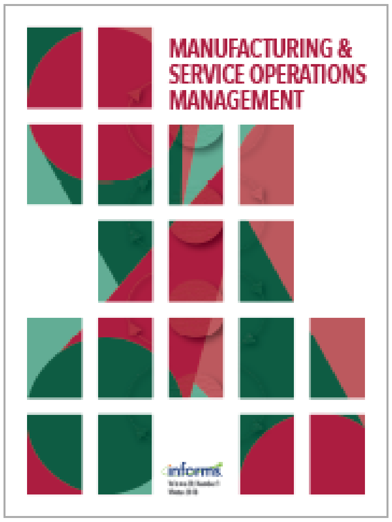


Manufacturing & Service Operations Management

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

The Basic Core of a Parallel Machines Scheduling Game

Tzvi Alon, Shoshana Anily

informs
https://pubsonline.informs.org/journal/msom
MANUFACTURING & SERVICE OPERATIONS MANAGEMENT
Vol. 25, No. 6, November–December 2023, pp. 2233–2248
ISSN 1523-4614 (print), ISSN 1526-5498 (online)

The Basic Core of a Parallel Machines Scheduling Game

Tzvi Alon,[a] Shoshana Anily[b,*]

[a] Department of Statistics and Federmann Center for the Study of Rationality, Hebrew University of Jerusalem, 9190501 Jerusalem, Israel;
[b] Coller School of Management, Tel Aviv University, 6997801 Tel Aviv, Israel

*Corresponding author

Contact: alontzvi@gmail.com, https://orcid.org/0000-0002-3847-8005 (TA); anily@post.tau.ac.il, https://orcid.org/0000-0001-8527-1659 (SA)

Received: July 22, 2021
Revised: December 8, 2021; September 21, 2022; May 10, 2023
Accepted: May 16, 2023
Published Online in Articles in Advance: July 20, 2023

https://doi.org/10.1287/msom.2021.0337

Abstract. ***Problem definition*:** We consider the parallel machine scheduling (PMS) under job-splitting game defined by a set of manufacturers where each holds uniform parallel machines and each is committed to produce some jobs submitted to her by her clients while bearing the cost of the sum of completion times of her jobs on her machines. An efficient algorithm for this scheduling problem is well known. We consider the corresponding cooperative game, where the manufacturers are players that want to join forces. We show that collaboration is profitable. Yet, the stability of the cooperation depends on the cost allocation scheme; we focus on the core of the game. ***Methodology/results*:** We prove that the PMS game is totally balanced and its core is infinitely large, by developing a sophisticated methodology of linear complexity that finds a line segment in its symmetric core. We call this segment the basic core of the game. ***Managerial implications*:** This PMS game has the potential for various applications both in traditional industry and in distributed computing systems in the hi-tech industry. The formation of a partnership among entrepreneurs, companies, or manufacturers necessitates not only a plan for joining forces toward the achievement of the ultimate goals, but also an acceptable agreement regarding the cost allocation among the partners. Core allocations guarantee the stability of the partnership as no subset of players can gain by defecting from the grand coalition.

Funding: This work was supported by the Henry Crown Israeli Institute for Business Research, the Coller Foundation, and the Israel Science Foundation [Grant 1489/19].

Keywords: cooperative games • core • parallel machines scheduling • job splitting • distributed computing

1. Introduction

In this paper, we deal with a cooperative game where several players own similar machines that process the same type of jobs, but they may differ in their speeds. Clients submit jobs to the players in order to process them. The goal of each player, if she works independently of the others, is to process her jobs on her machines so that the sum of completion times of her jobs is minimized. The type of jobs that we consider here allows for *job splitting* (i.e., any job can be split into parts that can be processed simultaneously on different machines). Our goal is to investigate the profitability of cooperation in such systems.

