Stochastics and Statistics

Incomplete information and multiple machine queueing problems

Manipushpak Mitra *

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 can find queueing problem with one machine that are first best implementable under certain cost conditions. 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

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 minimize the aggregate waiting cost. If the cost of waiting in the queue is private information then an individual, if asked, will announce her cost strategically so as to get her 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 minimize 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; 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 realizations 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 serve 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 generalized 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 minimizes 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 analyzed 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 et al. (1975) and flow shop problems with dominant machines by Krabbenborg et al. (1992) deal with finding algorithms to order (or queue) the n jobs in m machines in a way that minimizes 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 analyzed by Hamers et al. (1999). They analyzed 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 et al. (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 analyzed by Hamers et al. (1999) is identical to the multiple machine 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 generalize the non-cooperative sequencing situation, analyzed 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 non-trivial cost structure under which it is first best implementable. Finding an algorithm to obtain the cost minimizing 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 minimizing 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 organized in the following way: in Section 2 we develop the general class of problems, in Section 3 we derive some characterization results, in Section 4 we deal with separable cost multiple machine queueing problems and finally in the concluding remarks of Section 5 we summarize the results obtained in this paper.

2. The general class of problems

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 $k \in \{1, \ldots, M\}$. Let $\mathbf{R}_+$ be the non-negative orthant of the real line $\mathbf{R}$. A cost vector or type of an individual $j \in \mathbf{N}$ is a vector of the form $\underline{\theta}_j = (\theta_j(1), \ldots, \theta_j(M)) \in \mathbf{R}^M_+$. An ϵ-neighborhood around any vector $\underline{\theta}_j \in \mathbf{R}^M_+$ is the set $N_\epsilon(\underline{\theta}_j) = \{\underline{x} \in \mathbf{R}^M_+ : \|\underline{\theta}_j - \underline{x}\| < \epsilon\}$. [1] A typical domain of cost vectors of an individual $j \in \mathbf{N}$ is $\overline{\mathbf{\Theta}} \subset \mathbf{R}^M_+$. Observe that the domain $\overline{\mathbf{\Theta}}$ is assumed to be common for all individuals. For a domain $\overline{\mathbf{\Theta}}$, we denote its *interior* by $\mathbf{\Theta}$. A cost vector $\underline{\theta}_j \in \mathbf{\Theta}$, if $\exists\, \epsilon > 0$ such that $N_\epsilon(\underline{\theta}_j) \subset \overline{\mathbf{\Theta}}$. We assume that the interior $\mathbf{\Theta}$ (of the domain $\overline{\mathbf{\Theta}}$) is non-empty. Let $\Delta\underline{\theta}_j = (\Delta\theta_j(1), \ldots, \Delta\theta_j(M-1))$ represent the vector of first differences generated from the vector $\underline{\theta}_j$. Here $\Delta\theta_j(k) = \theta_j(k+1) - \theta_j(k)$ for all $k \in \{1, \ldots, M-1\}$. We say that $\Delta\underline{\theta}_j < \Delta\underline{\theta}'_j$ if and only if $\Delta\theta_j(k) \leqslant \Delta\theta'_j(k)$ for all $k \in \{1, \ldots, M-1\}$ and there exists at least one $k' \in \{1, \ldots, M-1\}$ such that

[1] Here $\|\underline{\theta}_j - \underline{x}\|$ denotes the Euclidean distance between the two cost vectors $\underline{\theta}_j = (\theta_j(1), \ldots, \theta_j(M))$ and $\underline{x} = (x(1), \ldots, x(M))$, that is $\|\underline{\theta}_j - \underline{x}\| = \sqrt{\sum_{k=1}^{M}(\theta_j(k) - x(k))^2}$.

$\Delta\theta_j(k') < \Delta\theta'_j(k')$. The three main assumptions we make on the common domain of preference $\overline{\Theta}$ are as follows.

Assumption 1. For all $j \in \mathbf{N}$ and for all cost vector $\underline{\theta}_j = (\theta_j(1), \ldots, \theta_j(M)) \in \overline{\Theta}$, $0 \leqslant \theta_j(1) \leqslant \theta_j(2) \leqslant \cdots \leqslant \theta_j(M)$.

Assumption 2. $\overline{\Theta}$ is *convex*, that is, if $\underline{\theta}_j \in \overline{\Theta}$ and $\underline{\theta}'_j \in \overline{\Theta}$, then $\lambda\underline{\theta}_j + (1-\lambda)\underline{\theta}'_j \in \overline{\Theta}$ for all $\lambda \in [0,1]$.

Assumption 3. $\overline{\Theta}$ is *sufficiently open* if for all $\underline{\theta}_j \in \dot{\overline{\Theta}}$, there exists $(\underline{\alpha}, \underline{\beta}) \in \overline{\Theta} \times \overline{\Theta}$ such that $\Delta\underline{\alpha} < \Delta\underline{\theta}_j < \Delta\underline{\beta}$. Moreover, $\Delta\theta_j(k) < \Delta\beta(k)$ for all $k \in \{1, \ldots, M-1\}$ and $\Delta\alpha(k) = 0$ for all $k \in \{1, \ldots, M-1\}$. [2]

Assumption 1 simply says that the individuals are impatient. Assumption 2 is the standard convexity requirement. Finally, Assumption 3 is an openness assumption on the cost domain which says that for any type $\underline{\theta}_j$, in the interior of the domain, there exists one type $\underline{\beta}$ in the domain that, in terms of difference, strictly dominates $\underline{\theta}_j$ and there exists another type $\underline{\alpha}$ with null first difference vector in the domain which is dominated, in terms of difference, by $\underline{\theta}_j$.

