BONN ECON DISCUSSION PAPERS

Discussion Paper 20/2001

Simple Sequencing Problems with Interdependent Costs

by

Roland Hain, Manipushpak Mitra

July 2001

Bonn Graduate School of Economics
Department of Economics
University of Bonn
Adenauerallee 24 - 42
D-53113 Bonn

The Bonn Graduate School of Economics is
sponsored by the

Deutsche Post World Net

MAIL EXPRESS LOGISTICS FINANCE

SIMPLE SEQUENCING PROBLEMS WITH INTERDEPENDENT COSTS

Roland Hain & Manipushpak Mitra *

Department of Economics, University of Bonn,
D-53113 Bonn, Germany.
Email: Roland.Hain@wiwi.uni-bonn.de
m.mitra@wiwi.uni-bonn.de
Fax: +49-228-739493
JULY 2001

Abstract

In this paper we analyze sequencing situations under incomplete information where agents have interdependent costs. We first argue why Vickrey-Clarke-Groves (or VCG) mechanism fails to implement a simple sequencing problem in dominant strategies. Given this impossibility, we try to implement simple sequencing problems in ex-post equilibrium. We show that a simple sequencing problem is implementable if and only if the mechanism is a 'generalized VCG mechanism'. We then show that for implementable n agent simple sequencing problems, with polynomial cost function of order $(n-2)$ or less, one can achieve first best implementability. Moreover, for the class of simple sequencing problems with "sufficiently well behaved" cost function, this is the only class of first best implementable simple sequencing problems.

*The authors are grateful to Eric Maskin and Georg Nöldeke for their valuable suggestions. The authors would also like to thank Debashish Goswami and Arunava Sen for their helpful comments. Financial assistance from the German Research Foundation is also gratefully acknowledged.

1 Introduction

In this paper we study sequencing situation under incomplete information where agents have interdependent costs. In particular, we consider the problem of a planner who has to provide a facility to a finite group of agents. Each agent has one job to process using this 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. Therefore, the planner will have to order the agents in a queue. Waiting in the queue is costly for each agent. The planner's objective is to select an efficient queue to minimize the aggregate cost. Consider the situation where the planner knows how many agents require the facility but she does not know the job processing time of the agents. Moreover, if it is costly for the planner either 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 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 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. Thus, for a simple sequencing problem, we have mechanism design problem of a social planner under interdependent cost where the signals (or processing time of agents) are one dimensional.

Sequencing situations with interdependent cost can arise in many real life scenario. A central agent providing access to telephone, for local as well as long distance calls, to a set of individuals (in some locality) would be one example.[1] Another example would be a college (or any institute) that has

[1] A good evidence that can be sited in this context is one of Noa Jahan Begum who lives in the town of Bora, 20 miles from Bangladesh's capital Dhaka. She bought a cellular phone (by taking a loan from Gramin Bank) and began offering it for a fee to neighbours and other villagers who need to make calls. Sometimes there are 20 or 30 or more, sometimes even 50 or 60 people waiting to make calls (see remarks by William E. Kennard, Chairman, Federal Communications Commission to WIRELESS 98, Atlanta, Ga. February 23, 1998).

only one computer. If a set of individuals (students and teachers) want to use this computer to process their own data set then the college authorities will have to provide access to this computer sequentially. Providing access of one runway facility to aeroplanes, for landing and takeoff, is another example of sequencing. 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.

Sequencing problems in the absence of interdependent cost was analyzed, among others, by Suijs (1996) and Mitra (2001). In these papers the standard Vickrey-Clarke-Groves (or VCG) mechanisms (due to Vickrey (1961), Clarke (1971) and Groves (1973)) are the only class of mechanisms that lead to truthful revelation of private information in dominant strategies and efficiency of decision. In these problems, the cost parameter is private information to the agents and the job processing time, of each agent, is common knowledge. For a simple sequencing problem with interdependent costs, the standard VCG mechanisms fails to solve the incentive problem since the job processing time is private information. Hence, our first objective is to identify the class of mechanisms that implement a simple sequencing problem in ex-post equilibrium. By implementability in ex-post equilibrium we mean that one can find mechanisms that are efficient in terms of decision and ex-post incentive compatible. A mechanism is ex-post incentive compatible if truth-telling is a best response of an agent whenever others' are truthful. We identify the complete class of mechanisms that implements a simple sequencing problem in ex-post equilibrium. Our main and final objective is to identify the sub-class of first best implementable simple sequencing problems. A simple sequencing problem is first best implementable if there exists a mechanism that implements it in ex-post equilibrium with a transfer scheme that adds up to zero in all states. Our analysis is 'similar' to the analysis on first best implementability with VCG mechanisms under different private values set up (see Green and Laffont (1979), Laffont and Maskin (1982), Hurwicz and Walker (1990) and Walker (1980)).

We start our analysis on implementability of a simple sequencing problem by arguing why VCG mechanism fails to implement a simple sequencing problem in dominant strategies using arguments from Radner and Williams (1988). Given this impossibility, we try to implement simple sequencing problems in ex-post equilibrium. We show that a simple sequencing problem with interdependent costs is implementable if and only if the mechanism is a 'generalized VCG mechanism' as defined by Bergemann and Välimäki (2000). For a simple sequencing problem, a generalized VCG mechanism is such that each agent receives, as transfer, the sum of her own "maximum possible incremental loss" and that of all agents who precedes her in the queue, up to a constant. We then explore the possibility of first best implementability. We show that for implementable n agent simple sequencing problems, with polynomial cost function of order $(n-2)$ or less, one can achieve first best implementability. Moreover, for the class of simple sequencing problems with "sufficiently well behaved" cost function (that is, cost function with power series representation), this is the only class of 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 simple sequencing problems in section two. In section three we provide results on implementability of simple sequencing problems. In section four we address the issue of first best implementability. We conclude our analysis in section five. Most of the proofs are relegated in the appendix.

1.1 Related Literature

Mechanism design problems with interdependent valuation has been analyzed extensively in the context of auctions (see Ausubel (1999), Dasgupta and Maskin (2000), Jéhiel and Moldovanu (2001) and Perry and Reny (1998)). Trading problems with interdependent values was analyzed by Fieseler, Kittsteiner and Moldovanu (2001) and by Gresik (1991). In a general mechanism design setting, Bergemann and Välimäki (2000) address the issue of common (or interdependent) value by restricting signals to be one dimensional. Radner and Williams (1988) showed the existence problem of 'dominant mechanisms' when there exists a possibility of informational externality. By dominant mechanisms they mean mechanisms that satisfy dominant strategy incentive compatibility and efficiency of decision.

In the literature on mechanism design problem under interdependent valuations, finding mechanisms that implements a decision problem in ex-post equilibrium is not new. In the context of auctions with interdependent valuations and with one dimensional signals, the mechanisms provided in Ausubel (1999), Dasgupta and Maskin (2000), Jéhiel and Moldovanu (2001) and Perry and Reny (1998) are all ex-post incentive compatible. Ausubel (1999), by assuming valuation functions are know to the agents and to the auctioneer, define a 'generalized Vickrey auction' for multiple identical objects and show that truth-telling is an ex-post equilibrium. Ausubel's 'generalized Vickrey auction' is a generalization of the 'modified second price auction' of Maskin (1992). Dasgupta and Maskin (2000), by assuming that valuation functions are know to the agents but not to the auctioneer, achieve ex-post incentive compatibility via an indirect mechanism where agents use bids that depends on the valuations of other agents. Dasgupta and Maskin (2000) allow for objects to be non-identical. In the same informational structure but with identical objects, Perry and Reny (1998) achieve ex-post incentive compatibility in a two round bidding procedure. Jéhiel and Moldovanu (2001) show that, with multi-dimensional signals, ex-post incentive compatible direct mechanism do not exist in general. In the case of one-dimensional signals, they provide a direct mechanism that achieves ex-post incentive compatibility. In the trading context Fieseler, Kittsteiner and Moldovanu (2001) provide a 'generalized Groves mechanism' which satisfies ex-post incentive compatibility. The most related paper to our analysis on implementability in ex-post equilibrium is the one by Bergemann and Välimäki (2000). They consider a mechanism design setting where agents can acquire costly information, by receiving a noisy signal about the true state, before participating in the mechanism. What is important for our paper is their characterization results on implementability in ex-post equilibrium. Thus, our analysis on implementability of a simple sequencing problem in ex-post equilibrium can be considered as an application of the general characterization results of Bergemann and Välimäki (2000).

The main contribution of our paper is the characterization result on first best implementability of a simple sequencing problem in ex-post equilibrium. Analysis in this direction, in an interdependent value set up, is relatively scarce. In the context of auctions with interdependent values, budget balancedness is automatically satisfied as the seller is the residual claimant. In the partnership context, Fieseler, Kittsteiner and Moldovanu (2001) argue that, with their 'generalized Groves mechanism', it is possible to apply expected externality payments (à la Arrow (1979) and d'Aspremont Gérard-Varet (1979)) to achieve (ex-post) budget balancedness. They point out that, for the expected externality mechanism, truth-telling is a Bayesian but not an ex-post equilibrium. In this paper, we look for mechanisms that satisfy efficiency of decision, ex-post budget balancedness and ex-post incentive compatibility.

