CORC Technical Report TR-2005-3

Cooperation in Queues *

Akshay-Kumar Katta† Jay Sethuraman ‡

May 2005

Abstract

A set of agents arrive simultaneously to a service facility. Each agent requires the facility for a certain length of time and incurs a cost for each unit of time spent in queue. Attached to each agent is an index, which is the ratio of her waiting cost rate to processing time. Efficiency dictates that the agents be served in decreasing order of their indices. If monetary compensations are disallowed, *fairness* suggests that agents be served in a random order, each ordering of the agents being equally likely. The efficient ordering is unfair to agents with low indices; the random service order is typically extremely inefficient. It is well-known that this gap can be bridged by using monetary compensations. This paper is motivated by the need to design compensation schemes that are *fair* to all the agents.

Assuming quasi-linear preferences, we find money transfers for the efficient ordering of the agents so that every coalition of agents is at least as well off in the proposed solution as they are in the random service order. To that end, we propose two solution concepts (RP and CRP core), which serve to place upper bounds on the cost share of any coalition of agents. The solutions differ in the definition of the expected worth of a coalition, when the agents are served in a random order. A detailed study of these two concepts as well as their compatibility with other fairness criteria are the main focus of this paper. We show that the RP core is not convex, and that the CRP core is. Furthermore, we show that standard solution concepts like the Shapley solution (of a related game) and the equal gains solution are also in the CRP core. We describe an efficient algorithm to find the equal costs solution in this core. Finally, we axiomatically characterize the Shapley solution.

*This research was supported by an NSF grant DMI-0093981.

†Department of Industrial Engineering and Operations Research, Columbia University, New York, NY; email: ark2001@columbia.edu

‡Department of Industrial Engineering and Operations Research, Columbia University, New York, NY; email: jay@ieor.columbia.edu

1 Introduction

In this paper, we study a scheduling problem in which a set of agents requiring service arrive simultaneously to a service facility. The facility can serve only one agent at a time; moreover, once an agent begins service, she occupies the service facility until her job is completed. Each agent i requires a processing time p_i at the service facility, and incurs a waiting cost of c_i for each unit of time she spends in queue. (An agent does not incur any cost when being served.) Moreover, these parameters are common knowledge and do not need to be elicited from the agents. Since the server can serve only one agent at a given time, a queue needs to be formed. How should the agents be served? The answer to this question, of course, depends on the objective of the service provider or the planner. If *efficiency* is the goal, the agents must be served in decreasing order of their c/p ratios. However, if the service discipline must be *fair*, and if monetary compensations are disallowed, then a reasonable way to serve the agents is to serve the agents in random order, always picking the next agent to be served uniformly from the remaining set of agents. It is clear that the efficient solution is "unfair," especially to agents with low waiting costs or those with large processing times; similarly, it is easy to see that the "fair" service discipline—serving the agents in a random order—is highly inefficient. Our study is motivated by the following question: if monetary compensations are allowed, and if agents have quasi-linear preferences and are served in the efficient order, what compensations can be viewed as being *fair* ?

The cost sharing problem is a fundamental problem in economics, and has a rich literature [14]. A successful and often-used approach to the "fair" cost sharing problem is to associate with the problem an appropriate cooperative game, and then use a standard solution concept from the theory of cooperative games: for a good overview of this approach in the scheduling context, we refer the reader to Curiel et al. [5]. Our work is inspired by a recent paper of Maniquet [11], who considers the special case of the scheduling problem in which all the agents have identical processing times. In Maniquet [11], the "worth" of a coalition is defined as the least possible cost incurred by that coalition, if no other agents are present; then, monetary compensations are designed so that each agent's utility is precisely her Shapley value in this cooperative game. Chun [3] studies the same problem, but defines the worth of a coalition as the least possible cost incurred by the coalition, assuming the agents *not* in the coalition are served before the coalition members; again, money transfers are designed so as to yield the Shapley value utilities. Both Maniquet [11] and Chun [3] axiomatize the solutions. Curiel et al. [6] consider a model in which the initial ordering of the agents is known; if this initial ordering is not efficient, then the agents could achieve a lower cost by rearranging themselves in an efficient manner. Curiel et al. [6] consider equitable ways of distributing the savings among the agents; they show that the "mid-point" solution (in which each pair of agents split the savings they generate by cooperation) is in the core of an associated cooperative game.

In the model we study, we assume that the waiting costs and processing times are observable or common knowledge. In situations where this is not the case, strategic aspects have to be modeled as well. Dolan [7] provides an incentive compatible (but not budget-balanced) mechanism for revealing the waiting-costs of the agents; in addition to the static case which has a fixed number of agents, Dolan [7] examines the dynamic case where agents arrive over time. Suijs [18] proves the existence of budget balanced incentive-compatible mechanisms for the waiting-cost revelation when there are at least three agents. Mitra [13] considers the case of identical processing times, and identifies the cost structures for which the problems are *First best* implementable. In two recent papers, Moulin [15, 16] discusses ways of designing mechanisms to prevent agents from "merging" jobs or from "splitting" jobs; this is appropriate in situations where the service provider can monitor the processing time of a job, but not the identity of the agent.

Motivation. An important justification for the Shapley solution is the following principle: the cost borne by any subset S of agents should be *at least* their stand-alone cost $L(S)$, which is the least cost they incur if no other agents are present. For the scheduling problem considered here, $L(S)$ is supermodular, so the associated cooperative game is convex, and the Shapley value of this game—the Shapley solution— is always in the core (the LB core). In contrast, our work finds *upper bounds* on the cost of any coalition of agents.

The key motivation behind our approach is the following simple idea: if monetary compensations are not allowed, the only "fair" scheduling policy is the one that serves the agents in a random order, each ordering of the agents being equally likely. We take this to be the standard against which any "solution" to the problem must be measured. This leads naturally to the question of finding monetary compensations such that agents prefer the efficient order (along with the proposed compensation scheme) to the random service order. This question is straightforward to interpret for individual agents: each agent's utility under the proposed compensation scheme should be no smaller than her expected utility under the random service order. But if *arbitrary subsets* of agents can cooperate, then it becomes necessary to define the expected utility of a coalition of agents under the random service order; to do this, we need to specify precisely what an arbitrary coalition of agents can and cannot do. The two solutions concepts we propose, Random Priority (RP) and Constrained Random Priority (CRP), differ only in this respect, and are discussed next.

Consider an arbitrary subset S of agents. We can think of the agents in S as owning a random set $|S|$ positions, each subset of size $|S|$ equally likely. In RP, the agents are free to rearrange themselves in any manner as long as they use only the $|S|$ positions they jointly own; in CRP, however, the agents who rearrange themselves should form a contiguous block. In either case, the cooperating agents will rearrange themselves in an "efficient" queue because this will maximize their resulting savings. To clarify the difference between RP and CRP, suppose $S = \{1, 2, 4\}$ and the random ordering of the agents is (4132). Under RP, the agents in S can rearrange themselves in any manner using the first, second, and fourth position; under CRP, only agents 1 and 4 can rearrange themselves using the first two positions, because agent 3, a non-coalition member, is "in between" agent 1 and agent 2. This restriction imposed by CRP is especially meaningful if p_2 is much bigger than p_4 and p_1 (because allowing agent 2 to move ahead of agent 3 in that case will increase the waiting time of agent 3). The question about monetary compensations posed earlier now becomes the following: can one design compensation schemes in which the utility of every subset of agents is at least as much as their *expected utility* in the random service order? Solutions satisfying this property will be called RP and CRP core solutions respectively, depending on whether the expected utility is computed under RP or CRP. (The expected utility of a coalition under RP will in general be greater than its expected utility under CRP.) We view the presence in the RP or CRP core as our main fairness criterion. To select a solution in the core, we use additional fairness criteria such as equalizing costs, equalizing gains, etc.

Illustrative examples.

Example 1. Suppose there are 3 agents with identical processing times (normalized to 1), but with waiting cost rates $c_1 = c_2 = 12$, and $c_3 = 6$. Let (w_1, w_2, w_3) be the disutility vector of the agents in the proposed solution. The LB core constraints are:

$$w_1 + w_2 \geq 12, \quad w_1 + w_3 \geq 6, \quad w_2 + w_3 \geq 6,$$

$$w_1 + w_2 + w_3 = 24, \quad w_1, w_2, w_3 \geq 0.$$

The disutility vector associated with the Shapley solution is $(9, 9, 6)$. While this solution is "fair" in that it belongs to the LB core, one could argue that this solution overcharges agent 3: the joint cost of the coalition $\{2, 3\}$ is 15 in the Shapley solution, but their *expected* cost in a random ordering is easily seen to be 14. The solution with disutility vector $(10, 10, 4)$ is immune to such difficulties: it is easy to check that in this solution (a) no coalition is charged more than its expected cost in the random ordering; and (b) no coalition is charged less than its stand-alone cost. Thus, the latter solution is a more compelling way for the agents to share their joint costs.

Example 2. Suppose there are 3 agents with identical waiting costs (normalized to 1), but with processing times $p_1 = 4, p_2 = 6$, and $p_3 = 10$. Let (w_1, w_2, w_3) be the disutility vector of the agents in the proposed solution. The LB core constraints are:

$$w_1 + w_2 \geq 4, \quad w_1 + w_3 \geq 4, \quad w_2 + w_3 \geq 6,$$

$$w_1 + w_2 + w_3 = 14, \quad w_1, w_2, w_3 \geq 0.$$

The disutility vector associated with the Shapley solution is $(4, 5, 5)$. It is easy to check that this solution is also in the RP core. While this solution is "fair" in that it belongs to both the RP core and the LB core, there are "better" solutions. The disutility vector $(14/3, 14/3, 14/3)$ is also in the RP core as well as the LB core: moreover it equalizes the costs of all the agents. Clearly, the latter solution is a more compelling way for the agents to share their joint costs if the goal is to "equalize costs" among the core solutions.

To summarize, while the Shapley solution is attractive and enjoys several desirable properties, there could be other solutions to this class of fair cost-sharing problems that are at least as compelling. Our main objective in this paper is to explore such solutions systematically.

Contributions. In this paper we provide a thorough treatment of RP and CRP core solutions, and discuss their relationship to the solutions in the LB core. First we consider the special case in which the the processing times are identical. In this case, we show that the RP core is nonempty and contains all envy-free solutions. Moreover, we show that the Shapley solution and the equal gains solution (which distributes the net benefit equally among the agents) are not in the RP core, but the unique budget-balanced VCG solution is. Furthermore, we show that the RP core is in general incompatible with the LB core: we construct an example in which every RP core solution does not even satisfy the LB core condition for individual agents! In contrast, the CRP core is well-behaved: it gives rise to a convex game; is compatible with the LB core, and in fact contains the Shapley solution; moreover, it is possible to efficiently compute a solution in the CRP core that "equalizes costs," and this is the egalitarian solution proposed by Dutta and Ray [8]. We then turn to the case of identical waiting costs. For this special case, the RP core gives rise to a convex game. It is compatible with the LB core, and in fact contains the Shapley solution; both the equal gains solution (equal-gains) and the equal-cost solution are in the RP core. Since the CRP core contains the RP core, these solutions are all in the CRP core as well. In the general case when processing times and waiting costs are arbitrary, we get a sharp divide: the RP core could be empty; the CRP core, however, contains the Shapley solution (hence is compatible with the LB core), and the equal gains solution. Furthermore, the CRP core is convex, so the egalitarian solution proposed by Dutta and Ray [8] equalizes costs among all core solutions, and, in fact, Lorenz dominates all core solutions; we provide an efficient algorithm to find this solution. For the case with identical waiting costs and for the general case, we provide an axiomatic characterization of the Shapley solution along the lines of Maniquet [11]. Finally, we show how to find solutions in the intersection of the LB core and the CRP core, while satisfying auxiliary conditions such as equalizing costs, etc. These problems are submodular optimization problems that can be solved in polynomial time, but we provide more efficient algorithms for solving them.

The rest of the paper is organized as follows. In §2 we describe the model in detail and outline several desirable properties of solutions that we are interested in; in §3, we discuss two standard solution concepts that are used heavily in the rest of the paper: the Shapley solution and the equal gains solution. Sections 4 and 5 are devoted to the special cases of identical processing times and identical waiting cost rates respectively; in each of these sections we first discuss the RP core and then the CRP core, paying particular attention to the issue of selection from these cores. In §6 we study the general case in which both processing times and waiting costs are identical. We explore solutions that are at the intersection of the LB core and the CRP core in §7.

2 Model

2.1 Problem Description

A set of agents $N = \{1, 2, \ldots, n\}$ arrive to the service system simultaneously for service. Agent i needs to be served for p_i time units and incurs a cost of c_i for each unit of time she waits in queue. (We assume that agents incur a cost only when they wait in queue for service, and do not incur a cost when in service.) The service system can serve only one agent at a time. Furthermore, we assume that once the system starts serving an agent, she has to be served till completion; in standard scheduling terminology, this is the same as requiring non-preemptive service. As we shall see later, even if preemption is allowed, efficiency considerations will force the system manager to adopt a non-preemptive discipline, and so this last assumption is neither unreasonable nor particularly restrictive. Any "solution" to this problem consists of two parts: the order in which the agents are served, which is any permutation of the set of agents N, and the monetary compensations received by the agents from the system manager. Let σ_i be the position of agent i in the service order: thus $\sigma_i = k$ simply means that agent i is served as the kth agent. Also, let t_i be the money received by agent i from the system manager (if $t_i < 0$, then agent i pays the system manager $|t_i|$). Any solution to the problem is thus completely specified by the quantities $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_n)$ and $t = (t_1, t_2, \ldots, t_n)$. We assume that agents' utilities are quasi-linear, and so the utility of agent i is simply

$$u_i(\sigma; t) = t_i - c_i(\sum_{j:\sigma_j < \sigma_i} p_j).$$

Sometimes it will be convenient to work with *disutilities* instead of utilities; we let w_i denote the disutility of agent i, where $w_i := -u_i$. (Obviously, maximizing the sum of the utilities of the agents is the same as minimizing the sum of their disutilities.)