Let $\mathbf{\Theta}$ be the class of domains satisfying Assumptions 1–3. Let $\overline{\Theta}$ be any domain belonging to $\mathbf{\Theta}$ satisfying Assumptions 1–3. 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, $\theta_j(k)$ is the cost of individual j at the kth queue position and t_j is the transfer that individual j receives.

The server's aim is to achieve efficiency or minimize 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 σ is a one-to-one correspondence between the set of jobs $\mathbf{N} = \{1, \ldots, n\}$ and the queue positions $P(\mathrm{X}_{n,m})$, 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 kth 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 $\underline{\theta} = (\underline{\theta}_1, \ldots, \underline{\theta}_n) \in \overline{\Theta}^n$ where $\underline{\theta}_j$ is a $1 \times M$ vector.

Definition 2.1. Given a state $\underline{\theta} \in \overline{\Theta}^n$, a queue $\sigma^* \in P(\mathrm{X}_{n,m})$ is said to be *efficient* if $\sigma^* \in \arg\min_{\sigma \in P(\mathrm{X}_{n,m})} \sum_{j \in \mathbf{N}} \theta_j(\sigma_j)$.

An efficient queue σ^* 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 Mth queue position to the remaining $n-(M-1)m$ individuals in such a way that the aggregate cost is minimized. 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 σ^* 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 she can calculate an 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. We refer to such a problem as a *multiple machine queueing problem* under incomplete information and is written as $\Gamma = \langle \mathbf{N}, m, \overline{\Theta} \rangle$ where $\overline{\Theta} \in \mathbf{\Theta}$. Note that we are assuming that the domain

$\overline{\Theta}$ 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 $\overline{\Theta}$. Formally, a mechanism $\mathbf{M}$ is a pair $\langle \sigma, \mathbf{t} \rangle$ where $\sigma : \overline{\Theta}^n \to P(X_{n,m})$ and $\mathbf{t} \equiv (t_1, \ldots, t_n) : \overline{\Theta}^n \to \mathbf{R}^n$. 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 σ 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 her 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 $\Gamma = \langle \mathbf{N}, m, \overline{\Theta} \rangle$ is *implementable* if there exists an *efficient rule* $\sigma^* : \overline{\Theta}^n \to P(X_{n,m})$ and a mechanism $\mathbf{M} = \langle \sigma^*, \mathbf{t} \rangle$ such that for all $j \in \mathbf{N}$, for all pairs of announcement vectors $(\underline{\theta}_j, \underline{\theta}'_j) \in \overline{\Theta} \times \overline{\Theta}$ and for all announced $\underline{\theta}_{-j} \in \overline{\Theta}^{n-1}$, $U_j(\sigma^*_j(\underline{\theta}), t_j(\underline{\theta}); \underline{\theta}_j) \geqslant U_j(\sigma^*_j(\underline{\theta}'_j, \underline{\theta}_{-j}), t_j(\underline{\theta}'_j, \underline{\theta}_{-j}); \underline{\theta}_j)$.

This definition says that $\Gamma = \langle \mathbf{N}, m, \overline{\Theta} \rangle$ is implementable if there exists a direct mechanism, with an efficient queueing rule σ^* 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 $\Gamma = \langle \mathbf{N}, m, \overline{\Theta} \rangle$ is *first best implementable*, if there exists a mechanism $\mathbf{M} = \langle \sigma^*, \mathbf{t} \rangle$ such that (1) $\mathbf{M}$ implements Γ and (2) for all announcements $\underline{\theta} \in \overline{\Theta}^n$, $\sum_{j \in \mathbf{N}} t_j(\underline{\theta}) = 0$.

A multiple machine queueing problem $\Gamma = \langle \mathbf{N}, m, \overline{\Theta} \rangle$ is first best implementable if it can be implemented with a budget balancing transfer. Thus, if Γ 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 $\underline{\theta} \in \overline{\Theta}^n$, 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 her 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 σ^* is efficient 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 her 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 minimization) 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)). Since by Assumption 1, any domain $\overline{\Theta} \in \Theta$ is convex, all multiple machine queueing problems $\Gamma = \langle \mathbf{N}, m, \overline{\mathbf{\Theta}} \rangle$ with $\overline{\mathbf{\Theta}} \in \mathbf{\Theta}$ 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 domain restrictions 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 σ^* it follows that for all $\underline{\theta}$ and $\underline{\theta}'$, $C(\underline{\theta}) \leqslant C(\sigma^*(\underline{\theta}'); \underline{\theta})$.

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 $\Gamma = \langle \mathbf{N}, m, \overline{\mathbf{\Theta}} \rangle$ satisfies the *General Combinatorial Property* (or GCP) if for all $\theta_j \in \overline{\mathbf{\Theta}}$

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

where $\alpha(n, m) = \{\alpha(k; n, m)\}_{k=1}^{M}$ and

$$\alpha(k; n, m) = (-1)^{(k-1)m}\left\{\sum_{l=0}^{m-1} (-1)^l \binom{n-1}{(k-1)m+l}\right\} \quad \text{if } k \in \{1, \ldots, M-1\} \quad \text{and}$$

$$\alpha(k; n, m) = (-1)^{(M-1)m}\left\{\sum_{l=0}^{n-(M-1)m-1} (-1)^l \binom{n-1}{(M-1)m+l}\right\} \quad \text{if } k = M.$$

The following example illustrates the GCP.

