Cooperation in Queues *

Akshay-Kumar Katta† Jay Sethuraman ‡

March 2006

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. To that end, we propose two solution concepts (RP and CRP cores), 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 (such as the LB core, due to Maniquet) are the main focus of this paper. We show that the RP core may be empty; but the CRP core is always nonempty, is compatible with the LB core, and contains the equal gains solution. Moreover, we provide an efficient algorithm to find the the egalitarian solution proposed by Dutta and Ray [8], which equalizes costs among all core solutions. We also explore two natural special-cases: identical waiting cost rates and identical processing times. When the waiting cost rates are identical, we prove that the RP core is nonempty, compatible with the LB core, and in fact contains both the equal gains solution and the equal costs. For the case of identical processing times, we show that the RP core is nonempty and contains all envy-free solutions, but is, in general, incompatible with the LB core.

*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

Consider a set of agents arriving simultaneously to a service facility. The facility can serve only one agent at a time; moreover, once an agent begins service, he occupies the service facility until completion. Each agent i requires a processing time p_i at the service facility, and incurs a cost c_i for each unit of time spent in queue. (An agent does not incur any cost when being served.) Moreover, these parameters are observable and do not need to be elicited from the agents. Since the facility can serve only one agent at a 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 approach is to serve the agents in a 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 are served in an efficient order, what compensations can be viewed as being *fair*? Equivalently, how should the agents share their joint costs?

The cost sharing problem is a fundamental problem in economics, and has a rich literature [14]. A successful and often-used approach to "fair" cost sharing 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 their 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. If 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 eliciting the waiting-costs of the agents; in addition to the case with a fixed number of agents, Dolan [7] examines the case where agents arrive over time. Suijs [18] proves the existence of a budget balanced incentive-compatible mechanisms for 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" or "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. The starting point for our investigation is the solution proposed by Maniquet [11], which can be justified by 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 is always in its core. In the rest of this paper, we refer to the core of this game as the Lower Bound (LB) core; the Shapley value of this game will be called simply the *Shapley solution*.

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, and the agents do not cooperate among themselves, then a "reasonable" 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 is 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 cost under the proposed compensation scheme should be no larger than her expected cost under the random service order. But if *arbitrary subsets* of agents can cooperate, then it becomes necessary to define the expected cost 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, and let $s = |S|$. We view the agents in S as owning a random set of s positions, each subset of size s being 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 contiguous blocks. In either case, the cooperating agents will rearrange themselves so as to minimize their total waiting cost. 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 cost of *every subset* of agents is no more than their *expected cost* in the random service order? Solutions satisfying this property will be called RP and CRP core solutions respectively, depending on whether the expected cost is computed under RP or CRP. (The expected cost of a coalition under RP will in general be smaller than its expected cost 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 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 cost 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 cost 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 the coalition $\{2, 3\}$: the joint cost of this coalition is 15 in the Shapley solution, but its *expected* cost in a random ordering is easily seen to be 14. The solution with cost 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 cost 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 cost 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 cost 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 work we analyze the RP and CRP cooperative games, explore the properties of the corresponding cores, and discuss their relationship to the solutions in the LB core. First, we consider the RP game. When the waiting cost rates are identical, we prove that

- the RP game is convex; and
- the RP core is compatible with the LB core, and in fact contains the Shapley solution; both the equal gains solution and the equal costs solution are also in the RP core.

For the case of identical processing times, we show that

- the RP core is non-empty and contains all envy-free solutions;
- the Shapley solution, the equal gains solution (which distributes the net benefit equally among the agents) and the equal costs solution need not be in the RP core, but the unique anonymous budget-balanced VCG solution is; and
- the RP core is, in general, incompatible with the LB core: we construct an example in which none of the RP core solutions satisfies the LB core condition, even for individual agents!

However, even with identical processing times, the RP game may not be convex. Moreover, if processing times and waiting cost rates are arbitrary, the RP core could be empty! This motivates the need for the CRP game. We show that the CRP game is convex, even with arbitrary processing times and waiting costs. The CRP core contains the Shapley solution (hence is compatible with the LB core), and the equal gains solution. As the CRP game is convex, 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. We then 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 faster algorithms for solving them.

The rest of the paper is organized as follows. In §2 we describe the model in detail and formally define the RP and CRP games and their associated cores; in §3, we discuss three standard solution concepts that are used extensively in the rest of the paper: the Shapley solution, the equal costs solution, and the equal gains solution. Sections 4 and 5 are devoted to the analysis of the RP and CRP games, respectively. We conclude by exploring solutions that are at the intersection of the LB core and the CRP core.

2 Model

2.1 Problem Description

A set of agents $N = \{1, 2, \ldots, n\}$ arrive to a service facility 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 spent in queue. We assume that agents incur a cost only when they wait in queue, and do not incur a cost when in service. The facility can serve only one agent at a time. Furthermore, we assume that once the facility starts serving an agent, he 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 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 service provider. 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 service provider (if $t_i < 0$, then agent i pays the service provider $-t_i$). Any solution to the problem is thus completely specified by the quantities $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_n)$ and $t = (t_1, t_2, \ldots, t_n)$, so the cost of agent i is simply

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

A solution $z = (\sigma, t)$ is *feasible* if σ is a permutation of the agents and $\sum_{i=1}^n t_i \leq 0$. The net transfer from the service provider should be non-positive in a feasible solution because we do not want the service provider to incur a loss while serving all the agents. A feasible solution z is *efficient* whenever it minimizes the sum of the costs 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. This implies that finding an efficient solution to the scheduling problem is simple: sequence the agents according to Smith's rule, and let $t_1 = t_2 = \ldots = t_n = 0$. 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 zero cost, but the agent served last has a huge cost. In the next sections, we explore the possibility of finding efficient solutions that are also "fair."

2.2 Games and associated Cores

Given a finite set N of agents, a *cooperative game* with transferable utility is completely specified by its characteristic function $v(\cdot)$ on every non-empty subset of agents. In this section we define two natural cooperative games—RP and CRP games—that are used to find a solution to the scheduling problem.

Random Priority (RP) Game. Let S be a coalition of agents. In the RP game, the agents in S are allowed to exchange the (random) positions they own and rearrange themselves in a way that minimizes their collective cost; this minimum expected total cost is the "worth" of the coalition S in the RP game, and is denoted $C^{RP}(S)$. A cost vector $(w_1, w_2, \ldots, w_n)$ is in the *RP core* if and only if $\sum_{i \in N} w_i \;=\; C^{RP}(N)$, and

$$\sum_{i \in S} w_i \;\leq\; C^{RP}(S), \quad \forall S \subset N.$$

Constrained Random Priority (CRP) Game. Let S be a coalition of agents. In the CRP game, the agents in S are allowed to exchange the random positions they own, but with an additional restriction: only contiguous subsets of the agents in S agents can rearrange themselves. It is reasonable to assume that they rearrange themselves in a way that minimizes their collective cost; this minimum expected total cost (subject to the contiguity restriction) defines the "worth", $C^{CRP}(S)$, of the coalition S in the CRP game. A cost vector $(w_1, w_2, \ldots, w_n)$ is in the *CRP core* if and only if $\sum_{i \in N} w_i \;=\; C^{CRP}(N)$, and

$$\sum_{i \in S} w_i \;\leq\; C^{CRP}(S), \quad \forall S \subset N.$$

The RP and CRP core solutions find cost vectors that place an *upper bound* on the cost of any coalition of agents. In contrast, the *LB core* solution places a *lower bound* on the cost any coalition of agents should bear. The formal definition of the LB game described in Maniquet [11] is the following:

