Soc Choice Welfare (2007) 29:369–382
DOI 10.1007/s00355-006-0210-3

ORIGINAL PAPER

Cost sharing in a job scheduling problem

Debasis Mishra · Bharath Rangarajan

Received: 18 January 2006 / Accepted: 20 October 2006 / Published online: 17 November 2006

Abstract A set of jobs need to be served by a server which can serve only one job at a time. Every job has a processing time and incurs cost due to waiting (linear in its waiting time). The jobs share their costs using monetary transfers. We provide an axiomatic characterization of the Shapley value solution for this problem.

1 Introduction

In a scheduling problem, a set of jobs needs to be served by a server, which can process only one job at a time. Each job has a finite processing time and a per unit time waiting cost. Efficiency directs us to serve the jobs in decreasing order of the ratio of per unit time waiting cost and processing time. To compensate for waiting cost of jobs, monetary transfers to jobs are allowed. How should the jobs share the cost in a fair manner amongst themselves (through transfers)? The Shapley value solution is considered to be an appropriate solution for the fair division problem in general (Moulin 1992a), and for the scheduling problem in particular (Chun 2004a; Katta and Sethuraman 2005; Maniquet 2003; Moulin 2004).

In this work, we characterize the Shapley value solution for the scheduling problem. A paper by Maniquet (2003) is the closest to our model, and is the motivation behind our work. Maniquet (2003) studies a model where he

D. Mishra (✉)
Planning Unit, Indian Statistical Institute, 7, S.J.S. Sansanwal Marg, New Delhi 110 016, India
e-mail: dmishra@isid.ac.in

B. Rangarajan
Department of Mechanical Engineering, University of Minnesota, Minneapolis, MN, USA
e-mail: bharathr@me.umn.edu

assumes all jobs to have unit processing times, and characterizes the Shapley value solution. Using a different definition of worth of coalitions, Chun (2004a) derives a "reverse" rule for the same model.

The key difference between our model and the models in Maniquet (2003) and Chun (2004a) is that ours is a two dimensional model. In the one dimensional model discussed in Chun (2004a) and Maniquet (2003), all orderings are efficient if and only if all the jobs are identical, i.e., have the same parameters. In the two dimensional model, all orderings are efficient even if the jobs are not identical. For this reason, the axioms for the one dimensional model are insufficient for our two dimensional model. To deal with the cost sharing in the two dimensional model, we introduce new axioms. We characterize the Shapley value solution for this equal *priority* case (i.e., the case when all orderings are efficient) using these axioms. Using this as the springboard, we are able to characterize the Shapley value solution for the general instances of non-identical jobs.

Independent of our work, Chun (2004b) has developed a characterization of the Shapley value solution for the scheduling problem. Our characterization is different from his characterization in two aspects: (1) we do not make use of consistency and monotonicity axioms, the focus of his characterizations, (2) we do not assume *binary transfers* between jobs (jobs pay each other) like he does.

Another stream of literature is on "sequencing games", first introduced by Curiel et al. (1989). The model in Curiel et al. (1989) is similar to ours. Their notion of the worth of a coalition is very different from the one we consider here. They focus on sharing the *savings in costs* from a given initial ordering to the optimal ordering (also see Hamers et al. 1996).

Strategic aspects of queueing problems have also been studied (Mitra 2002; Suijs 1996). The general result in these studies is that incentive compatible, efficient, and budget-balanced cost sharing is possible only for linear cost functions, like in our scheduling problem. The Shapley value solution discussed in this work is not incentive compatible. However, Moulin (2004) studies strategic concepts such as *splitting* and *merging* in scheduling problems with equal per unit time waiting costs, and shows that the Shapley value solution is merge-proof, but not split-proof.

1.1 Our contribution

Our focus is the Shapley value solution and its axiomatic characterization in the scheduling problem. We show that the Shapley value of a job in the scheduling problem is the average of the cost it inflicts on other jobs and the cost inflicted to it by other jobs.

Our contribution lies in extending the characterizations of Maniquet (2003) for the one dimensional model to the general model of the scheduling problem. Our objective is to use, with appropriate generalizations, as many axioms as possible from Maniquet (2003). The axioms in Maniquet (2003) can be divided into two types: (1) classical axioms such as efficiency, Pareto indifference etc.

and (2) axioms specific to the scheduling problem (we call these axioms Maniquet's axioms). To characterize the Shapley value solution axiomatically for the general case, we first provide a set of axioms that characterize the Shapley value solution for the case when every ordering is efficient. This is a key step in our characterization for the general case. We achieve this by imposing an upper bound on the cost share of every job. We call this the expected cost bound (ECB) axiom. A classical axiom, equal treatment of equals, used by Maniquet (2003) is insufficient in our model, and is replaced by by the ECB axiom. The other classical axioms that we use, which are also used in Maniquet (2003), are efficiency and Pareto indifference. Efficiency and ECB characterize the Shapley value solution for the equal priority case.

