Mitra, Manipushpak

**Working Paper**

# Incomplete Information and Multiple Machine Queueing Problems

Bonn Econ Discussion Papers, No. 19/2001

**Provided in Cooperation with:**
Bonn Graduate School of Economics (BGSE), University of Bonn

*Suggested Citation:* Mitra, Manipushpak (2001) : Incomplete Information and Multiple Machine Queueing Problems, Bonn Econ Discussion Papers, No. 19/2001, University of Bonn, Bonn Graduate School of Economics (BGSE), Bonn

This Version is available at:
https://hdl.handle.net/10419/78417

Bonn Econ Discussion Papers

**Discussion Paper 19/2001**

# Incomplete Information and Multiple Machine Queueing Problems

by

**Manipushpak Mitra**

March 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

# INCOMPLETE INFORMATION AND MULTIPLE MACHINE QUEUEING PROBLEMS

Manipushpak Mitra *

Department of Economics, University of Bonn,
Adenauerallee 24-26, D-53113 Bonn,
Germany.

Revised on March 2001

**ABSTRACT**

In mechanism design problems under incomplete information, it is generally difficult to find decision problems that are first best implementable. A decision problem under incomplete information is first best implementable if there exists a mechanism that extracts the private information and achieves efficiency with a transfer scheme that adds up to zero in every state. One such problem is the queueing problem with one machine. In this paper we identify the conditions on cost structure for which queueing problems with multiple machines are first best implementable.

**Keywords:** Queueing Problems, First Best Implementability.

*The author is grateful to Arunava Sen, Jeroen Suijs and two anonymous referees for their invaluable advice. The author gratefully acknowledges the financial support from the Indian Statistical Institute and from the German Research Foundation (DFG).

# 1 Introduction

In a queueing problem with multiple machines, there is a server (for example, a computer server), with more than one identical machines (computers) which has to process a finite number of jobs for a set of individuals. The machines are identical in the sense that a given job takes the same length of time for completion. We assume that it takes one unit of time to complete one job. Each individual has one job to be processed. The server can serve one individual in one machine, that is, it takes one unit of time to process one job in a machine. If the number of jobs, to be processed, is more than the number of machines then individuals will have to wait in a queue. Waiting in a queue is costly for each individual. The server's objective is to order the individuals in a queue efficiently so as to minimise the aggregate waiting cost. If the cost of waiting in the queue is private information then an individual, if asked, will announce his cost strategically so as to get his job done as early as possible. Therefore, in the queueing scenario described above, the server's role is that of a planner who has to solve an incentive problem under incomplete information. More precisely, we have a mechanism design problem of a social planner (server in the queueing problem) whose objective is to extract the privately held information (true waiting cost) of each individual and select the efficient decision (to order the individuals in a queue so as to minimise the aggregate waiting cost) in each state.

One of the most significant achievements in the planner's mechanism design problems under incomplete information has been the existence of a class of mechanisms called Groves-Clarke mechanisms (see Clarke (1971) and Groves (1973)). These mechanisms achieve the twin objectives of truthful revelation of private information and efficiency of decisions provided the agents have quasi-linear preferences. Moreover, for a very broad class of preference structures, in a quasi-linear set up, Groves-Clarke mechanisms are the only class of mechanisms that achieve these objectives (see Holmström (1979)). However, the drawback of such mechanisms is that they are, in general, not Pareto-optimal. This means that there are preference real-

izations where the sum of Groves-Clarke transfers are non-zero. In the pure public goods problem, Hurwicz (1975), Green and Laffont (1979) and Walker (1980) proved the budget imbalance of a Groves-Clarke scheme. Hurwicz and Walker (1990) proved the impossibility result in the context of pure exchange economies (economies in which there are no production, no public goods and other externalities). The damaging nature of budget imbalance, in the public goods context, was pointed out by Groves and Ledyard (1977). They proposed, using a very simple model, that an alternative procedure based on majority rule voting may lead to an allocation of resources which is Pareto superior to the one produced by Groves mechanism. However, there are certain decision problems where first best or Pareto optimality can be achieved. In the public goods problem, Groves and Loeb (1975) have proved that if preferences are quadratic then we can find balanced Groves transfer. This result was generalized by Tian (1996) and Liu and Tian (1999). In a single server (one machine) queueing problem with linear cost, Mitra and Sen (1998) showed the existence of first best mechanisms. A problem similar to the queueing problem with linear costs in Mitra and Sen (1998) is the sequencing problem in Suijs (1996). Unlike the queueing problem, where it takes one unit of time to service one individual, in a sequencing problem the servicing time can differ from one individual to another. Therefore, while the linear cost queueing problem is a discrete time problem, the sequencing problem in Suijs (1996) is a continuous time problem. By assuming servicing time to be common knowledge, Suijs proved the existence of first best mechanisms for the sequencing problem. The existence result in Mitra and Sen (1998) was further generalised by Mitra (2001) for a broader class of cost structures. It was proved that the class of cost structures under which a 'one machine queueing problem' is first best implementable is 'fairly' large. In this paper, we deal with the question of first best implementability of queueing problems with multiple machine. Therefore, this paper is a generalization of the one machine queueing framework of Mitra (2001) to a multiple machine framework. A multiple machine queueing problem is first best implementable if there exists a mechanism that can extract the private information with a vector of transfers that add up to zero. This allows the

server to order the jobs in a way that minimises the aggregate cost. The most important implication of first best implementability is that the server can extract the private information costlessly. If a queueing problem is first best implementable then there is no welfare loss as the transfers used to extract the private information adds up to zero in all states.

A multiple machine queueing problem resembles some of the sequencing problems that are analysed in the operations research literature. Papers relating to sequencing $n$ jobs in $m$ machines by Dudek and Teuton Jr. (1964), flow shop sequencing problems with ordered processing time by Dudek, Panwalkar and Smith (1975) and flow shop problems with dominant machines by Krabbenborg, Potters and van den Nouweland (1992) deal with finding algorithms to order (or queue) the $n$ jobs in $m$ machines in a way that minimises the total elapsed time. However, in all these models, unlike multiple machine queueing problems, machines are not identical. The processing time for the same job can be different in different machines. Moreover, unlike a multiple machine queueing problem where cost parameter is private information, the cost structures in all the above mentioned sequencing problems are common knowledge.

In the incomplete information set up, sequencing problems were analysed by Hamers, Klijn and Suijs (1999). They analysed a multiple identical machine sequencing (or scheduling) problem with linear time cost. Therefore, their problem is a continuous time version of the multiple machine queueing problem with linear cost. Hamers et al. (1999) look at the $n$ jobs and $m$ identical machines sequencing situation both in a co-operative and non-cooperative environments. In a co-operative set up, a sequencing problem is called a sequencing game. Curiel, Pederzoli and Tijs (1989) analyzed the sequencing game in a one machine framework. Hamers et al.(1999), by extending the sequencing game of Curiel et al.(1989) to a multiple identical machine framework, addressed the issues of balancedness and non-emptiness of core in $m$-sequencing games. In a non-cooperative set up, they address the issue of first best implementability by assuming job processing time to be equal to one. Thus, in the non-cooperative set up, the sequencing situation analysed by Hamers et al.(1999) is identical to the multiple ma-

chine queueing problem described in this paper with the restriction that the cost is linear over time. *The main objective of this paper is to achieve first best implementability in multiple machine queueing problems. More precisely, this paper identifies the conditions on the cost structure that lead to first best implementability.* Therefore, in this paper, we also generalise the non-cooperative sequencing situation, analysed by Hamers et al.(1999), by allowing for a very general time cost. The results that we get suggest that first best implementability depends critically on the number of machines and the number of jobs to be processed on those machines. If the number of machines (that is $m$) is even or if the number of jobs to be processed is strictly greater than the number of machines but less than or equal to twice the number of machines, then a multiple machine queueing problem is not first best implementable. For all other multiple machine queueing problems there exists a cost structure under which it is first best implementable. Finding an algorithm to obtain the cost minimising queue is not a very important issue since the conditions on the cost structure under which a multiple machine queueing problem is first best implementable are such that the algorithm for finding the cost minimising queue is transparent. Therefore, obtaining a state contingent transfer scheme that extracts the private information while adding up to zero is of prime importance.

The paper is organised in the following way: in section two we develop the general problem, in section three we derive some characterization results, in section four we deal with an application of the general results and finally in the concluding remarks of section five we summarize the results obtained in this paper.

## 2 The General Problem

Let $\mathbf{N} = \{1, 2, \ldots, n\}$ be the set of individuals and $m(> 1)$ be the set of identical machines. Define $[x]_+$ to be the lowest integral value not less than $x$. For example, $[2.005]_+ = 3$ and $[3]_+ = 3$. Given $n$ and $m$, the total number of queue positions are $M = [n/m]_+$. Here the number of individuals (and hence the number of jobs) $n$ is strictly greater than the number of machines

$m$ in order to have a meaningful queueing problem. $\theta_j(k)$ measures the cost of waiting $k$ periods in the queue for individual $j$ where $k \in \{1, \ldots, M\}$. The type or cost vector of individual $j \in \mathbf{N}$ is the vector $\underline{\theta}_j = (\theta_j(1), \ldots, \theta_j(M))$. Let $\mathbf{R}_+$ represent the non-negative orthant of $\mathbf{R}$. Clearly, $\theta_j(k) \in \mathbf{R}_+$ for all $j \in \mathbf{N}$ and for all $k \in \{1, \ldots, M\}$. Any cost vector of an individual $j \in \mathbf{N}$ belongs to $\mathbf{\Theta}$ satisfying the following assumptions:

**Assumption I:** For all $j \in \mathbf{N}$ and for all $\underline{\theta}_j = (\theta_j(1), \ldots, \theta_j(M)) \in \mathbf{\Theta}$, $0 \leq \theta_j(1) \leq \theta_j(2) \leq \ldots \leq \theta_j(M)$.