2 Simple Sequencing Problems

Let $\mathbf{N} \equiv \{1, 2, \ldots, n\}$ be the set of agents in need of the facility provided by the planner. 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 planner will have to provide the facility to the agents in a queue. 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 $\mathcal{S}_j(\sigma) = \{q \in \mathbf{N} - \{j\} \mid \sigma_j < \sigma_q\}$ 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* in S_j. 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* (1) for all pairs of agents $\{j, i\}$ such that $s_j < s_i$, $\sigma^*_j < \sigma^*_i$ and (2) for all pairs of agents $\{j, i\}$ such that $s_j = s_i$, either $\sigma^*_j < \sigma^*_i$ or $\sigma^*_j > \sigma^*_i$. For example, consider the case where $n = 3$ and a state $s = (s_1, s_2, s_3)$ such that $s_3 < s_1 = s_2$. For the state $s = (s_1, s_2, s_3)$, the queue $\sigma = (\sigma_1 = 2, \sigma_2 = 3, \sigma_3 = 1)$ and the queue $\tilde{\sigma} = (\tilde{\sigma}_1 = 3, \tilde{\sigma}_2 = 2, \tilde{\sigma}_3 = 1)$ are both efficient. Therefore, we have an efficiency correspondence. An *efficient rule* is a single valued selection from the efficiency correspondence. One can always select an efficient rule from the efficiency correspondence by selecting an appropriate tie breaking rule. In this paper we will consider the following tie breaking rule: if $s_i = s_j$ then $\sigma^*_i < \sigma^*_j$ if $i < j$.

It is natural to assume that agents have private information about their own job processing time.[2] If the processing time vector $s = (s_1, \ldots, s_n)$ is private information, the planner's problem is to design a *mechanism* that will elicit this information truthfully. In this paper, we restrict our attention to *direct mechanisms* where each agent reports her own processing time (or type) and based on this report, the planner 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 $\sigma : (0, \overline{s}]^n \to \mathbf{\Sigma}$ and $\mathbf{t} \equiv (t_1, \ldots, t_n) : (0, \overline{s}]^n \to \mathbf{R}^n$. We represent a *simple sequencing problem with interdependent cost* by $\Gamma = \langle \mathbf{N}, F, (0, \overline{s}] \rangle$, where $\mathbf{N}$ is the number of agents, F is the common cost func-

[2] A person in need for the computer (telephone) has private information about how much time she will take to process her data (to make the call). The airport authority has little knowledge about how long an aeroplane will occupy its runway. If the planner wants to provide the facility in an efficient way then she will have to know the processing time of the agents before starting to provide the facility. This means that the planner will have to rely on the agents' announced processing time. It may be costly for the planner to monitor the actions of the agents. Moreover, it may be costly to verify whether the agent is truthful or not. For example, the college authorities (the central agent providing the telephone facility) may find it difficult to punish a teacher (an influential individual of the locality) for misreporting. Similarly, the airport authorities may find it difficult to punish an airline for misreporting because the airlines company might have paid a substantial amount for the maintenance of the facilities at the runway. Therefore, with either monitoring cost or verification cost there is an incentive problem as agents have an incentive to report strategically.

tion and $(0, \overline{s}]$ is the interval of job processing time. Under $\mathbf{M} = \langle \sigma, \mathbf{t} \rangle$, given an announcement $\hat{s} = (\hat{s}_1, \ldots, \hat{s}_n) \in (0, \overline{s}]^n$ in state $s = (s_1, \ldots, s_n) \in (0, \overline{s}]^n$, 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 by the planner based on 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 implementable in ex-post equilibrium and first best implementability of a simple sequencing problem.

DEFINITION 2.1 A simple sequencing problem with interdependent cost $\Gamma = \langle \mathbf{N}, F, (0, \overline{s}] \rangle$, is *implementable in ex-post equilibrium*, if there exists an efficient queue $\sigma^* : (0, \overline{s}]^n \rightarrow \mathbf{\Sigma}$ and a mechanism $\mathbf{M} = \langle \sigma^*, \mathbf{t} \rangle$ such that, for all $j \in \mathbf{N}$, for all $(s_j, s'_j) \in (0, \overline{s}]^2$ and for all true processing time vectors $s_{-j} \in (0, \overline{s}]^{n-1}$, $U_j(\sigma^*(s), t_j(s); s) \geq U_j(\sigma^*(s'_j, s_{-j}), t_j(s'_j, s_{-j}); s)$.

DEFINITION 2.2 A simple sequencing problem Γ is said to be *first best implementable* (*in ex-post equilibrium*) if there exists a mechanism $\mathbf{M} = \langle \sigma^*, \mathbf{t} \rangle$ that implements it in ex-post equilibrium with a budget balancing transfer.

3 Implementability in Ex-post Equilibrium

We start this section by providing the reason for the failure of VCG mechanisms to achieve efficiency of decision and dominant strategy incentive compatibility. A VCG mechanism means that each agent pays the cost of all other agents in the queue, up to a constant. A VCG mechanism leads to efficiency of decision as well as dominant strategy incentive compatibility only if it satisfies the 'independence property' (see 'Independence Property III' in Radner and Williams (1988)). In a simple sequencing problem this 'independence property' would mean the following: consider two true states $s = (s_j, s_{-j})$ and $s' = (s'_j, s_{-j})$ that differ by the processing time (or type) of only one agent $j \in \mathbf{N}$.[3] Moreover, let the queue position of agent j remain unchanged in the two states, that is let $\sigma_j^*(s) = \sigma_j^*(s')$. 'Independence property' requires that in such a situation the aggregate cost of all but agent j must remain unchanged in both the states. Observe that if agent j is not served last, that is if $\sigma_j^*(s) = \sigma_j^*(s') \neq n$, then $\mathcal{S}_j(\sigma^*(s)) = \mathcal{S}_j(\sigma^*(s')) \neq \phi$. For all agents $q \in \mathcal{S}_j(\sigma^*(s)) = \mathcal{S}_j(\sigma^*(s'))$, the cost in states s and s' are not the same, that is $F(S_q(\sigma^*(s); s)) \neq F(S_q(\sigma^*(s'); s'))$, since $j \in \mathcal{P}_q(\sigma^*(s)) = \mathcal{P}_q(\sigma^*(s'))$ and $s_j \neq s'_j$. If $\sigma_j^*(s) = \sigma_j^*(s') = n$ then $F(S_i(\sigma^*(s); s)) = F(S_i(\sigma^*(s'); s'))$ since $\mathcal{S}_j(\sigma^*(s)) = \mathcal{S}_j(\sigma^*(s')) = \phi$. Hence 'independence property' *does not hold* unless agent j has the last queue position. Thus, the VCG mechanism fails to meet the desired objectives. Given this impossibility, we try to implement a simple sequencing problem in ex-post equilibrium.

We first define 'set convexity' of a simple sequencing problem which is a necessary condition for implementability in ex-post equilibrium (see Bergemann and Välimäki (2000)). Consider a simple sequencing problem Γ. Fix the true processing time of all but agent j at $s_{-j} \in (0, \bar{s}]^{n-1}$. Given s_{-j}, consider the set $\hat{S}_j^k = \{s_j \in (0, \bar{s}] \mid \sigma^*(s_j, s_{-j}) = \sigma = (\sigma_j = k, \sigma_{-j})\}$. Therefore, the set $\hat{S}_j^k$ is the set of types for agent j that gives her the kth queue position. It is important to note that given s_{-j}, efficiency of decision implies that for all types of agent j, that keeps her queue position unchanged, the queue position of the remaining set of agents also remains unchanged. This is due to the common cost function of each agent. Therefore, sets $\{\hat{S}_j^k\}_{k=1}^n$ forms a partition of $(0, \bar{s}]$ for agent j. The collection $\{\hat{S}_j^k\}_{k=1}^n$ satisfies *set convexity* if for all queue positions $k \in \{1, \ldots, n\}$, $s_j \in \hat{S}_j^k$ and $s'_j \in \hat{S}_j^k \Rightarrow \lambda s_j + (1-\lambda)s'_j \in \hat{S}_j^k$ for all $\lambda \in [0, 1]$. Thus, set convexity requires that given the types of all but agent j, the set of types of agent j, that keeps her efficient queue position unchanged, must be convex. Moreover, this must be true for all queue positions of agent j and for all agents $j \in \mathbf{N}$. It is quite easy to verify that all simple sequencing problems satisfy set

[3] Here s_{-j} is of the form $(s_1, \ldots, s_{j-1}, s_{j+1}, \ldots, s_n)$ and hence $s = (s_j, s_{-j}) = (s_1, \ldots, s_{j-1}, s_j, s_{j+1}, \ldots, s_n)$, $s' = (s'_j, s_{-j}) = (s_1, \ldots, s_{j-1}, s'_j, s_{j+1}, \ldots, s_n)$ and $s_j \neq s'_j$.

convexity.[4] Set convexity is not enough to implement a simple sequencing problem. The following proposition provides the other necessary condition for implementability of a simple sequencing problem. Before stating the Proposition, we define the *first order incremental loss of amount h at x* as $\Delta(h)F(x) = F(x+h) - F(x)$.

