trade. J. Econ. Theory. In press.

Green, J., Laffont, J.J., 1979. Incentives in Public Decision Making. North-Holland, Amsterdam.

Gresik, T.A., 1991. Ex-ante incentive efficient trading mechanisms without the private valuation restriction. J. Econ. Theory 55, 41–63.

Groves, T., 1973. Incentives in teams. Econometrica 41, 617–631.

Hurwicz, L., Walker, M., 1990. On the generic non-optimality of dominant strategy allocation mechanisms: a general theorem that includes pure exchange economies. Econometrica 58, 683–704.

Jéhiel, P., Moldovanu, B., 2001. Efficient design with interdependent valuations. Econometrica 69, 1237–1259.

Laffont, J.J., Maskin, E., 1982. The theory of incentives: an overview. In: Hildenbrand, W. (Ed.), Advances in Economic Theory.

[10] A Stirling number of the first kind, $s(m, q)$, is the coefficient of x^q in the expansion of $[x]_m = x(x-1)\cdots(x-m+1)$, that is $[x]_m = \sum_{q=1}^{m} s(m, p)x^q$ (see Tomescu, 1985).

Liu, L., Tian, G., 1999. A characterization of the existence of optimal dominant strategy mechanisms. Rev. Econ. Design 4, 205–218.

Mitra, M., 2001. Mechanism design in queueing problems. Econ. Theory 17, 277–305.

Mitra, M., 2002. Achieving the first best in sequencing problems. Rev. Econ. Design 7, 75–91.

Mitra, M., 2003. Incomplete information and multiple machine queueing problems. Discussion papers ERU/2003-3. Indian Statistical Institute, Kolkata.

Perry, M., Reny, P.J., 2002. An efficient auction. Econometrica 70, 1199–1212.

Suijs, J., 1996. On incentive compatibility and budget balancedness in public decision making. Econ. Design 2, 193–209.

Tian, G., 1996. On the existence of optimal truth-dominant mechanisms. Econ. Lett. 53, 17–24.

Tomescu, I., 1985. Problems in Combinatorics and Graph Theory. Wiley–Interscience. Translated by Melter, R.A.

Vickrey, W., 1961. Counterspeculation, auctions and competitive sealed tenders. J. Finance 16, 8–37.

Walker, M., 1980. On the non-existence of dominant strategy mechanisms for making optimal public decisions. Econometrica 48, 1521–1540.