Assumption I simply says that the individuals are impatient. We are assuming that the domain of cost vectors is identical for all individuals. Let $\bar{\mathbf{\Theta}}$ be the largest domain satisfying assumption I. Therefore, a domain $\mathbf{\Theta}$ to be a subset of $\bar{\mathbf{\Theta}}$. For all $j \in \mathbf{N}$, $\underline{\theta}_j \in \bar{\mathbf{\Theta}}$, the utility of each individual $j$ is assumed to be quasi-linear and is of the form: $U_j(k, t_j; \underline{\theta}_j) = v_j - \theta_j(k) + t_j$ where $v_j (> 0)$ is the gross benefit derived by individual $j$ from the service and $t_j$ is the transfer that individual $j$ receives.

The server's aim is to achieve efficiency or minimise the aggregate cost. To define what we mean by efficiency in a queueing problem with $m$ machines, we need to develop the concept of a *multi-set*. A multi-set is a set where all elements may not be distinct. For example, $X = \{1, 1, 1, 3, 6, 6, 9\}$ is a multi-set. Given a queueing problem with $n$ individuals, $m$ machines and hence $M = [n/m]_+$ queue positions, consider the multi-set $\mathrm{X}_{n,m}$ of the form $\mathrm{X}_{n,m} = \{\underbrace{1, \ldots, 1}_{m}, \underbrace{2, \ldots, 2}_{m}, \ldots, \underbrace{M-1, \ldots, M-1}_{m}, \underbrace{M, \ldots, M}_{n-(M-1)m}\}$. Let $P(\mathrm{X}_{n,m})$ be the set of all possible permutations of the multi-set $\mathrm{X}_{n,m}$. In this problem, a queue $\sigma$ is a mapping from the set of individuals $\mathbf{N}$ to $P(\mathrm{X}_{n,m})$, that is, $\sigma = (\sigma_1, \ldots, \sigma_n) : \mathbf{N} \to P(\mathrm{X}_{n,m})$. Thus, $\sigma_j = k$ indicates that individual $j$ has the $k$th position in the queue. Given a queue $\sigma = (\sigma_1, \ldots, \sigma_n) (\in P(\mathrm{X}_{n,m}))$, the cost of an individual $j \in \mathbf{N}$ is $\theta_j(\sigma_j)$. A state of the world is $\underline{\theta} = (\underline{\theta}_1, \ldots, \underline{\theta}_n) \in \bar{\mathbf{\Theta}}^n$ where $\underline{\theta}_j$ is a $1 \times M$ vector.

**DEFINITION 2.1** *Given a state $\underline{\theta} \in \bar{\mathbf{\Theta}}^n$, a queue $\sigma^* \in P(X_{n,m})$ is efficient if $\sigma^* \in argmin_{\sigma \in P(X_{n,m})} \sum_{j \in \mathbf{N}} \theta_j(\sigma_j)$.*

An efficient queue $\sigma^*$ is an assignment that gives each individual exactly

one queue position and each of the first $M-1$ queue positions to exactly $m$ individuals and the $M$th queue position to the remaining $n-(M-1)m$ individuals in such a way that the aggregate cost is minimised. Observe, that there can be states with more than one efficient queue. So we have an efficiency correspondence. An *efficient rule* is a single valued selection from the efficiency correspondence. Note that efficiency of a queue $\sigma^*$ is a concept independent of transfers and gross benefits of all individuals.

If the server knows the true state $\underline{\theta} = (\underline{\theta}_1, \ldots, \underline{\theta}_n)$ then he can calculate the efficient queue. However, if $\underline{\theta}_j$ is private information for individual $j$, the server's problem then is to design a *mechanism* that will elicit this information truthfully. This problem is called a *multiple machine queueing problem* under incomplete information and is written as $\Gamma = \langle \mathbf{N}, m, \bar{\mathbf{\Theta}} \rangle$. Note that we are assuming that the domain $\bar{\mathbf{\Theta}}$ is common knowledge and that the cost vector of an individual is private information. Therefore, each individual, if asked, will announce a cost vector from the domain $\bar{\mathbf{\Theta}}$. Formally, a mechanism $\mathbf{M}$ is a pair $\langle \sigma, \mathbf{t} \rangle$ where $\sigma : \bar{\mathbf{\Theta}}^n \to P(\mathrm{X}_{n,m})$ and $\mathbf{t} \equiv (t_1, \ldots, t_n) : \bar{\mathbf{\Theta}}^n \to \mathbf{R}^n$. Thus, a mechanism $\mathbf{M}$ is a *direct revelation mechanism* where each individual $j \in \mathbf{N}$ announces a cost vector $\underline{\theta}_j = (\theta_j(1), \ldots, \theta_j(M))$ and based on the announcements of all individuals (that is, $\underline{\theta} = (\underline{\theta}_1, \ldots, \underline{\theta}_n)$), the planner (or server) specifies a queue $\sigma$ and a vector of transfers $\mathbf{t} = (t_1, \ldots, t_n)$. Under $\mathbf{M} = \langle \sigma, \mathbf{t} \rangle$, given all others' announcement $\underline{\theta}_{-j}$, the utility of individual $j$ of type $\underline{\theta}_j$ when his announcement is $\underline{\theta}_j{}'$ is given by $U_j(\sigma_j(\underline{\theta}_j{}', \underline{\theta}_{-j}), t_j(\underline{\theta}_j{}', \underline{\theta}_{-j}); \underline{\theta}_j) = v_j - \theta_j(\sigma_j(\underline{\theta}_j{}', \underline{\theta}_{-j})) + t_j(\underline{\theta}_j{}', \underline{\theta}_{-j})$.

**DEFINITION 2.2** *A multiple machine queueing problem $\Gamma = \langle \mathbf{N}, m, \bar{\mathbf{\Theta}} \rangle$ is implementable if there exists an efficient rule $\sigma^* : \bar{\mathbf{\Theta}}^n \to P(X_{n,m})$ and a mechanism $\mathbf{M} = \langle \sigma^*, \mathbf{t} \rangle$ such that for all $j \in \mathbf{N}$, for all pair of announcement vectors $(\underline{\theta}_j, \underline{\theta}_j{}') \in \bar{\mathbf{\Theta}}^2$ and for all announced $\underline{\theta}_{-j} \in \bar{\mathbf{\Theta}}^{n-1}$, $U_j(\sigma_j^*(\underline{\theta}), t_j(\underline{\theta}); \underline{\theta}_j) \geq U_j(\sigma_j^*(\underline{\theta}_j{}', \underline{\theta}_{-j}), t_j(\underline{\theta}_j{}', \underline{\theta}_{-j}); \underline{\theta}_j)$.*

This definition says that $\Gamma = \langle \mathbf{N}, m, \bar{\mathbf{\Theta}} \rangle$ is implementable if there exists a direct mechanism, with an efficient queueing rule $\sigma^*$ and a vector of transfers, that induces each individual to tell the truth independent of others' report.

**DEFINITION 2.3** *A multiple machine queueing problem* $\Gamma = \langle \mathbf{N}, m, \bar{\mathbf{\Theta}} \rangle$ *is first best implementable, if there exists a mechanism* $\mathbf{M} = \langle \sigma^*, \mathbf{t} \rangle$ *such that (1)* $\mathbf{M}$ *implements* $\Gamma$ *and (2) for all announcements* $\underline{\theta} \in \bar{\mathbf{\Theta}}^n$, $\sum_{j \in \mathbf{N}} t_j(\underline{\theta}) = 0$.

A multiple machine queueing problem $\Gamma = \langle \mathbf{N}, m, \bar{\mathbf{\Theta}} \rangle$ is first best implementable if it can be implemented with a budget balancing transfer. Thus, if $\Gamma$ is first best implementable then incomplete information does not impose any welfare loss.

# 3 Characterization Results

**DEFINITION 3.4** *A mechanism* $\mathbf{M} = \langle \sigma, \mathbf{t} \rangle$ *is a Groves-Clarke mechanism if for all* $j \in \mathbf{N}$ *and for all* $\underline{\theta} \in \bar{\mathbf{\Theta}}^n$, *the transfer is of the form*

$$t_j(\underline{\theta}) = -\sum_{l \neq j} \theta_l(\sigma_l(\underline{\theta})) + \gamma_j(\underline{\theta}_{-j}) \tag{3.1}$$

In a Groves-Clarke mechanism, the transfer of any individual $j \in \mathbf{N}$ in any state $\underline{\theta}$ is the negative of aggregate cost plus the cost of individual $j$ (that is $-\sum_{l \in \mathbf{N}} \theta_l(\sigma_l(\underline{\theta})) + \theta_j(\sigma_j(\underline{\theta})) = -\sum_{l \neq j} \theta_l(\sigma_l(\underline{\theta}))$), plus a constant $\gamma_j(\underline{\theta}_{-j})$. The utility of individual $j$ with a Groves-Clarke transfer is his gross benefit $v_j$ less the aggregate cost in state $\underline{\theta}$ plus the constant. We now proceed to verify that given a mechanism $\mathbf{M} = \langle \sigma^*, \mathbf{t} \rangle$ where the queue satisfies an efficient rule and the transfer satisfies condition (3.1), truth-telling is a dominant strategy. Suppose, it were not the case. Then there exists an individual $j$ with true cost $\underline{\theta}_j$ and there exists a report $\underline{\theta}_{-j}$ such that individual $j$ strictly benefits by misreporting his cost to be some $\underline{\theta}'_j (\neq \underline{\theta}_j)$. That is, $U_j(\sigma_j^*(\underline{\theta}), t_j(\underline{\theta}); \underline{\theta}_j) < U_j(\sigma_j^*(\underline{\theta}_j{}', \underline{\theta}_{-j}), t_j(\underline{\theta}_j{}', \underline{\theta}_{-j}); \underline{\theta}_j)$. Simplifying this inequality after substituting the Groves-Clarke transfer, we get $\sum_{j \in \mathbf{N}} \theta_j(\sigma_j^*(\underline{\theta})) > \sum_{j \in \mathbf{N}} \theta_j(\sigma_j^*(\underline{\theta}'_j, \underline{\theta}_{-j}))$. This contradicts efficiency of decision (or aggregate cost minimisation) in state $\underline{\theta}$. Hence, the Groves-Clarke transfer leads to truth-telling in dominant strategies.