PROPOSITION 3.1 A simple sequencing problem Γ is implementable in ex-post equilibrium **only if** the cost function F is weakly concave.

The proof of this Proposition is provided in the Appendix. The requirement of weak concavity of the cost function is due to the nature of the incentive constraint in a simple sequencing problem. To meet the incentive constraint, the planner will have to compensate an agent j for her *aggregate incremental loss* in the queue σ. The aggregate incremental loss of agent j in the queue 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 this aggregate incremental loss must be *non-increasing*, we need weak concavity of the cost function.

We restrict our attention to the class of simple sequencing problems for which the cost function is weakly concave and derive our implementability result. To do that we define the *generalized VCG mechanisms*, following Bergemann and Välimäki (2000). 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 $\sigma^*(s)$. For a simple sequencing problem Γ, a mechanism $\mathbf{M} = \langle \sigma^*, \mathbf{t}\rangle$ is said to be a generalized VCG mechanism if, for all $j \in \mathbf{N}$ and for all announcement vectors $\hat{s}_{-j} \in (0, \bar{s}]^{n-1}$, the following two conditions are satisfied:
(i) For announcements $(\hat{s}_j, \hat{s}'_j) \in (0, \bar{s}] \times (0, \bar{s}]$ 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 $(\hat{s}_j, \hat{s}'_j) \in (0, \bar{s}] \times (0, \bar{s}]$ with $\sigma^*_j(\hat{s}) = \sigma^*_j(\hat{s}'_j, \hat{s}_{-j}) - 1$,

$$t_j(\hat{s}'_j, \hat{s}_{-j}) - t_j(\hat{s}) = C_{-j}(\sigma^*(\hat{s}); \tilde{s}_j, \hat{s}_{-j}) - C_{-j}(\sigma^*(\hat{s}'_j, \hat{s}_{-j}); \tilde{s}_j, \hat{s}_{-j}) \quad (3.1)$$

[4] Consider the true type of all but agent 1 to be s_{-1} such that $s_2 < s_3 < \ldots < s_n$. Given s_{-1}, from the efficient rule it follows that $\hat{S}_1^1 = (0, s_2]$, $\hat{S}_1^k = (s_k, s_{k+1}]$ for all $k \in \{2, \ldots n-1\}$ and $\hat{S}_1^n = (s_n, \bar{s}]$. Now one can easily verify that given s_{-1}, the collection $\{\hat{S}_1^k\}_{k=1}^n$ satisfies set convexity since each $\hat{S}_1^k$ is an interval. This argument can be easily be generalized to verify that a simple sequencing problem satisfies set convexity.

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 $\sum_{i \in \mathbf{N}} F(S_i(\sigma^*(\hat{s}); \tilde{s}_j, \hat{s}_{-j})) = \sum_{i \in \mathbf{N}} F(S_i(\sigma^*(\hat{s}'_j, \hat{s}_{-j}); \tilde{s}_j, \hat{s}_{-j}))$. Using conditions (i) and (ii), we derive the form of the class of generalized VCG mechanisms in the next Proposition.

PROPOSITION 3.2 For a simple sequencing problem Γ, a mechanism $\mathbf{M} = \langle \sigma^*, \mathbf{t} \rangle$ is a generalized VCG mechanism **if and only if** for all announced processing time vectors $\hat{s} \in (0, \overline{s}]^n$ and for all $j \in \mathbf{N}$,

$$t_j(\hat{s}) = \begin{cases} \sum\limits_{p \in \mathcal{P}_j(\sigma^*(s))} V_p(\hat{s}) + h_j(\hat{s}_{-j}) & \text{if } \sigma^*_j(\hat{s}) \neq 1 \\ h_j(\hat{s}_{-j}) & \text{if } \sigma^*_j(\hat{s}) = 1 \end{cases} \tag{3.2}$$

where $V_p(\hat{s}) = \Delta(\hat{s}_p) F(S_p(\sigma^*(\hat{s}); \hat{s}))$.

The proof of Proposition 3.2 is provided in the Appendix. The importance of the class of generalized VCG mechanism in implementing a simple sequencing problem is captured in the next Proposition.

PROPOSITION 3.3 All simple sequencing problems with weakly concave cost functions are implementable in ex-post equilibrium. Moreover, a simple sequencing problem Γ is implementable in ex-post equilibrium **if and only if** the mechanism is a generalized VCG mechanism.

Observe that to prove Proposition 3.3, it is enough to prove the second statement in the Proposition, using weak concavity of the cost function. We provide the proof of Proposition 3.3 in the Appendix. Here we try to provide the reason behind implementability property of the generalized VCG mechanism. Let $j(p)$ be the immediate predecessor of agent j in the queue σ, that is $j(p) = \{i \in \mathcal{P}_j(\sigma) \mid \sigma_i = \sigma_j - 1\}$. We define the *incremental loss* of agent j, in state s and in queue σ, as

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

Thus, the incremental loss of agent j is the additional cost that j incurs due to the presence of her immediate predecessor $j(p)$ in the queue σ. Consider a state s and an efficient queue $\sigma^*(s)$. For the state s, the incremental loss of agent j is $\mathcal{V}_j(\sigma^*(s);s) = \Delta(s_{j(p)})F(\sum_{q\in\mathcal{P}_{j(p)}(\sigma^*(s))} s_q + s_j)$ if $\sigma^*_j(s) = k \neq 1$ and $s_{j(p)} \leq s_j$ and $\mathcal{V}_j(\sigma^*(s);s) = 0$ otherwise. Consider all types $s'_j \in \hat{S}^k_j$ of agent j and define her *maximum possible incremental loss* as $\mathcal{V}^*_j(\sigma^*(s);s) = max_{s'_j\in\hat{S}^k_j}\mathcal{V}_j(\sigma^*(s'_j,s_{-j});s'_j,s_{-j})$. Due to weak concavity of the cost function F, it follows that

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

Therefore, the maximum possible incremental loss of an agent $j \in \mathbf{N}$, in queue position $\sigma^*_j(s) = k \neq 1$, is the first order difference of amount $s_{j(p)}$ at time point $S_{j(p)}(\sigma(s);s)$ 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 $\Delta(\sum_{p\in\mathcal{P}_j(\sigma^*(s))} s_p)F(s_j) = \sum_{p\in\mathcal{P}_j(\sigma^*(s))}\Delta(s_p)F(S_p(\sigma^*(s);s)+s_j)$.[5] Using weak concavity of F we get $\sum_{p\in\mathcal{P}_j(\sigma^*(s))}\Delta(s_p)F(S_p(\sigma^*(s);s)+s_j) \leq \sum_{p\in\mathcal{P}_j(\sigma^*(s))}\Delta(s_p)F(S_p(\sigma^*(s);s)) = \sum_{p\in\mathcal{P}_j(\sigma^*(s))} V_p(s)$. The last inequality states that 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 generalized VCG mechanism is the maximum possible incremental loss of agent j and that of all her predecessors in the queue, up to a constant. Thus, the generalized VCG transfer is enough to compensate an agent j for her aggregate incremental loss in the queue. However, one can verify that truth-telling is not a dominant strategy. This point will be more explicit from the following example.

EXAMPLE 3.1 Consider the simple sequencing problem with the cost function $F^L(x) = x$, that is consider $\Gamma = \langle\mathbf{N}, F^L, (0,\bar{s}]\rangle$. Since the cost

[5] 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.

function F^L is linear, $\Delta(x)F^L(y) = x$ for all x and y. Therefore, the transfer given by condition (3.2) is

$$t_j(\hat{s}) = \begin{cases} \sum\limits_{p \in \mathcal{P}_j(\sigma^*(\hat{s}))} \hat{s}_p + h_j(\hat{s}_{-j}) & \text{if } \sigma_j^*(\hat{s}) \neq 1 \\ h_j(\hat{s}_{-j}) & \text{otherwise} \end{cases}$$

for all $\hat{s} = (\hat{s}_1, \ldots, \hat{s}_n) \in (0, \bar{s}]^n$. Consider a state $s = (s_1, \ldots, s_n)$ such that $s_1 < s_2 < \ldots < s_n$. Let the announced processing vector of all but agent 1 be $\hat{s}_{-1} = (\hat{s}_2, \ldots, \hat{s}_n)$ such that $\hat{s}_2 < s_1$ and $\hat{s}_i = s_i$ for all $i \in \mathbf{N} - \{1, 2\}$. Thus, in the set $\mathbf{N} - \{j\}$, all but agent 2 are truthful. The utility of agent 1, if she announces truthfully, is $U_1(\sigma^*(s_1, \hat{s}_2, s_{-1-2}), t_1(s_1, \hat{s}_2, s_{-1-2}); s) = v_1 - (s_1 + s_2) + \hat{s}_2 + h_1(\hat{s}_2, s_{-1-2})$ (since for agent 1, $\sigma_1^*(s_1, \hat{s}_2, s_{-1-2}) = 2$ and for agent 2, $\sigma_2^*(s_1, \hat{s}_2, s_{-1-2}) = 1$). What is crucial here is that the efficient queue is calculated on the basis of the announcement $(s_1, \hat{s}_2, s_{-1-2})$ and the cost of agent 1, in the efficient queue, depends on her own processing time s_1 and the actual processing time s_2 (and not announced processing time $\hat{s}_2$) of agent 2. Now suppose that agent 1 announces $\hat{s}_1 (< \hat{s}_2)$ instead of her true processing time s_1. Then her utility is $U_1(\sigma^*(\hat{s}_1, \hat{s}_2, s_{-1-2}), t_1(\hat{s}_1, \hat{s}_2, s_{-1-2}); s) = v_1 - s_1 + h_1(\hat{s}_2, s_{-1-2})$ since $\sigma_1^*(\hat{s}_1, \hat{s}_2, s_{-1-2}) = 1$. Observe that the benefit from misreporting is $s_2 - \hat{s}_2 > 0$. Thus, truth-telling is *not* a dominant strategy for agent 1.