The *Queueing problem* we consider is completely specified by n (or N), $P = (p_1, p_2, \ldots, p_n)$ and $C = (c_1, c_2, \ldots, c_n)$, so we denote any given problem by $q = (N; C, P)$. As stated earlier, any *solution* to the queueing problem q is given by $z = (\sigma, t)$ where σ_i denotes the position of agent i in the service order and t_i is the money received by agent i from the system manager.

A solution z is *feasible* for $q = (N; C, P)$ if σ is a permutation of the agents and $\sum_{i=1}^{n} t_i \leq 0$. The net transfer from the system manager should be non-positive in a feasible solution because we do not want the system manager to incur a loss while serving all the agents. Let $\Sigma(q)$ denote the set of all feasible solutions to a given problem q. A feasible solution z is *efficient* for q whenever it maximizes the sum of the utilities of all the agents among all feasible solutions. It is not difficult to see that a feasible solution $z = (\sigma, t)$ is efficient if and only if (i) σ minimizes the sum of the waiting costs of the agents over all possible permutations of the agents; and (ii) the net transfer $\sum_{i=1}^{n} t_i$ is zero. Smith [17] proved that for this simple scheduling problem the sum of the waiting costs of the agents is minimized by serving them in decreasing order of the ratios c_i/p_i, with ties broken arbitrarily. Let $\Sigma^*(q)$ denote the set of efficient queues for the problem q. Note that the set of efficient queues is always non-empty; moreover, it consists of a single permutation if and only if the ratios c_i/p_i are all distinct. This implies that finding an efficient solution to the problem q is a simple matter: we could choose any $\sigma \in \Sigma^*(q)$ and set t to be the zero vector. Although this is an efficient solution, it could be very unfair: as an extreme example, suppose all the agents have identical p and c values; in the proposed solution, the agent served first has a utility of zero, but the agent served last has a huge negative utility. In the next sections, we explore the possibility of finding efficient solutions that are also "fair." This motivates the next subsection where we define several criteria to capture such properties of interest.

2.2 Fairness Criteria

A *rule* φ associates with each problem q a non-empty subset $\varphi(q)$ of solutions. (The system manager then chooses one solution from this set of solutions. We shall not be concerned about how this choice is made.) A rule φ is *feasible* if for any q, every solution $z \in \varphi(q)$ is feasible; it is *efficient* if every $z \in \varphi(q)$ is efficient. In this paper we shall only consider *feasible* rules; for the most part (but not always), we will also require our rules to be *efficient*. We now define additional desirable properties of rules.

Consider any two solutions z and z' such that $u_i(z) = u_i(z')$ for every $i \in N$. In this case, we view z and z' to be equally desirable, and so we would like the rule to include both z and z', or to exclude both of them. This property is called *Pareto Indifference* (PI). Formally,

Pareto Indifference (PI): A rule φ satisfies PI if for any q, and any $z, z' \in \Sigma(q)$ such that $u_i(z) = u_i(z')$ for every $i \in N$,

$$z \in \varphi(q) \Leftrightarrow z' \in \varphi(q).$$

Next, we turn to two classical equity axioms: *Anonymity* (A) and *Equal Treatment of Equals* (ETE). Anonymity requires the rule to be independent of the identity (names) of the agents, whereas ETE requires that "identical" agents receive identical utilities in any solution. Formally:

Anonymity (A): A rule φ is anonymous if for any $q = (N; C, P)$, and any permutation π of the agents, we have

$$z \in \varphi(N; C, P) \implies \pi(z) \in \varphi(N; \pi(C), \pi(P)).$$

Equal Treatment of Equals (ETE): A rule φ satisfies ETE if and only if for any $q = (N; C, P)$, any $z \in \varphi(q)$,

$$(c_i, p_i) = (c_j, p_j) \Rightarrow u_i(z) = u_j(z), \quad \forall i, j \in N.$$

Suppose the agents in the system cannot reach an agreement or refuse to cooperate with each other. An appealing solution, then, is to serve the agents according to a random permutation, each of the $n!$ permutations being equally likely. The next property—*Fair Share* (FS)—requires that each agent be at least as well off. Note that since all permutations are equally likely, the expected waiting time in queue of a given agent i is simply $\sum_{j:j \neq i} p_j/2$, as any other agent is as likely to precede i as she is to succeed i. Therefore:

Fair Share (FS): A rule φ satisfies FS if for any $q = (N; C, P)$, and any $z \in \varphi(q)$,

$$u_i(z) \geq -c_i \sum_{j:j\in N, j\neq i} \frac{p_j}{2}, \quad \forall i \in N.$$

The next property—*Ranking* (R)—is meant to capture the "fair" requirement that agents who impose a greater strain on the system bear a greater burden of the cost. For a given set of agents N, let $Z(N)$ be maximum possible sum of the utilities of all the agents among all feasible solutions. ($Z(N)$ is unique.) Define the marginal cost of agent i to be $Z(N) - Z(N \setminus \{i\})$. A rule satisfies (R) if agents with a lower marginal cost have a higher utility. Specifically:

Ranking (R): For any problem $q = (N; C, P)$, and any $z = (\sigma, t)$,

$$c_i \sum_{k:\sigma_k<\sigma_i} p_k + p_i \sum_{k:\sigma_k>\sigma_i} c_k \leq c_j \sum_{k:\sigma_k<\sigma_j} p_k + p_j \sum_{k:\sigma_k>\sigma_j} c_k \implies u_i(z) \geq u_j(z), \quad \forall i, j \in N.$$

The next set of properties we discuss deal with how changes in the parameters of one of the agents impacts the utilities of the other agents. First, suppose the processing time of one of the agents, say agent i, increases. Then, the overall waiting costs increase. Specifically, the waiting costs increase because of the increased average waiting time of the group consisting of agent i and all the agents served after i. However, the cost imposed by any agent served before i on everyone else remains the same. The property *Independence of Succeeding Agents' Processing Times* (ISAPT) states that agents served before i should not bear any of the additional cost.

ISAPT: Consider two problems $q = (N; C, P)$ and $q' = (N; C, P')$ such that $p_k < p'_k$ for some $k \in N$ and $p_i = p'_i$ for the other agents. A rule φ satisfies ISAPT if for all $z = (\sigma, t) \in \varphi(q)$, and $z' = (\sigma', t') \in \varphi(q')$,

$$\sigma_i < \sigma_k \implies u_i(z) = u_i(z').$$

By the same reasoning, if the waiting cost of agent i increases, then agents served after i should not bear any of the extra cost. This property, called *Independence of Preceding Agents' Impatience* (IPAI), is formally stated as:

IPAI: Consider two problems $q = (N; C, P)$ and $q' = (N; C', P)$ such that $c_k < c'_k$ for some $k \in N$ and $c_i = c'_i$ for the other agents. A rule φ satisfies IPAI if for all $z = (\sigma, t) \in \varphi(q)$, and $z' = (\sigma', t') \in \varphi(q')$,

$$\sigma_i > \sigma_k \implies u_i(z) = u_i(z').$$

Our final property deals with how a change in the set of agents affects the overall utilities. Suppose agents i and j are served before an agent k. Then, the waiting cost incurred by k due to i is proportional to p_i and the cost incurred due to j is proportional to p_j. Now consider the situation in which the last agent quits the society. The order of the remaining agents does not change. The only change is that the transfer that was allocated to the last agent has to be redistributed among the remaining agents. If a rule redistributes this transfer among the remaining agents in amounts proportional to their processing times, it satisfies ***Proportional Responsibility*** (PR).

3 Standard solution concepts

In this section, we review two standard solution concepts in cooperative game theory and show how one can compute them explicitly for the model discussed here. This serves multiple purposes: first, for some special cases of the model, similar approaches have been used to find compelling rules; and second, we shall use many of these ideas in the latter sections. To avoid cumbersome notation, we assume (in this section) that agents are served in the order $(1, 2, \ldots, n)$, and that this is an efficient queue. In particular, the agents are labeled so that $c_1/p_1 \geq c_2/p_2 \geq \ldots \geq c_n/p_n$.

3.1 Shapley solution

One way to find a good rule for the scheduling problem described here is to view it as a cooperative game among the agents, and find a solution in which each agent's utility is exactly her Shapley value in this cooperative game. As mentioned earlier, for the special case of this model with identical processing times, the papers of Maniquet [11] and Chun [3] do just that. It is a simple matter to extend those ideas to the case when processing times are not identical. We shall concentrate on extending the solution proposed by Maniquet [11], as that will be useful in later sections of the paper as well; computing the analog of Chun's solution is a straightforward exercise.

A cooperative game is defined by specifying the worth of every possible coalition $S \subseteq N$ of agents. For the scheduling problem we consider, this worth can be defined in many ways; for instance, the work of Maniquet [11] and Chun [3] (for a special case of the scheduling problem) differ primarily in the way the worth of a coalition is defined. Following Maniquet [11], we define the worth of a coalition $S \subseteq N$ as the minimum possible sum of its members' waiting costs, if they were the only agents present (equivalently, if the agents in S had the power to be served before the agents not in S). That is,

$$v_q(S) = -\sum_{i \in S} c_i(\sum_{j:j<i} p_j).$$

Given this definition, the Shapley value of agent an $i \in N$ is

$$SV_i \;=\; \sum_{S \subseteq N-i} \frac{|S|!(|N|-|S|-1)!}{|N|!}[v_q(S \cup \{i\}) - v_q(S)].$$

The Shapley value can be computationally expensive to compute, but has an appealing interpretation, which is often useful in computing it. Suppose the agents "arrive" in a random order, every ordering of the agents being equally likely. Suppose also that each agent is asked to pay for the additional cost she imposes on the agents who are already present. The Shapley value of an agent i is then simply her expected payment: this is because i pays $(v_q(S \cup \{i\}) - v_q(S))$ if and only if the set of agents arriving before her is exactly the agents in S (in any order), and this event has a probability of $|S|!(|N|-|S|-1)!/|N|!$. Therefore, the Shapley value of agent i is simply

$$SV_i \;=\; E[v_q(S \cup \{i\}) - v_q(S)],$$

where the expectation is with respect to S, the (random) set of agents who arrive before i in a uniform random ordering of N. We now turn to computing the Shapley value of an agent using this interpretation.

Recall that the agents are served in the order $(1, 2, \ldots, n)$ in an efficient queue. Observe that if a subset S of $N \setminus \{i\}$ is chosen uniformly at random, then for any agent $j \neq i$, $\Pr[j \in S] = \Pr[j \notin S] = 1/2$. Also, given a subset S of agents not containing i, the marginal contribution of agent i to S consists of two parts: first, the waiting cost of agent i herself; and second, the *additional* cost imposed by agent i on all the members of S. Thus,

$$v_q(S \cup \{i\}) - v_q(S) \quad = \quad -c_i(\sum_{j \in S: j<i} p_j) - p_i(\sum_{j \in S: j>i} c_j). \tag{1}$$

Taking expectations in Eq. 1, we have

$$SV_i(v_q) \quad = \quad -\frac{1}{2}\big[c_i(\sum_{j:j<i} p_j) + p_i(\sum_{j:j>i} c_j)\big]. \tag{2}$$

The Shapley value rule for a problem $q = (N; C, P)$ finds an *efficient solution* in which each agent's utility is exactly her Shapley value. Let (σ, t) be such a solution. Assuming $\sigma = (1, 2, \ldots, n)$, we shall find the transfer vector $(t_1, t_2, \ldots, t_n)$ that achieves the Shapley value utility for each agent. In other words, we want

$$SV_i \; = \; u_i(\sigma, z) \quad = \quad -c_i(\sum_{j:j<i} p_j) + t_i. \tag{3}$$

Using Eq. (2) in Eq. (3), we find

$$t_i \quad = \quad = \frac{1}{2}\big[c_i(\sum_{j:j<i} p_j) - (p_i \sum_{j:j<i} c_j)\big]. \tag{4}$$

It is an elementary exercise to check that the Shapley solution satisfies all the fairness criteria discussed in §2.2.

3.2 Equal Gains solution

We now turn to another well-studied solution concept in games—the equal gains solution. If the agents do not cooperate and we implement the random service order solution, then agent i's expected waiting time is $\sum_{j:j\neq i} p_j/2$. Letting

$$d_i = -c_i \sum_{j:j\neq i} p_j/2 \tag{5}$$

the vector of utilities in the random service order is $(d_1, d_2, \cdots, d_n)$.

If the agents cooperate, however, then any vector of utilities $(u_1, u_2, \ldots, u_n)$ is achievable as long as

$$\sum_{i:i\in N} u_i \quad = \quad -\sum_{i\in N} \sum_{j:j<i} c_i p_j \tag{6}$$

The RHS of Eq. (6) is simply the (negative of the) minimum possible total waiting cost of the agents. Since an efficient solution has zero net transfer, the total sum of utilities of the agents in any efficient solution must equal the RHS of Eq. (6). Thus, the net benefits from cooperation is given by

$$B \quad = \quad \frac{1}{2}\sum_{k=1}^{n} c_k \sum_{j\neq k} p_j - \sum_{k=1}^{n} c_k \sum_{j<k} p_j. \tag{7}$$

The equal gains solution divides these benefits equally among all the agents. Thus, the equal gains utility of agent i is $u_i = d_i + B/n$, where d_i and B are given by Eq. (5) and Eq. (7) respectively. Since the waiting cost of agent i in an efficient queue is $c_i \sum_{j<i} p_j$, the transfer

$$t_i = d_i + B/n + c_i \sum_{j<i} p_j$$

achieves the equal gains utility vector for agent i.