Because of the simplicity and generality of the system described, it has a potential for various interesting applications in both traditional and hi-tech industries. For a recent collaboration in the production industry, consider the pharmaceutical giants Sanofi and Novartis, which used their manufacturing capabilities to produce the mRNA vaccine developed by Pfizer and BionTech during the coronavirus disease 2019 pandemic. In service systems, capacity sharing among firms is a common practice that helps manufacturers and service providers to cope with fluctuations in demands, maintain their clients, and increase their sales; see Yu et al. (2015). These two applications allow for job splitting of the type discussed here (i.e., where the parts of a job can be processed simultaneously).

The modern hi-tech world requires analysis of vast amounts of data. To carry this out effectively, one must achieve the goals of maximizing throughput while minimizing latency and response time. Analyses of large amounts of data are employed in machine learning models as well as in comparing implementations of an algorithm and in verifying software and hardware implementations. In the last two decades, as a result of the development and spread of high-bandwidth networks, distributed computing systems have evolved, consisting of multiple computers (processors) working together in order to accomplish a single task. These systems offer the most effective way to reach the goals. The processors, each having its own memory, are connected by high-bandwidth networks and may therefore be located far away from one another, possibly in different continents, and they may be owned by different entities. Load-balancing algorithms determine the load allocation among the various processors in order to minimize the completion time of the tasks. Our model can be beneficial for pooling of resources in distributed computing systems. For example, pooling the processors of a multidivision hi-tech company or of a university with several departments will accelerate the accomplishment of tasks as described. For a review of load-balancing schemes for distributed computing systems, see Kushwaha and Gupta (2015). For the use of cooperative and noncooperative game theory in modeling load-balancing problems in distributed computing systems, see Penmatsa and Chronopoulos (2011) and the references therein.

The formation of a partnership among companies or other entities necessitates not only a plan for joining forces toward the achievement of the ultimate goals but also an agreement regarding the revenue, cost, or load allocation among the partners. Here is where the theory of cooperative games can be helpful as it provides a number of revenue and cost allocation schemes that guarantee the stability of the partnership or satisfy some fairness properties that players would like to have. The decision regarding the allocation scheme to be used must be acceptable by all players as otherwise, the alliance is put at risk.

A cooperative game is defined by a set of players and a *characteristic (coalitional) function* that returns a cost for any coalition of players. The main tasks while analyzing a cooperative game are (i) to predict the formation of coalitions and then, for any possible coalition, (ii) to propose an allocation of its total cost among its members. If it turns out that all players cooperate, then a single coalition called the *grand coalition*, which contains all the players, is formed. In such a case, the next question that arises is how to allocate the total cost among the players. Several cost allocation mechanisms have been proposed in the literature, such that either fairness or stability of the cooperation is achieved. Here, we focus on the *core*, a notion proposed by Gillies (1953); the core guarantees the stability of the grand coalition by requiring that (i) the total cost incurred by the grand coalition is allocated among all the players and (ii) the total cost allocated to the members of any proper subcoalition does not exceed the cost that the subcoalition would have paid if it defected from the grand coalition. Thus, the set of core cost allocation vectors is defined by 2^n linear constraints, where n is the number of players in the grand coalition. Researchers usually face a real challenge to (partially) characterize the core or even to prove that the core is nonempty, in view of the exponential number of constraints that define the core.

In this paper, we consider the cooperative game of a parallel machines scheduling (PMS) problem under *job splitting*, where the characteristic function is to minimize the scheduling cost, which is proportional to the sum of completion times of the jobs. The game is defined by a set of players, where each player is a manufacturer that owns a nonempty set of machines. All machines have the same capabilities, although their speeds may be machine dependent. Each manufacturer is also associated with a possibly empty set of jobs (i.e., orders submitted to him by clients). All machines are capable of producing any job. The jobs differ in their *processing requirement* time. The jobs can be partitioned into any number of disjoint segments that can be produced on different machines at any order and even simultaneously. Such jobs are called *splitting jobs*, and the respective PMS problem is said to allow for job splitting. There are interesting applications for splitting jobs in the industry, where the procedure of splitting a job is called *lot sizing* and the split parts are also called *sublots*; see Potts and Van Wassenhove (1992), which considers the case where a job consists of many identical items. An application of scheduling looms in the textile industry where splitting jobs is allowed is described in Serafini (1996). The most prevalent application of job splitting is in the hi-tech industry in distributed data processing, which is discussed above.