According to a well known result of Holmström (see Holmström (1979)), decision problems with "convex" domains are implementable if and only if

the mechanism is a Groves-Clarke mechanism (see Theorem (2) in Holmström (1979)). Note that the domain under consideration in all multiple machine queueing problems $\Gamma = \langle \mathbf{N}, m, \bar{\mathbf{\Theta}} \rangle$ are "convex" domains. Hence multiple machine queueing problems are implementable if and only if the mechanism is a Groves-Clarke mechanism. Therefore, the question of first best implementability of a multiple machine queueing problem reduces to finding conditions on the domain under which we can find a balanced Groves-Clarke transfer.

Let $C(\sigma^*(\underline{\theta}'); \underline{\theta}) = \sum_{j \in \mathbf{N}} \theta_j(\sigma_j^*(\underline{\theta}'))$ where, as stated earlier, $\sigma^*(\underline{\theta}')$ is an efficient queue for the announced state $\underline{\theta}'$. Thus, $C(\sigma^*(\underline{\theta}'); \underline{\theta})$ is the minimum aggregate cost with respect to the announced state $\underline{\theta}'$ when the actual state is $\underline{\theta}$. For notational simplicity we define $C(\underline{\theta}) \equiv C(\sigma^*(\underline{\theta}); \underline{\theta})$ to be the minimum aggregate cost with respect to the actual state $\underline{\theta}$ when the announced state is also $\underline{\theta}$.

**REMARK 3.1** *From the definition of efficiency of the queue $\sigma^*$ it follows that for all $\underline{\theta}$ and $\underline{\theta}'$, $C(\underline{\theta}) \leq C(\sigma^*(\underline{\theta}'); \underline{\theta})$.*

When is a multiple machine queueing problem first best implementable? In our first Theorem we show that the General Combinatorial Property, defined below, is necessary for first best implementability of a multiple machine queueing problem.

**DEFINITION 3.5** *A multiple machine queueing problem $\Gamma = \langle \mathbf{N}, m, \mathbf{\Theta} \rangle$ satisfies the General Combinatorial Property (or GCP) if for all $j \in \mathbf{N}$, $\underline{\theta}_j \in \mathbf{\Theta}$ is such that*

$$\sum_{k=1}^{M} \alpha(k; n, m)\theta_j(k) = 0 \tag{3.2}$$

*where the coefficient vector $\alpha(n, m) = \{\alpha(k; n, m)\}_{k=1}^{M}$ and $\alpha(k; n, m) = (-1)^{(k-1)m} \left\{ \sum_{l=0}^{m-1} (-1)^l \binom{n-1}{(k-1)m+l} \right\}$ if $k \in \{1, \ldots, M-1\}$ and $\alpha(k; n, m) = (-1)^{(M-1)m} \left\{ \sum_{l=0}^{n-(M-1)m-1} (-1)^l \binom{n-1}{(M-1)m+l} \right\}$ if $k = M$.*

The following example illustrates the GCP.

**EXAMPLE 3.1** *Consider* $\hat{\Gamma} = \langle \mathbf{N} = \{1, \ldots, 10\}, m = 3, \mathbf{\Theta} \rangle$. *Here the number of queue positions is* $M = [10/3]_+ = 4$ *and* $\alpha(10,3) = (\alpha(1;10,3) = 28, \alpha(2;10,3) = -84, \alpha(1;10,3) = 57, \alpha(1;10,3) = -1)$. *Therefore,* $\hat{\Gamma}$ *satisfies the GCP if for all* $j \in \{1, \ldots, 10\}$, $\underline{\theta}_j = (\theta_j(1), \theta_j(2), \theta_j(3), \theta_j(4)) \in \mathbf{\Theta}$ *is such that* $28\theta_j(1) - 84\theta_j(2) + 57\theta_j(3) - \theta_j(4) = 0$. *Note that the coefficient vector* $\alpha(10,3) = (28, -84, 57, -1)$ *is such that* $\sum_{k=1}^{M=4} \alpha(k;10,3) = 0$.

From Example 3.1 it is quite obvious that for a multiple machine queueing problem $\Gamma$, the coefficient vector $\alpha(n,m) = \{\alpha(k;n,m)\}_{k=1}^{M}$ is such that

$$\sum_{k=1}^{M} \alpha(k;n,m) = \sum_{p=1}^{n} (-1)^{p-1} \binom{n-1}{p-1} = 0 \tag{3.3}$$

We define the first order difference at $k$ as $\Delta\theta_j(k) = \theta_j(k+1) - \theta_j(k)$. The first order difference at queue position $k$ represents the increase in queueing cost for individual $j$ if he is moved from $k$th queue position to $(k+1)$th queue position. By simplifying equation (3.2) using $\Delta\theta_j(k)$ we get

$$\sum_{k=1}^{M-1} z(k;n,m)\Delta\theta_j(k) = 0 \tag{3.4}$$

where the partial sum coefficient vector $z(n,m) = \{z(k;n,m)\}_{k=1}^{M-1}$ is such that $z(k;n,m) = \sum_{r=1}^{k} \alpha(r;n,m)$ for all $k \in \{1, \ldots, M-1\}$. From the mathematical identity $\sum_{q=0}^{r} (-1)^q \binom{a}{p} = (-1)^r \binom{a-1}{r}$, it follows that $z(k;n,m) = (-1)^{km-1} \binom{n-2}{km-1}$ for all $k \in \{1, \ldots, M-1\}$ (see Tomescu (1985)). For example consider the multiple machine queueing problem $\hat{\Gamma}$ in Example 3.1. Recall that in the co-efficient vector $\alpha(10,3) = (28, -84, 57, -1)$. In $\hat{\Gamma}$, the elements of the partial sum coefficient vector $z(10,3)$ are $z(1;10,3) = \alpha(1;10,3) = 28 = (-1)^2 \binom{8}{2}$, $z(2;10,3) = \alpha(1;10,3) + \alpha(2;10,3) = -56 = (-1)^5 \binom{8}{5}$ and $z(3;10,3) = \sum_{k=1}^{3} \alpha(k;10,3) = 1 = (-1)^8 \binom{8}{8}$.

**THEOREM 3.1** *A multiple machine queueing problem* $\Gamma = \langle \mathbf{N}, m, \bar{\mathbf{\Theta}} \rangle$ *is first best implementable only if it satisfies the GCP.*

Before proving Theorem 3.1, a Lemma due to Walker (1980) is stated below. Consider two profiles $\underline{\theta} = (\underline{\theta}_1, \ldots, \underline{\theta}_n)$ and $\underline{\theta}' = (\underline{\theta}'_1, \ldots, \underline{\theta}'_n)$. Define

for $S \subseteq \mathbf{N}$, a type $\underline{\theta}_j(S) = \underline{\theta}_j$ if $j \notin S$ and $\underline{\theta}_j(S) = \underline{\theta}'_j$ if $j \in S$. Thus for each $S \subseteq \mathbf{N}$, we have a state $\underline{\theta}(S) = (\underline{\theta}_1(S), \ldots, \underline{\theta}_n(S))$.

**LEMMA 3.1** *A multiple machine queueing problem* $\Gamma = \langle \mathbf{N}, m, \bar{\mathbf{\Theta}} \rangle$ *is first best implementable only if for all pair of states* $(\underline{\theta}, \underline{\theta}') \in \bar{\mathbf{\Theta}}^n \times \bar{\mathbf{\Theta}}^n$, $\sum_{S \subseteq \mathbf{N}} (-1)^{|S|} C(\underline{\theta}(S)) = 0$.[1]

By adding the Groves-Clarke transfer of all individuals and setting it to zero we get $(n-1)C(\underline{\theta}) = \sum_{j \in \mathbf{N}} \gamma_j(\underline{\theta}_{-j})$ (see Holmström (1977) for a more general result). Thus, for all $\{\underline{\theta}, \underline{\theta}'\} \in \bar{\mathbf{\Theta}}^n \times \bar{\mathbf{\Theta}}^n$, $\sum_{S \subseteq \mathbf{N}} (-1)^{|S|} C(\underline{\theta}(S)) = \frac{1}{(n-1)} \sum_{j \in \mathbf{N}} \sum_{S \subseteq \mathbf{N}} (-1)^{|S|} \gamma_j(\underline{\theta}_{-j}(S)) = 0$ (see Walker (1980)).