Example 3.1. Consider $\widehat{\Gamma} = \langle \mathbf{N} = \{1, \ldots, 10\}, m = 3, \overline{\mathbf{\Theta}} \rangle$. Here the number of queue positions is $M = [10/3]_+ = 4$ and $\alpha(10, 3) = (\alpha(1; 10, 3) = 28, \alpha(2; 10, 3) = -84, \alpha(3; 10, 3) = 57, \alpha(4; 10, 3) = -1)$. Thus, $\widehat{\Gamma}$ satisfies GCP if $\forall j \in \{1, \ldots, 10\}$, $\underline{\theta}_j = (\theta_j(1), \theta_j(2), \theta_j(3), \theta_j(4)) \in \overline{\mathbf{\Theta}}$ 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 Definition 3.5 it is quite obvious that for a multiple machine queueing problem Γ, the coefficient vector $\alpha(n, m) = \{\alpha(k; n, m)\}_{k=1}^{M}$ is such that

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

The first-order difference at queue position k (that is, $\Delta\theta_j(k) = \theta_j(k+1) - \theta_j(k)$) represents the increase in queueing cost for individual j if he is moved from kth queue position to $(k+1)$th queue position. By simplifying Eq. (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 $k\in\{1,\dots,M-1\}$. From the mathematical identity $\sum_{q=0}^{r}(-1)^q\binom{a}{q}=(-1)^r\binom{a-1}{r}$, it follows that $z(k;n,m)=(-1)^{km-1}\binom{n-2}{km-1}$ for all $k\in\{1,\dots,M-1\}$ (see Tomescu, 1985).

Theorem 3.1. *A multiple machine queueing problem $\Gamma=\langle\mathbf{N},m,\overline{\mathbf{\Theta}}\rangle$ 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 $\underline{\theta}=(\underline{\theta}_1,\dots,\underline{\theta}_n)$ and $\underline{\theta}'=(\underline{\theta}'_1,\dots,\underline{\theta}'_n)$. 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 $\underline{\theta}(S)=(\underline{\theta}_1(S),\dots,\underline{\theta}_n(S))$.

Lemma 3.1. *A multiple machine queueing problem $\Gamma=\langle\mathbf{N},m,\overline{\mathbf{\Theta}}\rangle$ is first best implementable only if for all pairs $(\underline{\theta},\underline{\theta}')\in\overline{\mathbf{\Theta}}^n\times\overline{\mathbf{\Theta}}^n$, $\sum_{S\subseteq\mathbf{N}}(-1)^{|S|}C(\underline{\theta}(S))=0$.* [3]

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 $(\underline{\theta},\underline{\theta}')\in\overline{\mathbf{\Theta}}^n\times\overline{\mathbf{\Theta}}^n$, $\sum_{S\subseteq\mathbf{N}}(-1)^{|S|}C(\underline{\theta}(S))=\frac{1}{(n-1)}\sum_{j\in\mathbf{N}}\sum_{S\subseteq\mathbf{N}}(-1)^{|S|}\gamma_j(\underline{\theta}_{-j}(S))=0$ (see Walker, 1980).

Proof of Theorem 3.1. Consider a multiple machine queueing problem $\Gamma=\langle\mathbf{N},m,\overline{\mathbf{\Theta}}\rangle$. We start with a given type $\underline{\theta}_1$ for individual 1 in the interior of the domain 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 $\underline{\theta}_1=(\theta_1(1),\dots,\theta_1(k),\dots,\theta_1(M))\in\dot{\overline{\mathbf{\Theta}}}$. Given $\underline{\theta}_1$, we consider two states $\underline{\theta}=(\underline{\theta}_1,\underline{\theta}_2,\dots,\underline{\theta}_n)$ and $\underline{\theta}'=(\underline{\theta}'_1,\dots,\underline{\theta}'_n)$ of the following type: for all $k=1,\dots,M-1$, $\Delta\theta_1(k)<\Delta\theta_j(k)<\Delta\theta_{j+1}(k)$ for all $j\in\{2,\dots,n-1\}$ and $\theta'_j(k)=\eta$, for all $j\in\mathbf{N}$. Therefore, $\underline{\theta}'_j=(\eta,\eta,\dots,\eta)$ for all $j\in\mathbf{N}$ which implies that $\Delta\theta'_j(k)=0$ for all $k\in\{1,\dots,M-1\}$. Note that this sort of construction is possible due to Assumptions 1 and 3. 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 kth position and $(j+1)$th individual gets the $(k+1)$th position, then the costs for these two positions add up to $\{\theta_j(k)+\theta_{j+1}(k+1)\}$. If, on the other hand, the positions of j and $(j+1)$ are interchanged then the costs add up to $\{\theta_j(k+1)+\theta_{j+1}(k)\}$. The former cost strictly exceeds the latter for all $k=1,\dots,M-1$ since, by construction, $\Delta\theta_j(k)<\Delta\theta_{j+1}(k)$ for all $j=1,\dots n-1$. Thus, the queue that minimizes the aggregate cost requires that, $\sigma^*_{j+1}(\underline{\theta})\leqslant\sigma^*_j(\underline{\theta})$ for all $j=1,\dots,n-1$. In other words, $\sigma^*_j(\underline{\theta})=[(n+1-j)/m]_+$ for all $j\in\mathbf{N}$. Now consider profiles $\underline{\theta}(S)=(\underline{\theta}_1(S),\dots,\underline{\theta}_n(S))$ 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 $\underline{\theta}$ it follows that if $j,l\notin S$ and $j<l$, then $\sigma^*_l(\underline{\theta}(S))\leqslant\sigma^*_j(\underline{\theta}(S))$. Again, given any $S\subseteq\mathbf{N}$ and $S\neq\phi$, $\sigma^*_j(\underline{\theta}(S))\leqslant\sigma^*_s(\underline{\theta}(S))$ for all $(s,j)\in S\times\mathbf{N}-S$. This is because, for any given queue position $k\in\{1,\dots,M-1\}$, the incremental loss of any individual $j\notin S$ (that is, $\Delta\theta_j(k)$) is strictly more than that of any individual $s\in S$. Note that for all queues $\sigma^*(\underline{\theta}(S))$, satisfying (1) $\sigma^*_j(\underline{\theta}(S))\leqslant\sigma^*_s(\underline{\theta}(S))$ for all $(s,j)\in S\times\mathbf{N}-S$ and (2) $\sigma^*_l(\underline{\theta}(S))\leqslant\sigma^*_j(\underline{\theta}(S))$ 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.