It is straightforward to check that the equal gains solution satisfies Pareto Indifference, Anonymity, Equal Treatment of Equals, and Fair Share; it is also clear that the equal gains solution does not satisfy ISAPT, IPAI, PR, and Ranking. We also note that, by definition, each agent is at least as well off under the equal gains solution as she is under the random service order. While coalitions of agents do not play a role in this solution, we shall later show that even *arbitrary* coalitions of agents are at least as well off in the equal gains solution as they would be if the service order is random.

4 Identical Processing times

In this section we consider the special case in which all the jobs are of unit length. This particular special case has received a lot of attention in the recent literature. We assume that the processing-time requirements of the agents' jobs are identical; without loss of generality we take this common job-length to be 1. As before, we assume (throughout this section) that the agents are labeled so that $c_1 \geq c_2 \geq \ldots \geq c_n$.

If monetary compensations are not permitted, one "fair" solution to the scheduling problem is to serve the agents in a random order, each ordering being equally likely. This observation leads to the following natural question: if monetary compensations are permitted, can we design a compensation scheme such that the total utility of any coalition of agents is *at least* its expected utility in the random ordering? To fully answer this question, we have to describe precisely what the expected utility of a coalition is under the random order. As mentioned earlier, the two core solutions we propose—RP and CRP core—differ only in this respect. Any coalition S of agents will occupy a random set of $|S|$ slots. In RP, the agents are free to rearrange themselves (and redistribute the resulting savings among themselves in any way they like) in any order as long as they use the same $|S|$ slots; in CRP, only contiguous subsets of agents can rearrange themselves using the slots they occupy. In either case, the cooperating agents will rearrange themselves in an "efficient" queue because this will maximize their resulting savings.

4.1 Random Priority (RP)

As stated earlier, in RP, any cooperating subset of agents can rearrange themselves using the slots they occupy. This does not affect the welfare of any non-coalition member, regardless of where such agents appear, making the rules of cooperation under RP appealing and realistic. Our goal is to characterize the set of transfers $(t_1, t_2, \ldots, t_n)$ such that in the efficient queue $\sigma^* = (1, 2, \ldots, n)$, the utility of every subset of agents is at least as much as its expected utility in a random ordering. Such a solution (σ^*, t) is said to be in the RP core.

4.1.1 The RP core

Consider a coalition, $S = \{a_1, a_2, \ldots, a_s\}$, of s members with $c_{a_1} \geq c_{a_2} \geq \ldots \geq c_{a_s}$. Now suppose that the processing order is chosen uniformly at random from the $n!$ possible orderings. Clearly, the cooperating coalition members will rearrange themselves in the order $(a_1, a_2, \ldots, a_s)$ using the time slots they occupy in the chosen processing order. For $i = 1, 2, \ldots, s-1$, let N_i be the (random) number of non-coalition members between a_i and a_{i+1}; also, let N_0 be the number of non-coalition members before a_1, and N_s be the number of non-coalition members after a_s. Clearly, the $(s+1)$ random variables $N_0, N_1, \ldots, N_s$ are identically distributed, and add up to $n-s$, so $E[N_i] = (n-s)/(s+1)$. Thus, the expected number of agents served before a_i, which is also the expected waiting time of a_i, is $(i-1)+i(n-s)/(s+1) = (n-s)/(s+1) \, + \, (i-1)(n+1)/(s+1)$, for $i = 1, 2, \ldots, s$.

Using all this, we have

$$C(S) \quad = \quad \frac{n-s}{s+1}\sum_{i=1}^{s} c_{a_i} + \frac{n+1}{s+1}\sum_{i=2}^{s-1}(i-1)c_{a_i}. \tag{8}$$

Our main result is that the RP core is nonempty. We prove this result by showing that it contains all *envy-free* solutions, defined as follows: Given a solution $z = (\sigma, t)$, we say i does not envy j in z if i prefers her allocation to that of j's, that is, if

$$-(\sigma_i - 1)c_i + t_i \geq -(\sigma_j - 1)c_i + t_j.$$

A solution z is *envy-free* if no agent envies any other agent. Note that in the definition of envy-freeness, we assume that transfers are attached to the "positions," not the individual agents, so t_j is the transfer to the agent served at position σ_j. We are now ready to state and prove the main result of this section. The proof is similar to a proof of Klijn et al. [10] for a different problem.

Theorem 1 *Every envy-free solution is in the RP core. In particular, the RP core is nonempty.*

Proof. Let (σ, t) be an envy-free solution to the scheduling problem. Let Π be the set of all the random orderings of N; since each random ordering is equally likely, for any $\pi \in \Pi$, $Pr[\pi] = 1/N!$. Recall that the utility of agent i is $u_i \; = \; -(\sigma_i - 1)c_i + t_i$.

For any $\pi \in \Pi$, let $\pi_S(i)$ be the position of agent i when the cooperating coalition of agents is S. (Thus, the agents in S will rearrange themselves in an efficient way—agents with higher cost will appear earlier— using the slots they occupy in π.) Then, the expected waiting cost of coalition S is given by

$$C(S) \; = \; \frac{1}{N!}\sum_{\pi\in\Pi}\sum_{i\in S}(\pi_S(i) - 1)c_i.$$

This expression will be the same as the expression in Eq. (8), but it will be convenient here to think of it in these terms. Also, note that the worth of a coalition S is just $-C(S)$. We have:

$$
\begin{aligned}
\sum_{i\in S} u_i &= \sum_{i\in S}(-(\sigma_i - 1)c_i + t_i) \\
&= \frac{1}{N!}\sum_{\pi\in\Pi}\sum_{i\in S}(-(\sigma_i - 1)c_i + t_i) \\
&= \frac{1}{N!}\sum_{\pi\in\Pi}\sum_{i\in S}(-(\sigma_i - 1)c_i + t_i - t_{\pi_S(i)}) + \frac{1}{N!}\sum_{\pi\in\Pi}\sum_{i\in S} t_{\pi_S(i)} \\
&\geq \frac{1}{N!}\sum_{\pi\in\Pi}\sum_{i\in S} -(\pi_S(i) - 1)c_i + \frac{1}{N!}\sum_{\pi\in\Pi}\sum_{i\in S} t_{\pi_S(i)} \\
& \quad \text{(envy-freeness implies } -(\sigma_i - 1)c_i + t_i \geq -(\pi_S(i) - 1)c_i + t_{\pi_S(i)} \text{)} \\
&= -C(S) + \frac{1}{N!}\sum_{\pi\in\Pi}\sum_{i\in S} t_{\pi_S(i)}
\end{aligned}
$$

But, note that $\frac{1}{N!}\sum_{\pi\in\Pi}\sum_{i\in S} t_{\pi_S(i)} = 0$, because every position will be occupied the same number of times on average and $\sum_{i=1}^{n} t_i = 0$. So, for any envy-free solution (σ, t), and any coalition of agents S,

$$\sum_{i\in S} u_i \geq -C(S).$$

Therefore all envy-free solutions are in the RP core.

It is easy to verify that the efficient queue $(1, 2, \ldots, n)$, and the transfer vector given by

$$
\begin{aligned}
t_1 &= -\frac{(n-1)c_2 + (n-2)c_3 + \ldots c_n}{n}, \\
t_i &= t_1 + c_2 + \ldots + c_i, \quad i = 2, \ldots, n,
\end{aligned}
$$

constitutes an efficient, envy-free solution. ■

Remarks.

(a) Chun [4] has explored the role of envy in such scheduling problems. It is not hard to show that the solution $(\sigma = (1, 2, \ldots, n), t)$ is envy-free if and only if

$$c_{i+1} \leq t_{i+1} - t_i \leq c_i, \quad \forall i = 1, 2, \ldots, n-1.$$

(b) The converse of Theorem 1 is not true: not all core solutions are envy-free. For instance, suppose there are three agents with $c_1 > c_2 > c_3$. Consider the solution given by

$$t_1 = -\frac{2c_2 + c_3}{3}, t_2 = \frac{c_2 - c_3}{3} + \delta, t_3 = \frac{c_2 + 2c_3}{2} - \delta$$

It is easy to check that this solution is in the RP core for a small enough $\delta > 0$. But agent 3 envies agent 2 in this solution.

4.1.2 Selection from the RP core

We know that the RP core contains all envy-free solutions and some other solutions as well. We now consider the problem of selecting an appealing solution (or subset of solutions) from the RP core.

Shapley value and the LB core. Recall that the RP core constraints give an upper bound on the disutility of each coalition of agents; in contrast, the LB core constraints give a lower bound on the disutility of each coalition of agents. A particularly appealing class of solutions would have been those that are members of both cores. When there are only two agents, it is easy to check that the disutility vector $(c_2/2, c_2/2)$ (achieved by $\sigma = (1, 2)$, $t_2 = -t_1 = c_2/2$) is in both the RP core and the LB core. Unfortunately, if there are 3 or more agents, it is sometimes impossible to find such a solution: there are instances of the scheduling problem in which any solution in the RP core is not in the LB core. As we show next, such solutions do not even satisfy the LB core constraints for *individual* agents. (These constraints require no agent to be subsidized by the system for participation.)

Example 3. Suppose there are three agents with costs $c_1 \geq c_2 \geq c_3$. The RP core constraints are:

$$w_1 \leq c_1 \quad w_2 \leq c_2 \quad w_3 \leq c_3,$$

$$w_1 + w_2 \leq \frac{c_1}{3} + \frac{5}{3}c_2 \quad w_1 + w_3 \leq \frac{c_1}{3} + \frac{5}{3}c_3 \quad w_2 + w_3 \leq \frac{c_2}{3} + \frac{5}{3}c_3,$$

and

$$w_1 + w_2 + w_3 \ = \ c_2 + 2c_3.$$

The LB core constraints for individual agents simply say that $w_1, w_2, w_3 \geq 0$. Suppose $c_1 = c_2 = 3$. If $c_3 = 0$, we see that w_3 is forced to be 0, which implies $w_1 \leq 1$ and $w_2 \leq 1$. But, the last constraint of the RP core requires $w_1 + w_2 + w_3 = 3$, which is not possible.

Since the Shapley value solution discussed in §3.1 is always in the LB core, the same example proves that the Shapley value solution need not always be in the RP core.

Equal gains solution. The equal gains solution may not always be in the RP core either. Suppose there are four agents with costs $c_1 \geq c_2 \geq c_3 \geq c_4$. The equal gains solution to the scheduling problem is

$$\begin{aligned}
w_1 &= \frac{9}{8}c_1 - \frac{1}{8}c_2 + \frac{1}{8}c_3 + \frac{3}{8}c_4, \\
w_2 &= -\frac{3}{8}c_1 + \frac{11}{8}c_2 + \frac{1}{8}c_3 + \frac{3}{8}c_4, \\
w_3 &= -\frac{3}{8}c_1 - \frac{1}{8}c_2 + \frac{13}{8}c_3 + \frac{3}{8}c_4, \\
w_4 &= -\frac{3}{8}c_1 - \frac{1}{8}c_2 + \frac{1}{8}c_3 + \frac{15}{8}c_4.
\end{aligned}$$

The RP core constraint for the coalition $\{1, 2, 3\}$ of agents is

$$w_1 + w_2 + w_3 \leq \frac{c_1}{4} + \frac{3}{2}c_2 + \frac{11}{4}c_3.$$

If $c_1 = 8$, and $c_2 = c_3 = c_4 = 0$, the equal gains solution has $w_1 = 9$, $w_2 = w_3 = w_4 = -3$, so $w_1 + w_2 + w_3 = 3$; but this violates the RP core constraint.

Thus, even though the RP core is always non-empty, neither the Shapley value solution nor the equal gains solution may be in it. Also, the function $C(S)$ representing the upper bound on the disutility of coalition S is not a submodular function, so the corresponding "game" is not well-behaved. Moreover, for the case with arbitrary processing times and waiting cost rates, allowing coalition members to trade positions may force the RP core to be empty (as we show later in §6). All of these considerations motivate the CRP solution, which is discussed in the next section. We end this section, however, on a positive note: we show that the (unique) budget-balanced VCG solution is in the RP core.

Budget-balanced VCG solution. When there are at least three agents, Suijs [18] showed that the budget-balanced VCG transfers are given by

$$t_i \ = \ \frac{1}{2}\sum_{j<i} c_j - \frac{1}{2}\sum_{j>i} c_j - \frac{1}{n-2}\sum_{j:j\neq i}\ \sum_{(k:k\neq i,k<j)} \frac{1}{2}(c_k - c_j).$$

To show that this solution is in the RP core, it is enough to show that it is envy-free, which follows (by Chun [4]) if $t_{i+1} - t_i \in [c_{i+1}, c_i]$. We have:

$$t_{i+1} - t_i = \frac{c_i}{2} + \frac{c_{i+1}}{2} + \frac{2i-n}{2(n-2)}[c_i - c_{i+1}] = \frac{n-i-1}{n-2}c_i + \frac{i-1}{n-2}c_{i+1}. \tag{9}$$

Thus, $t_{i+1} - t_i$ is a convex combination of c_i and c_{i+1}; the solution induced by these transfers is envy-free and so is in the RP core.