Once the characterization for the equal priority case is achieved, we need appropriate generalizations of the following Maniquet's axioms along with Pareto indifference to characterize the Shapley value solution for our problem. The independence axioms: cost share of a job is independent of preceding jobs' per unit time waiting cost and following jobs' processing time. Broadly, the independence axioms say that the cost share of a job should not change if the parameters of other jobs are changed in a way such that its "externality" (the cost it inflicts to other jobs and the cost it incurs due to other jobs) is unchanged. The proportional responsibility axioms: the transfer of an additional job removed from the end (beginning) of a queue is shared by the jobs before (after) it in proportion to their processing times (per unit time waiting costs). The proportionate responsibility axioms are generalizations of the *equal responsibility* axiom in Maniquet (2003). We characterize the Shapley value solution in three different ways using these axioms. In all the characterizations efficiency, Pareto indifference, and ECB are imposed. Besides these, we either need the independence axioms or one of the proportional responsibility axioms in place of one of the independence axioms. This shows that these axioms are substitutable in the presence of efficiency, Pareto indifference, and ECB.

The rest of the paper is organized as follows. Section 2 describes the model and Sect. 3 discusses the Shapley value solution for the model. In Sect. 4, we discuss our axioms. The characterization results involving axioms appear in Sect. 5.

2 The model

There are n jobs waiting to be served by a server. The server can process only one job at a time. The set of jobs are denoted as $N := \{1, \ldots, n\}$. An ordering of the jobs is given by a one to one map $\sigma : N \to N$ and σ_i denotes the position of job i in that order. Given an ordering σ, define the followers of job i by $F_i(\sigma) := \{j \in N : \sigma_j > \sigma_i\}$ and the predecessors of job i by $P_i(\sigma) := \{j \in N : \sigma_j < \sigma_i\}$. We assume that for any $i \in N$ and any σ, if $F_i(\sigma)$ or $P_i(\sigma)$ is the empty set, then any summation over such sets gives zero value.

Every job i is identified by two parameters: (p_i, θ_i), where p_i is the processing time and θ_i is the per unit time waiting cost of job i. Thus, a *queueing problem* is defined by a list $q = (N, p, \theta) \in \mathbb{Q}$, where $\mathbb{Q}$ is the set of all possible lists. We will denote $\gamma_i = \theta_i / p_i$. We call γ_i, the *priority* of job i. Given an ordering of jobs σ, the waiting cost incurred by job i is given by $c_i(\sigma) = \theta_i \sum_{j \in P_i(\sigma)} p_j$. The total waiting cost incurred by all jobs due to an ordering σ can be thought of in two ways: (1) by summing the cost incurred by every job and (2) by summing the costs inflicted by a job on jobs behind it due to its own processing time.[1]

$$C(N, \sigma) = \sum_{i \in N} c_i(\sigma) = \sum_{i \in N} \Big[\theta_i \sum_{j \in P_i(\sigma)} p_j\Big].$$
$$= \sum_{i \in N} \Big[p_i \sum_{j \in F_i(\sigma)} \theta_j\Big].$$

An *efficient ordering* σ^* is one that minimizes the total cost incurred by all jobs. So, $C(N, \sigma^*) \leq C(N, \sigma)\ \forall\ \sigma \in \Sigma$, where Σ is the set of all orderings. For notational simplicity, we will write the total cost in an efficient ordering of jobs from N as $C(N)$ whenever it is not confusing. Sometimes, we will deal only with a subset of jobs $S \subseteq N$. The ordering σ will then be defined only on the jobs in S and we will write $C(S)$ for the total cost from an efficient ordering of jobs in S. It is well known that jobs are ordered in non-increasing priority in an efficient ordering. This is also known as the *weighted shortest processing time* rule (Smith 1956).

An *allocation* for $q = (N, p, \theta) \in \mathbb{Q}$ has two components: an ordering σ and a transfer t_i for every job $i \in N$. The payment received by job i is denoted by t_i. Given a transfer t_i and an ordering σ, the *cost share* of job i is defined as, $\pi_i = c_i(\sigma) - t_i = \theta_i \sum_{j \in P_i(\sigma)} p_j - t_i$.

An allocation (σ, t) is *efficient* for $q = (N, p, \theta)$ whenever σ is an efficient ordering and $\sum_{i \in N} t_i = 0$. $\sigma^*(q)$ will be used to denote an efficient ordering of jobs in queue q (σ^* will be used when q is understood from the context). It is easy to see that for two different efficient orderings, the cost share vector in one efficient allocation is possible to achieve in the other by appropriately modifying the transfers.

Depending on the transfers, the cost shares in different efficient allocations may differ. An *allocation rule* ψ associates with every $q \in \mathbb{Q}$ a non-empty subset $\psi(q)$ of allocations.

3 Cost sharing using the Shapley value

In this section, we define the cost of a coalition of jobs and find the Shapley value of this game. Given a problem $q \in \mathbb{Q}$, the cost of a coalition of $S \subseteq N$ jobs in the queue is defined as the cost incurred by jobs in S if these are the only jobs served in the queue using an efficient ordering. Formally, the cost of a coalition $S \subseteq N$ is,

[1] Since a job is responsible for its own processing cost, we assume that this cost component of a job is not shared with other jobs. For this reason, we do not consider the processing costs of jobs in the total cost.

$$C(S) = \sum_{i \in S} \theta_i \sum_{j \in P_i(\sigma^*)} p_j,$$

where σ^* $(= \sigma^*(S))$ is an efficient ordering considering jobs from S only. The worth of a coalition of S jobs is just $-C(S)$. This way of defining the worth of a coalition assumes that each job in a coalition S is served before any job outside of S is. In a sense, it puts a natural lower bound on the cost share of a coalition of jobs.[2]

The Shapley value (or cost share) of a job i is defined as (Shapley 1953),

$$Sh_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(|N| - |S| - 1)!}{|N|!} \Big[C(S \cup \{i\}) - C(S) \Big]. \tag{1}$$

The Shapley value rule says that jobs are ordered using an efficient ordering and transfers are assigned to jobs such that the cost share of job i is given by Eq. (1), which can be simplified further for the scheduling problem.