Lower Bound (LB) Game. The worth of a coalition S of agents is their minimum cost, assuming the agents in S are served in the first $|S|$ positions. Let $L(S)$ denote the worth of coalition S in the LB game. A cost vector $(w_1, w_2, \ldots, w_n)$ is in the *LB core* if and only if $\sum_{i \in N} w_i \;=\; L(N)$, and

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

3 Standard solution concepts

In this section we review three standard solution concepts in cooperative game theory and show how to compute them explicitly for the model discussed here. This serves two purposes: first, for some special cases of the model, similar approaches have been used to find compelling solutions; and second, we shall use many of these ideas in the latter sections. To avoid cumbersome notation, we adopt the following convention throughout the paper:

Convention. The agents are served in the order $(1, 2, \ldots, n)$, and 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

Recall that the Shapley solution is the Shapley value of the LB game. Thus, when the ratios c_i/p_i are decreasing in i, the worth of a coalition S is given by

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

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

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

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 $(v(S \cup \{i\}) - v(S))$ 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

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

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. If S is the subset of agents before i, in a uniformly random order, 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,

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

Taking expectations in Eq. (1), we have

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

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 solution utility for each agent. In other words, we want

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

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

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

We end this section by giving an alternative expression for SV_i that will be helpful in establishing the presence of the Shapley solution in CRP core later. Equation (2) can be rewritten as:

$$\begin{aligned} SV_i &= \frac{c_i}{2} \sum_{j:j\neq i} p_j - \frac{1}{2} \sum_{j:j>i} (c_i p_j - c_j p_i) \\ &= c_i \frac{P - p_i}{2} - \frac{1}{2} \sum_{j:j>i} (c_i p_j - c_j p_i) \end{aligned} \tag{5}$$

where $P = \sum_{j\in N} p_j$.

3.2 Equalizing costs

In this section we consider the objective of *equalizing costs*. The aim is to distribute the overall waiting costs as "equally" as possible among all the agents. We first define the notion of *Lorenz domination*, followed by *strict* and *weak* notions of equalizing costs.

Let $x = (x_1, x_2, \ldots, x_n)$ and $y = (y_1, y_2, \ldots, y_n)$ be two given vectors. Let $x^{(i)}$ and $y^{(i)}$ denote the i^{th} smallest component of x and y respectively. We say that x *Lorenz dominates* y if and only if

$$\sum_{i=1}^{k} x^{(i)} \geq \sum_{i=1}^{k} y^{(i)}, \quad k = 1, 2, \ldots, n.$$

Suppose $\mathcal{W}$ is a set of acceptable cost vectors. (For example, $\mathcal{W}$ may be the set of CRP core solutions.) We say that $w \in \mathcal{W}$ *equalizes costs* in the *strict* sense if w Lorenz dominates every $w' \in \mathcal{W}$. We say that $w \in \mathcal{W}$ *equalizes costs* in the *weak* sense if there is no $w' \in \mathcal{W}$ that Lorenz dominates w. Note that a solution that equalizes costs in the strict sense also equalizes costs in the weak sense, but not vice-versa.

The *equal costs* solution is one in which the overall cost $W = \sum_{i\in N} c_i\big(\sum_{j<i} p_j\big)$ is split equally among all agents, i.e. $w_1 = w_2 = \ldots = w_n = W/n$. We note that if the equal costs solution is in $\mathcal{W}$, then it equalizes costs in the strict sense.

3.3 Equalizing Gains

Analogous to equalizing costs, we define the notion of equalizing gains. 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

$$d_i = c_i \sum_{j:j\neq i} p_j/2 \tag{6}$$

the vector of costs in the random service order is $(d_1, d_2, \cdots, d_n)$. Now, if the vector of costs is $(w_1, w_2, \ldots, w_n)$, then agent i gains $(d_i - w_i)$ by cooperating. Thus, every cost vector has a gains vector associated with it. Recall that, in any efficient solution, $\sum_{i:i\in N} w_i = \sum_{i\in N} \sum_{j:j<i} c_i p_j$. Hence, the net benefits from cooperation is given by

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

In the equalizing gains objective, our aim is to distribute this overall gain B as "equally" as possible among all the agents. Again, it is possible to define the notion of equalizing gains in the weak and strict senses. The cost vector $w = (w_1, w_2, \ldots, w_n)$ equalizes gains in the *strict* sense if its associated gain vector Lorenz dominates the gain vector associated with any other acceptable cost vector. The cost vector w equalizes gains in the *weak* sense if its associated gain vector is not Lorenz dominated by the gain vector associated with any other acceptable cost vector.

Paralleling the definition of the equal costs solution, we obtain the *equal gains* solution by letting $w_i = d_i - B/n$, for $i = 1, 2, \ldots, n$. Clearly, if the equal gains solution is among the acceptable solutions, then it equalizes gains in the strict sense.

4 RP Game

In this section we study the RP game. We first consider the case with identical waiting cost rates, followed by the case with identical processing times. In both cases we show that the RP core is nonempty, and discuss the structure of the core solutions. We end this section by showing that the core may be empty for the case with arbitrary waiting cost rates and processing times.

4.1 Identical waiting-cost rates

Consider the case in which agents have identical waiting cost rates, which, without loss of generality, we take to be 1. Recall that, by our convention, agents are labeled so that $p_1 \leq p_2 \leq \ldots \leq p_n$. When the agents of a coalition S rearrange themselves, they 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 being placed earlier, the waiting time of any non-coalition agent can only go down. Therefore, the rules of cooperation under RP are realistic.

Computing $C^{RP}(\cdot)$**.** Consider a coalition $S = \{a_1, a_2, \ldots, a_s\}$ with $p_{a_1} \le p_{a_2} \le \ldots \le p_{a_s}$. By definition, $C^{RP}(S)$ is the expected total cost 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. 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 the processing times of all the jobs. Thus, in the absence of any cooperation, the total expected waiting cost of the agents in S is

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

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 coalitional agents. Note that here "adjacent" does not necessarily mean that a_i and a_j are in positions that are actually adjacent to each other; it just means that there are no other coalitional members between them at the time of exchange. Since any pair of agents will 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,

$$\begin{aligned}
C^{RP}(S) &= \frac{1}{2}\big[sP - \sum_{i \in S} p_i - \sum_{i,j \in S: i<j}(p_j - p_i)\big] && (8)\\
&= \frac{1}{2}\big[sP - \sum_{i=1}^{s} p_{a_i} - \sum_{i=1}^{s-1}\sum_{j=i+1}^{s}(p_{a_j} - p_{a_i})\big] && (9)\\
&= \frac{1}{2}\big[sP + \sum_{i=1}^{s}(s-2i)p_{a_i}\big] && (10)\\
&= \frac{s}{2}\sum_{j: j \notin S} p_j + \sum_{i=1}^{s}(s-i)p_{a_i} && (11)
\end{aligned}$$

We are now ready to prove the main result of this section.

Theorem 1 *The RP game is convex and hence its core is non-empty.*

Proof. The Theorem follows if we show that $C^{RP}(\cdot)$ is a submodular set function. Let $S \subset T$, and let $k \notin T$. Submodularity of $C^{RP}(\cdot)$ follows if we prove the following inequality:

$$C^{RP}(S \cup \{k\}) - C^{RP}(S) \;\ge\; C^{RP}(T \cup \{k\}) - C^{RP}(T). \tag{12}$$

From equation (8), we have

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

and

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

Since $\sum_{i \in S} |p_k - p_i| \ \leq \ \sum_{i \in T} |p_k - p_i|$, inequality (12) holds. ■

4.1.1 Selection from the RP core

We now turn to the problem of selecting an appealing solution in the RP core. We show that the RP core contains the Shapley solution, the equal gains solution, and the equal costs solution.