**PROOF OF THEOREM 3.1:** We start with a given type for individual 1 (that is $\underline{\theta}_1$) and construct $\underline{\theta}_{-1}$ and $\underline{\theta}'$. Then we apply Lemma 3.1 due to Walker (1980) to derive the result. Consider individual 1 and any announcement $\underline{\theta}_1 = (\theta_1(1), \ldots, \theta_1(k), \ldots, \theta_1(M))$. Let $\{\epsilon_2, \ldots, \epsilon_n, \eta\}$ be a set of numbers such that $\epsilon_2 < \ldots < \epsilon_n$ and $0 \leq \eta \leq \theta_1(1)$. Given $\underline{\theta}_1$, we consider two states $\underline{\theta} = (\underline{\theta}_1, \underline{\theta}_2, \ldots, \underline{\theta}_n)$ and $\underline{\theta}' = (\underline{\theta}'_1, \ldots, \underline{\theta}'_n)$ of the following type: for all $k = 1, \ldots, M$, $\theta_j(k) = \theta_1(k) + k\epsilon_j$ for all $j \neq 1$ and $\theta'_j(k) = \eta$, for all $j \in \mathbf{N}$. Therefore, $\underline{\theta}_j = (\theta_1(1) + \epsilon_j, \theta_1(2) + 2\epsilon_j, \ldots, \theta_1(M) + M\epsilon_j)$ for all $j \neq 1$ and $\underline{\theta}'_j = (\eta, \eta, \ldots, \eta)$ for all $j \in \mathbf{N}$. Consider any two queue positions $k$ and $k+1$ and any two individuals $j$ and $j+1$ with types $\underline{\theta}_j$ and $\underline{\theta}_{j+1}$, respectively. Note that from the construction of $\underline{\theta}$, on the one hand, it follows that if individual $j$ gets the $k$th position and $(j+1)$th individual gets the $(k+1)$th position, then the costs for these two positions add up to $\{2\theta_1(k) + k\epsilon_j + (k+1)\epsilon_{j+1}\}$. If, on the other hand, the positions of $j$ and $(j+1)$ are interchanged then the costs add up to $\{2\theta_1(k) + (k+1)\epsilon_j + k\epsilon_{j+1}\}$. The former cost exceeds the latter for all $k = 1, \ldots, M-1$ since from the construction it follows that $\epsilon_j < \epsilon_{j+1}$ for all $j = 1, \ldots n-1$. Thus the queue that minimises the aggregate cost requires that, $\sigma^*_{j+1}(\underline{\theta}) \leq \sigma^*_j(\underline{\theta})$ for all $j = 1, \ldots, n-1$. In other words, $\sigma^*_j(\underline{\theta}) = [(n+1-j)/m]_+$ for all $j \in \mathbf{N}$. Now consider profiles $\underline{\theta}(S) = (\underline{\theta}_1(S), \ldots, \underline{\theta}_n(S))$ where $\underline{\theta}_j(S) = \underline{\theta}_j$ if $j \notin S$ and $\underline{\theta}_j(S) = \underline{\theta}'_j$ if $j \in S$. Observe that from the arguments applied to find the efficient queue in state

[1] Here $|X|$ denotes the cardinality of $X$.

$\underline{\theta}$ it follows that if $\{j,l\} \notin S$ and $j < l$, then $\sigma_l^*(\underline{\theta}(S)) \leq \sigma_j^*(\underline{\theta}(S))$. Again, given any $S \subseteq \mathbf{N}$ and $S \neq \phi$, $\sigma_j^*(\underline{\theta}(S)) \leq \sigma_s^*(\underline{\theta}(S))$ for all $\{s,j\} \in S \times \mathbf{N} - S$. This is because the queueing costs of all individuals $j \notin S$, in all queue positions, is not less than the queueing costs of all individual $s \in S$. Note that all queue $\sigma^*(\underline{\theta}(S))$, satisfying (1) $\sigma_j^*(\underline{\theta}(S)) \leq \sigma_s^*(\underline{\theta}(S))$ for all $\{s,j\} \in S \times \mathbf{N} - S$ and (2) $\sigma_l^*(\underline{\theta}(S)) \leq \sigma_j^*(\underline{\theta}(S))$ for all $\{j,l\} \in \mathbf{N} - S \times \mathbf{N} - S$, $j < l$, are efficient since the cost of all individuals $s \in S$ are identical. From this argument it follows that if $S = \mathbf{N}$, that is, if $\underline{\theta}(S) = \underline{\theta}'$, then any queue is efficient.

We now consider the sum $\sum_{S \subseteq \mathbf{N}} (-1)^{|S|} C(\underline{\theta}(S))$. Consider an individual $j \in \mathbf{N}$ with $j \neq 1$. Let $P_j = \{p : p > j\}$ be the set of individuals with the higher ranking index than $j$. Consider all sets $\bar{S}$ such that $j \notin \bar{S}$ and there are $x$ number of individuals from the set $P_j$ and $|\bar{S}| - x$ number of individuals from the set $\mathbf{N} - \{P_j \cup j\}$. The queue position of individual $j$ for all such $\bar{S}$ is $\sigma_j^*(\underline{\theta}(\bar{S})) = [(n+1-j-x)/m]_+$. By collecting all such sets $\bar{S}$, that is, by considering the coefficient of the term $\theta_j([(n+1-j-x)/m]_+)$, in the sum $\sum_{S \subseteq \mathbf{N}} (-1)^{|S|} C(\underline{\theta}(S))$ we get $(-1)^x \binom{n-j}{x} \sum_{T \subseteq \mathbf{N} - \{P_j \cup j\}} (-1)^{|T|}$. Note that $\sum_{T \subseteq \mathbf{N} - \{P_j \cup j\}} (-1)^{|T|} = \sum_{r=0}^{j-1} (-1)^r \binom{j-1}{r} = (1 + (-1))^{j-1} = 0$ since $j \neq 1$. Therefore, in $\sum_{S \subseteq \mathbf{N}} (-1)^{|S|} C(\underline{\theta}(S))$, the coefficient of a term $\theta_j([(n+1-j-x)/m]_+)$ is zero for all $j (\neq 1)$ and for all $x (\leq |P_j|)$. Hence, *the sum* $\sum_{S \subseteq \mathbf{N}} (-1)^{|S|} C(\underline{\theta}(S))$ *is independent of all elements in the set of vectors* $\{\underline{\theta}_2, \ldots, \underline{\theta}_n\}$. Observe that by adding the cost of an individual $j \in \mathbf{N}$ for all $\hat{S} (\subseteq \mathbf{N})$ such that $j \in \hat{S}$, we get $\sum_{j \in \hat{S} \subseteq \mathbf{N}} (-1)^{|\hat{S}|} \eta = \sum_{s=1}^{n} (-1)^s \binom{n-1}{s-1} \eta = -(1 + (-1))^{n-1} \eta = 0$. Therefore, *the sum* $\sum_{S \subseteq \mathbf{N}} (-1)^{|S|} C(\underline{\theta}(S))$ *is also independent of* $\eta$.

Finally, from the above observations, we get $\sum_{S \subseteq \mathbf{N}} (-1)^{|S|} C(\underline{\theta}(S)) = \sum_{S \subseteq \mathbf{N} - \{1\}} (-1)^{|S|} \theta_1(\sigma_1^*(\underline{\theta}_1, \underline{\theta}_{-1}(S)))$. For individual 1, with type $\underline{\theta}_1$, we get $\sigma_1^*(\underline{\theta}_1, \underline{\theta}_{-1}(S)) = [(n - |S|)/M]_+$ for all $S \subseteq \mathbf{N} - \{1\}$. Therefore,

$$\sum_{S \subseteq \mathbf{N} - \{1\}} (-1)^{|S|} \theta_1(\sigma_1^*(\underline{\theta}_1, \underline{\theta}_{-1}(S))) = \sum_{k=1}^{M} \left[ \sum_{x:\, [(n-x)/m]_+ = k} (-1)^x \binom{n-1}{x} \right] \theta_1(k) \quad (3.5)$$

Simplifying condition (3.5) using $\binom{n-1}{x} = \binom{n-1}{n-x}$ and then by applying Lemma 3.1 we get $(-1)^{n-1} \sum_{k=1}^{M} \alpha(k; n, m) \theta_1(k) = 0$. Since the selection of

individual 1, for the above construction, was arbitrary the result follows. ∎

Note that the GCP is a restriction on $\bar{\mathbf{\Theta}}$. It may be the case that with the GCP and assumption I, the only admissible cost vectors are those that are time independent. This would mean that there is no incentive problem. To eliminate such possibilities we consider the assumption of non-trivial domain (or NTD).

**Assumption NTD:** For all $j \in \mathbf{N}$, $\mathbf{\Theta}$ contains at least one cost vector $\underline{\theta}_j = (\theta_j(1), \ldots, \theta_j(M))$, such that $\theta_j(1) < \theta_j(M)$.

Let $\hat{\mathbf{\Theta}}(\subset \bar{\mathbf{\Theta}})$ be the largest domain satisfying the GCP. We now consider all multiple machine queueing problems $\Gamma = \langle \mathbf{N}, m, \hat{\mathbf{\Theta}} \rangle$. Observe that, for a multiple machine queueing problem $\Gamma = \langle \mathbf{N}, m, \hat{\mathbf{\Theta}} \rangle$, Groves-Clarke mechanisms are the only class of mechanisms that lead to truthful revelation of private information since $\hat{\mathbf{\Theta}}$ is a "convex" domain. Thus, for the existence of a first best implementable multiple machine queueing problem, it is necessary that the domain $\hat{\mathbf{\Theta}}$ satisfies assumption NTD.