Lemma 1 *Let σ be an efficient ordering of jobs in the set N. For all $i \in N$ the Shapley value of i is given by,*

$$Sh_i = \frac{1}{2} \Big[\sum_{j \in F_i(\sigma)} p_i \theta_j + \sum_{j \in P_i(\sigma)} \theta_i p_j \Big].$$

Proof The proof follows from an alternative interpretation of the Shapley value solution, where we choose a uniform random ordering and the cost share of a job is the marginal increase in the cost of jobs preceding it due to its presence. The expected cost from such a randomized allocation rule is exactly the Shapley value solution.

The marginal cost due to i for coalition $S \subseteq (N \setminus \{i\})$, assuming σ' to be an efficient ordering of jobs in $S \cup \{i\}$, can be written as:

$$C(S \cup \{i\}) - C(S) = \sum_{j \in F_i(\sigma')} p_i \theta_j + \sum_{j \in P_i(\sigma')} \theta_i p_j.$$

If we choose any set of agents $S \subseteq (N \setminus \{i\})$ uniformly at random, for any $j \neq i$, Probability$(j \in S)$ = Probability$(j \notin S) = \frac{1}{2}$. So, taking expectation over all $S \subseteq (N \setminus \{i\})$, we get the desired result. □

Another easy method to prove Lemma 1 is to use the inductive formula of the Shapley value. Denote $\text{Sh}_i(S)$ as the Shapley value of job i when jobs in $S \subseteq N$ $(i \in S)$ only are present. Then $\text{Sh}_i(N) = \frac{1}{n}[C(N) - C(N \setminus \{i\})] + \frac{1}{n} \sum_{j \neq i} \text{Sh}_i(N \setminus \{j\})$. A straightforward induction argument proves Lemma 1.

[2] In Chun (2004a), the worth of a coalition is calculated by assuming that the jobs in the coalition are served after the jobs not in the coalition, which puts a natural upper bound on the cost share of a coalition.

By Lemma 1, the transfer corresponding to the Shapley value of job i is given by,

$$t_i = \frac{1}{2}\Big[\sum_{j\in P_i(\sigma)} p_j\theta_i - \sum_{j\in F_i(\sigma)} p_i\theta_j\Big]. \tag{2}$$

4 The axioms

In this section, we will define several axioms on fairness and later characterize the Shapley value rule using them. For a given $q \in \mathbb{Q}$, we will denote $\psi(q)$ as the set of allocations from allocation rule ψ. Also, we will denote the cost share vector associated with an allocation rule (σ, t) as π and that with allocation rule (σ', t') as π' etc.

Our axioms fall into three classes: (1) classical axioms (efficiency, Pareto indifference, and equal treatment of equals); (2) new axioms; and (3) axioms generalized from Maniquet (2003).

4.1 Classical axioms

First, we define the efficiency axiom. It states that an efficient ordering should be selected and the transfers of jobs should add up to zero (*budget balance*).

Definition 1 *An allocation rule* ψ *is* efficient *if for every* $q \in \mathbb{Q}$ *and* $(\sigma, t) \in \psi(q)$, (σ, t) *is an efficient allocation.*

The second axiom says that the allocation rule should not discriminate between two allocations which are equivalent to each other in terms of cost shares of jobs.

Definition 2 *An allocation rule* ψ *satisfies* Pareto indifference *if for every* $q \in \mathbb{Q}$, $(\sigma, t) \in \psi(q)$, *if there exists another allocation* (σ', t') *such that* $\Big[\pi_i = \pi'_i \;\; \forall\, i \in N\Big]$, *then* $(\sigma', t') \in \psi(q)$.

The next axiom is classical in literature and says that two jobs with equal parameters should be compensated such that their cost shares are also equal.

Definition 3 *An allocation rule* ψ *satisfies equal treatment of equals (ETE) if for all* $q \in \mathbb{Q}$, $(\sigma, t) \in \psi(q)$, $i, j \in N$, *then*

$$\Big[p_i = p_j; \theta_i = \theta_j\Big] \Rightarrow \Big[\pi_i = \pi_j\Big].$$

ETE directs us to share costs equally between jobs if they have identical set of parameters. At the same time, it is silent about the cost shares of two jobs i and j that are indistinguishable with respect to an efficient ordering (with

$\gamma_i = \gamma_j$) but have different parameters ($(p_i, \theta_i) \neq (p_j, \theta_j)$). We introduce some new axioms to resolve this gap.