The Shapley solution. Recall that the RP core constraints give an upper bound on the cost of each coalition of agents; in contrast, the LB core constraints give a lower bound on the cost of each coalition of agents. Particularly appealing solutions are those that are members of both cores; we show next that the Shapley solution is one such solution.

When the waiting cost rates are identically one, equation (5) reduces to

$$SV_i = \frac{P - p_i}{2} - \frac{1}{2} \sum_{j:j>i} (p_j - p_i),$$

where $P := \sum_{j \in N} p_j$. Consider a coalition S, with $|S| = s$. The sum of the (Shapley value) costs of all the agents in S is given by

$$\begin{aligned}
\sum_{i \in S} SV_i &= \sum_{i \in S} \frac{P - p_i}{2} - \frac{1}{2} \sum_{i \in S} \sum_{j:j>i} (p_j - p_i) \\
&= \frac{1}{2}\big[sP - \sum_{i \in S} p_i\big] - \frac{1}{2} \sum_{i,j \in S: i<j} (p_j - p_i) - \frac{1}{2} \sum_{i \in S} \sum_{j:j>i, j \notin S} (p_j - p_i) \\
&\leq \frac{1}{2}\big[sP - \sum_{i \in S} p_i\big] - \frac{1}{2} \sum_{i,j \in S: i<j} (p_j - p_i) \qquad [\text{ as } p_j \geq p_i \quad \forall \ \ j : j > i] \\
&= C^{RP}(S). \qquad [\text{ By (8) }]
\end{aligned}$$

Since this is true for an arbitrary coalition S, it follows that the Shapley solution is in the RP core. Note that, by definition, the Shapley solution is in the LB core.

Equal gains solution. When all agents have a unit waiting cost rate, the cost of agent i in the equal gains solution is given by

$$w_i = \sum_{j:j \neq i} p_j/2 - B/n = (P - p_i)/2 - B/n,$$

where

$$\begin{aligned} B &= \frac{1}{2}\sum_{k=1}^{n}\sum_{j \neq k} p_j - \sum_{k=1}^{n}\sum_{j<k} p_j = \frac{1}{2}\sum_{k=1}^{n}\sum_{j>k} p_j - \frac{1}{2}\sum_{k=1}^{n}\sum_{j<k} p_j \\ &= \frac{1}{2}\sum_{i=1}^{n}(2i-1-n)p_i = \frac{1}{2}\sum_{i=1}^{\lfloor n/2 \rfloor}(n-2i+1)(p_{n-i+1} - p_i). \end{aligned} \quad (13)$$

Let $S = \{a_1, a_2, \ldots, a_s\}$ be any subset of agents. Let $D(S)$ be the cost savings achieved by the agents in S by cooperating among themselves in a random solution, i.e., let $D(S) = \sum_{i=1}^{s}(P - p_{a_i})/2 - C^{RP}(S)$. From equation (9), we have

$$\begin{aligned} D(S) &= \frac{1}{2}\sum_{i=1}^{s-1}\sum_{j=i+1}^{s}(p_{a_j} - p_{a_i}) = \frac{1}{2}\sum_{i=1}^{s}(2i-1-s)p_{a_i} \\ &= \frac{1}{2}\sum_{i=1}^{\lfloor s/2 \rfloor}(s-2i+1)(p_{a_{s-i+1}} - p_{a_i}). \end{aligned} \quad (14)$$

For the equal gains solution to be in the core, we need $\sum_{i=1}^{s} w_{a_i} \leq C^{RP}(S)$. Substituting for the w_{a_i} and $C^{RP}(S)$, we get to

$$\sum_{i=1}^{s}\frac{P - p_{a_i}}{2} - \frac{s}{n}B \leq \sum_{i=1}^{s}\frac{P - p_{a_i}}{2} - D(s),$$

which is equivalent to

$$B \geq \frac{n}{s}D(S) \quad (15)$$

Since $B \geq 0$, (15) holds for all single-person coalitions. Suppose $\mid S \mid = s \geq 2$. We have

$$
\begin{aligned}
B &= \frac{1}{2}\sum_{i=1}^{\lfloor s/2\rfloor}(n-2i+1)(p_{n-i+1}-p_i) \qquad \big[\text{ From equation (13) }\big]\\
&\geq \frac{1}{2}\sum_{i=1}^{\lfloor s/2\rfloor}(n-2i+1)(p_{a_{s-i+1}}-p_{a_i})\\
&\quad \big[\text{ as } p_{n-i+1}-p_i \geq p_{a_{s-i+1}}-p_{a_i} \quad \forall \quad i \leq \lfloor s/2\rfloor\big]\\
&= \frac{1}{2}\frac{n-1}{s-1}\sum_{i=1}^{\lfloor s/2\rfloor}\frac{(n-2i+1)(s-1)}{n-1}(p_{a_{s-i+1}}-p_{a_i})\\
&\geq \frac{1}{2}\frac{n-1}{s-1}\sum_{i=1}^{\lfloor s/2\rfloor}(s-2i+1)(p_{a_{s-i+1}}-p_{a_i})\\
&\quad \left[\text{ as } \frac{(n-2i+1)(s-1)}{n-1} \geq (s-2i+1) \ \ \forall \ \ n \geq s \ \text{ and } i \geq 1\right]\\
&= \frac{n-1}{s-1}D(S) \qquad \big[\text{ from equation (14) }\big]\\
&\geq \frac{n}{s}D(S) \qquad \left[\text{ as } \frac{n-1}{s-1} \leq \frac{n}{s} \ \ \forall \ \ s \leq n\right]
\end{aligned}
$$

Therefore, (15) holds for all coalitions, and so the equal gains solution is in the RP core.

Equal costs solution. The equal costs solution is given by $w_i = W/n$ for all i, where $W = \sum_{i=1}^{n}(n-i)p_i$. To show that this solution is in the RP core, we first prove the following result.

Lemma 2 *Let $C^{RP}(S)$ be the expected waiting cost of S under RP when the waiting costs are all identical. Let*

$$
g(s) \ = \ C^{RP}\left(\left\{1,2,\ldots,\left\lfloor\frac{s}{2}\right\rfloor\right\} \cup \left\{n,n-1,n-2,\ldots,n-\left\lceil\frac{s}{2}\right\rceil+1\right\}\right), \qquad s=1,2,\ldots,n.
$$

Then,

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

Proof. Consider the second term of the RHS of Eq. (10). 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 $C^{RP}(S)$, 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 coefficients. The result follows. ■

Note that in the equal costs solution, the cost of any coalition of s agents is sW/n. In particular, the cost of a coalition depends only on its cardinality. So to check that the equal costs solution is in the RP core, it is enough to prove that

$$\frac{s}{n}W \quad \leq \quad g(s), \qquad s = 1, 2, \ldots, n. \tag{16}$$

We do this by showing the following result.