We now consider the sum $\sum_{S\subseteq\mathbf{N}}(-1)^{|S|}C(\underline{\theta}(S))$. We break this sum in three parts in the following way: (i) $\sum_{j\neq1}\sum_{\overline{S}\subseteq\mathbf{N},j\notin\overline{S}}(-1)^{|\overline{S}|}\theta_j(\sigma^*_j(\underline{\theta}(\overline{S})))$, (ii) $\sum_{j\in\mathbf{N}}\sum_{\widehat{S}\subseteq\mathbf{N},j\in\widehat{S}}(-1)^{|\widehat{S}|}\theta_j(\sigma^*_j(\underline{\theta}(\widehat{S})))$ and (iii) $\sum_{\widetilde{S}\subseteq\mathbf{N}-\{1\}}(-1)^{|\widetilde{S}|}\,\theta_1(\sigma^*_1(\underline{\theta}_1,\underline{\theta}_{-1}(\widetilde{S})))$. We consider each of these parts separately in the next three paragraphs.

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

We first consider part (i). 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 $\overline{S}$ such that $j \notin \overline{S}$ and there are x number of individuals from the set P_j and $|\overline{S}| - x$ number of individuals from the set $\mathbf{N} - \{P_j \cup j\}$. The queue position of individual j for all such $\overline{S}$ is $\sigma_j^*(\underline{\theta}(\overline{S})) = [(n+1-j-x)/m]_+$. By collecting all such sets $\overline{S}$, 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

$$\sum_{T\subseteq \mathbf{N}-\{P_j\cup j\}} (-1)^{|T|} = \sum_{r=0}^{j-1} (-1)^r \binom{j-1}{r} = (1+(-1))^{j-1} = 0 \quad \text{since } j \neq 1.$$

Therefore, the coefficient of a term $\theta_j([(n+1-j-x)/m]_+)$ is zero for all $j(\neq 1)$ and for all $x(\leqslant |P_j|)$. Thus, $\sum_{j\neq 1} \sum_{\overline{S}\subseteq \mathbf{N}, j\notin \overline{S}} (-1)^{|\overline{S}|} \theta_j(\sigma_j^*(\underline{\theta}(\overline{S}))) = 0$.

We now consider part (ii). Observe that by adding the cost of an individual $j \in \mathbf{N}$ for all $\widehat{S}(\subseteq \mathbf{N})$ such that $j \in \widehat{S}$, 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, $\sum_{j\in \mathbf{N}} \sum_{\hat{S}\subseteq \mathbf{N}, j\in \hat{S}} (-1)^{|\hat{S}|} \theta_j(\sigma_j^*(\underline{\theta}(\widehat{S}))) = 0$.

Since the sums in (i) and (ii) are both zero, it follows that the sum $\sum_{S\subseteq \mathbf{N}} (-1)^{|S|} C(\underline{\theta}(S))$ is equal to the sum in (iii), that is $\sum_{S\subseteq \mathbf{N}} (-1)^{|S|} C(\underline{\theta}(S)) = \sum_{\tilde{S}\subseteq \mathbf{N}-\{1\}} (-1)^{|\tilde{S}|} \theta_1(\sigma_1^*(\underline{\theta}_1, \underline{\theta}_{-1}(\widetilde{S})))$. For individual 1, with type $\underline{\theta}_1$, we get $\sigma_1^*(\underline{\theta}_1, \underline{\theta}_{-1}(\widetilde{S})) = [(n-|\widetilde{S}|)/m]_+$ for all $\widetilde{S} \subseteq \mathbf{N} - \{1\}$. Therefore,

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

Simplifying condition (3.5) and then by applying Lemma 3.1 we get $\sum_{k=1}^{M} \alpha(k; n, m)\theta_1(k) = 0$.

Since the selection of individual 1, for the above construction, was arbitrary, the result follows. □

Note that the GCP is an additional restriction on the class of domains $\mathbf{\Theta}$ satisfying Assumptions 1–3. Let $\widehat{\mathbf{\Theta}}(\subset \mathbf{\Theta})$ be the class of domains satisfying the GCP and Assumptions 1–3.

Proposition 3.1. *If the number of machines m is even or if the number of queue positions* $M = [n/m]_+ = 2$, *then there is no first best implementable multiple machine queueing problem* $\Gamma = \langle \mathbf{N}, m, \overline{\mathbf{\Theta}} \rangle$ *such that* $\overline{\mathbf{\Theta}} \in \widehat{\mathbf{\Theta}}$.