We prove that in the PMS under the job-splitting game, it is most probable that all manufacturers will join forces and form the grand coalition and that the core of the game and the core of any of its subgames are all nonempty. If the management of the cooperation allocates the total scheduling cost among the manufacturers by using any core cost allocation, then it is most probable that the stability of the cooperation will not be put at risk as no coalition of manufacturers is able to gain by defecting from the grand coalition. The only assumption that is needed in our proof regarding the problem's parameters is that the speeds of all the machines are positive rational numbers.

The outline of the paper is as follows. Section 2 presents the relevant literature. Section 3 introduces some notations and preliminaries. Section 4 defines the PMS under the job-splitting game. Section 5 presents the methodology of proving the total balancedness of the PMS game in a number of subsections, where each is devoted to a specific step. The general idea behind our proof is to split the players of the game into *job players*, resulting in a *constrained PMS game* in which each job is a player, called a *job player*, where only certain coalitions of job players are feasible. Based on the constrained PMS game, we generate as many constrained *basic PMS games* as the number of job players, whose processing requirement vectors are linearly independent zero-one vectors. The constrained PMS game is then shown to be equal to a nonnegative linear combination of the constrained basic PMS games. Thereafter, each job player of any constrained basic PMS game is split into a number of unit-speed job players, each having a processing requirement of zero or one, resulting in a respective unit-speed constrained basic game whose only feasible coalitions are those that consist of all descendants of feasible coalitions of job players in the constrained PMS game. We show that even under the tighter assumption where all coalitions of unit-speed job players in any unit-speed constrained basic game are feasible, the symmetric core of the game is infinitely large. In fact, we identify a closed form nondegenerate line segment in the symmetric core of any unit-speed constrained basic game and therefore, also in the constrained basic game to which it is associated. By working backward, using the presentation of the constrained PMS game as a nonnegative linear combination of constrained basic games, we derive a subset of the symmetric core of the constrained PMS game. We complete the proof by merging back the job players of the constrained PMS game into the players of the PMS game while generating a subset of the symmetric core of the PMS game. Section 6 concludes the paper, and it proposes some future research directions.

2. Literature Review

There is a vast literature on scheduling; see Pinedo (2016) for a state-of-the-art review on the subject. The literature on cooperative scheduling games is divided into three types. The first type is called *permutation games*; see Curiel (2010, chapter 3 and the references therein). A permutation game is defined by n jobs, which need to be processed on a single machine, where all jobs share the same processing time. The underlying assumption in such games is that given a certain processing sequence of the jobs, the cost of processing job i, $1 \leq i \leq n$, depends only on its location in the sequence, and the total cost is additive in the jobs. Thus, the data of such a game consist of the cost parameters, which are given by a square nonnegative matrix with n rows; see Tijs et al. (1984). Another game that is closely related to permutation games is the *assignment game*, where the data matrix is not necessarily a square matrix (Shapley and Shubik 1972).