Lemma 3 *$g(s)$ is a concave function of s, that is,*

$$\frac{g(s-1) + g(s+1)}{2} \leq g(s), \quad s = 1, 2, \ldots, n-1.$$

Proof. Consider any $k < n/2$. From equation (11), we have

$$\begin{aligned} g(2k+1) - g(2k) &= \frac{2k+1}{2}\big[p_{k+1} + \cdots + p_{n-k-1}\big] + p_1 + \cdots + p_k + kp_{n-k} \\ &\quad -\frac{2k}{2}\big[p_{k+1} + \cdots + p_{n-k}\big] \\ &= \frac{1}{2}\big[p_{k+1} + \cdots + p_{n-k-1}\big] + p_1 + \cdots + p_{k-1} + p_k, \end{aligned}$$

and

$$\begin{aligned} g(2k) - g(2k-1) &= \frac{2k}{2}\big[p_{k+1} + \cdots + p_{n-k}\big] + p_1 + \cdots + p_{k-1} + kp_k \\ &\quad -\frac{2k-1}{2}\big[p_k + \cdots + p_{n-k}\big] \\ &= \frac{1}{2}\big[p_{k+1} + \cdots + p_{n-k-1}\big] + p_1 + \cdots + p_{k-1} + \frac{1}{2}p_{n-k} + \frac{1}{2}p_k. \end{aligned}$$

Using these expressions, we can check that

$$\frac{g(2k-1) + g(2k+1)}{2} - g(2k) \;=\; \frac{1}{4}(p_k - p_{n-k}) \;\leq\; 0,$$

and that

$$\frac{g(2k-2) + g(2k)}{2} - g(2k-1) \;=\; 0.$$

■

Note that the inequality (16) is satisfied as an equality for $s = 0$ and for $s = n$. Its LHS grows linearly in s, whereas its RHS is concave in s (due to Lemma 3). Therefore, for every intermediate value of s, the LHS can be no more than the RHS, and so (16) is satisfied for all values of s. It follows that the equal costs solution is in the RP core.

4.2 Identical Processing Times

We turn to the case in which jobs have identical processing times; without loss of generality we take this common processing time to be 1. Recall that, by our convention, the agents are labeled so that $c_1 \geq c_2 \geq \ldots \geq c_n$. When the agents of a coalition S rearrange themselves, they do 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.

Computing $C^{RP}(\cdot)$. 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

$$C^{RP}(S) \quad = \quad \frac{n-s}{s+1}\sum_{i=1}^{s} c_{a_i} + \frac{n+1}{s+1}\sum_{i=2}^{s-1}(i-1)c_{a_i}. \tag{17}$$

Our first main result is that the RP core is non-empty. 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

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

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 4 *Every envy-free solution is in the RP core. In particular, the RP core is non-empty.*

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

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

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

This expression will be the same as the expression in Eq. (17), but it will be convenient here to think of it in these terms. We have:

$$\begin{aligned}
\sum_{i\in S} w_i &= \sum_{i\in S}\Big((\sigma_i-1)c_i - t_i\Big) \\
&= \frac{1}{N!}\sum_{\pi\in\Pi}\sum_{i\in S}\big((\sigma_i-1)c_i - t_i\big) \\
&= \frac{1}{N!}\sum_{\pi\in\Pi}\sum_{i\in S}\big((\sigma_i-1)c_i - t_i + t_{\pi_S(i)}\big) - \frac{1}{N!}\sum_{\pi\in\Pi}\sum_{i\in S} t_{\pi_S(i)} \\
&\le \frac{1}{N!}\sum_{\pi\in\Pi}\sum_{i\in S}\big(\pi_S(i)-1\big)c_i - \frac{1}{N!}\sum_{\pi\in\Pi}\sum_{i\in S} t_{\pi_S(i)} \\
&\qquad \big[\text{ envy-freeness implies } (\sigma_i-1)c_i - t_i \le (\pi_S(i)-1)c_i - t_{\pi_S(i)}\big] \\
&= C^{RP}(S) - \frac{1}{N!}\sum_{\pi\in\Pi}\sum_{i\in S} t_{\pi_S(i)}
\end{aligned}$$

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,

$$\sum_{i\in S} w_i \le C^{RP}(S).$$

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] explores 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

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

(b) The converse of Theorem 4 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.

4.2.1 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.

The Shapley solution and the LB core. When there are only two agents, it is easy to check that the cost 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 solution discussed in §3.1 is always in the LB core, the same example proves that the Shapley solution need not always be in the RP core. Moreover, in Example 3 the equal costs solution—$w_1 = w_2 = w_3 = 1$— is not 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, it may not contain the Shapley solution, the equal costs solution, or the equal gains solution. In fact, $C^{RP}(S)$ may not be submodular as shown in the following example.

Example 4. Suppose $n = 3$, $c_1 = 3$, $c_2 = 0$, and $c_3 = 0$. Let $S = \{1\}, T = \{1, 2\}$ and $k = 3$. It is straightforward to verify that:

$$C^{RP}(S) = 3, \;\; C^{RP}(S \cup \{k\}) = 1, \;\; C^{RP}(T) = 1, \;\; C^{RP}(T \cup \{k\}) = 0.$$

Therefore, we have

$$C^{RP}(S \cup \{k\}) - C^{RP}(S) = -2, \;\; C^{RP}(T \cup \{k\}) - C^{RP}(T) = -1$$

Hence $C^{RP}(S \cup \{k\}) - C^{RP}(S) < C^{RP}(T \cup \{k\}) - C^{RP}(T)$, and so the function $C^{RP}(S)$ is not submodular. Thus the RP game is not convex.

We end this section, however, on a positive note: we show that the (unique) anonymous 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 (see Chun [4]) if $t_{i+1} - t_i \in [c_{i+1}, c_i]$. We have:

$$t_{i+1} - t_i = \frac{c_i}{2} + \frac{c_{i+1}}{2} + \frac{2i-n}{2(n-2)}[c_i - c_{i+1}] = \frac{n-i-1}{n-2}c_i + \frac{i-1}{n-2}c_{i+1}. \qquad (18)$$

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).

4.3 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. By our convention, the agents are labeled so that $c_1/p_1 \geq c_2/p_2 \geq \ldots \geq c_n/p_n$. 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 is too optimistic, and that distributing the costs among the agents to meet this optimistic bound for all coalitions may be impossible. This is formally shown in the following example.

Example 5. Suppose $n = 3$, and suppose $c_1 = 5$, $c_2 = 0.8$, $c_3 = 3$, $p_1 = 5$, $p_2 = 1$, and $p_3 = 4$. It is straightforward to verify that the RP core for this model is characterized by the constraints:

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

$$w_1 + w_2 \leq c_2 p_1 + p_3 \frac{c_1 + 2c_2}{3}, \quad w_1 + w_3 \leq c_3 p_1 + p_2 \frac{c_1 + 2c_3}{3}, \quad w_2 + w_3 \leq c_3 p_2 + p_1 \frac{c_2 + 2c_3}{3},$$

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)

$$w_3 \leq 9, \;\; w_1 + w_2 \leq 12.8, \;\; w_1 + w_2 + w_3 = 22,$$

and this subsystem of inequalities is inconsistent.

This example justifies the need for the CRP game, which we analyze next.

5 CRP game

Recall that in the CRP game, only *contiguous* subsets of cooperating agents can rearrange themselves using the slots they occupy. It is straightforward to see that the efficient ordering of the agents of a coalition S, subject to the contiguity requirement, can be achieved by a series of exchanges between (physically) adjacent coalitional members. Our goal in this section is to study the properties of the CRP core. As we shall see, the CRP game is convex even for the case of arbitrary waiting cost rates and processing times, so we analyze this general case directly. Recall that the agents are labeled so that $c_1/p_1 \geq c_2/p_2 \geq \ldots \geq c_n/p_n$.