Proof. If the number of machines m is even then for all $k \in \{1, \ldots, M-1\}$, $(-1)^{km-1} = -1$ because km is also even. From Assumption 1, it follows that for all $\underline{\theta}_j = (\theta_j(1), \ldots, \theta_j(M)) \in \overline{\mathbf{\Theta}}$, $\Delta\theta_j(k) \geqslant 0$ for all $k \in \{1, \ldots, M-1\}$. By substituting $(-1)^{km-1} = -1$ and $\Delta\theta_j(k) \geqslant 0$ in (3.4) we get $\Delta\theta_j(k) = 0$ for all $k \in \{1, \ldots, M-1\}$. Hence, the GCP implies that $\theta_j(1) = \cdots = \theta_j(M)$ for all $\underline{\theta}_j = (\theta_j(1), \ldots, \theta_j(M)) \in \overline{\mathbf{\Theta}}$ and we have a violation of Assumption 3.

If $M = 2$, then GCP implies $\alpha(1; n, m)\theta_j(1) + \alpha(2; n, m)\theta_j(2) = 0$ for all $\underline{\theta}_j = (\theta_j(1), \theta_j(2)) \in \overline{\mathbf{\Theta}}$ and condition (3.3) implies $\alpha(1; n, m) + \alpha(2; n, m) = 0$. Therefore, the GCP implies that $\theta_j(1) = \theta_j(2)$ for all $\underline{\theta}_j = (\theta_j(1), \theta_j(2)) \in \overline{\mathbf{\Theta}}$. This again is a violation of Assumption 3. □

So far we have imposed restrictions on individual preferences. The next property is a restriction on group preferences.

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

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

The GIP for a multiple machine queueing problem Γ implies that if for any pair of individuals $(j,l) \in \mathbf{N} \times \mathbf{N}$, $j \neq l$, the respective cost vectors $\theta_j = (\theta_j(1), \ldots, \theta_j(M)) \in \overline{\mathbf{\Theta}}$ and $\theta_l = (\theta_l(1), \ldots, \theta_l(M)) \in \overline{\mathbf{\Theta}}$ 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 $\Delta\theta_j(k) \geqslant \Delta\theta_l(k)$ 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* $\Gamma = \langle \mathbf{N}, m, \overline{\mathbf{\Theta}} \rangle$ *with* $M = 3$, *GCP* $\Rightarrow$ *GIP.*

Proof. Consider any multiple machine queueing problem $\Gamma = \langle \mathbf{N}, m, \overline{\mathbf{\Theta}} \rangle$ such that $M = 3$ and $\overline{\mathbf{\Theta}}$ satisfies the GCP. Consider $\underline{\theta}_j = (\theta_j(1), \theta_j(2), \theta_j(3))$ and $\underline{\theta}_l = (\theta_l(1), \theta_l(2), \theta_l(3))$ for individuals j and l respectively. Since $\overline{\mathbf{\Theta}}$ satisfies the GCP, from condition (3.4) 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 $i \in \{j, l\}$, $\Delta\theta_i(1) = P.\ \Delta\theta_i(2)$ where $P = -\frac{z(2;n,m)}{z(1;n,m)} > 0$. Thus, $\Delta\theta_j(1) < (>)\Delta\theta_l(1)$ if and only if $\Delta\theta_j(2) < (>)\Delta\theta_l(2)$. $\square$

Consider $\widehat{\Gamma} = \langle \mathbf{N} = \{1, \ldots, 10\}, m = 3, \overline{\mathbf{\Theta}} \rangle$ 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 $\bar{\underline{\theta}}_j = (1, 3, 4, 4) \in \overline{\mathbf{\Theta}}$ and $\bar{\underline{\theta}}_l = (1, 2, 3, 31) \in \overline{\mathbf{\Theta}}$ 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* $\widehat{\Gamma}$, GCP $\nRightarrow$ GIP.

Let $\mathbf{\Gamma}$ represent the class of multiple machine queueing problems with odd number of machines and with at least three queue positions and satisfying the GCP, the GIP and Assumptions 1–3. Therefore, a multiple machine queueing problem $\overline{\Gamma} = \langle \mathbf{N}, m, \widetilde{\mathbf{\Theta}} \rangle$ belongs to $\mathbf{\Gamma}$ if $\widetilde{\mathbf{\Theta}} (\subseteq \widehat{\mathbf{\Theta}})$ is a domain satisfying the GCP, the GIP and Assumptions 1–3. We now derive the efficient rule for any multiple machine queueing problem $\overline{\Gamma} = \langle \mathbf{N}, m, \widetilde{\mathbf{\Theta}} \rangle \in \mathbf{\Gamma}$. Before doing that we give some more relevant notations and definitions. Consider $\overline{\Gamma} = \langle \mathbf{N}, m, \widetilde{\mathbf{\Theta}} \rangle$. For a state $\underline{\theta} \in \widetilde{\mathbf{\Theta}}^n$, define $Q_j(\underline{\theta}) = [l \in \mathbf{N} - \{j\}$ such that *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 $\underline{\theta} = (\underline{\theta}_1, \ldots, \underline{\theta}_n)$. Using this definition of ranking we state and prove an *efficient rule* (that is, a single-valued selection from the efficiency correspondence) for $\overline{\Gamma} = \langle \mathbf{N}, m, \widetilde{\mathbf{\Theta}} \rangle \in \mathbf{\Gamma}$.