4.2 New axioms

We would like to introduce the idea of merging jobs with respect to job i when all jobs have equal priority.[3] Suppose job i is in position σ_i in an ordering σ of the queue. There are two costs by which it interacts with the rest of the system. First is the waiting cost of job i that appears due to the processing times of jobs before it and second is the contribution to the waiting cost of jobs placed behind job i due to the processing time of job i. When we consider the waiting cost of job i, it is immaterial how job i came to wait that length of time: whether it was due to a single job with large processing time or multiple jobs with smaller processing times. In the same vein, the cost job i imposes on the jobs behind it depends only on the sum of their per unit time waiting costs and not on how these per unit time waiting costs were distributed among those jobs. Hence, as far as job i is concerned we can merge all jobs before it with a processing time of $\sum_{j \in P_i(\sigma)} p_j$ and all jobs behind it with a per unit time waiting cost of $\sum_{j \in F_i(\sigma)} \theta_j$. By merging, we would like to think of these merged jobs as a single job with the above specified processing time (or per unit time waiting cost). However, to preserve the priority (γ) of jobs that we started out with we set the per unit time waiting cost of the merged unit before as $\sum_{j \in P_i(\sigma)} \theta_j$ and processing time of the merged unit after as $\sum_{j \in F_i(\sigma)} p_j$. This means that the relative ordering remains intact; the jobs before (after) job i that were merged can be placed before (after) job i. Since in the modified queue set up (with only three jobs) the "world view" of job i with respect to its waiting cost or the cost it inflicts does not change (i.e., the "externalities" of job i is unchanged), we would expect that it still preserves its cost share. The remaining jobs receive a transfer together that must then be shared by all of them internally. This is the idea captured by our next axiom. We can generalize this idea of merging (with the same justification as above) to account for merging any subset of the jobs that are placed before or after i. We now present the technical definitions and details.

When any set of consecutive jobs $S \subseteq N$ are merged, they are treated like a single job with processing time $p_S := \sum_{i \in S} p_i$ and per unit time waiting cost $\theta_S := \sum_{i \in S} \theta_i$. We will denote the new (merged) job as $< S >$. Assume that we are given an efficient ordering σ and a job $i \in N$. We will only consider mergers of consecutive jobs $S \subseteq F_i(\sigma)$ (or $T \subseteq P_i(\sigma)$). A merger S (or T) is said to be a valid merger, if the new jobs are created by merging consecutive jobs and they have the parameters: $\sum_{j \in S} \theta_j$ and $\sum_{j \in S} p_j$ (or $\sum_{j \in T} \theta_j$ and $\sum_{j \in T} p_j$). A queue instance created by a particular choice of S and T (S or T can be $\emptyset$) is denoted by $q(S, T)$ and $M(\sigma, i)$ denotes the set of all such queue instances created using valid mergers. We recall here that (under the equal priority assumption) the

[3] For a strategic treatment of the merging concept, see (Moulin 2004), who considers a model where jobs have equal θ.

choice of parameters for the new job ensures that $\gamma_i = \gamma_{<S>} = \gamma_{<T>}$ and hence the relative ordering still remains efficient.[4]

Definition 4 *An allocation rule* ψ *satisfies* independence of valid merging (IVM) *if for all* $q = (N, p, \theta) \in \mathbb{Q}$ *with* $\gamma_1 = \ldots = \gamma_n$*,* $(\sigma, t) \in \psi(q)$*,* $i \in N$*,* $q(S, T) \in M(\sigma, i)$*, and* $(\sigma', t') \in \psi(q(S, T))$*, we have* $\pi_i = \pi_i'$*, where* π_i *is the cost share of job i in* (σ, t) *and* π_i' *is the cost share of job i in* (σ', t')*.*

To motivate our next axiom, let us consider the case of two jobs with equal γ. There are only two possible orderings σ with $\sigma_i = i$ for $i \in \{1, 2\}$ and the *reverse* ordering, denoted by σ'. In both the orderings the waiting cost of the second job is the same ($p_1\theta_2$ in σ and $p_2\theta_1 = p_1\theta_2$ in σ') and the first job does not incur any waiting cost. Both the jobs are equivalent in the sense that they each can inflict the same waiting cost on the other job when placed first in the queue. Our next axiom says that in this case the jobs should equally divide this externality, i.e., their cost shares should be equal.

Definition 5 *An allocation rule* ψ *satisfies* equal division for two equal priority jobs (ED2) *if for all* $q = (N, p, \theta) \in \mathbb{Q}$ *with* $N = \{1, 2\}$ *and* $\gamma_1 = \gamma_2$*, for any* $(\sigma, t) \in \psi(q)$*, we have* $c_1(\sigma) - t_1 = c_2(\sigma) - t_2$*.*

The following Lemma characterizes the cost share of jobs when they have equal priority under efficiency, IVM, and ED2.