Computing $C^{CRP}(\cdot)$. We begin by computing the characteristic function of the CRP game. Consider a coalition, S, of agents with $|S| = s$. If the agents are ordered uniformly at random, the expected total cost of the agents in S in the absence of cooperation will be

$$\sum_{i \in S} c_i \frac{(P - p_i)}{2},$$

where P is the sum of processing times of all the jobs. By cooperating the agents in S generate some cost-savings, which can be found as follows: Consider two agents i and j, with $i < j$. (By our convention, $c_i/p_i \geq c_j/p_j$.) 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 in the random order 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 $[i, j, N \setminus S, S \setminus \{i, j\}]$. We first place agent i, we then place agent j randomly in one of the two possible slots, after which we sequentially place each of the agents of $N \setminus S$ (the order within this set does not matter) randomly in one of the available slots and finally we sequentially place the agents of $S \setminus \{i, j\}$ (again the order within this set does not matter) randomly in one of the available slots. For i and j to exchange positions, 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 $1/(n - s + 2)$. Also, since i and j will be in adjacent slots when they swap their positions, the cost-savings from a rearrangement between i and j is $c_i p_j - c_j p_i$; therefore the expected savings due to the rearrangement of i and j will be $(c_i p_j - c_j p_i)/(n - s + 2)$. Therefore

$$C^{CRP}(S) = \sum_{i \in S} c_i \frac{(P - p_i)}{2} - \frac{1}{n + 2 - s} \sum_{i,j \in S, i<j} (c_i p_j - c_j p_i). \tag{19}$$

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

Theorem 5 *The CRP game is convex and hence its core is non-empty.*

Proof. We prove the Theorem by showing that $C^{CRP}(\cdot)$ is a submodular set function, that is,

$$C^{CRP}(S \cup \{k\}) - C^{CRP}(S) \geq C^{CRP}(T \cup \{k\}) - C^{CRP}(T), \quad \forall S \subset T, \;\; k \notin S.$$

Observe that

$$
\begin{aligned}
C^{CRP}(S\cup\{k\})-C^{CRP}(S) &= c_k\frac{P-p_k}{2}-\frac{1}{n+1-s}\sum_{i\in S}|c_kp_i-c_ip_k| \\
&\quad -\left(\frac{1}{n+1-s}-\frac{1}{n+2-s}\right)\sum_{i,j\in S,i<j}(c_ip_j-c_jp_i),
\end{aligned}
$$

and

$$
\begin{aligned}
C^{CRP}(T\cup\{k\})-C^{CRP}(T) &= c_k\frac{P-p_k}{2}-\frac{1}{n+1-t}\sum_{i\in T}|c_kp_i-c_ip_k| \\
&\quad -\left(\frac{1}{n+1-t}-\frac{1}{n+2-t}\right)\sum_{i,j\in T,i<j}(c_ip_j-c_jp_i).
\end{aligned}
$$

For $S\subset T$, $s<t$; it is easy to verify that each term in $C^{CRP}(S\cup\{k\})-C^{CRP}(S)$ dominates the corresponding term in $C^{CRP}(T\cup\{k\})-C^{CRP}(T)$. The submodularity of $C^{CRP}(\cdot)$ follows. ■

Remark. A closely related result is by Curiel et al. [6], where they establish a similar result with a fixed initial order. We can prove Theorem 5 by averaging over all initial orders.

We now consider the problem of identifying "good" solutions in the CRP core. As before, we explore the Shapley solution, and the goals of equalizing gains and equalizing costs.

5.1 The Shapley solution.

By definition, the Shapley solution is in the LB core. Now, we show that it is in the CRP core as well. Consider a coalition S with $|S|=s$. Substituting from equation (5), we have

$$
\begin{aligned}
\sum_{i\in S}SV_i &= \sum_{i\in S}c_i\frac{P-p_i}{2}-\frac{1}{2}\sum_{i\in S}\sum_{j:j>i}(c_ip_j-c_jp_i) \\
&= \sum_{i\in S}c_i\frac{P-p_i}{2}-\frac{1}{2}\sum_{i,j\in S,i<j}(c_ip_j-c_jp_i)-\frac{1}{2}\sum_{i\in S}\sum_{j:j>i,j\notin S}(c_ip_j-c_jp_i) \\
&\le \sum_{i\in S}c_i\frac{P-p_i}{2}-\frac{1}{2}\sum_{i,j\in S,i<j}(c_ip_j-c_jp_i) \qquad \left[\text{ as } c_ip_j\ge c_jp_i \quad \forall \;\; j:j>i\right] \\
&\le \sum_{i\in S}c_i\frac{P-p_i}{2}-\frac{1}{n+2-s}\sum_{i,j\in S,i<j}(c_ip_j-c_jp_i) \\
&\quad \left[\text{ as } c_ip_j\ge c_jp_i \quad \forall \;\; i<j \qquad \text{and} \qquad n+2-s\ge 2 \;\right] \\
&= C^{CRP}(S) \qquad (20)
\end{aligned}
$$

Since S was an arbitrary coalition, it follows that the Shapley solution is in the core.

5.2 Equalizing gains

In the equal gains solution, the cost of agent i is $w_i = \sum_{j:j\neq i} c_i p_j/2 - B/n = c_i(P - p_i)/2 - B/n$, where B is given by:

$$B := \sum_{i,j\in S, i<j} \frac{(c_i p_j - c_j p_i)}{2}.$$

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

$$\frac{s}{n} B \geq \frac{1}{n-s+2} \sum_{i,j\in S, i<j} (c_i p_j - c_j p_i). \tag{21}$$

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:

$$\begin{aligned}
\frac{s}{n} B &= \frac{s}{n} \sum_{i,j\in N, i<j} \frac{c_i p_j - c_j p_i}{2} \\
&= \frac{s}{n} \frac{n-s+2}{2} \frac{1}{n-s+2} \sum_{i,j\in N, i<j} (c_i p_j - c_j p_i) \\
&\geq \frac{1}{n-s+2} \sum_{i,j\in N, i<j} (c_i p_j - c_j p_i) \qquad \left[\text{as } \frac{s}{n}\frac{n-s+2}{2} \geq 1 \quad \forall \;\; 2 \leq s \leq n\right] \\
&\geq \frac{1}{n-s+2} \sum_{i,j\in S, i<j} (c_i p_j - c_j p_i)
\end{aligned}$$

Thus, equation (21) holds and the equal gains solution is in the CRP core.

5.3 Equalizing costs

It is not difficult to show that the equal costs solution is *not* in the CRP core. Thus, our goal is to find a core solution that Lorenz dominates every other solution in the CRP core. For convex games, Dutta and Ray [8] propose an algorithm to find such a solution (called the "egalitarian" solution). As the CRP game is convex, we can use the Dutta-Ray algorithm to compute the equal costs solution in the CRP core. Their algorithm, adapted to our problem, is as follows:

(i) Identify *the* largest coalition S^* that minimizes $C^{CRP}(S)/|S|$, over all non-empty subsets S of N, and let $w_i = C^{CRP}(S^*)/|S^*|$ for all $i \in S^*$;

(ii) Define $\hat{N} = N \setminus S^*$, and for every $S \subseteq \hat{N}$, define $\hat{C}(S) \leftarrow C^{CRP}(S \cup S^*) - C^{CRP}(S^*)$; and

(iii) Apply (i) recursively, with $\hat{N}$ playing the role of N, and $\hat{C}(\cdot)$ playing the role of $C^{CRP}(\cdot)$.

Note that submodularity of $C^{CRP}(\cdot)$ 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 here, as we show next.