Remark. The expressions for the transfers are valid only when $n \geq 3$. If $n = 2$, the transfer vector $t_2 = -t_1 = c_2/2$ is in the RP core (and in the LB core).

4.2 Constrained Random Priority (CRP)

In CRP, only *contiguous* subsets of cooperating agents can rearrange themselves using the slots they occupy; the requirement of contiguity distinguishes this from the RP core. Our goal, again, is to characterize the set of transfers $(t_1, t_2, \ldots, t_n)$ such that in the efficient queue $\sigma^* = (1, 2, \ldots, n)$, the utility of every subset of agents is at least as much as its expected utility in a random ordering. Such a solution (σ^*, t) is said to be in the CRP core.

4.2.1 The CRP core

Consider a coalition, S, of agents with costs $c_{a_1} \geq c_{a_2} \geq \ldots \geq c_{a_s}$, where $s = |S|$. In a uniform random ordering of the agents, the expected cost of the agents in S is

$$\frac{n-1}{2}(c_1 + \ldots + c_s).$$

However, if the agents in S cooperate, they will generate some cost-savings by rearranging themselves in the efficient queue $(a_1, a_2, \ldots, a_s)$ using the slots they occupy. We assume that *any* smaller subset of agents in S who are adjacent can still rearrange their positions, even if the entire set of agents in S do not form a contiguous block. For instance, let $n = 4$, and let agents $\{1, 3, 4\}$ be a cooperating coalition. If the random order is 3241, then agents 1 and 4 are allowed to swap their positions to generate a net savings of $c_1 - c_4$. We next derive a simple expression for the cost savings achieved by S.

Consider two agents i and j, with $i < j$. (By our convention, $c_i \geq c_j$.) If both i and j are part of a cooperating coalition S, under what conditions will they change their relative order? This will occur if and only if (i) j is before i in the random order; and (ii) every agent between j and i is a member of S. (Recall that not all the members of S need to be contiguous for i and j to exchange their positions.) Since these events do not depend on the identity of S, only on its cardinality, we denote this probability by K_s. We compute K_s from the following simple observation: one way to generate a uniform random ordering of the agents is to start with a fixed ordering of the agents, and place them in "queue" randomly, one at a time; for $k \geq 2$, the kth agent is placed in any one of the k possible slots, each possibility being equally likely. Suppose we start with the ordering $[i, j, N \setminus S, S \setminus \{i, j\}]$. For the event of interest to occur, j must appear before i, and every agent between i and j should be a member of S. The former has a probability of 1/2; the latter occurs only if each of the $n - s$ members of $N \setminus S$ is *not* placed between j and i, which has a probability of $2/3 \times 3/4 \times \ldots \times (n-s+1)/(n-s+2) = 2/(n-s+2)$. Hence, $K_s = 1/(n - s + 2)$. Since our argument did not depend on the identity of i and j, we may conclude that the probability of an exchange between any two members of the coalition S is the same, and equals $1/(n - s + 2)$. Therefore, the expected waiting cost of the coalition S is given by

$$\begin{aligned} C(S) &= \frac{n-1}{2}(c_{a_1} + \ldots + c_{a_s}) - \frac{1}{n+2-s} \sum_{i,j \in S, i<j} (c_{a_i} - c_{a_j}) \\ &= \frac{n-1}{2}(c_{a_1} + \ldots + c_{a_s}) - \frac{1}{n+2-s} \sum_{k=1}^{s} (s - 2k + 1) c_{a_k}. \end{aligned} \tag{10}$$

In contrast to the RP core, the CRP core is well-behaved: the function $C(\cdot)$ is a submodular function, and so the associated cooperative game is a *convex* game.

Proposition 1 *Let $C(S)$ denote the expected waiting cost of a coalition S of agents under CRP. $C(S)$ is a submodular set function.*

Proof. To prove that $C(\cdot)$ is a submodular set function, we show that

$$C(S \cup \{k\}) - C(S) \geq C(T \cup \{k\}) - C(T), \quad \forall S \subset T, \;\; k \notin S \tag{11}$$

We have

$$C(S \cup \{k\}) - C(S) = \frac{n-1}{2} c_k - (\frac{1}{n+1-s} - \frac{1}{n+2-s}) \sum_{i,j \in S, i<j} (c_i - c_j) - \frac{1}{n+1-s} \sum_{i \in S} |c_k - c_i|$$

and

$$C(T \cup \{k\}) - C(T) = \frac{n-1}{2} c_k - (\frac{1}{n+1-t} - \frac{1}{n+2-t}) \sum_{i,j \in T, i<j} (c_i - c_j) - \frac{1}{n+1-t} \sum_{i \in T} |c_k - c_i|$$

Since $S \subset T$, $s < t$; under these conditions, it is easy to verify that each term in $C(S \cup \{k\}) - C(S)$ is at least as big as the corresponding term in $C(T \cup \{k\}) - C(T)$. The submodularity of $C(\cdot)$ follows. ■

4.2.2 Selection from the CRP core

We now turn to the problem of finding an appealing solution in the CRP core.

Equalizing costs. Our goal is to find a solution such that the components of the associated disutility vector $(w_1, w_2, \ldots, w_n)$ are as equal as possible. Recall that

$$W := \sum_i w_i = c_2 + 2c_3 + \ldots + (n-1)c_n$$

for all achievable disutility vectors. Thus, the ideal disutility vector from this objective's point of view would be $(W/n, W/n, \ldots, W/n)$, but this vector may not be in the CRP core. The natural question, then, is one of identifying a vector in the CRP core that equalizes the disutilities as much as possible. Since $C(\cdot)$ is submodular, the egalitarian solution proposed by Dutta and Ray [8] exists, is in the core of the associated game, and Lorenz dominates every other core solution; it is easy to see that this solution will also equalize costs in the sense just described. Dutta and Ray [8] propose a general algorithm to compute the egalitarian solution of any convex game. Their algorithm (adapted to this "cost" version) is as follows: (i) identify *the* largest coalition S^* that minimizes $C(S)/|S|$, over all nonempty subsets S of N, and let $w_i = C(S^*)/|S^*|$ for all $i \in S^*$; (ii) define $\hat{N} = N \setminus S^*$, and for every $S \subseteq \hat{N}$, define $\hat{C}(S) \leftarrow C(S \cup S^*) - C(S^*)$; and (iii) apply (i) recursively, with $\hat{N}$ playing the role of N, and $\hat{C}(\cdot)$ playing the role of $C(\cdot)$. Note that submodularity of $C(\cdot)$ implies the existence of S^* in step (i), and the submodularity of the $\hat{C}(\cdot)$ function defined in step (ii). The most expensive computation in this algorithm is the identification of the "bottleneck" set S^* in step (i). While this problem can always be solved as a submodular optimization problem for any convex game, it can be solved much more efficiently for the scheduling problem, as the following simple proposition shows:

Proposition 2 *Let $C(S)$ be as in Eq. (10). Then,*

$$C(\{s+1,\ldots,n\}) \;=\; \min_{S\subseteq N, |S|=n-s} C(S), \quad \forall s = 0,1,\ldots,n-1.$$

Proof. Consider the expression for $C(S)$:

$$C(S) \;=\; \frac{n-1}{2}(c_{a_1} + \ldots + c_{a_s}) - \frac{1}{n+2-s}\sum_{k=1}^{s}(s-2k+1)c_{a_k}. \tag{12}$$

The coefficient of c_{a_k} in the second term increases as k increases, so c_{a_1} has the smallest coefficient, which is $(n-1)/2-(s-1)/(n+2-s)$; this itself decreases as s increases, so its smallest value is zero, which occurs when $s = n$. Thus, we see that the coefficient of c_{a_i} in the expression for $C(S)$ is always non-negative, and increasing with i. So to minimize $C(S)$ among all sets of cardinality S, it is optimal to include the s smallest cost coefficients. ■

By Proposition 2, the bottleneck set in the original problem can be found by examining at most n sets, each of the form $\{s, s+1, \ldots, n\}$, for $s = 1, 2, \ldots, n$. In the recursive application of the algorithm, the same argument holds because the function $\hat{C}(\cdot)$ is obtained from $C(\cdot)$ by subtracting the same constant from every coalition. This algorithm can be restated in the following constructive form:

- Let the initial disutility of al the agents be zero; start increasing all of them at the same rate. When the disutility of all the agents reaches $\frac{n-1}{2}c_i$, it is no longer possible to increase the disutility of agent n without going out of the core. This point will definitely be reached because the Maniquet solution is in the **CRP core**. At this point, fix the disutility of agent n and increase the disutilities of the rest of the agents at the same rate.

- When the disutilities reach a certain threshold value, we would be at the core boundary once again. The set which reaches this threshold will be of the form $S = \{n, n-1, ..., n+1-s\}$. Hence the threshold value can be calculated b considering all the possible values of s.

- Once we find the threshold value, we fix the disutilities of the agents in S and increase the disutilities of the remaining agents. We repeat like in step 2 until the set N becomes the bottleneck set.

As mentioned earlier, the solution that equalizes costs Lorenz dominates every other solution in the CRP core. By standard results in submodular optimization [9, Theorem 8.2, Theorem 10.1, Corollary 10.2], this solution maximizes the minimum disutility, minimizes the maximum disutility; moreover, the disutility vector found is also lexicographically minimal among all disutility vectors in the core.

Equalizing gains. The goal here is to find a disutility vector that equalizes the "gains" of the agents as much as possible. Recall that the gain of agent i in a solution $(w_1, w_2, \ldots, w_n)$ is $(n-1)c_i/2 - w_i$. In §6, we shall prove that the equal gains solution is in the CRP core. Since the equal gains solution distributes the "gains" equally among all the agents, this will be the solution equalizing the gains in the CRP core.

Shapley value. We show that the Shapley value solution described in §3.1 is in the CRP core. Again, the proof is postponed to §6, where this result is shown to hold in the more general setting in which agents have arbitrary processing times and waiting costs. In view of the characterization results of Maniquet [11], this also means that the CRP core is well-behaved; in particular, it allows for solutions that meet all the criteria discussed in §2.2.

5 Identical Waiting Cost Rates

We now turn to the case in which agents have jobs with different processing times, but their waiting cost rates are identical. We take this common waiting cost rate to be 1. Let the processing time of i's job be p_i; without loss of generality, we assume that $p_1 \leq p_2 \leq \ldots \leq p_n$, which implies $\sigma^* = (1, 2, \ldots, n)$ is an efficient queue. For this model, we consider the *RP* and *CRP* solutions.

5.1 Random Priority (RP)

Recall that in RP, any cooperating subset of agents can rearrange their positions to lower their joint costs. Since jobs have different processing times, so this may affect the welfare of some of the agents who are not in the coalition. However, since any efficient rearrangement will result in jobs with smaller processing times placed earlier, the disutility of any non-coalition agent can only go down. Therefore, RP is a realistic model in this setting. As before, a solution is in the RP core if the utility of every subset of agents is at least as much as their expected utility in a uniform random ordering.

5.1.1 The RP core

Consider a coalition $S = \{a_1, a_2, \ldots, a_s\}$, of s members, with $p_{a_1} \leq p_{a_2} \leq \ldots \leq p_{a_s}$. We first determine $C(S)$, the expected total disutility of the agents in S, when the agents are ordered uniformly at random, and when the members of S rearrange themselves in the order $(a_1, a_2, \ldots, a_s)$ using the positions they occupy. We say that $C(S)$ is the expected waiting cost of S under RP. In the absence of any cooperation, the expected cost of agent i is $\sum_{j \in N, j \neq i} p_j/2$, which can be rewritten as $(P - p_i)/2$, where $P := \sum_{j \in N} p_j$ is the sum of all the processing times of the agents' jobs. Thus, in the absence of any cooperation, the total waiting cost of the agents in S is

$$\frac{1}{2}[sP - \sum_{i=1}^{s} p_{a_i}].$$

By cooperating, the agents in S generate some cost savings, which can be found as follows: suppose $i < j$, and a_i appears after a_j, and no agent between them is in S. Then, by exchanging their positions, a_i and a_j generate a saving of $(p_{a_j} - p_{a_i})$. Given any random order, it is straightforward to see that the agents in S can rearrange themselves in an efficient queue by making such exchanges between "adjacent" pairs of agents. Since any pair of agents need to swap positions with probability $1/2$, the expected savings generated by S is simply

$$\frac{1}{2}\sum_{i=1}^{s-1}\sum_{j=i+1}^{s}(p_{a_j} - p_{a_i}).$$

Therefore,

$$\begin{aligned} C(S) &= \frac{1}{2}\big[sP - \sum_{i=1}^{s} p_{a_i} - \sum_{i,j\in S, i<j}(p_{a_j} - p_{a_i})\big] \\ &= \frac{1}{2}\big[sP + \sum_{i=1}^{s}(s-2i)p_{a_i}\big] \qquad (13) \\ &= \frac{s}{2}\sum_{j:j\notin S} p_j + \sum_{i=1}^{s}(s-i)p_{a_i} \qquad (14) \end{aligned}$$

Proposition 3 *Let $C(S)$ be the expected waiting cost of S under RP. Then $C(S)$ is a submodular set function.*

Proof. Let $S \subset T$, and let $k \notin T$. The result follows if we prove the following inequality:

$$C(S\cup\{k\}) - C(S) \;\geq\; C(T\cup\{k\}) - C(T).$$

$$C(S\cup\{k\}) - C(S) = \frac{1}{2}\big[(P - p_k) - \sum_{i\in S}|p_k - p_i|\big],$$

and, similarly,

$$C(T\cup\{k\}) - C(T) = \frac{1}{2}\big[(P - p_k) - \sum_{i\in T}|p_k - p_i|\big].$$

Since

$$\sum_{i\in S}|p_k - p_i| \;\leq\; \sum_{i\in T}|p_k - p_i|,$$

the inequality follows. ■

We end this section with the following result, which will be useful later on.