**PROPOSITION 3.1** *If the number of machines $m$ is even or if the number of queue positions $M = [n/m]_+ = 2$, then $\Gamma = \langle \mathbf{N}, m, \hat{\mathbf{\Theta}} \rangle$ is not first best implementable.*

**PROOF:** If the number of machines $m$ is even then for all $k \in \{1, \ldots, M\}$, $(-1)^{km-1} = -1$ because $km$ is also even. Moreover, from assumption I it follows that for all $\underline{\theta}_j$, $\Delta\theta_j(k) \geq 0$ for all $k \in \{1, \ldots, M-1\}$. By substituting $(-1)^{km-1} = -1$ and $\Delta\theta_j(k) \geq 0$ in (3.4) we get $\Delta\theta_j(k) = 0$ for all $k \in \{1, \ldots, M-1\}$. Hence, $\theta_j(1) = \ldots = \theta_j(M)$ and we have a violation of assumption NTD.

If $M = 2$, then GCP implies $\alpha(1; n, m)\theta_j(1) + \alpha(2; n, m)\theta_j(2) = 0$ and condition (3.3) implies $\alpha(1; n, m) + \alpha(2; n, m) = 0$. Therefore, $\theta_j(1) = \theta_j(2)$. This again is a violation of assumption NTD. ∎

Given the impossibility results of Proposition 3.1, we consider the class multiple machine queueing problems with *odd* number of machines and with *at least three* queue positions. So far we have imposed a restriction on individual preferences (that is, the GCP). The next property is a restriction

on group preferences.

**DEFINITION 3.6** *A multiple machine queueing problem* $\Gamma = \langle \mathbf{N}, m, \mathbf{\Theta} \rangle$ *satisfies the General Independence Property (or GIP) if for all pairs* $\{j, l\} \in \mathbf{N} \times \mathbf{N}$, $j \neq l$, *all cost vectors* $\{\underline{\theta}_j, \underline{\theta}_l\} \in \mathbf{\Theta}^2$ *are such that one and only one of the following two conditions holds:*

1. $\Delta\theta_j(k) \geq \Delta\theta_l(k)$ *for all* $k \in \{1, 2, \ldots, M-1\}$
2. $\Delta\theta_j(k) \leq \Delta\theta_l(k)$ *for all* $k \in \{1, 2, \ldots, M-1\}$.

The GIP for a multiple machine queueing problem $\Gamma$ implies that if for any pair of individuals $\{j, l\} \in \mathbf{N} \times \mathbf{N}$, $j \neq l$, the respective cost vectors $\theta_j = (\theta_j(1), \ldots, \theta_j(M)) \in \mathbf{\Theta}$ and $\theta_l = (\theta_l(1), \ldots, \theta_l(M)) \in \mathbf{\Theta}$ are such that there exists a $\bar{k} \in \{1, \ldots, M-1\}$ such that $\Delta\theta_j(\bar{k}) > \Delta\theta_l(\bar{k})$, then $\Delta\theta_j(k) \geq \Delta\theta_l(k)$ for all $k \in \{1, \ldots, M-1\}$. The relationship between the GCP and the GIP is captured in the next proposition.

**PROPOSITION 3.2** *For* $\Gamma = \langle \mathbf{N}, m, \mathbf{\Theta} \rangle$ *with* $M = 3$, *GCP*$\Rightarrow$*GIP.*

**PROOF:** Consider $\Gamma = \langle \mathbf{N}, m, \mathbf{\Theta} \rangle$ such that $M = 3$ and any two cost vectors $\underline{\theta}_j = (\theta_j(1), \theta_j(2), \theta_j(3))$ and $\underline{\theta}_l = (\theta_l(1), \theta_l(2), \theta_l(3))$ for individuals $j$ and $l$ respectively. From condition 3.3 we know that $z(1; n, m)\Delta\theta_i(1) + z(2; n, m)\Delta\theta_i(2) = 0$ for all $i \in \{j, l\}$. Therefore, for all $i \in \{j, l\}$, $\Delta\theta_i(1) = \alpha\Delta\theta_i(2)$ where $\alpha = -\frac{z(2;n,m)}{z(1;n,m)} > 0$. Thus, $\Delta\theta_j(1) < (>)\Delta\theta_l(1)$ if and only if $\Delta\theta_j(2) < (>)\Delta\theta_l(2)$. ∎

Consider $\hat{\Gamma} = \langle \mathbf{N} = \{1, \ldots, 10\}, m = 3, \mathbf{\Theta} \rangle$ of Example 3.1. Here the number of queue positions $M = 4$ and $\alpha(10, 3) = (28, -84, 57, -1)$. Consider individuals $j$ and $l$ with costs $\underline{\bar{\theta}}_j = (1, 3, 4, 4) \in \mathbf{\Theta}$ and $\underline{\bar{\theta}}_l = (1, 2, 3, 31) \in \mathbf{\Theta}$ respectively. Observe that for $i \in \{j, l\}$, $\sum_{k=1}^{4} \alpha(k; 10, 3)\bar{\theta}_i(k) = 0$. However, $\Delta\bar{\theta}_j(1) = 2 > \Delta\bar{\theta}_l(1) = 1$ and $\Delta\bar{\theta}_j(3) = 0 < \Delta\bar{\theta}_l(3) = 28$. Therefore, the GIP is not satisfied. Hence *for* $\hat{\Gamma}$, *GCP* $\not\Rightarrow$ *GIP*. Let $\mathbf{\Gamma}$ represent the class of multiple machine queueing problems (with odd number of machines and with at least three queue positions) satisfying both the GCP and the GIP. Therefore, $\bar{\Gamma} = \langle \mathbf{N}, m, \tilde{\mathbf{\Theta}} \rangle$ belongs to $\mathbf{\Gamma}$ if $\tilde{\mathbf{\Theta}} (\subseteq \hat{\mathbf{\Theta}})$ is a domain satisfying

the GIP and assumption NTD. Note that $\tilde{\mathbf{\Theta}}$ is any domain (and not necessarily the largest domain) which is a subset of $\hat{\mathbf{\Theta}}$ and satisfies the GIP and assumption NTD. We now derive the efficient rule for any multiple machine queueing problem $\bar{\Gamma} = \langle \mathbf{N}, m, \tilde{\mathbf{\Theta}} \rangle \in \mathbf{\Gamma}$. Before doing that we give some more relevant notations and definitions. Consider $\bar{\Gamma} = \langle \mathbf{N}, m, \tilde{\mathbf{\Theta}} \rangle$. For a state $\underline{\theta} \in \tilde{\mathbf{\Theta}}^n$, define $Q_j(\underline{\theta}) = [l \in \mathbf{N} - \{j\}$ s.t. *either* $\exists k \in \{1, \ldots, M-1\}$ and $\{\Delta\theta_l(k) > \Delta\theta_j(k)\}$ *or* $\forall k \in \{1, 2, \ldots, M-1\}$, $\{\Delta\theta_l(k) = \Delta\theta_j(k)$ and $l < j\}]$. Let $R_j(\underline{\theta}) = 1 + |Q_j(\underline{\theta})| (\in \{1, \ldots, n\})$ be the *rank of individual $j$ in state $\underline{\theta}$*. Observe that the way we have specified the ranking, there is no possibility of a tie in the ranking of different individuals in any given state. Therefore, $R_j(\underline{\theta})$ measures the rank of individual $j$ in state $\underline{\theta}$. Using this definition of ranking we state and prove an *efficient rule* (that is, a single-valued selection from the efficiency correspondence) for $\bar{\Gamma} = \langle \mathbf{N}, m, \tilde{\mathbf{\Theta}} \rangle \in \mathbf{\Gamma}$.

**PROPOSITION 3.3** *Consider $\bar{\Gamma} = \langle \mathbf{N}, m, \tilde{\mathbf{\Theta}} \rangle \in \mathbf{\Gamma}$. For all $\underline{\theta} \in \tilde{\mathbf{\Theta}}^n$, let $\sigma^*(\underline{\theta}) = (\sigma_1^*(\underline{\theta}), \ldots, \sigma_n^*(\underline{\theta}))$ be the queue such that $\sigma_j^*(\underline{\theta}) = [R_j(\underline{\theta})/m]_+$ for all $j \in \mathbf{N}$. The queue $\sigma^*(\underline{\theta}) = (\sigma_1^*(\underline{\theta}), \ldots, \sigma_n^*(\underline{\theta}))$ is efficient.*

**PROOF:** We prove Proposition 3.3 by contradiction. If the statement in Proposition 3.3 is false, then there exists a state $\underline{\theta} \in \tilde{\mathbf{\Theta}}^n$, such that the queue $\sigma^*(\underline{\theta}) = (\sigma_1^*(\underline{\theta}), \ldots, \sigma_n^*(\underline{\theta}))$ is not efficient. Therefore, there exists another ordering $\hat{\sigma} = (\hat{\sigma}_1, \ldots, \hat{\sigma}_n) \in P(\mathrm{X}_{n,m}) - \{\sigma^*(\underline{\theta})\}$ which is efficient and hence yields a strictly lower cost than $\sigma^*(\underline{\theta})$. It must be the case that for $\hat{\sigma}$, there exists at least one pair $\{j, l\}$, $j \neq l$ and a queue position $k \in \{1, \ldots, M-1\}$, such that $\Delta\theta_j(k) > \Delta\theta_l(k)$ and $\hat{\sigma}_j = \hat{\sigma}_l + 1 > \hat{\sigma}_l = k$.[2] Now, consider another queue $\bar{\sigma} \in P(\mathrm{X}_{n,m})$ such that $\hat{\sigma}_j = \bar{\sigma}_l$, $\hat{\sigma}_l = \bar{\sigma}_j$ and $\hat{\sigma}_i = \bar{\sigma}_i$ for all $i \neq j \neq l \neq i$. Thus, $\bar{\sigma}$ is obtained from $\hat{\sigma}$ by interchanging the queue positions of $j$ and $l$ only. Note that the cost difference $C(\hat{\sigma}; \underline{\theta}) - C(\bar{\sigma}; \underline{\theta}) = \Delta\theta_j(k) - \Delta\theta_l(k) > 0$. This contradicts our assumption that $\hat{\sigma}$ is efficient. ∎