Let $w = (w_1, w_2, \ldots, w_n)$ denote the vector of costs. Recall that the CRP core constraints are

$$\sum_{i \in S} w_i \leq C^{CRP}(S), \quad \forall S \subset N,$$

where $C^{CRP}(S)$ is given by (19). Define $z_i = (\sum_{j=1}^n p_j)c_i/2 - c_ip_i/2 - w_i$; note that $(z_1, \ldots, z_n)$ is just the vector of gains 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. Replacing w_i by $z_i + c_ip_i/2 - (\sum_{j=1}^n p_j)c_i/2$, and substituting for $C^{CRP}(S)$, the CRP core constraints can be rewritten as

$$-\sum_{i \in S} z_i + \frac{p_1 + \cdots + p_n}{2} \sum_{i \in S} c_i - \frac{\sum_{i \in S} c_ip_i}{2}; \leq \ C^{CRP}(S).$$

Substituting for $C^{CRP}(S)$ from Eq. (19), the core constraints can be expressed in terms of the z_i as

$$\sum_{i \in S} z_i \ \geq \ \frac{1}{n+2-s} \sum_{i,j \in S : i<j} g_{ij}, \tag{22}$$

where $g_{ij} = |c_ip_j - c_jp_i|$. Defining $G(S) = \sum_{i,j \in S, i<j} g_{ij}$, and $F(S) = G(S)/(n - |S| + 2)$, the core constraints become

$$\sum_{i \in S} z_i \ \geq \ F(S).$$

Now consider the network shown in Figure 1. 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. 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}; and for each node q_{ij}, there are 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.

Let Cuts(S) be the set of cuts in which the nodes $r_i, i \in S$ are on the source side of the cut and the nodes $r_i, i \in N \setminus S$ are on the sink side. Let MinCut(S) be a cut with minimum capacity among all the cuts in Cuts(S). Since none of the infinite capacity edges can be in a minimum-cut, all the nodes of the type q_{ij}, where either $i \in S$ or $j \in S$, must be on the source side of $MinCut(S)$; all the remaining nodes of the type q_{ij} will be on the sink-side. Therefore, the capacity of MinCut(S) will be

$$\sum_{j \in N \setminus S} 2z_j + \sum_{j \in N \setminus S} sz_j + G(N) - G(N \setminus S) \tag{23}$$

where the second term comes from the observation that each node $r_i, i \in S$ is connected to each node $r_j, j \in N \setminus S$ by an edge of capacity z_j. Notice that a minimum cut in the network has to be one among the set of cuts $\{MinCut(S) : S \subseteq N\}$. We now show how the equal costs solution can be calculated by solving a parametric maximum flow problem on this network.

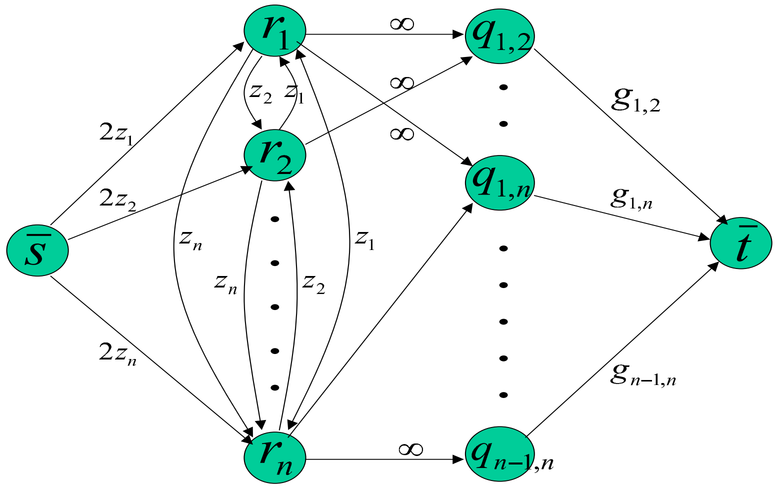

Figure 1: Network for equalizing costs

Implementing the initial step. Let $z_i = c_i(p_1 + \cdots + p_n)/2 - c_i p_i/2 - \delta \ \ \forall \ \ i \in N$; thus we will have $w_i = \delta \ \ \forall \ \ i \in N$. First, suppose $\delta = 0$. Then, for any $S \subseteq N$, we have

$$\begin{aligned}
\sum_{j \in N \setminus S} (s+2) z_j + G(N) - G(N \setminus S) &\geq \sum_{j \in N \setminus S} 2 z_j + G(N) - G(N \setminus S) \qquad [\text{ as } s \geq 0] \\
&= \frac{p_1 + \cdots + p_n}{2} c_i - \frac{c_i p_i}{2} + G(N) - \sum_{i,j \in N \setminus S} g_{ij} \\
&\quad [\text{ substituting for } z_i \text{ and } G(N \setminus S)] \\
&\geq G(N)
\end{aligned}$$

Thus, the capacity of MinCut(S) for any $S \subseteq$ N, is at least $G(N)$. Moreover, the capacity of MinCut(N) is exactly $G(N)$. Thus, when $\delta = 0$, MinCut(N) is a minimum capacity cut in the network and its capacity is $G(N)$.

Increase δ gradually increased from zero, and let δ_1 be the smallest positive value of δ for which some cut MinCut(S) with $S \subset N$ is also a minimum-cut of the network (along with MinCut(N)). Assume that $S_1 \subset N$ is the minimum cardinality set among all such sets; thus MinCut(S_1) is a minimum-cut of the network when $\delta = \delta_1$, and δ_1 is the smallest break-point of the parametric maximum flow problem associated with this network. Moreover, as MinCut(N) is still a minimum cut of the network, the capacity of the minimum-cut in the network is $G(N)$. Therefore, when $\delta = \delta_1$, the capacity of

MinCut(S) is at least $G(N)$ and so

$$
\begin{aligned}
& \sum_{j \in N \setminus S} (s+2) z_j + G(N) - G(N \setminus S) \geq G(N) \\
\Longrightarrow \quad & \sum_{j \in N \setminus S} z_j \geq F(N \setminus S) \\
\Longrightarrow \quad & \frac{p_1 + \cdots + p_n}{2} \sum_{i \in N \setminus S} c_i - \frac{\sum_{i \in N \setminus S} c_i p_i}{2} - (n-s)\delta_1 \geq F(N \setminus S) \\
\Longrightarrow \quad & \delta_1 \leq \frac{\frac{p_1 + \cdots + p_n}{2} \sum_{i \in N \setminus S} c_i - \frac{\sum_{i \in N \setminus S} c_i p_i}{2} - F(N \setminus S)}{n-s} = \frac{C^{CRP}(N \setminus S)}{n-s}
\end{aligned}
$$

As all the above inequalities are tight for $S = S_1$, denoting $N \backslash S$ by T, we have $\delta_1 = \min_{T \subset N} C^{CRP}(T)/|T|$. This completes the first step in determining the egalitarian solution.

Implementing the subsequent steps. Let

$$z_i \;=\; \frac{p_1 + \cdots + p_n}{2} c_i - i \frac{c_i p_i}{2} - \delta_1, \quad \forall i \in N \setminus S_1,$$

and let

$$z_i \;=\; \frac{p_1 + \cdots + p_n}{2} c_i - i \frac{c_i p_i}{2} - \delta, \quad \forall i \in S_1.$$