The second type of scheduling games is called *sequencing games*; see Curiel (2010, chapter 4 and the references therein). The class of sequencing games generalizes the class of permutation games by allowing the jobs to be associated with job-dependent processing times, and the cost of each job depends on its completion time, which consists of its waiting and service times. This stream of research was initiated in Curiel et al. (1989), which considers a game where each player has a single job and a player-dependent linear cost function of its completion time. Given an initial sequence of the jobs, the profit of any coalition is defined as the maximum possible savings that the coalition can get by reordering the jobs in the *connected* parts of the coalition, implying that the completion times of the jobs that are not members of the coalition are not affected. Reordering the jobs is done iteratively, where at each iteration, two consecutive jobs in a connected part of the coalition are switched. The gain achieved by any such switch is equally divided between the two players that own the jobs that were switched. Accordingly, the algorithm is called the *equal gain splitting* (EGS) rule. For a generalization of this paper to general additive weakly increasing cost functions, see Curiel et al. (1994). Other sequencing games vary by their characteristic function; the set of requirements regarding the jobs, like *ready times* and *due dates*; and/or allowed actions while forming coalitions. For more details, see Curiel et al. (2002). Although the vast majority of papers on sequencing games deal with single-machine models, there are also some papers that consider the parallel machines case; see Hamers et al. (1999) and Slikker (2006).

The third type of scheduling games is called *cooperative PMS games*, where players are manufacturers that own machines. In addition, each manufacturer is committed to produce a (possibly empty) set of jobs. The machines of all manufacturers have the same capabilities (i.e., each can produce any job), and they differ only in their speeds. PMS games allow the manufacturers to collaborate in order to lower their production cost. Such games differ in their characteristic function and the type of jobs. In this paper, we consider the characteristic function that returns, for any coalition of manufacturers, the minimum sum of completion times of their jobs by using their machines. The jobs are assumed to fulfill the *job-splitting* property (i.e., they can be split into any number of disjoint segments that can be allocated to the machines in any possible order and even simultaneously on a number of machines). For polynomially solvable PMS problems allowing job splitting, see Xing and Zhang (2000) and Tahar et al. (2006).

Another PMS game that has been analyzed is the one where the characteristic function minimizes the *makespan*. The makespan of a schedule that starts at time 0 is the point of time when the schedule ends. In the single-machine case, the optimal makespan is independent of the processing order of the segments of the jobs in contrast to the case where the objective function minimizes the sum of the jobs' completion times, where the cost is heavily order dependent, even in the single-machine case. Although the PMS game under the makespan where job splitting is allowed was not directly mentioned in the literature, its solution can be deduced from another game, which has the same form, namely the $M/M/1$ queueing game analyzed in Anily and Haviv (2010). Therefore, we conclude that the PMS game under makespan belongs to the class of *centralizing aggregation games*, which has been proved in Anily (2018) to have a nonempty core whose nonnegative part is fully characterized. The PMS game under makespan, where the property of job splitting is replaced by the more restrictive property of *preemptive jobs* (i.e., it allows for job preemptions but not for simultaneous processing of a job on several machines), is also considered in Anily (2018). The paper proves that this game, although is not a centralizing aggregation game, has a nonempty core, and a polytope, which is a subset of its nonnegative core, is fully characterized. To the best of our knowledge, the two versions of the PMS makespan game mentioned above are the only cooperative PMS games that have been analyzed. Yet, the properties of the core of each of these games are different from the properties proved here for the PMS under the job-splitting game, where the characteristic function minimizes the sum of completions times of the jobs. Although a subset of the nonnegative core is fully characterized for the PMS games under the makespan, the nonnegative core of the PMS under the job-splitting game analyzed here might be empty, as will be shown in Example 4. This difference calls for a different type of analysis of the current game.

Interestingly, the property of job splitting also attracts the attention of researchers from the perspective of noncooperative games, where no central controller that has complete information on all the players exists and issues of incentive incompatibility may arise; consider a parallel multiprocessor computing system, where the schedule prioritizes jobs according to a monotone order of a certain property, like the remaining processing time of the jobs. The players, in such a game, may manipulate the system by splitting, merging, or partially transferring some of their jobs to other players, disabling the central controller monitoring the identity of the players. See Moulin (2007, 2008) in the context of PMS with job splitting.

3. Notations and Preliminaries

This section starts by presenting the PMS under the job-splitting model, hereafter called the *PMS game*. We summarize some concepts and preliminaries of the theory of cooperative games and add a new definition that will be helpful in the sequel.