REMARK 3.1 Linearity of the cost function is not crucial for the impossibility argument in Example 3.1. Consider a simple sequencing problem $\Gamma = \langle \mathbf{N}, F, (0, \bar{s}] \rangle$ with a strictly increasing and weakly concave cost function F and consider the same construction as in Example 3.1. What we require for this impossibility argument to work is that there exists a selection $0 < \hat{s}_2 < s_1 < s_2 < \bar{s}$ such that $\Delta(s_2)F(s_1) > \Delta(\hat{s}_2)F(\hat{s}_2)$. Since the cost function is strictly increasing, by selecting sufficiently small positive numbers $\hat{s}_2$ and s_1, satisfying $\hat{s}_2 < s_1$, and by selecting a sufficiently large positive number $s_2 (< \bar{s})$, one can always obtain the required inequality. Moreover, for all implementable simple sequencing problems with $|\mathbf{N}| \geq 2$, the impossibility argument is valid.

4 First Best Implementability

Consider any generalized VCG mechanism for an implementable simple sequencing problem. Observe that for each state $s \in (0, \bar{s}]^n$, if we add up the generalized VCG transfer (3.2) for all agents and set it to zero, we get $\mathbf{V}(s)+\sum_{i\in \mathbf{N}} h_i(s_{-i}) = 0$ where $\mathbf{V}(s) = \sum_{j\in \mathbf{N}}(n-\sigma_j^*(s))V_j(s)$ is the *weighted aggregate maximum possible incremental loss* in state s. The implication of the budget balancedness condition (that is $\mathbf{V}(s) + \sum_{i\in \mathbf{N}} h_i(s_{-i}) = 0$) can be summarized by the Cubical Array Lemma which is due to Walker (1980). Before stating the Lemma we provide some more notations. Consider two profiles $s = (s_1, \dots, s_n)$ and $s' = (s'_1, \dots, s'_n)$. For $P \subseteq \mathbf{N}$, define a type $s_j(P) = s_j$ if $j \notin P$ and $s_j(P) = s'_j$ if $j \in P$. Therefore, $s(P) = (s_1(P), \dots, s_n(P))$ for all $P \subseteq \mathbf{N}$.

LEMMA 4.1 A simple sequencing problem Γ is first best implementable *only if* for all $\{s, s'\} \in (0, \bar{s}]^n \times (0, \bar{s}]^n$, $\sum_{P\subseteq \mathbf{N}}(-1)^{|P|}\mathbf{V}(s(P)) = 0$.[6]

Walker (1980) proved Lemma 4.1 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 generalized VCG mechanism, in the context of simple sequencing problems, depends on the type of all agents and hence the Lemma applies for simple sequencing problems as well. Lemma 4.1 will be used in proving our main Theorem. Before stating our main Theorem, we provide two relevant definitions. The second definition will be used in our main Theorem.

DEFINITION 4.3 A function f is said to be *well behaved* if it is infinitely differentiable in it's open domain.

[6] For any set X, let |X| denote the cardinality of X. Observe that for two agents, budget balancedness implies that for all $\{s = (s_1, s_2), s' = (s'_1, s'_2)\}$, $\mathbf{V}(s)-\mathbf{V}(s'_1, s_2)-\mathbf{V}(s_1, s'_2)+\mathbf{V}(s') = 0$. Similarly, for three agents, it implies that for all $\{s = (s_1, s_2, s_3), s' = (s'_1, s'_2, s'_3)\}$, $\mathbf{V}(s)-\mathbf{V}(s'_1, s_2, s_3)-\mathbf{V}(s_1, s'_2, s_3)-\mathbf{V}(s_1, s_2, s'_3)+\mathbf{V}(s'_1, s'_2, s_3)+\mathbf{V}(s'_1, s_2, s'_3)+\mathbf{V}(s_1, s'_2, s'_3) - \mathbf{V}(s') = 0$. This Lemma is a generalization of this condition for n agents.

DEFINITION 4.4 A function f is said to be *sufficiently well behaved* if it has a power series representation in it's entire open domain, that is there exists y_0 in it's open domain such that the function f has the form $f(y) = \sum_{l=0}^{\infty} c_l (y - y_0)^l$.

A sufficiently well behaved function is well behaved but the converse is not true. In our main Theorem, we provide a necessary condition by restricting the cost function to be sufficiently well behaved.

THEOREM 4.1 A simple sequencing problem is first best implementable **if** the cost function F is a polynomial of order $(n-2)$ or less. Moreover, a simple sequencing problem with sufficiently well behaved cost function is first best implementable **only if** F is a polynomial of order $(n-2)$ or less.

The proof of Theorem 4.1 is provided in the Appendix. Here we first state and prove another Lemma that will also be used in proving Theorem 4.1 and then provide the idea of the proof of Theorem 4.1. We define the *second order cross-partial difference* at x of amounts (a_1, a_2) as $\Delta(a_1)\Delta(a_2)F(x) = \Delta(a_1)[F(x+a_2) - F(x)] = F(x + a_1 + a_2) - F(x + a_2) - F(x + a_1) - F(x)$. Similarly, we define *third order cross-partial difference* at x of amounts (a_1, a_2, a_3) as $\Delta(a_1)\Delta(a_2)\Delta(a_3)F(x) = \Delta(a_1)\Delta(a_2)[\Delta(a_3)F(x)]$ and so on. In general the mth order cross-partial difference at x of amount $(a_1, \ldots, a_m)$ is given by $\Pi_{i=1}^{m}\Delta(a_i)F(x)$. Observe that for a linear function $F^1(x) = b_0 + b_1 x$, the second order cross partial difference of amounts (a_1, a_2) at 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 $F^2(x) = b_0 + b_1 x + b_2 x^2$), 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 4.2 If F is a polynomial function of order $m(= 0, 1, \ldots)$, then for any set of numbers $\{a_1, \ldots, a_{m+1}, x\}$, $\Pi_{r=1}^{m+1}\Delta(a_r)F(x) = 0$.

PROOF: For a polynomial function of order m or less, any cross-partial difference of order $m+1$ is zero. Hence the result. ■

Idea of the proof of Theorem 4.1: To prove the first part of the Theorem, we construct a generalized VCG mechanism for a simple sequencing problem Γ with a polynomial cost function F of order $(n-2)$ and show that it is budget balancing. Let $\mathbf{M}^* = \langle \sigma^*, \mathbf{t}^* \rangle$ be the generalized VCG mechanism such that for all $j \in \mathbf{N}$ and for all $s_{-j} \in (0, \bar{s}]^{n-1}$, $h_j^*(s_{-j}) = -\sum_{i \neq j} g_{ij}(s_{-j})$ where

$$g_{ij}(s_{-j}) = \sum_{r=1}^{\sigma_i^*(s_{-j})} \left\{ \frac{(-1)^{\sigma_i^*(s_{-j})-r}(\sigma_i^*(s_{-j})-r)!(n-\sigma_i^*(s_{-j})-1)!}{(n-r-1)!} \right\} z_{ir}(s_{-j}) \tag{4.3}$$

$$z_{ir}(s_{-j}) = \sum_{\mathcal{P}_{i,r-1}(\sigma^*(s_{-j})) \subset P_i(\sigma^*(s_{-j}))} \Delta(s_i) F\left(\sum_{q \in \mathcal{P}_{i,r-1}(\sigma^*(s_{-j}))} s_q + s_i \right) \tag{4.4}$$

$\mathcal{P}_{i,\alpha}(\sigma^*(s_{-j}))$ is a subset of $\mathcal{P}_i(\sigma^*(s_{-j}))$ of size α and $\sigma^*(s_{-j})$ is the efficient queue in the absence of agent j.[7] Using Lemma 4.2 we prove that $\sum_{j \in \mathbf{N}-\{i\}} g_{ij}(s_{-j}) = (n - \sigma_i^*(s))\Delta(s_i)F^*(S_i(\sigma^*(s); s_i))$. Using this result we get $\mathbf{V}(s) + \sum_{j \in \mathbf{N}} h_j^*(s_{-j}) = 0$. This proves that the generalized VCG mechanism $\mathbf{M}^*$ is budget balancing.[8]