Lemma 2 *Consider* $q \in \mathbb{Q}$ *such that* $\gamma_1 = \cdots = \gamma_n$*. In an efficient allocation rule* ψ *satisfying IVM and ED2, for every* $i \in N$ *the cost share of i is* $\frac{1}{2}\theta_i \sum_{j \neq i} p_j = \frac{1}{2}\left[\theta_i \sum_{j \in P_i(\hat{\sigma})} p_j + p_i \sum_{j \in F_i(\hat{\sigma})} \theta_j\right]$*, where* $\hat{\sigma}$ *is any ordering of jobs in N.*

Proof Consider the allocation $(\sigma, t) \in \psi(q)$. Let i be the first job in σ. Now, perform a valid merging of jobs in $F_i(\sigma)$ to form the new queue q' with jobs i and $< F_i(\sigma) >$. The equal γ case is preserved by the valid merging as the new job $< F_i(\sigma) >$ has a processing time of $\sum_{j \neq i} p_j$ and per unit time waiting cost of $\sum_{j \neq i} \theta_j$ and $\gamma_i = \frac{\sum_{j \neq i} \theta_j}{\sum_{j \neq i} p_j}$. By ED2, the cost share of job i in any allocation in $\psi(q')$ is $\frac{1}{2} p_i \sum_{j \neq i} \theta_j$. By IVM, the cost share of job i is same in every allocation in $\psi(q)$ and $\psi(q')$. So, the cost share of job i is $\frac{1}{2} p_i \sum_{j \neq i} \theta_j$.

Similarly, if k is the last job in σ, we can merge jobs in $P_k(\sigma)$, and use IVM and ED2 to show that its cost share equals $\frac{1}{2} p_{<P_k(\sigma)>} \theta_k = \frac{1}{2} \theta_k \sum_{j \neq k} p_j$. For any arbitrary job k that is not the fist or the last job in σ, we can merge jobs in $P_k(\sigma)$ and $F_k(\sigma)$ to form a queue with three jobs: $< P_k(\sigma) >, k$, and $< F_k(\sigma) >$. The cost share of jobs $< P_k(\sigma) >$ and $< F_k(\sigma) >$ (the first job and

[4] Even if the jobs are not of equal priority, then also such merging of jobs results in an ordering that is efficient. In fact our valid merging axiom holds in the Shapley value solution for the general case when jobs are not of equal priority. But to characterize the Shapley value solution, we only need it to hold for the equal priority case.

the last job in the new queue) can be computed as before, and they are respectively $\frac{1}{2}p_{<P_k(\sigma)>}[\theta_k + \theta_{<F_k(\sigma)>}]$ and $\frac{1}{2}\theta_{<F_k(\sigma)>}[p_k + p_{<P_k(\sigma)>}]$. Hence the total cost share of the first job and the last job is $p_{<P_k(\sigma)>}\theta_{<F_k(\sigma)>} + \frac{1}{2}p_k\theta_{<F_k(\sigma)>} + \frac{1}{2}\theta_k p_{<P_k(\sigma)>}$. By efficiency and IVM, the cost share of job k in the original queue is $\frac{1}{2}p_k\theta_{<F_k(\sigma)>} + \frac{1}{2}\theta_k p_{<P_k(\sigma)>} = \frac{1}{2}\big[p_k \sum_{j\in F_k(\sigma)} \theta_j + \theta_k \sum_{j\in P_k(\sigma)} p_j\big]$. □

Lemma 2 is the stepping stone to our axiomatic characterization results for the general two parameter case. It characterizes the cost share of jobs for the equal priority case. Observe that in the model where all jobs have the same processing time, the equal priority case reduces to the identical job case for which, by the ETE axiom, the total cost is shared equally among the jobs.

We present an alternative, but an intuitive, axiom to characterize the cost shares of jobs when $\gamma_1 = \cdots = \gamma_n$ and hence prove a lemma analogous to Lemma 2. If transfers were not allowed, then a fair allocation rule in this setting would be to choose every ordering with equal probability. In such a case, the cost incurred by every job i is $\frac{1}{2}\theta_i \sum_{j\neq i} p_j$. We impose this as an upper bound on cost share when transfers are allowed. Such bounds on cost shares (utilities) are often imposed through individual rationality axioms in many cost sharing problems (see individual rationality axioms in **?** as an example).

Definition 6 *An allocation rule ψ satisfies expected cost bound (ECB) if for all $q \in \mathbb{Q}$ with $\gamma_1 = \cdots = \gamma_n$, for every $i \in N$, for any $(\sigma, t) \in \psi(q)$, $\pi_i \leq \frac{1}{2}\theta_i \sum_{j\neq i} p_j$, where π_i is the cost share of job i in allocation (σ, t).*

Expected cost bound can be thought as a generalization of the ED2 axiom from the two job case to any number of jobs case. Using ECB, we can immediately obtain a lemma analogous to Lemma 2.