Proposition 3.3. *Consider* $\overline{\Gamma} = \langle \mathbf{N}, m, \widetilde{\mathbf{\Theta}} \rangle \in \mathbf{\Gamma}$. *For all* $\underline{\theta} \in \widetilde{\mathbf{\Theta}}^n$, *let* $\sigma^*(\underline{\theta})$ *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.*

For any state $\underline{\theta} = (\underline{\theta}_1, \ldots, \underline{\theta}_n) \in \widetilde{\mathbf{\Theta}}^n$, the queue $\sigma^*(\underline{\theta}) = (\sigma_1^*(\underline{\theta}), \ldots, \sigma_n^*(\underline{\theta}))$, defined in Proposition 3.3, is obtained in the following way: the individuals having rank 1 to m get the first position, the individuals having rank $m+1$ to $2m$ get the second position and this goes on till the individuals having rank $m(M-2)+1$ to $m(M-1)$ get the $(M-1)$th position and finally the remaining individual(s) having rank higher than $m(M-1)$ get the Mth position.

Proof of Proposition 3.3. Consider a state $\underline{\theta} = (\underline{\theta}_1, \ldots, \underline{\theta}_n) \in \widetilde{\mathbf{\Theta}}^n$ and any queue σ. We define a sequence $s(\sigma)$ as an ordering of the jobs obtained from σ by placing the jobs in the first position up to first m slots, the jobs in the second position in the following m slots and so on. [4] For a sequence $s(\sigma)$, let $j, l \in \mathbf{N}$ be neighbors with l in front of j, that is $s_\sigma(j) = s_\sigma(l) + 1$. If l and j switches their position (by keeping the position of all other jobs in the sequence unchanged), then the total cost changes by the amount $D(\sigma, \sigma'; \underline{\theta}) \equiv C(\sigma; \underline{\theta}) - C(\sigma'; \underline{\theta})$. We can have two types of switch-one that leaves the queue positions unaltered and one where the queue position of only the neighbors in question (that is, l and j) change. Therefore,

$$D(\sigma, \sigma'; \underline{\theta}) = \begin{cases} 0 & \text{if } \sigma = \sigma' \text{ (i.e. } \sigma_j = \sigma_l = k \text{ for some } k \in \{1, \ldots, M\}), \\ \Delta\theta_j(k) - \Delta\theta_l(k) & \text{if } \sigma \neq \sigma' \text{ (i.e. } \sigma_j = \sigma_l + 1 \text{ and } \sigma_l = k \text{ for some } k \neq M). \end{cases}$$

Given that the domain $\widetilde{\mathbf{\Theta}}$ satisfies the GIP, from $D(\sigma, \sigma'; \underline{\theta})$ it follows that switching l and j is weakly cost reducing *if and only if* $\Delta\theta_j(k) \geqslant \Delta\theta_l(k)$ for all $k \in \{1, \ldots, M-1\}$. Observe that the queue $\sigma^*(\underline{\theta})$ in state $\underline{\theta}$ is such that if $R_j(\underline{\theta}) < R_l(\underline{\theta})$ then (a) $\Delta\theta_j(k) \geqslant \Delta\theta_l(k)$ for all $k \in \{1, \ldots, M-1\}$ and (b) $\sigma^*_j(\underline{\theta}) \leqslant \sigma^*_l(\underline{\theta})$. Hence for $s(\sigma^*(\underline{\theta}))$, it is not possible to find neighbors for whom switching is cost reducing. This means that $\sigma^*(\underline{\theta})$ is efficient in state $\underline{\theta}$. Since the selection of $\underline{\theta} \in \widetilde{\mathbf{\Theta}}$ was arbitrary, the result follows. □

Proposition 3.3 shows that finding an efficient queue is quite transparent if $\overline{\Gamma} \in \mathbf{\Gamma}$. Observe that for $\overline{\Gamma} \in \mathbf{\Gamma}$, the 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 $\underline{\theta}'_{-j-l} \in \widetilde{\mathbf{\Theta}}^{n-2}$. 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 $M' \equiv \lceil \frac{n-1}{m} \rceil_+$ 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 $\overline{\Gamma} \in \mathbf{\Gamma}$, we define $Q_j(\underline{\theta}_{-l}) = [i \in \mathbf{N} - \{j, l\}$ such that *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 Γ 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 Γ 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 problem Γ satisfying the GIP when an individual is eliminated. Moreover, such a rank reversal can take place only if $M' = M - 1$ and Γ fails to

[4] For example, if there are only 2 machines and 4 jobs and jobs 1 and 2 are in front of machine 1 and jobs 3 and 4 are in front of machine 2 (i.e., if $\sigma = (\sigma_1 = 1, \sigma_2 = 2, \sigma_3 = 1, \sigma_4 = 2)$), then $s(\sigma) = (1, 3, 2, 4)$.

satisfy the GCP. If, a multiple machine queueing problem Γ 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 $\Delta\theta_j(M-1) = \Delta\theta_l(M-1)$. 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 Γ 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 $\overline{\Gamma} \in \mathbf{\Gamma}$, 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 $\underline{\theta} \in \widetilde{\mathbf{\Theta}}^n$,

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

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

Theorem 3.2. *A multiple machine queueing problem $\overline{\Gamma} \in \mathbf{\Gamma}$ 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 $\Gamma = \langle \mathbf{N}, m, \overline{\mathbf{\Theta}} \rangle$ satisfies the GCP, if and only if for all cost vector $\underline{\theta}_j \in \overline{\mathbf{\Theta}}$, 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 $\underline{\theta}_j \in \overline{\mathbf{\Theta}}$ 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 $p \in \{1, \ldots, n-1\}$,

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

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* Γ satisfies the GCP.