For the other part of the Theorem, we first construct two states $s = (s_1 = x, s_2 = 2x, \ldots, s_n = nx) \in (0, \bar{s}]^n$ and $s' = (s'_1 = nx, s'_2 = x, \ldots, s' = (n-1)x) \in (0, \bar{s}]^n$ and apply Lemma 4.1. This gives the following condition:

$$\Delta^{n-1}(x)F(w_1(n)x) = \Delta^{n-1}(x)F(w_2(n)x) + \Delta^{n-1}(x)F(w_3(n)x) \tag{4.5}$$

where $\Delta^m(x)F(y) = \underbrace{\Delta(x)\ldots\Delta(x)}_{m}F(y) = \sum_{\hat{m}=0}^{m}(-1)^{m-\hat{m}}\binom{m}{\hat{m}}F(y + \hat{m}x)$ is the *mth order partial difference of amount x at time point y*, $w_1(n) = \frac{(n-1)n}{2}$, $w_2(n) = \frac{(n-1)(n+2)}{2}$ and $w_3(n) = \frac{n(n+1)}{2}$. Condition (4.5) is a *general necessary condition* for first best implementability of a simple sequencing problem. From condition (4.5) we get the result using the fact that the cost function is sufficiently well behaved.

[7] Let $\tilde{\Sigma}$ be the set of all possible permutations of the set $\{1, \ldots, n-1\}$ and consider the mapping $\sigma : (0, \bar{s}]^{n-1} \to \tilde{\Sigma}$. Thus, for all $j \in \mathbf{N}$ and for all $s_{-j} \in (0, \bar{s}]^{n-1}$, the efficient queue in the absence of agent j is $\sigma^*(s_{-j}) \in argmin_{\sigma \in \tilde{\Sigma}} \sum_{q \in \mathbf{N}-\{j\}} F(S_q(\sigma; s_q))$.

[8] From now on we refer to $\mathbf{M}^* = \langle \sigma^*, \mathbf{t}^* \rangle$ as the first best mechanism.

We try to argue why polynomial cost of order $(n-2)$ is important for first best implementability of a simple sequencing problem. 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 $b < s_1 < a$. Note that 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 $F(|P|b + s_1)$.[9] This increase in agent 1's cost is due to the negative externality imposed by agents of the set P on agent 1. 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 $\sum_{P \subseteq \mathbf{N} - \{1\}} (-1)^{|P|} \binom{n-1}{|P|} F(|P|b + s_1) = \Delta^{n-1}(b)F(s_1) = \Pi_{i \neq 1} \Delta(\hat{s}_i) F(s_1) = 0$.[10] Observe that this group externality condition guarantees that the general necessary condition (given by condition (4.5)) is satisfied. Using Lemma 4.2, 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 $\Pi_{i \neq j} \Delta(\hat{s}'_i) F(s'_j) = 0$ if the cost function is a polynomial order $(n-2)$. This condition was used to show that the mechanism $\mathbf{M}^*$ is indeed budget balancing. *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*

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

[10] Note that these weights (that is 1 if group size is even and -1 if the group size is odd) is due to Lemma 4.1 which is a necessary condition for first best implementability.

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 implementability of a simple sequencing problem (in ex-post equilibrium) and hence justifies the need for a polynomial cost of order $(n-2)$.

REMARK 4.2 Without restricting the class of cost functions to be sufficiently well behaved one can prove, using condition (4.5), that any implementable simple sequencing problem with two agents is *not* first best implementable. Simplifying condition (4.5) for $n = 2$ we get $F(4x) - F(2x) = F(2x) - F(x)$ for all $x \in (0, \frac{\bar{s}}{2}]$. This condition holds *only if* F is a constant function and hence we have a violation of the fact that F is strictly increasing.

From Remark 4.2 it follows that none of the simple sequencing problems with two agents are first best implementable. With three agents, all simple sequencing problems with linear cost function are first best implementable. For four agents, consider the class of simple sequencing problems $\Gamma^* = \langle \mathbf{N} = \{1, 2, 3, 4\}, F^*, (0, \overline{s}]\rangle$ where $F^*(x) = a_1x + a_2x^2$ for all $x \in (0, n\overline{s}]$ and only one of the following two conditions holds: (1) $a_1 > 0$ and $a_2 = 0$ and (2) $a_1 > 0$, $a_2 < 0$, $\overline{s} < \infty$ and $a_1 \geq -2na_2\overline{s}$. It is easy to verify that this class is first best implementable. One can similarly obtain the class of first best implementable simple sequencing problems with more than four agents.

We conclude this section by providing one example where we consider the class of all linear cost simple sequencing problems. For a linear cost function F^l, the first order difference at some point x of amount y, that is $\Delta(y)F^l(x)$, depends only on y and not on x. This property of the linear cost makes the transfer scheme very transparent. It is easy to see that this property does not hold for any non-linear cost function.

EXAMPLE 4.2 Consider $\Gamma^l = \langle \mathbf{N}, F^l, (0, \bar{s}]\rangle$ where $|\mathbf{N}| \geq 3$, $F^l(x) = a_1x$ for all $x > 0$ and $a_1 > 0$. Consider the first best mechanism $\mathbf{M}^* = \langle \sigma^*, \mathbf{t}^*\rangle$ and a state $s = (s_1, \ldots, s_n) \in (0, \bar{s}]^n$. Observe that from (4.4) we get $z^l_{ir}(s_{-j}) = a_1\binom{\sigma^*_i(s_{-j})-1}{r-1}s_i$ since $\Delta(s_i)F\left(\sum_{q\in\mathcal{P}_{i,r-1}(\sigma^*(s_{-j}))} s_q + s_i\right) = a_1s_i$ due to linearity. By substituting $z^l_{ir}(s_{-j})$ in expression (4.3) and then simplifying it we get $g^l_{ij}(s_{-j}) = a_1 \binom{n-\sigma^*_i(s_{-j})-1}{n-2} s_i$.[11] Given $h^*_j(s_{-j}) = -\sum_{i\neq j} g^l_{ij}(s_{-j})$, we get $h^*_j(s_{-j}) = -a_1 \sum_{i\neq j} \binom{n-\sigma^*_i(s_{-j})-1}{n-2} s_i$. Observe that from the efficiency criterion it follows that $\sigma^*_i(s_{-j}) = \sigma^*_i(s)$ if $i \in \mathcal{P}_j(\sigma^*(s))$ and $\sigma^*_i(s_{-j}) = \sigma^*_i(s) - 1$ if $i \in \mathcal{S}_j(\sigma^*(s))$. Using this observation we get,

$$t^*_j(s) = a_1 \left\{ \sum_{p\in\mathcal{P}_j(\sigma^*(s))} \binom{\sigma^*_p(s) - 1}{n-2} s_p - \sum_{q\in\mathcal{S}_j(\sigma^*(s))} \binom{n - \sigma^*_q(s)}{n-2} s_q \right\} \quad (4.6)$$

Thus, the transfer of an agent j is the weighted sum of the processing cost of all other agents. The weights are positive for all agents who belong to the predecessor set of agent j and the weights are negative for all agents who belong to the successor set of j. It is quite easy to verify using (4.6) that $\sum_{j\in\mathbf{N}} t^*_j(s) = 0$ for all $s \in (0, \bar{s}]^n$.

5 Conclusion

In this paper we have analyzed the merit of ex-post equilibrium in implementing and first best implementing simple sequencing problems with interdependent costs. We showed that the class of generalized VCG mechanism (due to Bergemann and Välimäki (2000)) are the unique class of mechanisms that implements simple sequencing problem in ex-post equilibrium. A mechanism is a generalized VCG mechanism if each agent is paid the maximum possible incremental loss of her own and that of all agents who

[11] By substituting $z^l_{ir}(s_{-j})$ in expression (4.3) we get

$g^l_{ij}(s_{-j}) = a_1 \frac{1}{\binom{n-2}{\sigma^*_i(s_{-j})-1}} \sum_{r=1}^{\sigma^*_i(s_{-j})} (-1)^{\sigma^*_i(s_{-j})-r} \binom{n-2}{r-1} s_i$

or $g^l_{ij}(s_{-j}) = a_1 \frac{(-1)^{\sigma^*_i(s_{-j})-1}}{\binom{n-2}{\sigma^*_i(s_{-j})-1}} \sum_{r=1}^{\sigma^*_i(s_{-j})} (-1)^{r-1} \binom{n-2}{r-1} s_i$

or $g^l_{ij}(s_{-j}) = a_1 \frac{(-1)^{\sigma^*_i(s_{-j})-1}}{\binom{n-2}{\sigma^*_i(s_{-j})-1}} \sum_{k=0}^{\sigma^*_i(s_{-j})-1} (-1)^{k} \binom{n-2}{k} s_i$

or $g^l_{ij}(s_{-j}) = a_1 \frac{(-1)^{2(\sigma^*_i(s_{-j})-1)}}{\binom{n-2}{\sigma^*_i(s_{-j})-1}} \binom{n-3}{\sigma^*_i(s_{-j})-1} s_i$

or $g^l_{ij}(s_{-j}) = a_1 \binom{n-\sigma^*_i(s_{-j})-1}{n-2} s_i$. The penultimate step follows from the mathematical identity $\sum_{k=0}^{m} (-1)^k \binom{\hat{n}}{k} = (-1)^m \binom{\hat{n}-1}{m}$. One can easily prove this identity by applying induction on m.

precedes her in the queue, up to constant. Maximum possible incremental loss of an agent is the incremental loss that she incurs due to the presence of her immediate predecessor in the queue. We proved that a simple sequencing problem is first best implementable if the cost function is a polynomial of order $(n-2)$ or less. Moreover, for the class of simple sequencing problems with sufficiently well behaved cost function, this is the only first best implementable class. A polynomial cost of order $(n-2)$ guarantees that the weighted aggregate negative group externality that can be imposed on any agent j, with a given processing time, by all other agents and with all possible groups, while moving from a state where j is first in the queue to a state where she is last, must add up to zero. This group externality condition guarantees first best implementability of a simple sequencing problem. How will the results change in a more general sequencing set up is still an open question.