Note that the z_i for $i \in N \setminus S_1$ are *fixed*, whereas the z_i for $i \in S_1$ depend on δ, which is the parameter that will be varied in this iteration. Augment the network with infinite capacity arcs from the nodes $r_j : j \in N \setminus S_1$ to the sink node $\bar{t}$; thus these nodes will always be on the sink side of any min-cut in the augmented network. Note that, when $\delta = \delta_1$, the capacity of the cut $MinCut(S_1)$ in this augmented network is the same as its capacity in the original network. Moreover, the addition of positive-capacity arcs to to the network cannot reduce the minimum-cut capacity. Therefore, when $\delta = \delta_1$, MinCut(S_1) is still a minimum-cut of this augmented network with capacity $G(N)$. Increase δ gradually from δ_1, and let δ_2 be the smallest breakpoint of the parametric maximum flow problem of this modified network. When $\delta = \delta_2$, the modified network will have a minimum cut of the type MinCut(S), where $S \subset S_1$; let $S_2 \subset S_1$ be a minimum cardinality set among all such sets. Thus $MinCut(S_2)$ is a minimum-cut of the augmented network when $\delta = \delta_2$, and its capacity is $G(N)$. Therefore, the capacity of MinCut(S) is at least $G(N)$ and we will have

$$
\begin{aligned}
& \sum_{j \in N \setminus S} (s+2) z_j + G(N) - G(N \setminus S) \geq G(N) \\
\Longrightarrow \quad & \sum_{j \in N \setminus S} z_j \geq F(N \setminus S) \\
\Longrightarrow \quad & \frac{p_1 + \cdots + p_n}{2} \sum_{i \in N \setminus S} c_i - \frac{\sum_{i \in N \setminus S} c_i p_i}{2} - \sum_{i \in N \setminus S_1} \delta_1 - \sum_{i \in S_1 \setminus S} \delta_2 \geq F(N \setminus S) \\
\Longrightarrow \quad & \delta_2 \leq \frac{C^{CRP}(N \setminus S) - \sum_{i \in N \setminus S_1} \delta_1}{s_1 - s} = \frac{C^{CRP}(S_1 \setminus S \;\cup\; N \setminus S_1) - C^{CRP}(N \setminus S_1)}{s_1 - s}
\end{aligned}
$$

As all the above inequalities are tight for $S = S_2$, denoting $S_1 \setminus S$ by T, we have

$$\delta_2 = \min_{T \subset S_1} \frac{C^{CRP}(T \cup N \setminus S_1) - C^{CRP}(N \setminus S_1)}{|T|}.$$

This completes the second iteration. Continuing in the same way, we can find the core solution that equalizes costs in at most n iterations.

As mentioned earlier, the equalizing costs solution 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 cost as well as minimizes the maximum cost; moreover, the cost vector found is also lexicographically minimal among all cost vectors in the core. The following remarks discuss how the computation can be simplified when either the waiting costs or the processing times are identical.

Remarks.

(a) For the special case with identical waiting cost rates, we already showed that the equal costs solution, which splits the total cost W equally among all the agents, is in the RP core. As the CRP core contains the RP core, it follows that this solution is in the CRP core as well. (In particular, the Dutta-Ray procedure will find this solution!)

(b) Suppose the processing times are identically 1. Let $S = \{a_1, a_2, \ldots, a_s\}$, be a coalition of s members with $c_{a_1} \geq c_{a_2} \geq \ldots \geq c_{a_s}$. For coalition S, we have

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

Note that the coefficient of c_{a_k} in the second term increases as k increases, so c_{a_1} has the smallest coefficient. Also, the coefficient of c_{a_1} is $(n-1)/2 - (s-1)/(n+2-s)$, which decreases as s increases, so its smallest value is zero, which occurs when $s = n$. Thus, for any subset S, we will have $c_{a_1} \geq 0$. Hence, the coefficient of c_{a_i} in the expression for $C^{CRP}(S)$ is always non-negative, and increasing with i. So to minimize $C^{CRP}(S)$ among all sets of cardinality s, it is optimal to include the agents with the s smallest cost coefficients. Thus, we will have $C^{CRP}(\{s+1, \ldots, n\}) \;=\; \min_{S \subseteq N, |S| = n-s} C^{CRP}(S), \quad \forall s = 0, 1, \ldots, n-1$. From this discussion, it follows that in the first step of the Dutta and Ray algorithm, the set S^* will have the form $S^* = \{n, n-1, \ldots, i\}$ for some integer $1 \leq i \leq n$. Thus, we can identify the bottleneck set S^* by calculating the value of $C^{CRP}(n, n-1, \ldots, i)/(n+1-i)$ for all possible values of i and choosing the one with the smallest value. Thus the first step can be implemented in linear time. A similar argument establishes that the bottleneck sets in the intermediate steps consists of a sequence of contiguous elements, and so can be found in linear time. Thus each iteration of the algorithm can be implemented in $O(n)$ time, and there are at most n iterations.

6 Compatibility with the LB core

In the preceding sections, we have proposed two broad notions of equitable cost sharing: the RP and the CRP core solutions. Implicit in Maniquet [11] is another notion of fair cost sharing: the LB core solution. We have already seen the extent to which these two notions can be reconciled:

- The Shapley solution is a member of both the CRP core and the LB core. If the c_i are all identical, the Shapley 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.

This section is motivated by the search for other cost 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 solution is in both cores, it may not achieve auxiliary objectives such as equalizing costs or equalizing gains (among all the core solutions). Further, we saw that the equal gains solution may not be in the LB core, even in the two special cases (i.e. $p \equiv 1$ or $c \equiv 1$). Similarly, the solution that equalizes costs in the CRP core may not be in the LB core, even in the two special cases. 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 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. Again, we assume the agents are labeled so that $c_1/p_1 \geq c_2/p_2 \geq \ldots \geq c_n/p_n$.

6.1 Separation problem for 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.

6.1.1 Identical Processing Times

For any S,

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

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

$$\max_{S \subseteq N} \big[L(S) - \sum_{i \in S} w_i\big]. \tag{24}$$

If the optimal value of this optimization problem is non-positive, then w is in the LB core; otherwise, a set that achieves the maximum value will violate the core condition. Let R be the maximal set that achieves the optimal value for the optimization problem (24). We have the following result.

Lemma 6 *Let R be the maximal set that achieves the optimal value for the optimization problem (24). Then, $k \in R$ if and only if*

$$\sum_{i \in R; i>k} c_i + \sum_{i \in R; i<k} c_k \geq w_k \tag{25}$$

Proof. Suppose that $k \notin R$. Then we have

$$\begin{aligned} L(R \cup \{k\}) - \sum_{i \in R \cup \{k\}} w_i &= \sum_{i,j \in R; i<j} c_j + \sum_{i \in R; i>k} c_i + \sum_{i \in R; i<k} c_k - \sum_{i \in R} w_i - w_k \\ &= L(R) - \sum_{i \in R} w_i + \sum_{i \in R; i>k} c_i + \sum_{i \in R; i<k} c_k - w_k \end{aligned} \tag{26}$$

Since R is the maximal set that achieves the optimal value for problem (24), we have

$$\begin{aligned} L(R \cup \{k\}) - \sum_{i \in R \cup \{k\}} w_i &< L(R) - \sum_{i \in R} w_i \\ \Longrightarrow \sum_{i \in R; i>k} c_i + \sum_{i \in R; i<k} c_k &< w_k. \quad \text{(Substituting from equation 26)} \end{aligned} \tag{27}$$

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

$$\begin{aligned} L(R/\{k\}) - \sum_{i \in R/\{k\}} w_i &= \sum_{i,j \in R/k; i<j} c_j - \sum_{i \in R/\{k\}} w_i \\ &= \sum_{i,j \in R; i<j} c_j - \sum_{i \in R; i>k} c_i - \sum_{i \in R; i<k} c_k - \sum_{i \in R} w_i + w_k \\ &= L(R) - \sum_{i \in R} w_i - \sum_{i \in R; i>k} c_i - \sum_{i \in R; i<k} c_k + w_k \end{aligned} \tag{28}$$

Again, since R is the maximal set that achieves the optimal value for problem (24), we have

$$\begin{aligned} L(R/\{k\}) - \sum_{i \in R/\{k\}} w_i &\leq L(R) - \sum_{i \in R} w_i \\ \implies \sum_{i \in R; i > k} c_i + \sum_{i \in R; i < k} c_k &\geq w_k. \quad \text{(Substituting from equation 28)} \end{aligned} \tag{29}$$

Using (27) and (29), we can conclude that

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

■

We will now describe an algorithm, that uses Lemma 6, to check for the presence of w in the LB core in $O(n^2)$ time. Suppose the size of the set R is m. Then we can directly check whether agent n is present in R using Eq. (25), 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. (25): 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 (24). 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 $O(n^2)$.