The input of the PMS problem consists of (i) a given set of identical machines, which may differ only in their speeds; and (ii) a set of jobs, where each job can be processed by any machine. The jobs are defined by their *processing requirement*: namely, their processing time on a unit-speed machine. In addition, we assume that *job splitting* is allowed (i.e., the jobs can be partitioned and even processed simultaneously on different machines). The objective function is to minimize the scheduling cost, which is proportional to the sum of completion times of the jobs on the machines. Thus, in the sequel, we refer to the objective function of minimizing the sum of completion times of the jobs on the machines. We only consider the contribution of the machines because of their speed in achieving the ultimate goal of minimizing the sum of the completion times of the jobs.

The common notation for PMS problems, proposed in Graham et al. (1979), classifies problems by triplets of a three-field notation $\alpha|\beta|\gamma$, where (a) $\alpha \in \{P, Q, R\}$ defines the machines' environment, where $\alpha = P$ refers to identical machines, $\alpha = Q$ refers to the more general case of uniform machines where the machines are identical except for their speeds, and $\alpha = R$ refers to the most general case of unrelated machines. (b) β describes the jobs characteristics as, for example, $\beta = prmp$ if preemption is allowed. In Xing and Zhang (2000), $\beta = split$ is proposed for job splitting. (c) γ refers to the objective function. Some of the most common objective criteria include C_{max} for the makespan and $\sum_j C_j$ ($\sum_j w_j C_j$) for the (weighted) sum of completion times. Thus, the problem considered here is denoted by $Q|split|\sum_j C_j$. Let m be the number of machines and n be the number of jobs. In Xing and Zhang (2000), it is proven that for the $Q|split|\sum_j C_j$ problem, there exists an optimal schedule where each job is partitioned into m split parts that are processed simultaneously on all the machines. Therefore, the $Q|split|\sum_j C_j$ problem is reducible to a single-machine problem, namely $1|\cdot|\sum_j C_j$, where the speed of the machine is the sum of the speeds of the original machines, denoted by v, and the *processing time* on this machine of a job whose processing requirement is p is $\frac{p}{v}$.

Cooperative games with transferable utilities are coalitional games defined by a pair (N, G), where $N = \{1, \dots, n\}$ is a set of n *players* and the *characteristic function* $G: 2^N \to \Re$, is a set function that for any coalition $\emptyset \subseteq S \subseteq N$, returns a real number $G(S)$, where $G(\emptyset) = 0$. We refer to $G(S)$ as the cost of a set of players $S \subseteq N$ if its members cooperate and form a coalition. The cost imposed on a coalition is independent of what the players in $N \setminus S$ are doing. The coalition $S = N$ is called the *grand coalition*. A subgame (S, G) of a game (N, G), for any $S \subset N$, is the cooperative game whose set of players is S, and its characteristic function is the set function G reduced to all subsets of S. A game is called *monotone* if $G(S) \le G(T)$ for any $S \subseteq T \subseteq N$. Under any partition of the grand coalition into disjoint sets $S_1, \dots, S_K$, the total cost of the game is $\sum_{\ell=1}^{K} G(S_\ell)$, meaning that the total cost is additive in the coalitional structure. A necessary condition for all players of N to cooperate and form the grand coalition is subadditivity of the game. A game (N, G) is *subadditive* if and only if the characteristic function G is *subadditive*; that is, for any two disjoint coalitions $S, T \subset N$, $G(S \cup T) \le G(S) + G(T)$. Subadditivity implies that $G(N) \le \sum_{\ell=1}^{K} G(S_\ell)$ for any partition of N into disjoint coalitions $\{S_1, \dots, S_k\}$, $k \ge 1$, meaning that the grand coalition is an optimal formation of a coalitional structure.