6 APPENDIX

PROOF OF PROPOSITION 3.1: We first consider two states that differ only 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 $(0, \overline{s}]$ such that $a < s_j < s_i < s'_j < b$. 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 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 $U_j(\sigma^*(s), t_j(s); s) \geq U_j(\sigma^*(s'), t_j(s'); s)$ and $U_j(\sigma^*(s'), t_j(s'); s') \geq U_j(\sigma^*(s), t_j(s); s')$. Simplifying these two conditions we get that the difference $t_j(s'_j, s_{-j}) - t_j(s)$ must lie in the closed interval $[\Delta(s_i)F(a|P'| + s'_j), \Delta(s_i)F(a|P'| + s_j)]$ where $a|P'| = \sum_{p \in \mathcal{P}_j(\sigma^*(s))} s_p$. Therefore, it is necessary that $\Delta(s_i)F(a|P'| + s'_j) \leq \Delta(s_i)F(a|P'| + s_j)$. The last inequality implies weak concavity of F since $s_j < s'_j$. ■

PROOF OF PROPOSITION 3.2: To prove the necessity part of the Proposition we derive the explicit form of the transfer satisfying conditions (i) and (ii). 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 it follows that there exists an agent p such that $s_j \leq \hat{s}_p \leq s'_j$. From efficiency it also follows that at $\tilde{s}_j = \hat{s}_p$, $\sum_{i \in \mathbf{N}} F(S_i(\sigma^*(\bar{s}); \tilde{s}_j, \hat{s}_{-j})) = \sum_{i \in \mathbf{N}} F(S_i(\sigma^*(\bar{s}'); \tilde{s}_j, \hat{s}_{-j}))$. Using these observations and simplifying (3.1) we get

$$t_j(\bar{s}') - t_j(\bar{s}) = \Delta(\hat{s}_p)F(S_p(\sigma^*(\bar{s}); \bar{s}))(= V_p(\bar{s})) \tag{6.7}$$

Using condition (i) we write $t_j(\hat{s}) = h_j(\hat{s}_{-j})$ for $\sigma^*_j(\hat{s}) = 1$. Solving (6.7) 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 (3.2). The sufficiency part of the Proposition is obvious. ■

PROOF OF PROPOSITION 3.3: We start by proving the *necessity* of the Proposition. Consider a simple sequencing problem $\Gamma = \langle \mathbf{N}, F, (0, \overline{s}] \rangle$. 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 following form: $t_j(s) = \sum_{p \in \mathcal{P}_j(\sigma^*(s))} \Delta(s_p) F(S_p(\sigma^*(s); s_p)) + h_j(s)$. To prove the necessity part of the Proposition, we prove that for all $j \in \mathbf{N}$ and for all *true* $s_{-j} \in (0, \overline{s}]^{n-1}$, $h_j(s_j, s_{-j}) = h_j(s'_j, s_{-j})$ for all s_j and s'_j in $(0, \overline{s}]$. 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 $U_j(\sigma^*(s), t_j(s); s_j) \geq U_j(\sigma^*(s'), t_j(s'); s_j)$ and $U_j(\sigma^*(s'), t_j(s'); s'_j) \geq U_j(\sigma^*(s), t_j(s); s'_j)$. Simplifying, the two 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 using the general transfer $t_j(.)$, specified above, we get $0 \leq h_j(s_j, s_{-j}) - h_j(s'_j, s_{-j}) \leq 0$. Therefore, if s_j and s'_j are such that $\sigma^*(s_j, s_{-j}) = \sigma^*(s'_j, s_{-j})$, then $h_j(s_j, s_{-j}) = h_j(s'_j, s_{-j})$. Now we 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 sub-cases-(i) $\mathcal{P}_j(\sigma^*(s'_j, s_{-j})) - \mathcal{P}_j(\sigma^*(s_j, s_{-j})) = \{q'\}$ where $s_j \leq s_{q'} \leq s'_j$ (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 $s'_j \leq s_{p'} \leq s_j$ (with at least one strict inequality). We first consider sub-case (i). Applying the implementability requirement for agent $j \in \mathbf{N}$ and simplifying it, using the conditions in sub-case (i), we get $h_j(s'_j, s_{-j}) - h_j(s_j, s_{-j}) \in [A(s'_j), A(s_j)]$ where $A(x) = \Delta(s_{q'}) F(\sum_{p \in \mathcal{P}_j(\sigma^*(s_j, s_{-j}))} s_p + x) - \Delta(s_{q'}) F(\sum_{p \in \mathcal{P}_j(\sigma^*(s_j, s_{-j}))} s_p + s_{q'})$. Note that $A(x)$ is non-increasing in $x \in [s_j, s'_j]$ due to weak concavity of F. Moreover, $A(s'_j) \leq 0$, $A(s_j) \geq 0$ and $A(s_{q'}) = 0$. For all $\bar{s}_j \in [s_j, s_{q'})$, $h_j(\bar{s}_j, s_{-j}) = h_j(s_j, s_{-j})$ since $\sigma^*(s_j, s_{-j}) = \sigma^*(\bar{s}_j, s_{-j})$. 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 $\sigma^*(s'_j, s_{-j}) = \sigma^*(\tilde{s}_j, s_{-j})$. At $\hat{s}_j = s_{q'}$, depending on the tie breaking rule, either $\sigma^*(s_j, s_{-j}) = \sigma^*(\hat{s}_j, s_{-j})$ or $\sigma^*(s'_j, s_{-j}) = \sigma^*(\hat{s}_j, s_{-j})$. In either case, $h_j(s'_j, s_{-j}) - h_j(s_j, s_{-j}) = A(s_{q'}) = 0$. Therefore, $h_j(s'_j, s_{-j}) = h_j(s_j, s_{-j})$. We now consider sub-case (ii). Like sub-case (i) we get $h_j(s'_j, s_{-j}) - h_j(s_j, s_{-j}) \in [B(s'_j), B(s_j)]$ where $B(x) = \Delta(s_{p'}) F(\sum_{p \in \mathcal{P}_j(\sigma^*(s'_j, s_{-j}))} s_p + s_{p'}) - \Delta(s_{p'}) F(\sum_{p \in \mathcal{P}_j(\sigma^*(s'_j, s_{-j}))} s_p + x)$.