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

For $p = n$,

$$(n-p)h_j(p) + (p-1)h_j(p-1) = (n-1)h_j(n-1) = (n-1)\sum_{r=1}^{n-1}(-1)^{n-1-r}\frac{(n-2)!}{(r-1)!(n-r)!}\theta_j([r/m]_+)$$
$$= \sum_{r=1}^{n-1}(-1)^{n-1-r}\frac{(n-1)!}{(r-1)!(n-r)!}\theta_j([r/m]_+)$$
$$= \sum_{r=1}^{n-1}(-1)^{n-1-r}\binom{n-1}{r-1}\theta_j([r/m]_+)$$
$$= (-1)^{n-2}\sum_{k=1}^{M}\alpha(k;n,m)\theta_j(k) + \theta_j(M) \quad \text{(from the GCP)} = \theta_j(M).$$

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,

$$\sum_{p=1}^{n}(-1)^{p-1}\binom{n-1}{p-1}\theta_j([p/m]_+)$$
$$= \sum_{p=1}^{n}(-1)^{p-1}\binom{n-1}{p-1}\{(n-p)h_j(p) + (p-1)h_j(p-1)\}$$
$$= (n-1)\left\{\sum_{p=1}^{n-1}(-1)^{p-1}\binom{n-2}{p-1}h_j(p) + \sum_{p=2}^{n}(-1)^{p-1}\binom{n-2}{p-2}h_j(p-1)\right\} = 0. \qquad \square$$

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 $\overline{\Gamma} = \langle \mathbf{N}, m, \widetilde{\mathbf{\Theta}}\rangle \in \mathbf{\Gamma}$.

Proof of Theorem 3.2. Consider the sum $\sum_{l\neq j} h_j(R_j(\underline{\theta}_{-l}))$ in state $\underline{\theta} \in \widetilde{\mathbf{\Theta}}^n$ for individual $j \in \mathbf{N}$. From the GIP and Remark 3.2 we get

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

We consider a Groves–Clarke mechanism $\widehat{\mathbf{M}} = \langle \sigma^*, \hat{\mathbf{t}}\rangle$ 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

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

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

Observe that from Theorems 3.1 and 3.2 and from Proposition 3.2 it follows that *a multiple machine queueing problem satisfying A1–A3 and with three machines is first best implementable if and only if it satisfies the GCP.*

4. Separable cost multiple machine queueing problems

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. The following conditions describe a typical separable cost multiple machine queueing problem.

1. Define a real number $\bar{\theta} > 0$. Given $\bar{\theta} > 0$, the cost parameter or type of an individual belongs to the interval $[0, \bar{\theta}]$. Moreover, there exists a function $f : \{1, \ldots, M\} \to \mathbf{R}$ such that $f(k) \geqslant f(k-1)$ for all $k \in \{2, \ldots, M\}$ and $f(k^*) > f(k^*-1)$ for at least one $k^* \in \{2, \ldots, M\}$.
2. $\theta_j(k) = f(k)\theta_j$ for all $j \in \mathbf{N}$, for all $k \in \{1, 2, \ldots, M\}$ and for all $\theta_j \in [0, \bar{\theta}]$.

For a separable cost multiple machine queueing problem, the cost of each individual for each position is multiplicatively separated 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 θ_j represents the type (or cost parameter) of an individual that belong to the interval $[0, \bar{\theta}]$. In this set up a type vector of individual $j \in \mathbf{N}$ is given by $\bar{\theta}_j = (\theta_j(1) = f(1)\theta_j, \ldots, \theta_j(M) = f(M)\theta_j)$. Therefore, from now on we will write θ_j as the cost parameter or type of an individual. The cost parameter $\theta_j \in [0, \bar{\theta}]$ for all $j \in \mathbf{N}$ is private information. Finally, $\theta = (\theta_1, \ldots, \theta_n) \in [0, \bar{\theta}]^n$ represents a state of the world or a profile. It is important to observe that the domain $\Theta \equiv (f, [0, \bar{\theta}])$ of any separable cost multiple machine queueing problem satisfies Assumptions 1–3.

In this framework the set of individuals $\mathbf{N}$, $m(> 1)$ number of machines and $\Theta \equiv (f, [0, \bar{\theta}])$ define the *multiple machine separable cost queueing problem* $\widehat{\Gamma} = \langle \mathbf{N}, m, \Theta \rangle$. 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* $\widehat{\Gamma} = \langle \mathbf{N}, m, \Theta \rangle$ *satisfies the GIP.*

Proof. Consider any pair $(j, l) \in \mathbf{N} \times \mathbf{N}$, $j \neq l$, with cost parameters θ_j and θ_l respectively. It is obvious that either $\theta_j \geqslant \theta_l$ or $\theta_j \leqslant \theta_l$. Since $f(k+1) \geqslant f(k)$ for all $k \in \{1, \ldots, M-1\}$, it is also obvious that if $\theta_j \geqslant \theta_l$, then $\{f(k+1) - f(k)\}\theta_j \geqslant \{f(k+1) - f(k)\}\theta_l$ for all $k \in \{1, \ldots, M-1\}$. Therefore, $\Delta\theta_j(k) = \{f(k+1) - f(k)\}\theta_j \geqslant \Delta\theta_l(k) = \{f(k+1) - f(k)\}\theta_l$ for all $k \in \{1, \ldots, M-1\}$. Similarly, if $\theta_j \leqslant \theta_l$, then $\Delta\theta_j(k) \leqslant \Delta\theta_l(k)$ for all $k \in \{1, \ldots, M-1\}$. Thus, it follows that $\widehat{\Gamma} = \langle \mathbf{N}, m, \Theta \rangle$ 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 $\widehat{\Gamma} = \langle \mathbf{N}, m, \Theta \rangle$ 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 $\widehat{\Gamma} = \langle \mathbf{N}, m, \Theta \rangle$ 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 $\widehat{\Gamma} = \langle \mathbf{N}, m, \Theta \rangle$ is first best implementable if and only if the cost function satisfies the GCP.*