Therefore, finding a queue that ensures efficiency or minimises the aggregate cost is quite transparent if $\bar{\Gamma} \in \mathbf{\Gamma}$. Observe that for $\bar{\Gamma} \in \mathbf{\Gamma}$, the

[2] If not, then for all $\{j, l\} \in \mathbf{N} \times \mathbf{N}$, $j \neq l$, $\Delta\theta_j(k) \geq \Delta\theta_l(k) \Rightarrow \hat{\sigma}_j \leq \hat{\sigma}_l$. It is easy to check that in that case $C(\hat{\sigma}; \underline{\theta}) = C(\sigma^*(\underline{\theta}); \underline{\theta})$.

relative ranking of any two individuals $\{j,l\}$, for some given costs $\underline{\theta}_j$ and $\underline{\theta}_l$ respectively, is independent of the costs announced by all other individuals. Formally, if in state $\underline{\theta} = (\underline{\theta}_1, \ldots, \underline{\theta}_n)$, $R_j(\underline{\theta}) > R_l(\underline{\theta})$ for some $\{j,l\} \in \mathbf{N} \times \mathbf{N}$, $j \neq l$, then $R_j(\underline{\theta}_j, \underline{\theta}_l, \underline{\theta}'_{-j-l}) > R_l(\underline{\theta}_j, \underline{\theta}_l, \underline{\theta}'_{-j-l})$ for all $\underline{\theta}'_{-j-l} \in \tilde{\mathbf{\Theta}}^{n-2}$. Hence, what determines the efficient queue is the ranking that each individual gets in a given state. We now argue that if one individual is eliminated from the queue then the relative ranking of all other individuals remain unchanged. Before doing that we introduce some more relevant notations and definitions that captures the idea of elimination of an individual from the queue in any given state. Define $M' \equiv [\frac{n-1}{m}]_+$ to be the number of queue positions that remains in a multiple machine queueing problem with $n$ jobs and $m$ machines after one individual (and hence one job) is eliminated from the queue. Observe that $M' = M - 1$ if $n = rm + 1$ where $r = 2, 3, \ldots$ and $M' = M$ otherwise. Using the idea of ranking of individuals for $\bar{\Gamma} \in \mathbf{\Gamma}$, we define $Q_j(\underline{\theta}_{-l}) = [i \in \mathbf{N} - \{j,l\}$ s.t. *either* $\exists k \in \{1, \ldots, M' - 1\}$ and $\{\Delta\theta_i(k) > \Delta\theta_j(k)\}$ *or* $\forall k \in \{1, 2, \ldots, M' - 1\}$, $\{\Delta\theta_i(k) = \Delta\theta_j(k)$ and $i < j\}]$. Let $R_j(\underline{\theta}_{-l}) = 1 + |Q_j(\underline{\theta}_{-l})| (\in \{1, \ldots, n-1\})$. Therefore, in a state $\underline{\theta}$, $R_j(\underline{\theta}_{-l})$ measures the rank of individual $j$ in state $\underline{\theta}$ by eliminating the cost vector $\underline{\theta}_l$ of individual $l (\neq j)$.

**REMARK 3.2** *Consider any multiple machine queueing problem* $\Gamma$ *satisfying the GIP but not the GCP such that* $M' = M - 1$. *Consider a state* $\underline{\theta} = (\underline{\theta}_1, \ldots, \underline{\theta}_n)$ *such that* $\Delta\theta_1(k) = \Delta\theta_2(k)$ *for all* $k \in \{1, \ldots, M-2\}$ *and* $\Delta\theta_1(M-1) < \Delta\theta_2(M-1)$. *Moreover, assume that* $R_1(\underline{\theta}) = n$ *and* $R_2(\underline{\theta}) = n - 1$. *Hence* $R_1(\underline{\theta}) > R_2(\underline{\theta})$. *Observe, that for all* $i \in \mathbf{N} - \{1,2\}$, $R_1(\underline{\theta}_{-i}) = n - 2$ *and* $R_2(\underline{\theta}_{-i}) = n - 1$ *because* $\Delta\theta_1(k) = \Delta\theta_2(k)$ *for all* $k \in \{1, \ldots, M-2\}$ *and* $1 < 2$. *Therefore,* $R_1(\underline{\theta}) > R_2(\underline{\theta})$ *and for all* $i \in \mathbf{N} - \{1,2\}$, $R_1(\underline{\theta}_{-i}) < R_2(\underline{\theta}_{-i})$. *Therefore, for any multiple machine queueing problem* $\Gamma$ *satisfying the GIP but not the GCP, if* $M' = M - 1$, *then the above construction shows that there exist cost vectors for which the relative ranking of a pair of individuals can change if an individual outside the pair under consideration is eliminated from the queue. It is not hard to verify that the construction specified above is the only type of construction that can lead to such a rank reversal in a multiple machine queueing prob-*

*lem* $\Gamma$ *satisfying the GIP when an individual is eliminated. Moreover, such a rank reversal can take place only if* $M' = M - 1$ *and* $\Gamma$ *fails to satisfy the GCP. If, a multiple machine queueing problem* $\Gamma$ *satisfies the GIP and the GCP and if* $\Delta\theta_j(k) = \Delta\theta_l(k)$ *for all* $k \in \{1, \ldots, M-2\}$ *then the GCP implies that* $\Delta\theta_1(M-1) = \Delta\theta_2(M-1)$. *Therefore, the construction that led to rank reversal is not possible for a multiple machine queueing problem that satisfies the GCP. Hence, if a multiple machine queueing problem* $\Gamma$ *satisfies both the GCP and the GIP, then the relative ranking of any pair of individuals with any given pair of cost vectors remain unchanged if some other individual is eliminated from the queue. More formally, if* $\bar{\Gamma} \in \mathbf{\Gamma}$, *then we obtain the following relationship between* $R_j(\underline{\theta})$ *and* $R_j(\underline{\theta}_{-l})$. *For all* $\{j, l\} \in \mathbf{N} \times \mathbf{N}$, $j \neq l$ *and for all* $\underline{\theta} \in \tilde{\mathbf{\Theta}}^n$,

$$R_j(\underline{\theta}_{-l}) = \begin{cases} R_j(\underline{\theta}) & \text{if } R_j(\underline{\theta}) < R_l(\underline{\theta}) \\ R_j(\underline{\theta}) - 1 & \text{if } R_j(\underline{\theta}) > R_l(\underline{\theta}) \end{cases}$$

Using Remark 3.2 we derive the sufficiency condition under which a multiple machine queueing problem $\bar{\Gamma} \in \mathbf{\Gamma}$ is first best implementable.

**THEOREM 3.2** *A multiple machine queueing problem* $\bar{\Gamma} \in \mathbf{\Gamma}$ *is first best implementable.*

We first state and prove a lemma that will be used in proving Theorem 3.2.

**LEMMA 3.2** *A multiple machine queueing problem* $\Gamma = \langle \mathbf{N}, m, \mathbf{\Theta} \rangle$ *satisfies the GCP, if and only if for all cost vector* $\underline{\theta}_j \in \mathbf{\Theta}$, *there exists a unique* $1 \times (n-1)$ *vector* $H_j = \{h_j(1), \ldots, h_j(n-1)\}$ *such that for all* $p \in \{1, \ldots, n\}$,

$$\theta_j([p/m]_+) = (n-p)h_j(p) + (p-1)h_j(p-1). \tag{3.6}$$

**PROOF:** Consider a cost vector $\underline{\theta}_j \in \mathbf{\Theta}$ for individual $j \in \mathbf{N}$ that satisfies $\sum_{k=1}^{M} \alpha(k; n, m)\theta_j(k) = 0$. Define a vector $H_j = \{h_j(1), \ldots, h_j(n-1)\}$ such that for all $p \in \{1, \ldots, n-1\}$,

$$h_j(p) = \sum_{r=1}^{p} (-1)^{p-r} \frac{(k-1)!(n-p-1)!}{(r-1)!(n-r)!} \theta_j([r/m]_+) \tag{3.7}$$

We prove Lemma 3.2 in two steps. In the first step it is proved, using (3.7), that for all $p \in \{1, \ldots, n-1\}$, condition (3.6) holds. In the next step we prove that for $p = n$, condition (3.6) holds *only if* $\Gamma$ satisfies the GCP.

$$\begin{aligned}
&(n-p)h_j(p) + (p-1)h_j(p-1) \\
&\quad = (n-p) \sum_{r=1}^{p} (-1)^{p-r} \tfrac{(p-1)!(n-p-1)!}{(r-1)!(n-r)!} \theta_j([r/m]_+) \\
&\qquad +(p-1) \sum_{r=1}^{p-1} (-1)^{p-r-1} \tfrac{(p-2)!(n-p)!}{(r-1)!(n-r)!} \theta_j([r/m]_+) \\
&\quad = \sum_{r=1}^{p-1} \{(-1)^{p-r} + (-1)^{p-r-1}\} \tfrac{(p-1)!(n-p)!}{(r-1)!(n-r)!} \theta_j([r/m]_+) + \theta_j([p/m]_+) \\
&\quad = \theta_j([p/m]_+) \text{ (because } (-1)^{p-r} + (-1)^{p-r-1} = 0).
\end{aligned}$$