If the grand coalition is formed, the players start bargaining for a fair cost allocation scheme of the total cost $G(N)$. Let $\hat{x} = (x_1, \dots, x_n) \in \Re^n$ be a *cost allocation* *vector* where x_i, $i \in N$, is the cost allocated to player i. The *efficiency* condition, namely $\sum_{i=1}^{n} x_i = G(N)$, is preliminary for a cost allocation vector. As mentioned in Section 1, we focus here on the *core* of the game denoted by $\mathcal{C}(N,G)$. The core, a notion attributed to Gillies (1953), consists of all efficient cost allocation vectors $\hat{x} = (x_1, \ldots, x_n)$ that satisfy the additional $2^n - 1$ *coalitional rationality* conditions of the form, $\sum_{i \in S} x_i \le G(S)$, one for each proper subcoalition $S \subsetneq N$. As mentioned in Section 1, any cost allocation that satisfies this set of conditions suggests stability as no subset of players can reduce its cost by leaving the grand coalition. A cooperative game whose core is nonempty is said to be *balanced*, and if its core and the cores of all its subgames are nonempty, the game is *totally balanced*.

Except for the notion of job splitting, taken from the theory on PMS, we use in this paper a different type of splitting related to cooperative games, where each player is split into (at least one) subplayers. The game generated by the subplayers is defined on a subset of all the coalitions of subplayers called *feasible coalitions*, namely those coalitions of subplayers that consist of all the descendants of a certain coalition of players in the original game. A game that is defined on a subset of all possible coalitions of its grand coalition is called a *constrained game*. The grand coalition of the constrained game is a feasible coalition by definition. As we are going to see, splitting players of a PMS game may result in a more tractable game than the original one.

Definition 1. Given a cooperative game (N, G) with n players, the game $(\tilde{N}, \tilde{G})$ is said to be generated from the game (N, G) by splitting players of N if we have the following.

- There is a splitting scheme Π that maps N into $\tilde{N}$, which satisfies the following properties. (i) For any $i \in N$, $\Pi(i) \neq \emptyset$, (ii) for any $i, j \in N$, $i \neq j$, $\Pi(i) \cap \Pi(j) = \emptyset$, and (iii) $\cup_{i \in N} \Pi(i) = \tilde{N}$.
- For any coalition $S \subseteq N$, it holds that $G(S) = \tilde{G}(\cup_{i \in S} \Pi(i))$.

Except for the degenerate case where each player of the game (N, G) is split into a single subplayer, not all coalitions of the subplayers' game $(\tilde{N}, \tilde{G})$ correspond to a coalition of players in the game (N, G). We call a coalition of the game $(\tilde{N}, \tilde{G})$ a feasible coalition if it is a collection of all subplayers that are descendants of a certain coalition of players in the original game.

Definition 2. Let (N, G) be a cooperative game, and let $(\tilde{N}, \tilde{G})$ be a game that is generated from the game (N, G) by splitting players using a splitting scheme Π. For any coalition $S \subseteq N$, the set of subplayers $\cup_{i \in S} \Pi(i)$ is a feasible coalition of the game $(\tilde{N}, \tilde{G})$. Denote the set of all feasible coalitions of the game $(\tilde{N}, \tilde{G})$ by $\mathcal{C}$. Let the game $(\tilde{N}, \mathcal{C}, \tilde{G})$ be the *constrained game* of the game $(\tilde{N}, \tilde{G})$, to feasible coalitions of $\tilde{N}$.

The following definition follows naturally from Definition 2.

Definition 3. The constrained game $(\tilde{N}, \mathcal{C}, \tilde{G})$ is balanced if and only if the set of cost allocation vectors of the game $(\tilde{N}, \tilde{G})$ that satisfy the efficiency constraint and the coalitional rationality constraints for all feasible coalitions, that is, all coalitions of $\mathcal{C}$, is nonempty.