Note that $B(x)$ is non-decreasing in $x \in [s'_j, s_j]$ due to weak concavity of F. Moreover, $B(s'_j) \leq 0$, $B(s_j) \geq 0$ and $B(s_{p'}) = 0$. Thus, like sub-case (i) we get $h_j(s'_j, s_{-j}) - h_j(s_j, s_{-j}) = B(s_{p'}) = 0$ and hence, $h_j(s'_j, s_{-j}) = h_j(s_j, s_{-j})$. 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 $s'_j \in (0, \overline{s}]$ and for all $s_j \in (0, \overline{s}]$, $B(s'_j, s_j) \leq 0$. There are two possible sub-cases: (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 sub-case (a), $\sigma^*_j(s) > \sigma^*_j(s')$ and $B(s'_j, s_j) = \Delta\left(\sum_{q \in \bar{P}_j} s_q\right) F\left(\sum_{p \in \mathcal{P}_j(\sigma^*(s'))} s_p + s_j\right) - \sum_{q \in \bar{P}_j} \Delta(s_q) F\left(\sum_{r \in \mathcal{P}_q(\sigma^*(s))} s_r + s_q\right)$.[12] By repeatedly applying the relation $\Delta(h_1 + h_2)F(x) = \Delta(h_1)F(x + h_2) + \Delta(h_2)F(x)$ on the first term of $B(s'_j, s_j)$ and simplifying it we get $\Delta\left(\sum_{q \in \bar{P}_j} s_q\right) F\left(\sum_{p \in \mathcal{P}_j(\sigma^*(s'))} s_p + s_j\right) = \sum_{q \in \bar{P}_j} \Delta(s_q) F\left(\sum_{r \in \mathcal{P}_q(\sigma^*(s))} s_r + s_j\right)$. Thus, agent j's benefit is $B(s'_j, s_j) = \sum_{q \in \bar{P}_j} \Delta(s_q)\left[F\left(\sum_{r \in \mathcal{P}_q(\sigma^*(s))} s_r + s_j\right) - F\left(\sum_{r \in \mathcal{P}_q(\sigma^*(s))} s_r + s_q\right)\right]$. Clearly, $B(s'_j, s_j) \leq 0$ since $s_q \leq s_j$ for all $q \in \bar{P}_j$ and F is weakly concave. For sub-case (b), $\sigma^*_j(s) \leq \sigma^*_j(s')$ and the benefit of agent j from misreporting is given by $B(s'_j, s_j) = \sum_{q \in \hat{P}_j} \Delta(s_q) F\left(\sum_{r \in \mathcal{P}_q(\sigma^*(s'))} s_r + s_q\right) - \Delta\left(\sum_{q \in \hat{P}_j} s_q\right) F\left(\sum_{p \in \mathcal{P}_j(\sigma^*(s))} s_p + s_j\right)$.[13] By repeatedly applying the relation $\Delta(h_1 + h_2)F(x) = \Delta(h_1)F(x + h_2) + \Delta(h_2)F(x)$ on the first term of $B(s'_j, s_j)$ and simplifying it we get $\Delta\left(\sum_{q \in \hat{P}_j} s_q\right) F\left(\sum_{p \in \mathcal{P}_j(\sigma^*(s))} s_p + s_j\right) = \sum_{q \in \hat{P}_j} \Delta(s_q) F\left(\sum_{r \in \mathcal{P}_q(\sigma^*(s'))} s_r + s_j\right)$. Thus, the benefit of agent $j \in \mathbf{N}$, by deviating from her true processing time s_j to s'_j is given by $B(s'_j, s_j) = \sum_{q \in \hat{P}_j} \Delta(s_q)\left[F\left(\sum_{r \in \mathcal{P}_q(\sigma^*(s))} s_r + s_q\right) - F\left(\sum_{r \in \mathcal{P}_q(\sigma^*(s))} s_r + s_j\right)\right]$. Clearly, $B(s'_j, s_j) \leq 0$ since $s_j \leq s_q$ for all $q \in \hat{P}_j$ and F is weakly concave. ■

[12] Here $\bar{P}_j = \mathcal{P}_j(\sigma^*(s)) - \mathcal{P}_j(\sigma^*(s'))$.

[13] Here $\hat{P}_j = \mathcal{P}_j(\sigma^*(s')) - \mathcal{P}_j(\sigma^*(s))$.

PROOF OF THEOREM 4.1: We first prove the first part of the Theorem. To do that we construct a particular VCG mechanism for a simple sequencing problem 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 simple sequencing problem with a polynomial cost of order $(n-2)$ or less, consider the implementable 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 $g_{ij}(s_{-j}) = \sum_{r=1}^{\sigma^*_i(s_{-j})} (-1)^{\sigma^*_i(s_{-j})-r} \left\{\frac{(\sigma^*_i(s_{-j})-r)!(n-\sigma^*_i(s_{-j})-1)!}{(n-r-1)!}\right\} z_{ir}(s_{-j})$, $z_{ir}(s_{-j}) = \sum_{\mathcal{P}_{i,r-1}(\sigma^*(s_{-j})) \subset \mathcal{P}_i(\sigma^*(s_{-j}))} \Delta(s_i) F(\sum_{q \in \mathcal{P}_{i,r-1}(\sigma^*(s_{-j}))} s_q + s_i)$ and $\mathcal{P}_{i,\alpha}(\sigma^*(s_{-j}))$ is an α-element subset of $\mathcal{P}_i(\sigma^*(s_{-j}))$. We first prove that $\sum_{j \in \mathbf{N}-\{i\}} g_{ij}(s_{-j}) = (n - \sigma^*_i(s)) \Delta(s_i) F(S_i(\sigma^*(s); s))$ for all $\sigma^*_i(s) \neq n$. Since $\sum_{j \in \mathbf{N}-\{i\}} g_{ij}(s_{-j}) = \sum_{\substack{j \neq i \\ j \notin \mathcal{P}_i(\sigma^*(s_{-j}))}} g_{ij}(s_{-j}) + \sum_{j \in \mathcal{P}_i(\sigma^*(s_{-j}))} g_{ij}(s_{-j})$, 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

$$\sum_{\substack{j \neq i \\ j \notin \mathcal{P}_i(\sigma^*(s_{-j}))}} g_{ij}(s_{-j}) = \sum_{r=1}^{\sigma^*_i(s)} (-1)^{\sigma^*_i(s)-r} \left(\frac{(\sigma^*_i(s)-r)!(n-\sigma^*_i(s))!}{(n-r-1)!}\right) L(r,s) \tag{6.8}$$

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

$$\sum_{j \in \mathcal{P}_i(\sigma^*(s_{-j}))} g_{ij}(s_{-j}) = \sum_{r=1}^{\sigma^*_i(s)-1} (-1)^{\sigma^*_i(s)-r-1} \left(\frac{(\sigma^*_i(s)-r)!(n-\sigma^*_i(s))!}{(n-r-1)!}\right) L(r,s) \tag{6.9}$$

By adding the sums given by conditions (6.8) and (6.9) and then simplifying it, using $(-1)^{\sigma_i^*(s)-r} + (-1)^{\sigma_i^*(s)-r-1} = 0$, we get

$$\sum_{j \in \mathbf{N}-\{i\}} g_{ij}(s_{-j}) = (n - \sigma_i^*(s))\Delta(s_i)F\left(\sum_{j \in P_i(\sigma^*(s))} s_j + s_i\right) \tag{6.10}$$

Therefore, from condition (6.10) it follows that the sum $\sum_{j \in \mathbf{N}-\{i\}} g_{ij}(s_{-j}) = (n - \sigma_i^*(s))\Delta(s_i)F(S_i(\sigma^*(s); s))$ for all $i \in \mathbf{N}$ such that $\sigma_i^*(s) \neq n$. Now we consider $\sum_{j \in \mathbf{N}-\{i\}} g_{ij}(s_{-j})$ 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 $z_{ir}(s_{-j}) = \sum_{\mathcal{P}_{i,r-1}(\sigma^*(s_{-j})) \subset P_i(\sigma^*(s_{-j}))} \Delta(s_i)F\left(\sum_{q \in \mathcal{P}_{ir}(\sigma^*(s_{-j}))} s_q + s_i\right)$, we get $g_{ij}(s_{-j}) = \Pi_{l \neq j}\Delta(s_l)F(s_i)$. 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 4.2 we get $g_{ij}(s_{-j}) = 0$. Therefore, for an agent i such that $\sigma_i^*(s) = n$, $\sum_{j \in \mathbf{N}-\{i\}} g_{ij}(s_{-j}) = 0$. Thus, we get $\sum_{j \in \mathbf{N}-\{i\}} g_{ij}(s_{-j}) = (n - \sigma_i^*(s))\Delta(s_i)F(S_i(\sigma^*(s); s))$ 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 $\sum_{j \in \mathbf{N}} h_j(s_{-j}) = -\sum_{i \in \mathbf{N}-\{j\}} g_{ij}(s_{-j})$, we get

$$\sum_{j \in \mathbf{N}} h_j(s_{-j}) = -\sum_{j \in \mathbf{N}} \sum_{i \in \mathbf{N}-\{j\}} g_{ij}(s_{-j}) = -\sum_{i \in \mathbf{N}} \sum_{j \in \mathbf{N}-\{i\}} g_{ij}(s_{-j})$$

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

or $\sum_{j \in \mathbf{N}} h_j(s_{-j}) = -\mathbf{V}(s)$.
The last step guarantees budget balancedness of the transfer in $\mathbf{M}^*$ for any state $s \in (0, \bar{s}]^n$.

We now prove the second part of the Theorem. The first step will be to construct a pair of states and then apply Lemma 4.1 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. Consider an implementable simple sequencing problem Γ with sufficiently well behaved cost function.

STEP 1: 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 $(n-\sigma_j^*(s(P))) \in \{2, 3, \dots, n-1\}$. 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 $\alpha(k) = \frac{(n-1)(n+2)}{2} - k$. 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 Lemma 4.1, we get condition (4.5), that is

$$\Delta^{n-1}(x)F(w_1(n)x) = \Delta^{n-1}(x)F(w_2(n)x) + \Delta^{n-1}(x)F(w_3(n)x) \quad (6.11)$$

where $w_1(n) = \frac{(n-1)n}{2}$, $w_2(n) = \frac{(n-1)(n+2)}{2}$ and $w_3(n) = \frac{n(n+1)}{2}$. Condition (6.11) is a *general necessary condition* for first best implementability of any implementable simple sequencing problem.