Proposition 4 *Let $C(S)$ be the expected waiting cost of S under RP. Let*

$$\hat{S} \;=\; N \setminus \left\{\left\lfloor\frac{s}{2}\right\rfloor + 1, \left\lfloor\frac{s}{2}\right\rfloor + 2, \ldots, n - \left\lceil\frac{s}{2}\right\rceil\right\}.$$

Then,

$$C(\hat{S}) \;=\; \min_{S\subseteq N, |S|=s} C(S), \quad \forall s = 1, \ldots, n.$$

Proof. Consider the second term of the RHS of Eq. (13). For any fixed value of s, the first $\lfloor s/2\rfloor$ terms within the summation have a non-negative coefficient, whereas the remaining $\lceil s/2\rceil$ terms within the summation have a negative coefficient. Also, recall that the processing times are weakly increasing. To minimize $C(S)$, it is therefore optimal to match the *largest* $\lceil s/2\rceil$ processing times with the negative coefficients, and the *smallest* $\lfloor s/2\rfloor$ processing times with the non-negative coefficient. The result follows. ■

5.1.2 Selection from the RP core

We now turn to the problem of selecting an appealing solution in the RP core. We consider, in turn, equalizing costs, equalizing gains, and the Shapley solution.

Equalizing costs. We know that for any $(w_1, w_2, \dots, w_n)$ in the core,

$$W \;:=\; \sum_i w_i \;=\; \sum_{i=1}^{n}(n-i)p_i.$$

We show that the disutility vector in which every $w_i = W/n$ is in the RP core. Let

$$g(k) \;=\; C\left(\left\{1,2,\dots,\left\lfloor\frac{k}{2}\right\rfloor\right\} \cup \left\{n, n-1, n-2, \dots, n-\left\lceil\frac{k}{2}\right\rceil+1\right\}\right), \quad k = 1,2,\dots,n.$$

To prove that the vector $(W/n, W/n, \dots, W/n)$ is in the RP core, it is enough to prove

$$\frac{k}{n}W \quad\le\quad g(k), \quad k = 1,2,\dots,n. \tag{15}$$

We do this by showing the following result.

Proposition 5 *$g(k)$ is a concave function of k, that is,*

$$\frac{g(k-1)+g(k+1)}{2} \le g(k), \quad k = 1,2,\dots,n-1.$$

Proof. Consider any $s < n/2$. From equation (14), we have

$$\begin{aligned} g(2s+1) - g(2s) \;&=\; \frac{2s+1}{2}[p_{s+1} + \cdots + p_{n-s-1}] + p_1 + \cdots + p_s + sp_{n-s} - \frac{2s}{2}[p_{s+1} + \cdots + p_{n-s}] \\ &=\; \frac{1}{2}[p_{s+1} + \cdots + p_{n-s-1}] + p_1 + \cdots + p_{s-1} + p_s \end{aligned}$$

and

$$\begin{aligned} g(2s) - g(2s-1) \;&=\; \frac{2s}{2}[p_{s+1} + \cdots + p_{n-s}] + p_1 + \cdots + p_{s-1} + sp_s - \frac{2s-1}{2}[p_s + \cdots + p_{n-s}] \\ &=\; \frac{1}{2}[p_{s+1} + \cdots + p_{n-s-1}] + p_1 + \cdots + p_{s-1} + \frac{1}{2}p_{n-s} + \frac{1}{2}p_s \end{aligned}$$

Therefore we have $\frac{g(2s-1)+g(2s+1)}{2} - g(2s) = \frac{1}{4}(p_s - p_{n-s}) \implies \frac{g(2s-1)+g(2s+1)}{2} \le g(2s)$. Similarly it can be checked that we will have $\frac{g(2s-2)+g(2s)}{2} - g(2s-1) = 0$. Therefore it follows that $g(k)$ is concave in k. ■

The inequality (15) is satisfied as an equality for $k = 0$ and for $k = n$. Its LHS is linear in k, whereas its RHS is concave in k. Therefore it must be true for every intermediate value of k as well.

Equalizing gains. We next show that the equal gains solution is in the RP core, and so will be the solution that equalizes the "gains" within the CRP core. Let $S = \{a_1, a_2, \dots, a_s\}$ be any subset of agents. Let $D(S)$ be the cost savings achieved by the agents in S by cooperating. From our earlier discussion,

$$D(S) \;=\; \frac{1}{2}\sum_{i=1}^{s}(2i-1-s)p_{a_i} \;=\; \frac{1}{2}\sum_{i=1}^{\lfloor s/2\rfloor}(s-2i+1)(p_{a_{s-i+1}} - p_{a_i}).$$

The total gains achieved by the grand coalition, therefore, satisfies:

$$
\begin{aligned}
D(N) &= \frac{1}{2}\sum_{i=1}^{\lfloor n/2 \rfloor}(n-2i+1)(p_{n-i+1}-p_i) \\
&\geq \frac{1}{2}\sum_{i=1}^{\lfloor s/2 \rfloor}(n-2i+1)(p_{a_{s-i+1}}-p_{a_i}) \\
&= \frac{1}{2}\frac{n-1}{s-1}\sum_{i=1}^{\lfloor s/2 \rfloor}\frac{(n-2i+1)(s-1)}{n-1}(p_{a_{s-i+1}}-p_{a_i}).
\end{aligned}
$$

But

$$(n-2i+1)(s-1) \ = \ ns+2i-1-2is+s-n \ \geq \ ns+2i-1-2in+n-s \ = \ (s-2i+1)(n-1),$$

from which we have

$$D(N) \geq \frac{n-1}{s-1}D(S).$$

This shows that $D(S) \leq (s-1)/(n-1)D(N)$. But the savings achieved by S in the equal gains solution is $s/nD(N)$, which is greater than $(s-1)/(n-1)D(S)$. Therefore, the equal gains solution is in the RP core.

The Shapley value solution. We next show that the Shapley value solution is in the RP core. As stated earlier, the Shapley value solution is one in which all the gains from cooperation between a pair of agents is given to the higher-priority agent. We shall use this interpretation to give a short proof of its membership in the RP core.

Consider any coalition S. Recall that for any pair of agents $i < j$ in S, the Shapley solution assigns all the benefits their interaction to agent i, the higher priority agent. Consider any set of agents S. We claim that the net benefit that agents in S can generate for themselves if only they cooperate is at least as much as the net savings achieved by the members of S when all the agents cooperate. The claim follows from the simple observation that when all the agents cooperate, every pair of agents in S exchange positions with probability $1/2$; whereas if only the agents in S cooperate, pairs of agents in S exchange positions with a smaller probability. In the Shapley solution, the members of S keep all the gains from their cooperation. Therefore, the total utility of the agents in S in the Shapley value solution will be at least $-C(S)$, where $C(S)$ is the expected cost of S under RP (see Eq. (13). Note that the Shapley solution is also in the LB core: this follows because the associated cooperative game is convex, and the proposed solution is precisely the Shapley value of that game. We now turn to a characterization of the Shapley value solution along the lines of Maniquet [11].

Theorem 2 *Let φ be any allocation rule. Then, the following statements are equivalent:*

1. *For every q, $\varphi(q)$ selects* all *the allocations assigning the agents utilities corresponding to the Shapley value.*
2. *φ satisfies Efficiency, Anonymity, Equal Treatment of Equals and Independence of Succeeding Agents' Processing Time.*
3. *φ satisfies Feasibility, Pareto Indifference, Fair Share, Ranking and Proportional Responsibility.*

4. *Among all the allocation rules satisfying Efficiency, Pareto Indifference, Fair Share and Proportional Responsibility, φ maximizes the minimum utility among all the agents.*

Proof. From the definition of the Shapley value solution, **1** $\Rightarrow$ **2** and **1** $\Rightarrow$ **3** are immediate.

2 $\Rightarrow$ **1**

Consider an allocation rule φ that satisfies E, A, ETE and ISAP. Suppose there are n agents with processing times $p_1 \le p_2 \le \ldots \le p_n$. Consider a new problem q' where all agents have the same service requirement, i.e. $p'_i = p_1$. Let $z = (\sigma', t') \in \varphi(q')$. For each such z, all the agents will have the same utility (by ETE), and the net transfer will be zero (by E). Therefore, $u'_i(q) = -\frac{n-1}{2}p_1$. Suppose an agent $k \neq 1$ is the one who is served first. Permuting k and 1 does not change the processing time vector; by A, it follows that there exists a $z' \in \varphi(q')$ in which agent 1 is the one who is served first. Together with $u(1) = -\frac{n-1}{2}p_1$, this implies $t_1 = -\frac{n-1}{2}p_1$.

Now, consider a problem q'' in which agent 1 has a requirement p_1 and everybody else has a requirement of p_2. By ISAPT, we still have $u_1(q'') = t_1 = -\frac{n-1}{2}p_1$. By E and ETE, for all $i \ge 2$, we have

$$u_i(q'') \;=\; -(n-2)\frac{p_2}{2} - p_1 - \frac{t_1}{n-1} = -\frac{p_1}{2} - (n-2)\frac{p_2}{2}.$$

Again, by A, $\exists z'' \in \varphi(q'')$ such that agent 2 is served second. We can now continue this argument along the same lines. Note that, in each of the subsequent steps, we would only be increasing the processing times of agents after 2. Therefore, the utilities of agents 1 and 2 will remain fixed at $u_1(q')$ and $u_2(q'')$ respectively. By continuing the argument for n steps, the result is proved.

3 $\Rightarrow$ **1**

Consider an allocation rule φ that satisfies Feasibility, PI, FS, PR and Ranking. Suppose there are only two agents with processing times p_1 and p_2. By FS we have $u_2 \ge \frac{p_1}{2}$ and from Ranking we have $u_1 \ge u_2$. These along with the constraint $u_1 + u_2 \le -p_1$ (from feasibility) imply that we should have $u_1 = u_2 = -\frac{p_1}{2}$, proving the result for the case of two people. Assume the result also holds when there are n agents. Suppose there are $n+1$ agents with processing times $p_1 \le p_2 \le \ldots \le p_n \le p_{n+1}$. Then FS requires that we have

$$u_{n+1} \ge -\frac{p_1 + \ldots + p_n}{2}.$$

We claim that $u_{n+1} = -(p_1 + \ldots + p_n)/2$. Otherwise, we must have $u_{n+1} > -(p_1 + \ldots + p_n)/2$, and we argue next that this is impossible. First assume that agent $n+1$ is the last one to be served. By Ranking, we have $u_n > -(p_1 + \ldots + p_n)/2$. Therefore if agent $n+1$ leaves, the utility of agent n goes up by an amount less than $p_n/2$ (as utility of agent n in the resulting system will be $-(p_1 + \ldots + p_{n-1})/2$). By PR, the utility of agent i goes up by an amount less than $p_i/2$. Therefore the collective amount payed by all other agents to agent $n+1$ is less than $(p_1 + \ldots + p_n)/2$, which implies $u_{n+1} < -(p_1 + \ldots + p_n)/2$, a contradiction. Now assume $r \neq n+1$ is the last agent to be served. If agent r leaves the system, the utility of agent $n+1$ goes up by an amount less than $p_r/2$ (as the utility of agent $n+1$ in the resulting system will be $-(p_1 + \ldots + p_{r-1} + p_{r+1} + \ldots + p_n)/2$). Hence by PR, the utility of agent i goes up by an amount less than $p_r/p_{n+1} \times p_i/2 \;\le\; \frac{p_i}{2}$. Therefore the collective amount payed by all other agents to agent r is less than $(p_1 + \ldots + p_{r-1} + p_{r+1} + \ldots + p_{n+1})/2$, which implies that $u_r < -(p_1 + \ldots + p_{r-1} + p_{r+1} + \ldots + p_{n+1})/2$, a contradiction of FS. Therefore, it follows that we will have $u_{n+1} = -\frac{p_1 + \ldots + p_n}{2}$.

Having established that the utility of agent $(n+1)$ is her Shapley value utility, we proceed to do so for the other agents. Suppose agent $r \neq n+1$ is in the last position and t_r is her transfer amount. If agent r leaves the system, then the utility of agent $n+1$ goes up by $\frac{p_r}{2}$ (as utility of agent $n+1$ in the resulting system will be $-(p_1+\ldots+p_{r-1}+p_{r+1}+\ldots+p_n)/2$). By repeating the argument in the above paragraph, we have

$$t_r \le \frac{p_r}{p_{n+1}} \frac{p_1+\ldots+p_{r-1}+p_{r+1}+\ldots+p_{n+1}}{2} \le \frac{p_1+\ldots+p_{r-1}+p_{r+1}+\ldots+p_{n+1}}{2}.$$

For FS to be satisfied, we must have $p_r = p_{n+1}$, which implies $u_r = u_{n+1}$. Observe that if we switch the position and payments of agents r and $n+1$, the utilities of all the agents will still remain the same. Therefore, we can invoke PI to conclude that there is a $z' \in \varphi(q)$ in which agent $n+1$ is in the last position and u_{n+1} is as given above. Now, if agent $n+1$ leaves, then PR requires that the utility of agent $i \le n$ go up by $\frac{p_i}{2}$. Combining this with the induction basis, we have

$$u_i + \frac{p_i}{2} = -\frac{p_1+\ldots+p_{i-1}}{2} - \frac{n-i-1}{2} p_i,$$

which implies

$$u_i = -\frac{p_1+\ldots+p_{i-1}}{2} - \frac{n-i}{2} p_i.$$

The induction argument is complete.