Example 1. Let (N, G) be a cooperative game with two players, $N = \{a, b\}$. Suppose that player a is split into two players: a_1 and a_2. Let $\tilde{N} = \{a_1, a_2, b\}$. The constrained game $(\tilde{N}, \mathcal{C}, \tilde{G})$ is associated with three feasible coalitions in $\mathcal{C}$, namely $\{a_1, a_2\}, \{a_1, a_2, b\}$, and $\{b\}$, which correspond to the three coalitions of the game (N, G), whereas the additional four coalitions of the game $(\tilde{N}, \tilde{G})$, namely $\{a_1\}, \{a_2\}, \{a_1, b\}$, and $\{a_2, b\}$, are infeasible coalitions of $(\tilde{N}, \mathcal{C}, \tilde{G})$.

The following observation follows directly from Definition 3.

Observation 1. If the game $(\tilde{N}, \tilde{G})$ is totally balanced, then the constrained game $(\tilde{N}, \mathcal{C}, \tilde{G})$ is also totally balanced, and as the constrained game $(\tilde{N}, \mathcal{C}, \tilde{G})$ is equivalent to the game (N, G), the game (N, G) is also totally balanced.

The literature describes a few classes of games that have been proven to be totally balanced. The most structured class of games that allow for a full characterization of the core is the class of *concave games*.

Definition 4. A game (N, G) is concave if its characteristic function is concave (i.e., for any two coalitions $S \subset T \subset N$ and $i \in N \setminus T$, $G(S \cup \{i\}) - G(S) \ge G(T \cup \{i\}) - G(T)$).

Concave games are subadditive but not the other way around. It is shown in Shapley (1971) that the core of a concave game possesses $n!$ extreme points, each of which being the marginal contribution vector of the players for one of the $n!$ permutations of the players. A few cooperative games in operations management have been proven to be concave; see, for example, Anily and Haviv (2007) and its generalization in Zhang (2009) that considers joint replenishment models of one warehouse and several retailers. However, the $Q|split|\sum_j C_j$ game considered here is not concave, as will be proven in Section 4.

4. The PMS Under the Job-Splitting Cooperative Game

Let $N = \{1, \ldots, n\}$, $n \ge 2$, be a set of players. Each player $i \in N$ is associated with (1) a set $J(\{i\})$ of $\psi(\{i\}) \ge 1$ jobs and (2) a nonempty set $M(\{i\})$ of machines whose total speed is positive. We further assume that for each player $i \in N$, the speed of her machines in $M(\{i\})$ is a positive rational number, implying that the total speed of the machines in $M(\{i\})$, denoted by $v(\{i\})$, satisfies $v(\{i\}) \in \mathcal{Q}_{++}$. For a coalition $S \subseteq N$, let $J(S)$, $\psi(S)$, and $M(S)$ be the set of jobs, the number of jobs, and the set of machines of the players of S correspondingly, and let $v(S)$ be the total speed of the machines in $M(S)$. As discussed in Section 3, the number of machines in $M(\{i\})$ is immaterial (i.e., we can assume, without loss of generality (w.l.o.g.), that each player $i \in N$ is associated with a single machine or alternatively, with $\psi(\{i\})$ machines; i.e., one machine for each job). Note that our model allows the machines to also be associated with rational operating costs. Let $\hat{oc}(\{i\}) \in \mathcal{Q}_{++}$ be the operating cost of the machines of player i. In such a case, we redefine the speed of the machines of player i to be $v(\{i\})/\hat{oc}(\{i\})$. Therefore, in the sequel, we do not consider the machines' operating costs explicitly.

Any job $j \in J(N)$ belongs to a certain player in N, called its *father*, and therefore, is denoted by $f(j) \in N$. In addition, each job $j \in J(N)$ is associated with its *processing requirement*, denoted by $p_j \geq 0$, which is the processing time duration of the job on a unit-speed machine ($v = 1$). Players are allowed to have no jobs. In such a case, we say that the player has a single job whose processing requirement is zero, called an *empty job*. Other players, namely players that have real jobs, are not allowed to have empty jobs. Note that the players that have no jobs are the most valuable in the game as they contribute their resource (machines) for processing of the jobs of the other players without consuming any resources as they do not have jobs to process. As mentioned in Section 3, under the job-splitting property, there exists an optimal schedule where each job is partitioned into as many split parts as the number of available machines, and these parts are processed simultaneously on all the machines. Therefore, the *processing time* of a job $j \in J(N)$ on the machines of coalition S is $p_j/v(S)$.