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 (6.11) are of this form. Observe that $\Delta^{n-1}(x)F(wx) = \sum_{k=1}^{n}(-1)^{k-1}\binom{n-1}{k-1}F((w+k-1)x)$ where $F((w+k-1)x) = \sum_{l=0}^{\infty} c_l((w+k-1)x-y_0)^l = \sum_{l=0}^{n-2} c_l((w+k-1)x-y_0)^l + \sum_{l=n-2}^{\infty} c_l((w+k-1)x-y_0)^l$. Therefore, we have re-written $\Delta^{n-1}(x)F(wx)$ as the sum two polynomials. The first one is a polynomial of order $(n-2)$, that is $\sum_{k=1}^{n}(-1)^{k-1}\binom{n-1}{k-1}\left\{\sum_{l=0}^{n-2} c_l((w+k-1)x-y_0)^l\right\}$ and the second sum is a polynomial with all higher order terms, that is $\sum_{k=1}^{n}(-1)^{k-1}\binom{n-1}{k-1}\left\{\sum_{l=n-1}^{\infty} c_l((w+k-1)x-y_0)^l\right\}$. We first consider the sum $\sum_{k=1}^{n}(-1)^{k-1}\binom{n-1}{k-1}\left\{\sum_{l=0}^{n-2} c_l((w+k-1)x-y_0)^l\right\}$ 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 in the sum, we get $\sum_{k=1}^{n}(-1)^{k-1}\binom{n-1}{k-1}\left\{\sum_{l=0}^{n-2} c_l((w+k-1)x-y_0)^l\right\} = \sum_{l=0}^{n-2} c_l \sum_{m=0}^{l}\binom{l}{m}(d(wx))^{l-m}x^m\delta(m)$ where the term $\delta(m)$ is of the form $\delta(m) = \sum_{k=1}^{n}(-1)^{k-1}\binom{n-1}{k-1}(k-1)^m$. From Euler's identity we know that $\delta(m) = 0$ for all integers $m \in \{1, \dots, n-2\}$.[14] *Therefore, the first polynomial*

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

of order $(n-2)$*, that is* $\sum_{k=1}^{n}(-1)^{k-1}\binom{n-1}{k-1}\left\{\sum_{l=0}^{n-2}c_l((w+k-1)x-y_0)^l\right\} = 0$ *for any set of real numbers* $\{c_0,\ldots,c_{n-2}\}$. Thus, $\Delta^{n-1}(x)F(wx)$ is equal to the other polynomial with all higher order terms, that is $\Delta^{n-1}(x)F(wx) = \sum_{k=1}^{n}(-1)^{k-1}\binom{n-1}{k-1}\left\{\sum_{l=n-1}^{\infty}c_l((w+k-1)x-y_0)^l\right\}$. 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: $\sum_{k=1}^{n}(-1)^{k-1}\binom{n-1}{k-1}\left\{\sum_{l=n-1}^{\infty}c_l((w+k-1)x-y_0)^l\right\} = \sum_{l=n-1}^{\infty}c_l\sum_{m=0}^{l}\binom{l}{m}(-y_0)^{l-m}\alpha(w,m)x^m$. We now try to evaluate the value of $\alpha(w,m)$. By taking the Binomial expansion of $(w+(k-1))^m$ we get $\alpha(w,m) = \sum_{m_0=0}^{m}\binom{m}{m_0}w^{m-m_0}\delta(m_0)$. From Euler's identity we know that $\delta(m_0) = 0$ for all $m_0 \le n-2$. Hence, $\alpha(w,m) = \sum_{m_0=n-1}^{m}\binom{m}{m_0}w^{m-m_0}\delta(m_0)$. Now we calculate the value of the term $\delta(m_0) = \sum_{k=1}^{n}(-1)^{k-1}\binom{n-1}{k-1}(k-1)^{m_0}$ for $m_0 \ge n-1$. Expanding $(k-1)^{m_0}$, using Stirling number of the second kind and then simplifying it, we get $\delta(m_0) = (-1)^{n-1}(n-1)!S(m_0,n-1)$.[15] 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,

$$\Delta^{n-1}(x)F(wx) = (-1)^{n-1}(n-1)!\sum_{l=n-1}^{\infty}c_l\sum_{m=n-1}^{l}\binom{l}{m}(-y_0)^{l-m}G(w,m)x^m \tag{6.12}$$

By substituting condition (6.12) in condition (6.11) and simplifying it, using $(-1)^{n-1}(n-1)! \neq 0$, we get $\sum_{l=n-1}^{\infty}c_l\sum_{m=n-1}^{l}\binom{l}{m}(-y_0)^{l-m}\beta(m)x^m = 0$, where the term $\beta(m)$ is given by $\beta(m) = G(w_2,m)+G(w_3,m)-G(w_1,m) =$

---

[15] 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. $1 = S(m_0,1) < S(m_0,2) < \ldots < S(m_0,M(m_0)) > S(m_0,M(m_0)-1) > \ldots > S(m_0,m_0) = 1$ or
2. $1 = S(m_0,1) < S(m_0,2) < \ldots < S(m_0,M(m_0)-1) = S(m_0,M(m_0)) > \ldots > S(m_0,m_0) = 1$

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; $1 \le q \le m_0\}$. For a better understanding see Tomescu (1985).

$\sum_{m_0=n-1}^{m} \binom{m}{m_0}(w_2^{m-m_0}+w_3^{m-m_0}-w_1^{m-m_0})S(m_0,n-1)$. Note that $\beta(m)>0$ since $0<w_1<w_2<w_3$, $m-m_0\geq 0$ for all $m_0=n-1,\ldots,m$ and since $S(m_0,n-1)\geq 1$ for all integers $m_0\geq n-1$. 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 $B_r(=\frac{A_r}{\beta(r)}) = \sum_{l=r}^{\infty} c_l\binom{l}{r}(-y_0)^{l-r}=0$ 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 this way recursively, we get, for any $p=0,1,\ldots,\infty$, $\sum_{l=r}^{\infty} l^pc_l\binom{l}{r}(-y_0)^{l-r}=0$ for all $r=n-1,n,\ldots,\infty$. Thus, given any $p=0,1,\ldots,\infty$, we also get

$$\sum_{l=r}^{\infty}(l-h)^p c_l\binom{l}{r}(-y_0)^{l-r}=0 \tag{6.13}$$

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 $r=n-1,n,\ldots,\infty$.[16] From condition (6.13) it follows that $E_r=0$ for all r, since E_r can be written as $E_r=\sum_{p=0}^{l-r} s(l-r,p)\left\{\sum_{l=r}^{\infty} c_l\binom{l}{r}(l-r)^p(-y_0)^{l-r}\right\}$ 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 (6.11) 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 $l=n-1,n,\ldots,\infty$. ■

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

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. and Välimäki, J. (2000). "Information Acquisition and Efficient Mechanism Design", Discussion Paper # 1248, Yale University.

Clarke, E. H. (1971). "Multi-part Pricing of Public Goods," *Public Choice* **11,** 17-33.

Dasgupta, P. and Maskin, E. (2000). "Efficient Auctions," *The Quarterly Journal of Economics* **115,** 341-388.

d'Aspremont, C. and Gérard-Varet L. A. (1979). "Incentives and Incomplete Information," *Journal of Public Economics* **11,** 25-45.

Fieseler, Kittsteiner and Moldovanu (2000). "Partnerships, Lemons and Efficient Trade," SFB 504 Discussion paper 01-18, University of Mannheim.

Green, J. and Laffont, J. J. (1979). Incentives in public decision making. North Holland Publication, Amsterdam.

Gresik, T. A. (1991). "Ex-Ante Incentive Efficient Trading Mechanisms without the Private Valuation Restriction," *Journal of Economic Theory* **55,** 41-63.

Groves, T. (1973). "Incentives in Teams," *Econometrica* **41,** 617-631.

Hurwicz, L. and 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. and Moldovanu, B. (2001). "Efficient Design with Interdependent Valuations," *forthcoming in Econometrica.*

Laffont J. J. and Maskin E. (1982). "The theory of incentives: an overview" in W. Hildenbrand edited, *Advances in Economic Theory.*

Maskin, E. (1992). "Auctions and Privatization," in Horst Siebert edited, *Privatization: Symposium in Honor of Herbert Giersch*.

Mitra, M. (2001). "Achieving the First Best in Sequencing Problems," *forthcoming in Review of Economic Design*.

Perry, M. and Reny, P. J. (1999). "An Ex-Post Efficient Auction," Working Paper, Hebrew University and University of Chicago.

Radner, R., and Williams, S. R. (1988). "Informational Externalities and the Scope of Efficient Dominant Strategy Mechanisms," Discussion Paper # 761, Northwestern University.

Suijs, J. (1996). "On Incentive Compatibility and Budget Balancedness in Public Decision Making," *Economic Design* **2,** 193-209.

Tomescu, I. (1985). Problems in combinatorics and graph theory. Translated by R.A. Melter, A Wiley-Interscience Publication.

Vickrey, W.(1961) "Counterspeculation, Auctions and Competitive Sealed Tenders," *Journal of Finance* **16**, 8-37.

Walker, M. (1980) "On the Non-Existence of Dominant Strategy Mechanisms for Making Optimal Public Decisions," *Econometrica* **48**, 1521-1540.