4 $\iff$ 1

Suppose there are only 2 agents. By FS we have $u_1 \ge p_2/2$, and $u_2 \ge -p_1/2$. Therefore, the mechanism with the required properties is given by $u_1 = --p_1/2$ and $u_2 = -p_1/2$. Assume that this is true for n agents. Now consider the case when there are $n+1$ agents with $p_1 \le p_2 \le \ldots \le p_{n+1}$. By E and PI, we may assume that agent $n+1$ is served last. Let

$$t_{n+1} = \delta \frac{p_1+\ldots+p_n}{2},$$

be the money transfer of agent $n+1$. For FS to be satisfied, $\delta \ge 1$. Now, if the agent $n+1$ leaves, by PR, the utility of agent i in will be $u_i + \delta p_i/2$. By the induction hypothesis, this is the same as the Shapley value utility. Therefore,

$$u_i = -\frac{p_1+\ldots+p_{i-1}}{2} - \frac{n-i}{2} p_i - \delta \frac{p_i}{2}.$$

Thus, we can conclude that for any two agents i and j with $i < j$, we will have $u_i \ge u_j$. From the above argument, the utility of agent n will be $u_n = -(p_1+\ldots+p_{n-1})/2 - \delta p_n/2$. Notice that this is decreasing in δ. For $\delta = 1$, we will have $u_n = u_{n+1} = -(p_1+\ldots+p_n)/2$. Since $u_i \ge u_n$ for all the other agents, agents n and $n+1$ are the ones with the least utility. Now if $\delta > 1$, the utility of agent n will be less than $-(p_1+\ldots+p_n)/2$. Hence, the minimum utility among all agents is maximized when $\delta = 1$. This proves the result.

■

5.2 Constrained Random Priority (CRP)

Consider a coalition, S, of agents with processing times $p_{a_1} \le p_{a_2} \le \ldots \le p_{a_s}$, where $s = |S|$. If the agents are ordered uniformly at random, agent a_i will have an expected stand-alone cost of $(P - p_{a_i})/2$, so the expected total stand-alone cost of coalition S will be $sP/2 - \sum_{i=1}^{s} p_{a_i}/2$. But by cooperating, the agents in S will reduce their joint cost, and this reduction can be found (as in §4.2.1) by first finding the savings resulting from a pair of agents interacting. Consider agents a_i and a_j, and suppose $i < j$. From the argument in §4.2.1, agents i and j will exchange positions with probability $K_s = 1/(n-s+2)$, in which case they jointly save $(p_{a_j} - p_{a_i})$. Therefore, the cost of the coalition S is given by

$$
\begin{aligned}
C(S) &= \frac{1}{2}\big[sP - \sum_{i=1}^{s} p_{a_i}\big] - \frac{1}{n-s+2} \sum_{i,j\in S, i<j} (p_{a_j} - p_{a_i}) \\
&= \frac{1}{2}\big[sP - \sum_{i=1}^{s} p_{a_i}\big] - \frac{1}{n-s+2} \sum_{i=1}^{s} (2i-1-s) p_{a_i} \qquad (16)
\end{aligned}
$$

It is straightforward to show that the $C(\cdot)$ just defined is submodular. Observe also that this expression bears a close resemblance to the $C(\cdot)$ of the RP core, the only difference being the probability of a pair of agents exchanging positions: this probability is $1/2$ in the RP core, but is $1/(n-s+2) \le 1/2$ in the CRP core. Therefore, the CRP core contains the RP core. (That the CRP core contains the RP core can be seen directly also: the savings from cooperation in the RP core are greater by definition, because the agents are less constrained.) In particular, the equal gains solution, the Shapley value solution, and the equal cost solution are all in the CRP core.

6 Arbitrary processing times and waiting costs

We turn to the general case in which agent i has processing time c_i and processing time requirement p_i. We shall assume that agents are labeled so that $c_1/p_1 \ge c_2/p_2 \ge \ldots \ge c_n/p_n$. Our earlier discussion suggested that RP may not be reasonable in this case because allowing arbitrary coalition members to exchange their positions may increase the waiting time of an agent who is not part of the cooperating coalition. This suggests that the expected cost of a coalition S under RP will be too optimistic, and that distributing the costs among the agents to meet this optimistic bound for all coalitions may be impossible. Indeed this is the case: our first observation in this section is to show that the RP core may be empty.

Example 4. Suppose $n = 3$, and suppose $c_1 = 10$, $c_2 = 20$, $c_3 = 3$, $p_1 = 10$, $p_2 = 25$, and $p_3 = 5$. It is straightforward to verify that the RP core for this model is characterized by the constraints:

$$w_1 \le c_1 \frac{p_2 + p_3}{2}, \quad w_2 \le c_2 \frac{p_1 + p_3}{2}, \quad w_3 \le c_3 \frac{p_1 + p_2}{2},$$

$$w_1 + w_2 \le c_2 p_1 + p_3 \frac{c_1 + c_2}{2}, \quad w_1 + w_3 \le c_3 p_1 + p_2 \frac{c_1 + c_3}{2}, \quad w_2 + w_3 \le c_3 p_2 + p_1 \frac{c_2 + c_3}{2},$$

and

$$w_1 + w_2 + w_3 = c_2 p_1 + c_3(p_1 + p_2).$$

For the particular choice of numbers, we get (among other inequalities)

$$w_3 \le 52.5, \ \ w_1 + w_2 \le 275, \ \ w_1 + w_2 + w_3 = 305,$$

and this subsystem of inequalities is inconsistent.

This example formally justifies the need for CRP, which we discuss next.

6.1 Constrained Random Priority (CRP)

Recall that under CRP, any subset of a cooperating subset of agents can rearrange their positions as long as all such members are contiguous. As before, we first consider evaluating $C(S)$, the cost of a coalition S of agents under CRP.

Proposition 6 *Let $C(S)$ denote the expected waiting cost of a coalition S of agents under CRP. Then,*

$$C(S) \quad = \quad \sum_{i \in S} c_i \frac{(P - p_i)}{2} - \frac{1}{n - s + 2} \sum_{i,j \in S, i<j} (c_i p_j - c_j p_i).$$

Furthermore, $C(S)$ is a submodular set function.

Proof Sketch. The expression for $C(S)$ follows easily from the discussion of the CRP core for the identical processing times case in §4.2.1, with the following two modifications: the expected waiting time of agent i in the absence of any cooperation is $(P - p_i)/2$; and the savings achieved by a pair $i < j$ of cooperating agents is $c_i p_j - c_j p_i$. The submodularity of $C(\cdot)$ follows from an argument similar to the proof of Lemma 1. ■

6.2 Selection from the CRP core

We now consider to the problem of finding "good" solutions in the CRP core. Since the $C(\cdot)$ function is submodular, we know that the core in non-empty. As before, we explore the goals of equalizing costs, equalizing gains, etc.

Equalizing costs. Since the CRP core is submodular, we can use the general algorithm proposed by Dutta and Ray [8] (discussed briefly in section 4.2.2) to compute the egalitarian solution. This solution Lorenz dominates every other solution in the CRP core. We provide an efficient algorithm to find this solution. Let $w = (w_1, w_2, \ldots, w_n)$ denote the vector of disutilities. The CRP core conditions are:

$$\sum_{i \in S} w_i \leq C(S) = \frac{p_1 + \cdots + p_n}{2} \sum_{i \in S} c_i - \frac{\sum_{i \in S} c_i p_i}{2} - \frac{1}{n + 2 - s} \sum_{i,j \in S: i<j} (c_i p_j - c_j p_i)$$

Let $z_i = \frac{p_1 + \cdots + p_n}{2} c_i - \frac{c_i p_i}{2} - w_i \quad \forall i \in N$. Observe that $(z_1, \ldots, z_n)$ is just the vector of benefits of each agent. We may assume $z_i \geq 0$ for all agents i, otherwise the core constraint for some individual agent will be violated. The CRP core constraints can be rewritten as

$$\begin{aligned} & -\sum_{i \in S} z_i + \frac{p_1 + \cdots + p_n}{2} \sum_{i \in S} c_i - \frac{\sum_{i \in S} c_i p_i}{2} \leq C(S) \\ \Longleftrightarrow \quad & \sum_{i \in S} z_i \geq \frac{1}{n + 2 - s} \sum_{i,j \in S: i<j} g_{ij} = \frac{G(S)}{n - |S| + 2} =: F(S) \end{aligned}$$

where $g_{ij} = |c_i p_j - c_j p_i|$ for all $i, j \in N$.

Consider a network with a source node $\bar{s}$, a sink node $\bar{t}$, a node r_i for each agent $i \in N$, and a node q_{ij} for each pair of agents $i < j$ in N. The edges of the network are as follows: there are n edges of the form $(\bar{s}, r_i)$, with capacity $2z_i$; $n(n-1)/2$ edges of the form $(q_{ij}, \bar{t})$, with capacity g_{ij}; for each node q_{ij}, add the two infinite capacity edges (r_i, q_{ij}) and (r_j, q_{ij}); finally, there is an edge (r_i, r_j) for every $i, j \in N$, with capacity z_j.

Consider all the cuts in which the nodes $r_i,\ \forall i \in S$ are on the source side of the cut and the nodes $r_i,\ \forall i \in N\backslash S$ are on the sink side. Since none of the infinite capacity edges can be in the min-cut, for any r_i in S every q_{ik} for all $k \in N$ must also be in the source side of any such cut. Therefore, the minimum-capacity among all such cuts is clearly

$$\sum_{j\in N: r_j \notin S} 2z_j + \sum_{j\in N: r_j\notin S} sz_j + G(N) - G(N\setminus S),$$

where the second term comes from the observation that each node r_i in S is connected to each node r_j not in S by an edge of capacity z_j. We now show how the minimum-cost solution can be calculated by solving a parametric maximum flow problem on this network.

Suppose that we start with $z_i = \frac{p_1+\dots+p_n}{2}c_i - \frac{c_ip_i}{2} - \delta\ \forall\ i \in N$; then we will have $w_i = \delta\ \forall\ i \in N$. For δ small enough, the min-cut in which all the nodes $r_j : j \in N$ are on the source side will be the unique minimum-cut. Suppose that we start with such a value of δ and start increasing the value of δ. Suppose that δ_1 is the smallest break-point of the associated parametric maximum flow problem and let S_1 be the minimum cardinality set such that all the nodes in $r_j : j \in S_1$ are on the source side in a minimum-cut. Then, for any set S, we will have

$$\begin{aligned}
&\sum_{j\in N: r_j\notin S} (s+2)z_j + G(N) - G(N\setminus S) \geq G(N) \implies \sum_{j\in N\setminus S} z_j \geq F(N\setminus S)\\
\implies\ & \frac{p_1+\dots+p_n}{2}\sum_{i\in N\setminus S} c_i - \frac{\sum_{i\in N\setminus S} c_ip_i}{2} - (n-s)\delta_1 \geq F(N\setminus S)\\
\implies\ & \delta_1 \leq \frac{\frac{p_1+\dots+p_n}{2}\sum_{i\in N\setminus S} c_i - \frac{\sum_{i\in N\setminus S} c_ip_i}{2} - F(N\setminus S)}{n-s} = \frac{C(N\setminus S)}{n-s}
\end{aligned}$$

As the above inequality is tight for the set $S = S_1$ we have $\delta_1 = \min_T \frac{C(T)}{t}$, thus the first iteration of the equal costs solution is done. Now fix the values of the $z_i's$ for all $i \in N \setminus S_1$, the values of the $z_i's$ for $i \in S_1$ will still be dependent on δ. Also, add infinite capacity arcs from the nodes $r_j : j \in N \setminus S_1$ to the sink, thus these nodes will always be on the sink side of any min-cut. Let δ_2 be the smallest breakpoint (which will be greater than δ_1) of this modified network and let let S_2 be the minimum cardinality set such that all the nodes in $r_j : j \in S_2$ (where $S_2 \subset S_1$) are on the source side of a min-cut. For any set $S \subset S_1$, we will have

$$\begin{aligned}
\sum_{j\in N\setminus S} z_j \geq F(N\setminus S) \quad &\implies \quad \frac{p_1+\dots+p_n}{2}\sum_{i\in N\setminus S} c_i - \frac{\sum_{i\in N\setminus S} c_ip_i}{2} - \sum_{i\in N\setminus S} w_i \geq F(N\setminus S)\\
&\implies \quad \sum_{i\in S_1\setminus S} w_i \leq \frac{p_1+\dots+p_n}{2}\sum_{i\in N\setminus S} c_i - \frac{\sum_{i\in N\setminus S} c_ip_i}{2} - F(N\setminus S) - \sum_{i\in N\setminus S_1} w_i\\
&\implies \quad \delta_2 \leq \frac{C(N\setminus S) - C(N\setminus S_1)}{s_1 - s} = \frac{C(S_1\setminus S\ \cup\ N\setminus S_1) - C(N\setminus S_1)}{s_1 - s}
\end{aligned}$$

As the above inequality is tight for the set $S = S_2$ we have $\delta_2 = \min_{T\subset S_1} \frac{C(T\cup N\setminus S_1) - C(N\setminus S_1)}{t}$, thus the second iteration of the equal costs solution is done. Fixing the values of the $z_i's$ for all $i \in S_1 \setminus S_2$, and continuing as before, the equal costs solution can be found efficiently.