Lemma 3 *Let $\gamma_1 = \cdots = \gamma_n$. In an efficient allocation rule ψ satisfying ECB, for every $i \in N$, the cost share of i is $\pi_i = \frac{1}{2}p_i \sum_{j\neq i} \theta_j = \frac{1}{2}\Big[\theta_i \sum_{j\in P_i(\hat{\sigma})} p_j + p_i \sum_{j\in F_i(\hat{\sigma})} \theta_j\Big]$, where $\hat{\sigma}$ is any ordering of jobs in N.*

Proof By definition, π_i is just the average of job i's waiting costs over all orderings, when transfers are ruled out. In the equal γ case, each of these orderings is efficient. So, $\sum_{i\in N} \pi_i$ must be the cost of an efficient queue. Therefore, any efficient allocation ψ satisfying ECB must have cost share of job i equal to $\pi_i = \frac{1}{2}\Big[\theta_i \sum_{j\in P_i(\hat{\sigma})} p_j + p_i \sum_{j\in F_i(\hat{\sigma})} \theta_j\Big]$, where $\hat{\sigma}$ is any ordering of jobs in N. □

There are other axioms that one can impose to arrive at the Shapley value solution for the equal priority case. For example, observe that the cost inflicted by the first job in the queue to the remaining jobs is equal to the cost incurred by the last job in the queue from the preceding jobs in the equal priority case. We say an allocation rule satisfies zero-sum extreme transfers (ZET) if the transfers of the first and last jobs in the queue add up to zero (i.e., the transfer *received* by the last job in the queue equals the transfer *paid* by the first job in the queue) in the equal priority case. Using ZET in place of ECB, it is possible to arrive at a Lemma analogous to Lemma 3 (Mishra and Rangarajan 2005a). ZET and some other axioms that can provide Lemmas analogous to Lemma 3 are discussed in detail in Mishra and Rangarajan (2005a).

4.3 Generalization of Maniquet's axioms

Next, we generalize the axioms in Maniquet (2003). These axioms, like the axioms in Sect. 4.2, are specific to the scheduling problem.

In an axiom called *equal responsibility*, Maniquet (2003) says that, under equal processing time assumption, if a job is removed from the end of the queue, then the remaining jobs are equally responsible for the waiting cost of the last job and should share the transfer of the removed job equally. Since processing times are not equal in our problem, such an axiom needs an appropriate generalization. In our problem, jobs are responsible for the waiting cost of the last job in proportion to their processing times. To capture this proportional share in waiting cost of the last job, we consider the case when the last job quits the queue. Then, it is not necessary to change the ordering. But the transfer of the last job needs to be redistributed amongst the remaining jobs. *Proportionate responsibility of p* requires that we do this in proportion to their processing times.

Definition 7 *An allocation rule ψ satisfies* proportionate responsibility of p (PRp) *if for all $q \in \mathbb{Q}$, for all $(\sigma, t) \in \psi(q)$, $k \in N$ such that $\sigma_k = |N|$, $q' = (N \setminus \{k\}, p', \theta') \in \mathbb{Q}$, such that for all $i \in N \setminus \{k\}$: $\theta'_i = \theta_i$, $p'_i = p_i$, there exists $(\sigma', t') \in \psi(q')$ such that for all $i \in N \setminus \{k\}$: $\sigma'_i = \sigma_i$ and*

$$t'_i = t_i + t_k \frac{p_i}{\sum_{j \neq k} p_j}.$$

Analogously, the waiting cost inflicted by the first job due to its processing time influences the following jobs in proportion to their θ values. If we remove the first job from the system, the ordering of rest of the jobs do not change, but the transfer of the removed job needs to be redistributed amongst the remaining jobs. The following axiom says that the transfer needs to be shared in proportion to their θ values.[5]

Definition 8 *An allocation rule ψ satisfies* proportionate responsibility of θ (PRθ) *if for all $q \in \mathbb{Q}$, for all $(\sigma, t) \in \psi(q)$, $k \in N$ such that $\sigma_k = 1$, $q' = (N \setminus \{k\}, p', \theta') \in \mathbb{Q}$, such that for all $i \in N \setminus \{k\}$: $\theta'_i = \theta_i$, $p'_i = p_i$, there exists $(\sigma', t') \in \psi(q')$ such that for all $i \in N \setminus \{k\}$: $\sigma'_i = \sigma_i - 1$ and*

$$t'_i = t_i + t_k \frac{\theta_i}{\sum_{j \neq k} \theta_j}.$$

[5] Maniquet (2003) does not use this axiom in his characterizations. But it is a natural generalization of his equal responsibility axiom.

The next set of axioms are a generalization of *independence* axioms in Maniquet (2003). Roughly, these axioms say that if the parameter of a job is changed, then the cost share of every job whose "interaction cost" (i.e., the cost it inflicts on other jobs and the cost it incurs due to other jobs) is unchanged remains the same. In some sense, these axioms say that the cost share of a job depends only on the interaction cost.

The waiting cost of a job does not depend on the per unit time waiting cost of its preceding jobs. So, if we increase the per unit time waiting cost of a job served before job i, without changing parameters of other jobs, then waiting cost of job i is unchanged. Our first axiom says that in such a case the cost share of job i remains unchanged.

Definition 9 *An allocation rule* ψ *satisfies* independence of preceding jobs' θ (IPJθ) *if for all* $q = (N, p, \theta), q' = (N, p', \theta') \in \mathbb{Q}$, $(\sigma, t) \in \psi(q)$, $(\sigma', t') \in \psi(q')$, *if for all* $k \in N, i \in N \setminus \{k\}$: $\theta_i = \theta'_i$, $p_i = p'_i$ *and* $\gamma_k < \gamma'_k$, $p_k = p'_k$, *then for all* $j \in F_k(\sigma)$: $\pi_j = \pi'_j$, *where* π *is the cost share in* (σ, t) *and* π' *is the cost share in* (σ', t').