6.1.2 Identical Waiting Costs

For any S, in this case, $L(S) = \sum_{i,j \in S, i<j} p_i$. 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 again 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 achieves the maximum value will violate the core condition. The algorithm 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.

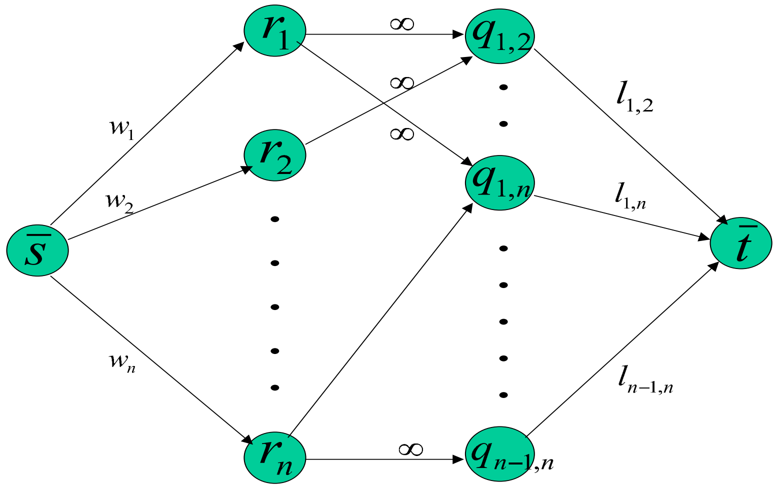

Figure 2: Network 5

6.1.3 Arbitrary processing times and waiting costs

In this case, for any S, we have

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

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

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

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.

Denote $l(i,j) = c_j p_i$. Consider the network shown in Figure 2. The 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. 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}, there are two infinite capacity edges (r_i, q_{ij}) and (r_j, q_{ij}). We now show how the separation problem can be solved by computing the maximum-flow in this network. First, note that the capacity of all the arcs that can bring flow into the sink is $L(N)$. Thus, the maximum flow in the network can be no more than $L(N)$.

Suppose that the maximum flow in the network is $L(N)$. We now argue that w is in the LB 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 $S \subset N$. Thus, the core constraints are satisfied and w is in the LB core.

Now suppose that 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. The minimum-capacity of such a cut is clearly $\sum_{i \notin S} w_i + \sum_{i \in S, j \in N} l_{ij} = \sum_{i \notin S} w_i + L(N) - L(N \setminus S)$. Since the minimum-cut capacity is at most $L(N)$, we have:

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

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

Thus, by solving the maximum flow problem on the network in Figure 2, we either conclude that the cost vector w is in the LB core or we exhibit a set $N \setminus S$ for which the core constraint is violated. Thus, the separation problem can be solved by computing the maximum flow on this network.

6.2 Separation problem for the CRP core

We present an algorithm to solve the separation problem for the general case in which agents have arbitrary processing times and waiting costs. Let $w = (w_1, w_2, \dots, w_n)$ be the given vector of costs; define $z_i = \frac{p_1 + \dots + p_n}{2} c_i - \frac{c_i p_i}{2} - w_i \quad \forall i \in N$. Recall, from equation (22), that the CRP core constraints can be rewritten as

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

where $G(S) = \sum_{i,j \in S, i<j} |c_i p_j - c_j p_i|$.

Consider the network in Figure 1, which was used in calculating the equalizing costs solution in CRP core. Recall, from equation (23), that the minimum-capacity among all cuts in which the nodes $r_i, i \in S$ 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 given by

$$\sum_{j \in N : r_j \notin S} (s+2) z_j + G(N) - G(N \setminus S) \tag{30}$$

We now show how the separation problem, for the CRP core, can be solved by computing the minimum-cut in this network (due to the max-flow min-cut theorem, this is the same as computing the maximum flow on this network). First, note that the capacity of the cut that separates the sink from all the other nodes is $G(N)$. Thus, the min-cut capacity can be no more than $G(N)$.

Suppose that the capacity of the minimum-cut is $G(N)$. Thus, the capacity of any cut in the network will be at least $G(N)$. In particular, for any subset S, the minimum-capacity among all cuts in which the nodes $r_i, i \in S$ are on the source side of the cut and the nodes $r_i, \forall i \in N \setminus S$ are on the sink side will be at least $G(N)$; substituting for this from equation (30), we get

$$\sum_{j \in N: r_j \notin S} (s+2)z_j + G(N) - G(N \setminus S) \geq G(N)$$
$$\implies \qquad \sum_{j \in N \setminus S} z_j \geq \frac{G(N \setminus S)}{n+2-(n-s)} = F(N \setminus S)$$

Thus the core constraint is satisfied for any set $N \setminus S$; since S is arbitrary, this implies all the core constraints are satisfied.

Suppose that the minimum-cut capacity is less than $G(N)$. Let S_1 be the set such that the nodes $r_i, i \in S_1$ are on the source side of a min-cut and the nodes $r_i, \forall i \in N \setminus S_1$ are on the sink side of this min-cut. Thus, the capacity of the minimum-cut can be got by substituting $S = S_1$ in equation (30). Since this capacity is less than $G(N)$, we get

$$\sum_{j \in N: r_j \notin S_1} (s_1+2)z_j + G(N) - G(N \setminus S_1) < G(N)$$
$$\implies \qquad \sum_{j \in N \setminus S_1} z_j < \frac{G(N \setminus S_1)}{n+2-(n-s_1)} = F(N \setminus S_1)$$

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

Thus, by solving the maximum flow (min-cut) problem on the network in Figure 1, we either conclude that the cost vector w is in the CRP core or we exhibit a set $N \setminus S$ for which the core constraint is violated. Thus, the separation problem can be solved by computing the maximum flow on this network.

6.3 Implementing the solution concepts

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

Equalizing costs. Suppose we want to equalize costs in a weak sense, subject to the restriction that the cost 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

$$\begin{aligned} & \max \quad x \\ s.t. \quad & L(S) \leq \sum_{i \in S} w_i \leq C^{CRP}(S) \\ & w_i \geq x, \quad i = 1, \dots, n. \end{aligned} \tag{31}$$

Since the Shapley solution is in the intersection of the LB and the CRP cores, the optimization problem (31) is always feasible. This problem can be solved in polynomial-time as the associated separation problem can be solved in polynomial time (as shown in sections 6.1 and 6.2). Let x^* be the value of the objective function in an optimal solution. Define S_1 to be the set of agents whose costs are x^* in all the optimal solutions of (31). Note that S_1 can be found efficiently by standard techniques.

Fixing the costs of the agents in S_1 at x^*, we then maximize the minimum cost of the remaining agents. This can be achieved by solving the optimization problem (31) with the following modifications: the constraint $w_i \geq x$ is replaced by $w_i = x^*$ for each agent $i \in S_1$. Since we had a feasible solution at the end of the first iteration, this new optimization problem is still feasible, and we will find an optimal solution. Let x^{**} be the value of the objective function in an optimal solution of this new problem, and let S_2 be the agents whose cost is x^{**} in all optimal solutions. 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 in a weak sense.

Equalizing costs. A similar procedure can be used to find a solution that equalizes the gains of the agents in a weak sense among all solutions that are in both the LB and CRP cores. The only difference is in the definition of the optimization problem, which is now

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

where d_i is the cost of agent i when the processing order is chosen uniformly at random.

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.