The necessity part of Proposition 3.1 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 $\mathbf{\Gamma}(S)(\subset \mathbf{\Gamma})$ be the class of multiple machine separable cost queueing problems where m is odd and $n > 2m$ and let $\mathbf{\Gamma}^*(\subset \mathbf{\Gamma}(S))$ 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 $\Gamma \in \mathbf{\Gamma}(S)$ such that $\Gamma \in \mathbf{\Gamma}^*$.*

Proof. Consider $\widehat{\Gamma}^0 = \langle \mathbf{N}, m, \Theta^o \equiv (f^0, [0, \overline{\theta}]) \rangle \in \mathbf{\Gamma}(S)$ with odd number of queue positions M and with f^0 of the following form: $f^0(1) = c \geqslant 0$ and $\Delta f^0(k) = f^0(k+1) - f^0(k) = 1 \Big/ \binom{n-2}{km-1}$ for all $k \in \{1, \ldots, M-1\}$. We will prove that $\widehat{\Gamma}^0 = \langle \mathbf{N}, m, (f^0, [0, \overline{\theta}]) \rangle \in \mathbf{\Gamma}^*$ by showing that f^0 satisfies condition (4.9). Observe first that $z(k; n, m)\Delta f^0(k) = (-1)^{km-1}$ for all $k \in \{1, \ldots, M-1\}$. Therefore, by substituting $z(k; n, m)\Delta f^0(k) = (-1)^{km-1}$ 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, $\widehat{\Gamma}^0 = \langle \mathbf{N}, m, (f^0, [0, \overline{\theta}]) \rangle \in \mathbf{\Gamma}^*$.

Similarly, consider $\widehat{\Gamma}^e = \langle \mathbf{N}, m, \Theta^e \equiv (f^e, [0, \overline{\theta}]) \rangle \in \mathbf{\Gamma}(S)$ with even number of queue positions M and with f^e of the following form: $f^e(1) = c \geqslant 0$ and $\Delta f^e(k) = f^e(k+1) - f^e(k) = 1 \Big/ \binom{n-2}{km-1}$ for all $k \in \{1, \ldots, M-2\}$ and $f^e(M-1) = f^e(M)$ (that is, $\Delta f^e(M-1) = 0$). Observe that $z(k; n, m)\Delta f^e(k) = (-1)^{km-1}$ for all $k \in \{1, \ldots, M-2\}$ and $z(M-1; n, m)\Delta f^e(M-1) = 0$. Therefore, by substituting $z(k; n, m)\Delta f^e(k)$ 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, $\widehat{\Gamma}^e \in \mathbf{\Gamma}^*$. □

Observe that given $\mathbf{\Gamma}^* \subset \mathbf{\Gamma}(S) \subset \mathbf{\Gamma}$, 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 Eq. (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 $f^l(k) = k$ for all k, then $\Gamma^l = \langle \mathbf{N}, m, \Theta^l \equiv (f^l, [0, \overline{\theta}]) \rangle \in \mathbf{\Gamma}^*$.*

5. Concluding remarks

We have obtained the following results regarding the first best implementability of the class of multiple machine queueing problems with domains satisfying Assumptions 1–3.

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 non-trivial 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.
4. If the number of machines m is odd and $M = [n/m]_+ \geqslant 4$ 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 non-trivial cost function for which a multiple machine queueing problem is first best implementable.
6. Finally, if m is odd, the number of queue positions $M = [n/m]_+ > 2$ is also odd and $n = m \cdot M$ 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.

Acknowledgements

The author is grateful to Arunava Sen, Jeroen Suijs and two anonymous referees for their invaluable advice. The author is thankful to Parikshit Ghosh and Suryapratim Banerjee for helpful discussions. The author is also thankful to the seminar participants at the European Summer Symposium in Economic Theory 2003 (held in Gerzensee, Switzerland). The author gratefully acknowledges the financial support from the Indian Statistical Institute and from the Deutsche Forschungsgemeinschaft Graduiertenkolleg 629 at the University of Bonn. The author is solely responsible for the errors that may still remain.

References

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

Curiel, I., Pederzoli, G., 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., 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., Ledyard, J.O., 1977. Some limitations of demand revealing processes. Public Choice 29, 107–124.

Groves, T., Loeb, M., 1975. Incentives and public inputs. Journal of Public Economics 4, 211–226.

Hamers, H., Klijn, F., 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., 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., van den Nouweland, A., 1992. Flow-shops with a dominant machine. European Journal of Operational Research 62, 38–46.

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

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

Mitra, M., Sen, A., 1998. Dominant Strategy Implementation of First Best Public Decisions, Mimeo., Indian Statistical Institute, New Delhi.

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. Economics Letters 53, 17–24.

Tomescu, I., 1985. Problems in Combinatorics and Graph Theory (Melter, R.A., Trans.) Wiley-Interscience, New York.

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