Similarly, the waiting cost inflicted by a job to its following jobs is independent of the processing times of the following jobs. So, if we increase the processing time of a job following job i, without changing parameters of other jobs, then waiting cost of job i is unchanged. This argument points to an axiom analogous to the previous axiom.

Definition 10 *An allocation rule* ψ *satisfies* independence of following jobs' p (IFJp) *if for all* $q = (N, p, \theta), q' = (N, p', \theta') \in \mathbb{Q}$, $(\sigma, t) \in \psi(q)$, $(\sigma', t') \in \psi(q')$, *if for all* $k \in N, i \in N \setminus \{k\}$: $\theta_i = \theta'_i$, $p_i = p'_i$ *and* $\gamma_k > \gamma'_k$, $\theta_k = \theta'_k$, *then for all* $j \in P_k(\sigma)$: $\pi_j = \pi'_j$, *where* π *is the cost share in* (σ, t) *and* π' *is the cost share in* (σ', t').

5 The characterization results

In this section, we propose three different ways to characterize the Shapley value solution using our axioms. All our characterizations involve efficiency, Pareto indifference, ECB (or, IVM with ED2 in place of ECB). Additionally, we use IPJθ with either of IFJp or PRp, or we use IFJp with either IPJθ or PRθ. Results in this section are built on Lemma 3, which in itself is a nontrivial extension of the ETE axiom in Maniquet (2003) to a more general case where jobs are indistinguishable in any efficient ordering, but are not identical.

Theorem 1 *The following statements are equivalent.*

1. *An allocation rule is the Shapley value rule.*
2. *An efficient allocation rule satisfies Pareto indifference, ECB, IPJ*θ*, and IFJ*p*.*
3. *An efficient allocation rule satisfies Pareto indifference, ECB, IPJ*θ*, and PR*p*.*
4. *An efficient allocation rule satisfies Pareto indifference, ECB, IFJ*p*, and PR*θ*.*

Proof From the definitions, the Shapley value satisfies all the axioms in (2), (3), and (4).

Now, define for any $i,j \in N$, $\theta^i_j = \gamma_i p_j$ and $p^i_j = \frac{\theta_j}{\gamma_i}$. Assume without loss of generality that σ is an efficient ordering with $\sigma_i = i \;\forall\; i \in N$ for $q = (N,p,\theta)$.

(2) implies (1): Consider the following $q' = (N,p',\theta')$ corresponding to job i with $p'_j = p_j$ if $j \leq i$ and $p'_j = p^i_j$ if $j > i$, $\theta'_j = \theta^i_j$ if $j < i$ and $\theta'_j = \theta_j$ if $j \geq i$. Observe that all jobs have the same γ: γ_i and thus, every ordering is efficient. By Pareto indifference and efficiency $(\sigma,t') \in \psi(q')$ for some set of transfers t'. Using Lemma 3, we get cost share of i from (σ,t') as $\pi_i = \frac{1}{2}\theta_i \sum_{j\neq i} p_j = \frac{1}{2}\Big[\theta_i \sum_{j<i} p_j + p_i \sum_{j>i} \theta_j\Big]$. Now, for any $j < i$, if we change θ'_j to θ_j without changing processing time, the new γ of j is $\gamma_j \geq \gamma_i$. Applying IPJθ, the cost share of job i should not change. Similarly, for any job $j > i$, if we change p'_j to p_j without changing θ_j, the new γ of j is $\gamma_j \leq \gamma_i$. Applying IFJp, the cost share of job i should not change. Applying this procedure for every $j < i$ with IPJθ and for every $j > i$ with IFJp, we reach $q = (N,p,\theta)$ and the payoff of i does not change from π_i. Using this argument for every $i \in N$ and using the expression for the Shapley value in Lemma 1, we get the Shapley value solution.

(3) implies (1): Consider a queue with jobs in set $K = \{1,\ldots,i,i+1\}$, where $i < n$. Define $q' = (K,p,\theta')$, where $\theta'_j = \theta^{i+1}_j \;\forall\; j \in K$. Define $\sigma'_j = \sigma_j \;\forall\; j \in K$. σ' is an efficient ordering for q'. By Pareto indifference and efficiency for some transfers t' we have $(\sigma',t') \in \psi(q')$. By Lemma 3 the cost share of job $i+1$ in any allocation rule in ψ must be $\pi_{i+1} = \frac{1}{2}\Big[\sum_{j<i+1} p_j\theta_{i+1}\Big]$. Now, consider $q'' = (K,p,\theta'')$ such that $\theta''_j = \theta^i_j \;\forall\; j \leq i$ and $\theta''_{i+1} = \theta_{i+1}$. σ' remains an efficient ordering in q'' and by IPJθ the cost share of $i+1$ remains π_{i+1}. In $q''' = (K \setminus \{i+1\},p,\theta'')$, we can calculate the cost share of job i using Lemma 3 as $\pi_i = \frac{1}{2}\sum_{j<i} p_j\theta_i$. So, using PR$p$ we get the new cost share of job i in q'' as

$$\pi'_i = \pi_i + t_{i+1}\frac{p_i}{\sum_{j<i+1} p_j} = \frac{1}{2}\Big[\sum_{j<i} p_j\theta_i + p_i\theta_{i+1}\Big]. \tag{3}$$