Equalizing gains. We show that the equal gains solution is in the CRP core. The net benefits from cooperation, denoted B, is clearly:

$$B := \sum_{i,j\in S, i<j} \frac{(c_ip_j - c_jp_i)}{2}.$$

In the equal gains solution, the disutility of agent i is

$$w_i = c_i \frac{(P - p_i)}{2} - \frac{B}{n}.$$

To show that this is in the CRP core, we need to show that for any coalition S of agents, $\sum_{i \in S} w_i \leq C(S)$, which is equivalent to proving

$$\frac{s}{n} B \;\geq\; \frac{1}{n-s+2} \sum_{i,j \in S, i<j} (c_i p_j - c_j p_i).$$

Since $B \geq 0$, it is evident that the core constraints will be satisfied for all single person coalitions. Suppose $|S| = s \geq 2$. Then, we have:

$$\begin{aligned}
\frac{s}{n} B &= \frac{s}{n} \sum_{i,j \in N, i<j} (c_i p_j - c_j p_i) \\
&= \frac{s}{n} \frac{n-s+2}{2} \frac{1}{n-s+2} \sum_{i,j \in N, i<j} (c_i p_j - c_j p_i) \\
&\geq \frac{s}{n} \frac{n-s+2}{2} \frac{1}{n-s+2} \sum_{i,j \in S, i<j} (c_i p_j - c_j p_i) \\
&\geq \frac{1}{n-s+2} \sum_{i,j \in S, i<j} (c_i p_j - c_j p_i),
\end{aligned}$$

where the last inequality uses the assumption that $s \geq 2$. Specifically, if $s \geq 2$, then

$$s(n-s+2) - 2n \;=\; (n-s)(s-2) \geq 0,$$

which implies $s(n-s+2) \geq 2n$. This completes the proof.

The Shapley value solution. We end this section by studying the Shapley value solution. First we show that the Shapley value solution is in the CRP core. Recall that for any pair of agents $i < j$, the Shapley solution assigns all the benefits their interaction to agent i, the higher priority agent. Consider any set of agents S. We claim that the net benefit that agents in S can generate for themselves if only they cooperate is at least as much as the net savings achieved by the members of S when all the agents cooperate. The claim follows from the simple observation that when all the agents cooperate, every pair of agents in S exchange positions with probability $1/2$; whereas if only the agents in S cooperate, pairs of agents in S exchange positions with a smaller probability. In the Shapley solution, the members of S keep all the gains from their cooperation, so they will have no incentive to deviate. As before, it is interesting to note that the Shapley solution is in the CRP core as well as the LB core; the latter follows because it is the Shapley value solution of a convex cooperative game. We provide a characterization that extends and complements the characterization of Maniquet [11]. Independent of our work, Misra and Rangarajan [12] provide a different characterization of the same solution.

Theorem 3 *Let φ be any allocation rule. Then, the following statements are equivalent:*

1. *φ selects all the allocations assigning to the agents utilities corresponding to the Shapley value.*

2. *φ satisfies Efficiency, Equal Treatment of Equals, Anonymity, Independence of Preceding agents' impatience , Independence of succeeding agents' processing times and Proportional Responsibility .*

3. *φ satisfies Efficiency, Pareto Indifference, Fair Share, Ranking, Independence of Preceding agents Impatience and Proportional Responsibility .*

4. *Among all the allocation rules satisfying Efficiency, Pareto Indifference, Fair Share, Independence of Preceding agents Impatience and Proportional Responsibility, φ maximizes the minimum among the utilities of all agents.*

Proof. From the definition of the Shapley value solution, **1** $\Rightarrow$ **2** and **1** $\Rightarrow$ **3** are immediate.

2 $\Rightarrow$ **1**

Suppose there are n agents with the agents labeled so that $c_1/p_1 \geq c_2/p_2 \geq \ldots \geq c_n/p_n$. We claim that $u_n = c_n(p_1 + p_2 + \ldots + p_{n-1})/2$. To see why, consider a problem $q' = (N; P', C')$ with $c'_1 = c'_2 = \ldots c'_n = c_n$, $p'_1 = p_1, p'_2 = p_2, \ldots, p'_{n-1} = p_{n-1}, p'_n > max\{p_1, \ldots, p_n\}$. Then, by corresponding result in Theorem 2, we can conclude that $u'_n = -c_n(p_1 + p_2 + \ldots + p_{n-1})/2$. Now, consider the problem q'' in which the processing times are the same as q', but the waiting costs of the agents are $c''_1 = c_1, \ldots c''_{n-1} = c_{n-1}, c''_n = c_n$. By IPAI, the utility of agent n remains the same as her utility in q', so $u''_n = u'_n$. Observe that the original problem q differs from q'' in only one respect: the processing time of agent 1 is p_n in q, but p'_n in q''. By ISAP, the utility of any agent $i \neq n$ is the same in the two problems q'' and q. As the sum of all the utilities is the same, this also agent n's utility in q is the same as her utility in q''.

If $n = 2$, the preceding discussion implies $u_2 = -c_2p_1/2$; since efficiency implies $u_1 + u_2 = -c_2p_1$, we have $u_1 = -c_2p_1/2$ in any efficient solution. So the result follows for $n = 2$. Suppose the result is true for $n = k$ agents. We show that the result remains valid when there are $k + 1$ agents. Given a problem q with $n = k + 1$ agents, consider another problem q' where the only change from q is that $p'_n > max\{p_1, \ldots, p_n\}$. In the problem q', agent n will be served last. Moreover, we know that $u'_n = -c_n(p_1 + p_2 + \ldots + p_{n-1})/2$, which implies $t'_n = c_n(p_1 + p_2 + \ldots + p_{n-1})/2$; by ISAP, the utilities of all the other agents remains the same as in the problem q. Now, suppose agent n leaves. The only change in the utility of agent i is the additional money she gets, because t'_n will now be redistributed among the agents $1, 2, \ldots, n - 1$. By PR,

$$u(i) + \frac{p_i c_n}{2} = -\frac{(p_1 + \ldots + p_{i-1})c_i}{2} - \frac{p_i(c_{i+1} + \ldots + c_{n-1})}{2}.$$

Simplifying,

$$u(i) = -\frac{(p_1 + \ldots + p_{i-1})c_i}{2} - \frac{p_i(c_{i+1} + \ldots + c_n)}{2},$$

which is identical to i's utility in the Shapley value solution. This establishes the result for $n = k + 1$ agents.

3 $\Rightarrow$ **1**

Suppose there are $n+1$ agents with $c_1/p_1 \geq c_2/p_2 \geq \ldots \geq c_n/p_n = c_{n+1}/p_{n+1}$. , where $p_{n+1} = max\{p_1, \ldots, p_n\}$. By E and PI, we can assume that agent $n+1$ is served last. Next, we claim that $u_{n+1} = -c_{n+1}(p_1+p_2+\ldots+p_n)/2$. This can proved by an argument similar to the one just employed in proving the previous case **2** $\Rightarrow$ **1**. By E and PI, there will be a solution in $\varphi(q)$ in which agent n (not $n+1$)) is served last; it is clear that there will be a set of transfers in which agent $(n+1)$'s utility is still u_{n+1} in this solution. Now, suppose agent n leaves. The utility of agent $n+1$ in the resulting problem q' will be $u'_{n+1} = -c_{n+1}(p_1+p_2+\ldots+p_{n-1})/2$, i.e., the utility of agent $n+1$ goes up by $c_{n+1}p_n/2$, which is also the same as $c_np_{n+1}/2$. By PR, the the utility of agent i will go up by $c_np_i/2$, which implies the utility of agent n in the original problem should have been $u_n = -c_n(p_1+p_2+\ldots+p_{n-1}+p_{n+1})/2$. Again, by E and PI, there will be a solution in which agent $n + 1$ is served last and the utilities of all the other agents remain the same. If agent $n+1$ leaves, by PR, we will have $u''_n = -c_n(p_1 + p_2 + \ldots + p_{n-1})/2$. So, in a system with only n agents, the utility of agent n is $-c_n(p_1 + p_2 + \ldots + p_{n-1})/2$.

If there are only 2 agents, this analysis shows $u_2 = -c_2 p_1/2$, which, along with E, implies $u_1 = -c_2 p_1/2$. Therefore, the result is true for 2 agents. Assuming the result for $n-1$ agents, consider the case when there are n agents. By E and PI, there will be a solution in which agent n is served last and has utility $-c_n(p_1 + p_2 + \ldots + p_{n-1})/2$. If agent n leaves, the only change in the utility of an agent i is due to the redistribution of t_n among all the agents. By PR,

$$u_i + \frac{p_i c_n}{2} = -\frac{(p_1 + \ldots + p_{i-1})c_i}{2} - \frac{p_i(c_{i+1} + \ldots + c_{n-1})}{2}.$$

Rewriting this expression, we have

$$u_i \;=\; -\frac{(p_1 + \ldots + p_{i-1})c_i}{2} - \frac{p_i(c_{i+1} + \ldots + c_n)}{2},$$

which is exactly agent i's Shapley value utility.

4 $\Longleftrightarrow$ **1**

The proof is the same as the above part. Instead of invoking the **3** $\Rightarrow$ **1** of Theorem 2 , we will invoke **4** $\Rightarrow$ **1**.

■

7 Equitable cost sharing: Another view

In this section, we provide another perspective on equitable cost sharing. We have considered both RP and CRP cores, each of which provides an upper bound on the cost-shares of any coalition S. Any core solution, thus, is *fair*, in the sense that the cost borne by any coalition of agents is no more than the least-cost that coalition can achieve for itself. Another notion of equitable cost sharing is as follows: consider any coalition S of agents, and suppose no other agents are present; suppose $L(S)$ is the least possible total cost of S (obtained when the agents in S are ordered efficiently). Then, if other agents are present, the total disutility of the agents in S should be at least $L(S)$; in other words, the sum of the disutilities of the agents in S should be at least their best-achievable cost, if no other agents are present. (No coalition of agents is "subsidized.") We refer to this latter concept as the "LB" core. From the earlier sections, we have the following results:

- The Shapley value solution is a member of both the CRP core and the LB core. If the c_i are all identical, the Shapley value solution is in the RP core as well.
- The RP core may be empty in the general case, but is always non-empty when the p_i are identical. However, in the latter case, there may not be a solution that is in both the RP core and the LB core.
- The Equal Gains solution may not be in the LB core, even in the two special cases. Similarly, the solution that equalizes costs in the CRP core may not be in the LB core, even in the two special cases.

This section is motivated by the search for other disutility vectors that are members of both the LB core and the CRP core. (In view of the results mentioned earlier, we do not consider the RP core.) While the Shapley value solution is in both cores, it may not achieve auxiliary objectives such as equalizing costs or equalizing gains (among all the core solutions). Thus, our goal is to find an "optimal" $(w_1, w_2, \ldots, w_n)$ among all such vectors in the intersection of the CRP core and the LB core. Assuming the objective is a linear (or "linearizable") function of the variables $w_1, w_2, \ldots, w_n$, this is simply a linear programming problem; the only difficulty is that it has exponentially many constraints, so conventional methods will be inefficient. Fortunately, from standard results in the theory of linear programming [2, section 8.5], we know that such large linear programs can be solved in polynomial time, provided one can solve the associated *separation problem* efficiently. In this case, the *separation problem* is the following: given a candidate $(w_1, w_2, \ldots, w_n)$ vector, either *prove* that it satisfies all the constraints, or exhibit a violating constraint. The separation problem is easily seen to be equivalent to minimizing a submodular function, and so can be solved in polynomial-time [9]. In this section, we discuss more efficient algorithms for the separation problem for the LB core and the CRP core. Our algorithms are based on solving the maximum-flow problem in an appropriately defined network. While the construction for the LB core appears to be standard, the construction for the CRP core is new. As always, we assume the agents are labeled so that $c_1/p_1 \geq c_2/p_2 \geq \ldots \geq c_n/p_n$.

7.1 The LB core

Given a vector $(w_1, w_2, \ldots, w_n)$, we wish to either conclude that it is in the LB core, or we wish to exhibit a violated constraint. For the general case, we can solve this as a simple maximum-flow problem [1]. For the two special cases, however, much faster algorithms can be given. We start with the special cases, and then describe the general case.

7.1.1 Identical Processing Times

For any S, let

$$L(S) := \sum_{i,j \in S, i<j} c_j.$$

Let $w = (w_1, w_2, \ldots, w_n)$ be given, and suppose $\sum_{i=1}^n w_i = L(N)$. The core constraints are:

$$\sum_{i \in S} w_i \geq L(S), \quad \forall S \subseteq N$$

The separation problem can be reformulated as the following optimization problem:

$$\max_{S \subseteq N} \left[L(S) - \sum_{i \in S} w_i\right] \tag{17}$$

If the optimal value of this optimization problem is non-positive, then w is in the LB core; otherwise, a set that the maximum value will violate the core condition.