For $p = n$,

$$\begin{aligned}
&(n-p)h_j(p) + (p-1)h_j(p-1) \\
&\qquad = (n-1)h_j(n-1) \\
&\qquad = (n-1) \sum_{r=1}^{n-1} (-1)^{n-1-r} \tfrac{(n-2)!}{(r-1)!(n-r)!} \theta_j([r/m]_+) \\
&\qquad = \sum_{r=1}^{n-1} (-1)^{n-1-r} \tfrac{(n-1)!}{(r-1)!(n-r)!} \theta_j([r/m]_+) \\
&\qquad = \sum_{r=1}^{n-1} (-1)^{n-1-r} \tbinom{n-1}{r-1} \theta_j([r/m]_+) \\
&\qquad = (-1)^{n-2} \sum_{k=1}^{M} \alpha(k; n, m)\theta_j(k) + \theta_j(M) \text{ (From the GCP)} \\
&\qquad = \theta_j(M).
\end{aligned}$$

Therefore, the last step not only proves the necessity of the GCP but also guarantees that for $\underline{\theta}_j$, the $1 \times (n-1)$ vector $H_j$ is unique.

We now prove the other part of Lemma 3.2. Observe that the sum $\sum_{k=1}^{M} \alpha(k;n,m)\theta_j(k) = \sum_{p=1}^{n}(-1)^{p-1}\binom{n-1}{p-1}\theta_j([p/m]_+)$. Therefore,

$$\begin{aligned}
&\sum_{p=1}^{n}(-1)^{p-1}\tbinom{n-1}{p-1}\theta_j([p/m]_+) \\
&\qquad = \sum_{p=1}^{n}(-1)^{p-1}\tbinom{n-1}{p-1}\{(n-p)h_j(p) + (p-1)h_j(p-1)\} \\
&\qquad = (n-1)\left\{\sum_{p=1}^{n-1}(-1)^{p-1}\tbinom{n-2}{p-1}h_j(p) + \sum_{p=2}^{n}(-1)^{p-1}\tbinom{n-2}{p-2}h_j(p-1)\right\} \\
&\qquad = 0.
\end{aligned}$$

∎

Lemma 3.2 gives rise to a particular type of separability (as given by condition 3.6) that will be used in deriving the explicit form of the transfer that first best implements any $\bar{\Gamma} = \langle \mathbf{N}, m, \tilde{\mathbf{\Theta}} \rangle \in \mathbf{\Gamma}$.

**PROOF OF THEOREM 3.2:** Consider the sum $\sum_{l\neq j} h_j(R_j(\underline{\theta}_{-l}))$ in state $\underline{\theta} \in \tilde{\mathbf{\Theta}}^n$ for individual $j \in \mathbf{N}$. From the GIP and Remark 3.2 we get

$$\begin{aligned}
\sum_{l\neq j} h_j(R_j(\underline{\theta}_{-l})) &= (n - R_j(\underline{\theta}))h_j(R_j(\underline{\theta})) + (R_j(\underline{\theta}) - 1)h_j(R_j(\underline{\theta}) - 1) \\
&= \theta_j([R_j(\underline{\theta})/m]_+) \text{ (From condition (3.6) in Lemma 3.2).}
\end{aligned}$$

We consider a Groves-Clarke mechanism $\hat{\mathbf{M}} = \langle \sigma^*, \hat{\mathbf{t}} \rangle$ where the term independent of $j$'s announcement is $\hat{\gamma}_j(\underline{\theta}_{-j}) = (n-1)\sum_{l\neq j} h_l(R_l(\underline{\theta}_{-j}))$. Then it follows that

$$\begin{aligned}
\sum_{j\in\mathbf{N}} \hat{\gamma}_j(\underline{\theta}_{-j}) &= (n-1)\sum_{j\in\mathbf{N}}\sum_{l\neq j} h_l(R_l(\underline{\theta}_{-j})) \\
&= (n-1)\sum_{j\in\mathbf{N}}\{\sum_{l\neq j} h_j(R_j(\underline{\theta}_{-l}))\} \\
&= (n-1)\sum_{j\in\mathbf{N}} \theta_j([R_j(\underline{\theta})/m]_+) \\
&= (n-1)C(\underline{\theta}).
\end{aligned}$$

Observe that the last step follows from the efficiency rule of Proposition 3.3. The last step implies that for all $\underline{\theta} \in \tilde{\mathbf{\Theta}}^n$, the sum of transfers $\sum_{j\in\mathbf{N}} \hat{t}_j(\underline{\theta}) = -(n-1)C(\underline{\theta}) + \sum_{j\in\mathbf{N}} \hat{\gamma}_j(\underline{\theta}_{-j}) = 0$. ∎

Observe that from Theorems 3.1 and 3.2 and from Proposition 3.2 it follows that *a multiple machine queueing problem* $\bar{\Gamma} \in \mathbf{\Gamma}$ *with* $M = 3$ *is first best implementable if and only if it satisfies the GCP.*

## 4 Applications

In this section we first define a class of multiple machine queueing problems with separable cost and then verify under what conditions these problems are first best implementable. For separable cost multiple machine queueing problems, $\theta_j(k)$ satisfies the following conditions:

1. $\theta_j(k) = f(k)\theta_j$ for all $j \in \mathbf{N}$, for all $k \in \{1, 2, \ldots, M\}$ and for all $\theta_j \in \Theta$. Here $\Theta$ is assumed to be an interval in $\mathbf{R}_+$.

2. Finally, $f(k) \geq f(k-1)$ for all $k \in \{2, \ldots, M\}$ and $\theta_j \geq 0$ for all $\theta_j \in \Theta$.

Condition (1) multiplicatively separates the cost of each individual for each position into two parts. The first part is a function $f$ that depends on the queue position. Observe that the functional form $f$ is assumed to be identical for all individuals $j \in \mathbf{N}$. Moreover, we assume that $f$ is common knowledge. The second part which is a non-negative number $\theta_j$ represents the type (or cost parameter) of an individual and it is independent of queue position. In this set up a type vector of individual $j \in \mathbf{N}$ is given by $\underline{\theta}_j = (\theta_j(1) = f(1)\theta_j, \ldots, \theta_j(M) = f(M)\theta_j)$. Therefore, from now on we will write $\theta_j$ as the cost parameter or type of an individual. Condition (2) implies assumption I, that is, $\theta_j(k+1) \geq \theta_j(k)$ for all $k \in \{1, \ldots, M-1\}$. The cost parameter $\theta_j \in \Theta$ for all $j \in \mathbf{N}$ is private information. Finally, $\theta = (\theta_1, \ldots, \theta_n) \in \Theta^n$ represents a state of the world or a profile.

In this framework the set of individuals $\mathbf{N}$, $m(> 1)$ number of machines and $(f, \Theta)$ define the *multiple machine separable cost queueing problem* $\hat{\Gamma} = \langle \mathbf{N}, m, (f, \Theta) \rangle$. We will completely characterize the class of first best implementable multiple machine separable cost queueing problems. We start by showing that these problems satisfy the GIP.

**PROPOSITION 4.4** *A multiple machine separable cost queueing problem* $\hat{\Gamma} = \langle \mathbf{N}, m, (f, \Theta) \rangle$ *satisfies the GIP.*

**PROOF:** Consider any pair $\{j, l\} \in \mathbf{N} \times \mathbf{N}$, $j \neq l$, with cost parameters $\theta_j$ and $\theta_l$ respectively. It is obvious that either $\theta_j \geq \theta_l$ or $\theta_j \leq \theta_l$. Since $f(k+1) \geq f(k)$ for all $k \in \{1, \ldots, M-1\}$, it is also obvious that if $\theta_j \geq \theta_l$, then $\{f(k+1) - f(k)\}\theta_j \geq \{f(k+1) - f(k)\}\theta_l$ for all $k \in \{1, \ldots, M-1\}$. Therefore, $\Delta\theta_j(k) = \{f(k+1) - f(k)\}\theta_j \geq \Delta\theta_l(k) = \{f(k+1) - f(k)\}\theta_l$ for all $k \in \{1, \ldots, M-1\}$. Similarly, if $\theta_j \leq \theta_l$, then $\Delta\theta_j(k) \leq \Delta\theta_l(k)$ for all $k \in \{1, \ldots, M-1\}$. Thus, it follows that $\hat{\Gamma} = \langle \mathbf{N}, m, (f, \Theta) \rangle$ satisfies the GIP. ∎

The next two remarks follow trivially from the discussion of the GCP in the previous section.

**REMARK 4.3** *A multiple machine separable cost queueing problem* $\hat{\Gamma} = \langle \mathbf{N}, m, (f, \Theta) \rangle$ *satisfies the GCP if*

$$\sum_{k=1}^{M} \alpha(k; n, m) f(k) = 0 \tag{4.8}$$

*Using* $\Delta f(k) = f(k+1) - f(k)$ *and simplifying equation (4.8) we get*

$$\sum_{k=1}^{M-1} z(k; n, m) \Delta f(k) = 0 \tag{4.9}$$

*where* $z(k; n, m) = \sum_{r=1}^{k} \alpha(r; n, m) = (-1)^{km-1} \binom{n-2}{km-1}$.