Now, we can set $K = K \cup \{i+2\}$ and consider $\hat{q} = (K,p,\hat{\theta})$, where $\hat{\theta}_j = \theta^i_j$ $\forall\; j \leq i$ and $\hat{\theta}_j = \theta_j$ for $j \in \{i+1,i+2\}$. As before, using Pareto indifference, efficiency, Lemma 3, and IPJθ, we can find cost share of $i+2$ in the queue $\hat{\theta}$ as $\pi_{i+2} = \frac{1}{2}\Big[\sum_{j<i+2} p_j\theta_{i+2}\Big]$. The cost share of i in the queue $q'' = (K\setminus\{i+2\},p,\hat{\theta})$ is $\frac{1}{2}\Big[\sum_{j<i} p_j\theta_i + p_i\theta_{i+1}\Big]$ (Equation 3). Using PRp we get the new cost share of job i in the queue $\hat{q}$ as $\pi_i = \frac{1}{2}\Big[\sum_{j<i} p_j\theta_i + p_i\theta_{i+1} + p_i\theta_{i+2}\Big]$. This process can be repeated till we add job n at which point cost share of i is $\frac{1}{2}\Big[\sum_{j<i} p_j\theta_i + \sum_{j>i} p_i\theta_j\Big]$. Then, we can adjust the θ of preceding jobs of i to their original value to get the original queue (N,p,θ), and the payoffs of jobs i through n will not change due to IPJθ. This gives us the Shapley values of jobs i through n. Setting $i = 1$, we get cost shares of all the jobs from ψ as their respective Shapley values.

(4) implies (1): This proof mirrors the proof of (3) implies (1). We provide a short sketch. As before, θs are kept equal to original data and processing times are initialized to p_j^{i+1}. This allows us to use IFJp. Also, we consider $K = \{i, i+1, \ldots, n\}$ and repeatedly add jobs to the beginning of the queue maintaining the same efficient ordering. So, we add the cost components of preceding jobs to the cost share of jobs in each iteration and arrive at the Shapley value solution. □

Some comments about our characterization results and the characterization of Maniquet (2003) for the case of $p_1 = \cdots = p_n = 1$ are in order. Observe that we do not use the ETE axiom in our characterizations. But Maniquet uses the ETE axiom for his model. It is clear that identical jobs ought to have identical bargaining power (ETE axiom). But it is not clear as how bargaining power is distributed among jobs of equal priority. We have tried to establish this through IVM with ED2 or ECB. In a sense, the ETE axiom in Maniquet's model makes the cost share of a job single-valued when every ordering of jobs is efficient. This cannot be achieved in our model using the ETE axiom. But it is achieved using ECB (Lemma 3) or IVM with ED2 (Lemma 3) for our model. The "identical preferences lower bound" axiom used in Maniquet (2003) is not satisfied by the Shapley value solution in our model. So, no characterization is possible using it.

Acknowledgements Most of this research was done when the authors were visiting Center for Operations Research and Econometrics (CORE), Belgium. The authors thank two anonymous referees and an associate editor for their comments. The authors are also grateful to François Maniquet, Hervé Moulin, Eilon Solan, Rakesh Vohra, Anna Bogomolnaia, Sidartha Gordon, Jay Sethuraman, and the seminar participants at CORE, Stony Brook, and Caen for their feedback on this work. Few results from this work has appeared as an extended abstract in the sixth ACM conference on Electronic Commerce (Mishra and Rangarajan 2005b).

References

Chun Y (2004a) A note on Maniquet's characterization of the Shapley value in queueing problems. Working Paper, Rochester University

Chun Y (2004b) Consistency and monotonicity in sequencing problems. Working Paper, Seoul National University

Curiel I, Pederzoli G, Tijs S (1989) Sequencing games. Eur J Oper Res 40:344–351

Hamers H, Suijs J, Tijs S, Borm P (1996) The split core for sequencing games. Games Econ Behav 15:165–176

Katta A-K, Sethuraman J (2005) Cooperation in queues. Working Paper, Columbia University

Maniquet F (2003) A characterization of the Shapley value in queueing problems. J Econ Theory 109:90–103

Mishra D, Rangarajan B (2005a) Cost sharing in a job scheduling problem. Technical report, CORE, Discussion Paper 2005-53

Mishra D, Rangarajan B (2005b) Cost sharing in a job scheduling problem using the Shapley Value. In: Proceedings of 6th ACM conference on electronic commerce (EC' 05)

Mitra M (2002) Achieving the first best in sequencing problems. Rev Econ Des 7:75–91

Moulin H (1992a) An Application of the Shapley value to fair division with money. Econometrica 6(60):1331–1349

Moulin H (2004) On scheduling fees to prevent merging, splitting and transferring of jobs. Working Paper, Rice University

Shapley LS (1953) Contributions to the theory of games II. In: Kuhn HW, Tucker AW (eds) Value for *n*-person games. Annals of mathematics studies, pp 307–317, chapt A

Smith WE (1956) Various optimizers for single-stage production. Naval Res Logistics Quarterly 3:59–66

Suijs J (1996) On incentive compatibility and budget balancedness in public decision making. Econ Des 2:193–209