Let R be the maximal set that achieves the optimal value for the optimization problem (19). Suppose that $k \notin R$. Then we have

$$\begin{aligned} L(R \cup \{k\}) - \sum_{i \in R \cup \{k\}} w_i &= \sum_{i,j \in R; i<j} \min\{c_i, c_j\} + \sum_{i \in R; i>k} c_i + \sum_{i \in R; i<k} c_k - \sum_{i \in R} w_i - w_k \\ &= L(R) - \sum_{i \in R} w_i + \sum_{i \in R; i>k} c_i + \sum_{i \in R; i<k} c_k - w_k \\ &< L(R) - \sum_{i \in R} w_i \\ \implies \sum_{i \in R; i>k} c_i + \sum_{i \in R; i<k} c_k &< w_k \end{aligned}$$

Now suppose that $k \in R$. Then, we have

$$\begin{aligned}
L(R/\{k\}) - \sum_{i \in R/\{k\}} w_i &= \sum_{i,j \in R/k; i<j} \min\{c_i, c_j\} - \sum_{i \in R/\{k\}} w_i \\
&= \sum_{i,j \in R; i<j} \min\{c_i, c_j\} - \sum_{i \in R; i>k} c_i - \sum_{i \in R; i<k} c_k - \sum_{i \in R} w_i + w_k \\
&= L(R) - \sum_{i \in R} w_i - \sum_{i \in R; i>k} c_i - \sum_{i \in R; i<k} c_k + w_k \\
&\leq L(R) - \sum_{i \in R} w_i \\
\implies \sum_{i \in R; i>k} c_i + \sum_{i \in R; i<k} c_k &\geq w_k
\end{aligned}$$

Thus we can conclude that

$$k \in R \iff \sum_{i \in R; i>k} c_i + \sum_{i \in R; i<k} c_k \geq w_k \tag{18}$$

Now we will describe how the above equations can be used to check the presence in the lower core in $O(n^2)$ time. Suppose the size of the set R is m. Then we can directly check whether the agent n is present in R using Eq. (18), because there is no agent following n. Once n's membership in R is known, we can check the status of agent $(n-1)$, again using Eq. (18): this is because the only agent following $n-1$ is n, and we already know whether or not she is in R. By working our way backwards, we will find a set of agents in R. If the number of agents found is not equal to m, we know that R cannot have cardinality m, so we discard the set just found; if not, we call this set R^m. By doing this for $m = 1, 2, \ldots, n$, we will have a collection of at most n sets, one of which optimizes (17). For each fixed m, the algorithm runs in $O(n)$ time, and there are n possible values of m, so the overall running time is $O(n^2)$.

7.1.2 Identical Waiting Costs

For any S, let

$$L(S) := \sum_{i,j \in S, i<j} p_i.$$

Let $w = (w_1, w_2, \ldots, w_n)$ be given, and suppose $\sum_{i=1}^{n} w_i = L(N)$. The separation problem can be reformulated as

$$\max_{S \subseteq N} \left[L(S) - \sum_{i \in S} w_i \right]$$

If the optimal value of this optimization problem is non-positive, then w is in the LB core; otherwise, a set that the maximum value will violate the core condition.

The argument in this case is very similar to the one used for the case of identical processing times; the only change is the condition used to check whether $k \in R$, which now reads:

$$k \in R \iff \sum_{i \in R; i>k} p_k + \sum_{i \in R; i<k} p_i \quad \geq \quad w_k$$

To check this condition, we proceed forwards (instead of backwards), starting with agent 1, then agent 2, etc. Therefore, as before, we can solve the separation problem in $O(n^2)$ time.

7.1.3 Arbitrary processing times and waiting costs

For any S, let

$$L(S) := \sum_{i,j \in S, i<j} c_j p_i.$$

Let $w = (w_1, w_2, \ldots, w_n)$ be given, and suppose $\sum_{i=1}^{n} w_i = L(N)$. The core conditions are

$$\sum_{i \in S} w_i \geq L(S).$$

Let $l(i,j) = c_j p_i$. A fairly standard construction can be used to show that the separation problem can be solved as a maximum flow problem on an appropriately defined network (we refer the reader to the text by Ahuja et al. [1]. We provide the construction and the proof here for the sake of completeness.

Consider a network with a source node $\bar{s}$, a sink node $\bar{t}$, a node r_i for each agent $i \in N$, and a node q_{ij} for each pair of agents $i < j$ in N. The edges of the network are as follows: there are n edges of the form $(\bar{s}, r_i)$, with capacity w_i; $n(n-1)/2$ edges of the form $(q_{ij}, \bar{t})$, with capacity l_{ij}; and for each node q_{ij}, add the two infinite capacity edges (r_i, q_{ij}) and (r_j, q_{ij}). We claim that the given vector w is in the LB core if and only if the maximum flow in this network is $L(N)$. Suppose that the maximum flow in this network is $L(N)$. We argue that w is in the core. Observe that the sum of capacities of all the arcs into the sink $\bar{t}$ is $L(N)$; therefore the flow in each of these arcs has to be at its capacity. For any subset of agents S, consider the set of nodes $\bar{S} = \{(i,j) : i,j \in S, i < j\}$. Note that the sum of the capacities of all the arcs from these nodes to the sink $\bar{S}$ is $L(S)$; hence $L(S)$ units of flow passes through these nodes. But the only nodes that have arcs to the nodes in $\bar{S}$ are the nodes r_i for $i \in S$, which proves that $\sum_{i \in S} w_i \;\geq\; L(S)$, for all $S \subset N$. To see the converse, suppose the maximum flow in this network is less than $L(N)$. By the max-flow min-cut theorem, the minimum-cut capacity is less than $L(N)$. Consider any minimum cut, and let S index all the i such that r_i is in the source side of the cut. Since none of the infinite capacity edges can be in the cut, for any r_i in S every q_{ik} for all $k \in N$ must also be in the source side of the cut. Therefore, the minimum-capacity of such a cut is clearly $\sum_{i \notin S} w_i + L(N) - L(N \setminus S)$. Since the minimum-cut capacity is at most $L(N)$, we have:

$$\sum_{i \notin S} w_i + L(N) - L(N \setminus S) < L(N),$$

so the core constraint is violated for the set $N \setminus S$.

7.2 The CRP core

We present an algorithm for the general case in which agents have arbitrary processing times and waiting costs. Let $w = (w_1, w_2, \ldots, w_n)$ be the given vector of disutilities. Recall, from the equalizing costs solution of section 6.2, that if we have $z_i = \frac{p_1 + \cdots + p_n}{2} c_i - \frac{c_i p_i}{2} - w_i \quad \forall i \in N$, then the CRP core constraints can be rewritten as

$$\sum_{i \in S} z_i \geq \frac{1}{n+2-s} \sum_{i,j \in S: i<j} g_{ij} = \frac{G(S)}{n+2-s} =: F(S)$$

where $g_{ij} = |c_i p_j - c_j p_i|$ for all $i, j \in N$.

Consider the same network as in section 6.2. This network has a source node $\bar{s}$, a sink node $\bar{t}$, a node r_i for each agent $i \in N$, and a node q_{ij} for each pair of agents $i < j$ in N. The edges of the network are as follows:

there are n edges of the form $(\bar{s}, r_i)$, with capacity $2z_i$; $n(n-1)/2$ edges of the form $(q_{ij}, \bar{t})$, with capacity g_{ij}; for each node q_{ij}, add the two infinite capacity edges (r_i, q_{ij}) and (r_j, q_{ij}); finally, there is an edge (r_i, r_j) for every $i, j \in N$, with capacity z_j. From section 6.2, we know that the minimum-capacity among all cuts in which the nodes $r_i, \ \ \forall i \in S$ are on the source side of the cut and the nodes $r_i, \ \ \forall i \in N \setminus S$ are on the sink side is

$$\sum_{j \in N : r_j \notin S} 2z_j + \sum_{j \in N : r_j \notin S} s z_j + G(N) - G(N \setminus S),$$

We claim that the given vector z is in the CRP core if and only if the maximum flow in this network is $G(N)$. We prove this below.

If the minimum-cut capacity is $G(N)$, we have

$$\sum_{j \in N : r_j \notin S} (s+2) z_j + G(N) - G(N \setminus S) \geq G(N),$$

which, when rewritten, becomes

$$\sum_{j \in N \setminus S} (n - (n-s) + 2) z_j \geq G(N \setminus S),$$

Thus the core constraint is satisfied for any set $N \setminus S$; since S is arbitrary, this implies all the core constraints are satisfied. If the minimum-cut capacity is less than $G(N)$, and S_1 is the set on the source side of the min-cut, we have:

$$\sum_{j \in N : r_j \notin S_1} (s_1 + 2) z_j + G(N) - G(N \setminus S_1) < G(N),$$

which, when rewritten, becomes

$$\sum_{j \in N \setminus S_1} (n - (n - s_1) + 2) z_j < G(N \setminus S_1),$$

a violation of the core constraint for the set $N \setminus S_1$.

7.3 Implementing the solution concepts

In this section we discuss how auxiliary criteria such as equalizing costs, equalizing gains, etc. can be incorporated.

7.3.1 Equalizing costs

Suppose we want to equalize costs subject to the restriction that the disutility vector be a member of both the LB core and the CRP core. We can achieve this as follows: first we maximize the minimum cost among all agents subject to the core conditions, i.e., we solve the following problem

$$\begin{aligned} & \max \quad x \\ s.t. \qquad & L(S) \leq \sum_{i \in S} w_i \leq C(S) \\ & x \leq w_i \quad \forall \quad i = 1, \ldots, n \end{aligned}$$

This problem can be solved in polynomial-time as the associated separation problem can be solved in polynomial time. Let an optimal solution be x^*. Let S_1 be the set of agents whose utilities are x^* in all the optimal solutions;

this can again be done efficiently. Fixing the costs of these agents to x^*, we then maximize the minimum cost of the remaining agents by solving the above optimization problem with the additional constraints $w_i = x^* \;\; \forall i \in S_1$ and removing the constraints $x \leq w_i \forall i \in S_1$. Since we had a feasible solution at the end of the first iteration, this second optimization problem is feasible, and we will find an optimal solution. Now, we find the set S_2 of agents whose costs have to be held fixed. Continuing this procedure for at most n iterations, we will find a solution that is a member of both the LB core and the CRP core that equalizes costs. Note that this solution also Lorenz dominates all other solutions in the common core.

7.3.2 Equalizing gains

Suppose we want to equalize the gains of the agents subject to the condition that the disutility vector be in both the LB core and the CRP core. Let $(w_1, w_2, \ldots, w_n)$ be a a disutility vector. Then the gain of agent i is given by $b_i - w_i$; where b_i is the disutility of agent i in the solution in which the processing order is chosen uniformly at random. First we maximize the minimum benefit among all agents subject to the core conditions, i.e., we solve the following problem

$$\begin{aligned} & \max \quad x \\ s.t. \quad & L(S) \leq \sum_{i \in S} w_i \leq C(S) \\ & x \leq b_i - w_i \quad \forall \quad i = 1, \ldots, n \end{aligned}$$

From here we proceed in exactly the same manner as described in the equal costs solution.

Remark. While finding the equal costs solution or the equal gains solution, we can impose the *Ranking* criterion described in §2.2. As ranking simply requires that the agents' disutilities be ordered in a particular way, this constraint can be easily added to the optimization. Moreover, the Shapley solution satisfies this property, so we know that the resulting optimization problem will have at least one solution. Another criterion that can be incorporated is the one in which the cost of any agent cannot exceed a fixed fraction K of her default cost. One needs to be careful in choosing these fractions, though. For instance, the Shapley solution achieves the value $K = 1/2$ exactly for all the agents; it is easy to check that $K < 1/2$ is impossible, and that the Shapley solution is the only one achieving $K = 1/2$.

Acknowledgments

A version of this paper was presented at the Spring 2005 Midwest Economic Theory Conference at Vanderbilt University, Tennessee. We gratefully acknowledge the comments of the participants of that conference. We thank Herve Moulin for many enlightening conversations about the content of this paper. In addition, we thank Rakesh Vohra and Chung-Piaw Teo for their comments on an earlier draft of this paper.

References

[1] Ahuja, Magnanti and Orlin ,“ Network Flows: Theory, Algorithms, and Applications”, Prentice Hall, 1993.

[2] D. Bertsimas and J. Tsitsiklis, "Introduction to Linear Optimization," Athena Scientific, 1997.

[3] Y.Chun (2004), "A note on Maniquet's Characterization of the Shapley Value in Queueing Problems", Working paper, Rochester University.

[4] Y.Chun (2004), "No-envy in Queuing Problems", Working paper, Rochester University.

[5] I.Curiel, H.Hammers and F.Klijn (2002), "Sequencing Games : A Survey", In P.Borm and H.Peters,editors,*Chapter in Game Theory.* Theory and Decision Library, Kulwer Academic Publishers, 2002.

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

[7] R.Dolan (1978),"Incentive mechanisms for priority queuing problems",*Bell Journal of Economics*,421-436.

[8] B. Dutta and D. Ray (1989), "A concept of Egalitarianism under participation constraints",*Econometrica*,57(3), 615-635.

[9] S. Fujishige (1988),*Submodular functions and optimization*, North-Holland, Editor : P.L.Hammer .

[10] Klijn F., S. Tijs, and H. Hamers (2000): "A Simple Proof of the Total Balancedness of Permutation Games", *Economics Letters*, 69, 323-326

[11] F.Maniquet (2003),"A characterization of the Shapley Value in Queuing Problems",*Journal of Economic Theory*, 109, 90-103.

[12] D.Mishra, B.Rangarajan (2005),"Cost Sharing in a Job Scheduling Problem",*working paper*, Universite Catholique de Louvain.

[13] M.Mitra (2001),"Mechanism Design in Queuing Problems", *Economic Theory*, 17(2), 277-305.

[14] H.Moulin , *Handbook of Social choice and welfare*, Chapter: Axiomatic Cost and Surplus sharing, North Holland, 2002.

[15] H.Moulin (2004),"On scheduling fees to prevent merging, splitting and transferring of jobs ",*Working paper*, Rice University.

[16] H.Moulin (2004),"Split-proof Probabilistic Scheduling ",*Working paper*, Rice University.

[17] W.E.Smith (1956), "Various Optimizers for single stage Production", *Naval Research Logistics Quarterly*, 3, 59-66, 1956.

[18] J.Suijs (1996), "On Incentive Compatibility and budget balancedness in public decision making", *Economic Design*, 2. 193-209.