**REMARK 4.4** *From condition (3.6) it follows that* $\hat{\Gamma} = \langle \mathbf{N}, m, (f, \Theta) \rangle$ *satisfies the GCP, if and only if there exists a unique vector* $H = \{h(1), \ldots, h(n-1)\}$ *such that for all* $p \in \{1, \ldots, n\}$,

$$f([p/m]_+) = (n-p)h(p) + (p-1)h(p-1) \tag{4.10}$$

*where* $h(p) = \sum_{r=1}^{p} (-1)^{p-r} \frac{(p-1)!(n-p-1)!}{(r-1)!(n-r)!} f([r/m]_+)$.

The next result completely characterizes the class of first best implementable multiple machine separable cost queueing problems.

**PROPOSITION 4.5** *A multiple machine separable cost queueing problem* $\hat{\Gamma} = \langle \mathbf{N}, m, (f, \Theta) \rangle$ *is first best implementable if and only if the cost function satisfies the GCP.*

The necessity part of Proposition 4.5 is similar to that of Theorem 3.1 and the sufficiency part is similar to Theorem 3.2. Therefore, we omit the proof of this proposition.

It is easy to verify that Proposition 3.1 is also true for separable cost queueing problems. Therefore, *all multiple machine separable cost queueing problems with either (1) even number of machines or (2) two queue positions are not first best implementable.* Let $\mathbf{\Gamma}(\mathbf{S})(\subset \mathbf{\Gamma})$ be the class of multiple machine separable cost queueing problems where $m$ is odd and $n > 2m$ and let $\mathbf{\Gamma}^*(\subset \mathbf{\Gamma}(\mathbf{S}))$ be the class of first best implementable multiple machine separable cost queueing problems. The next proposition proves the existence of $\mathbf{\Gamma}^*$.

**PROPOSITION 4.6** *There exists* $\Gamma \in \mathbf{\Gamma}(\mathbf{S})$ *such that* $\Gamma \in \mathbf{\Gamma}^*$.

**PROOF:** Consider $\hat{\Gamma}^* = \langle \mathbf{N}, m, (f^*, \Theta) \rangle \in \mathbf{\Gamma}(\mathbf{S})$ with odd number of queue positions $M$ and with $f^*$ of the following form: $f^*(1) = c \geq 0$ and $\Delta f^*(k) = f^*(k+1) - f^*(k) = \frac{1}{\binom{n-2}{km-1}}$ for all $k \in \{2, \ldots M-1\}$. We will prove that $\hat{\Gamma}^* = \langle \mathbf{N}, m, (f^*, \Theta) \rangle \in \mathbf{\Gamma}^*$ by showing that $f^*$ satisfies condition (4.9). Observe first that $z(k; n, m)\Delta f^*(k) = (-1)^{km-1}$ for all $k \in \{1, \ldots, M-1\}$. Therefore, by substituting $z(k; n, m)\Delta f^*(k) = (-1)^{km-1}$ in the left hand side of condition (4.9) we get $\sum_{k=1}^{M-1}(-1)^{km-1}$. Since both $m$ and $M$ are odd, it is obvious that, $\sum_{k=1}^{M-1}(-1)^{km-1} = 0$. Thus, $\hat{\Gamma}^* = \langle \mathbf{N}, m, (f^*, \Theta) \rangle \in \mathbf{\Gamma}^*$.

Similarly, consider $\hat{\Gamma}^{**} = \langle \mathbf{N}, m, (f^{**}, \Theta) \rangle \in \mathbf{\Gamma}(\mathbf{S})$ with even number of queue positions $M$ and with $f^{**}$ of the following form: $f^{**}(1) = c \geq 0$ and $\Delta f^{**}(k) = f^{**}(k+1) - f^{**}(k) = \frac{1}{\binom{n-2}{km-1}}$ for all $k \in \{2, \ldots M-2\}$ and $f^{**}(M-1) = f^{**}(M)$ (that is, $\Delta f^{**}(M-1) = 0$). Observe that $z(k; n, m)\Delta f^{**}(k) = (-1)^{km-1}$ for all $k \in \{1, \ldots, M-2\}$ and $z(M-1; n, m)\Delta f^{**}(M-1) = 0$. Therefore, by substituting $z(k; n, m)\Delta f^{**}(k)$ for all $k \in \{1, \ldots, M-1\}$ in the left hand side of condition (4.9) we get $\sum_{k=1}^{M-2}(-1)^{km-1}$. Here $\sum_{k=1}^{M-2}(-1)^{km-1} = 0$ because $m$ is odd and $M$ is even. Thus, $\hat{\Gamma}^{**} \in \mathbf{\Gamma}^*$. ∎

Observe that given $\mathbf{\Gamma}^* \subset \mathbf{\Gamma}(\mathbf{S}) \subset \mathbf{\Gamma}$, it follows from Proposition 4.6 that *there exist first best implementable multiple machine queueing problems in* $\mathbf{\Gamma}$. We conclude this section with an important observation.

**Observation [1]:** Given the co-efficient vector $\alpha(n,m)$, it follows that if $m$ is odd, $M = 2q+1$ and $n = m \times (2q+1)$ (where $q \in \{1,2,\ldots\}$) then $\alpha(k;n,m) = \alpha(2q+2-k;n,m)$ for all $k \in \{1,\ldots,q\}$. Using this result and by substituting $\alpha(q+1;n,m) = -2\sum_{k=1}^{q}\alpha(k;n,m)$ in equation (3.2) we get

$$\sum_{k=1}^{q}\alpha(k;n,m)\{\theta_j(k)+\theta_j(2q+2-k)-2\theta_j(q+1)\}=0 \tag{4.11}$$

Observe that if $\theta_j(k) = k\theta_j$ for all $k \in \{1,\ldots,M\}$, then condition (4.11) holds. Thus, *if* $m$ *is odd,* $n = mM$*,* $M$ *is also odd and* $\tilde{f}(k) = k$ *for all* $k$*, then* $\Gamma^L = \langle \mathbf{N}, m, (\tilde{f}, \Theta)\rangle \in \mathbf{\Gamma}^*$.

## 5 Concluding Remarks

We have obtained the following results regarding the first best implementability of a multiple machine queueing problem.

1. A multiple machine queueing problem is first best implementable only if it satisfies the GCP.

2. If the number of machines is even or if there are only two queue positions, then a multiple machine queueing problem fails to satisfy the GCP and hence is not first best implementable.

3. If the number of machines $m$ is odd and $M = [n/m]_+ = 3$ then a multiple machine queueing problem is first best implementable if and only if it satisfies the GCP and the GIP.

4. If the number of machines $m$ is odd and $M = [n/m]_+ \geq 4$ then a multiple machine queueing problem is first best implementable if it satisfies the GCP and the GIP.

5. For all $n > 2m$ such that $m$ is odd, there exists a cost function for which a multiple machine queueing problem is first best implementable.

6. Finally, if $m$ is odd, the number of queue positions $M = [n/m]_+$ is also odd and $n = m.M$ then a multiple machine queueing problem with linear cost function is first best implementable.

Thus, first best implementability of a multiple machine queueing problem depends heavily on the number of machines and on the number of jobs.

# REFERENCES


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

Curiel, I., Pederzoli, G., and Tijs, S. (1989). "Sequencing Games," *European Journal of Operational Research* **40,** 344-351.

Dudek, R. A., Panwalkar, S., Smith, M. (1975). "Flow-shop Sequencing Problem with Ordered Processing Time Matrices," *Management Science* **21,** 544-549.

Dudek, R. A., Teuton Jr., O. F. (1964). "Development of M-stage Decision Rule for Scheduling $n$ jobs through $M$ Machines," *Operations Research* **12,** 471-497.

Green, J., and Laffont, J. J. (1979). *Incentives in Public Decision Making.* North Holland Publication, Amsterdam.

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

Groves, T., and Ledyard, J. O. (1977). "Some Limitations of Demand Revealing Processes," *Public Choice* **29,** 107-124.

Groves, T., and Loeb, M. (1975). "Incentives and Public Inputs," *Journal of Public Economics* **4,** 211-226.

Hamers, H., Klijn, F., and Suijs, J. (1999). "On the Balancedness of Multiple Machine Sequencing Games," *European Journal of Operational Research* **119,** 678-691.

Holmström, B. (1977). "On Incentives and Control in Organisation", Ph.D Dissertation, Stanford Graduate School of Business.

Holmström, B. (1979). "Groves' Schemes on Restricted Domains," *Econometrica* **47,** 1137-1144.

Hurwicz, L. (1975). "On the Existence of Allocation Systems Whose Manipulative Nash Equilibria are Pareto Optimal," *mimeo* University of Minnesota.

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.

Krabbenborg, M., Potters, J., and van den Nouweland, A. (1992). "Flowshops with a Dominant Machine," *European Journal of Operational Research* **62,** 38-46.

Liu, L., and Tian, G. (1999). "A Characterization of the Existence of Optimal Dominant Strategy Mechanisms," *Review of Economic Design* **4,** 205-218.

Mitra, M., and Sen, A. (1998). "Dominant Strategy Implementation of First Best Public Decisions," *mimeo*: Indian Statistical Institute, New Delhi.

Mitra, M. (2001). "Mechanism Design in Queueing Problems," *Economic Theory* **17,** 277-305.

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

Tian, G. (1996). "On the Existence of Optimal Truth-Dominant Mechanisms," *Economic Letters* **53,** 17-24.

Tomescu, I. (1985). "Problems in Combinatorics and Graph Theory", *translated by R.A. Melter*, A Wiley-Interscience Publication.

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