The jobs in the set $J(N)$ are indexed from one up to $\psi(N)$ in a nondecreasing order of their processing requirements. Let $(p_1, \ldots, p_{\psi(N)})$ be the processing requirement vector of the jobs of $J(N)$, where ties are broken arbitrarily. Let $p_0 = 0$. Note that the order of the jobs of $J(N)$ is preserved in all subsets of $J(N)$, and that is, if the index of job j precedes the index of job k in $J(N)$, then this will be the case in all subsets of $J(N)$ that contain both jobs j and k. Accordingly, the jobs of $J(S)$, for any coalition $S \subseteq N$, are indexed from one up to $\psi(S)$. In particular, let p_k^S be the processing requirement of the kth job of $J(S)$. Let j_k^S, $0 \leq k \leq \psi(S)$, be the index in $J(N)$ of the kth job of $J(S)$, where $j_0^S = 0$. Thus, the processing requirement p_k^S of the kth job in $J(S)$, $1 \leq k \leq \psi(S)$, is equal to $p_{j_k^S}$, where $1 \leq k \leq j_k^S \leq \psi(N) - (|S| - k)$. As a consequence, the sequence $(p_j)_{j=1}^{\psi(N)}$ coincides with the sequence $(p_j^N)_{j=1}^{\psi(N)}$.

According to the *shortest processing time* (*SPT*) rule (see Smith 1956), the minimum sum of completion times of jobs on a single machine is achieved by processing the jobs in a nondecreasing order of their processing requirements. Let $P : 2^N \to \Re$ be the set function that returns, for any coalition of players $S \subseteq N$, the minimum sum of completion times of the jobs of $J(S)$ on the machines in the set $M(S)$. Following the results in Xing and Zhang (2000) and as explained, $P(S)$ is achieved by splitting up the jobs of $J(S)$ into $|M(S)|$ split parts that are processed simultaneously, according to the SPT rule, on all the machines in the set $M(S)$, whose total speed is $v(S)$.

In Section 4.1, we present a few properties of the PMS game (N, P) that are based on the optimal solution of the scheduling problem $Q|split|\sum_j C_j$. In Section 4.2, we first present the characteristic function P of the PMS game that follows directly from the structure of the optimal solution of the PMS $Q|split|\sum_j C_j$ problem. Then, we present an equivalent formulation of the PMS game as a constrained PMS game whose players are the jobs of $J(N)$, under a set of feasible coalitions of $J(N)$.

4.1. Some Properties of the PMS Game (*N*, *P*)

Claim 1. The PMS game (N, P) is subadditive.

Proof. In order to prove the subadditivity of the PMS game (N, P), we need to show that for any two disjoint coalitions of players $S, T \subset N$, $P(S \cup T) \leq P(S) + P(T)$. Note that $P(S \cup T)$ is the solution of a minimization problem of the sum of completion times of the jobs in $J(S \cup T)$ by the machines in $M(S \cup T)$. The solution of $P(S) + P(T)$, on the other hand, is achieved by a schedule of the jobs $J(S \cup T)$ on the machines of $M(S \cup T)$ under the restriction that the jobs of S and T are assigned according to the set they belong to, so that the jobs of $J(S)$ are processed by the machines of $M(S)$ and the jobs of $J(T)$ are processed by the machines of $M(T)$. Thus, the cost $P(S) + P(T)$ is the cost of optimally scheduling $J(S)$ on $M(S)$ and $J(T)$ on $M(T)$, implying that it is the cost of a feasible but not necessarily optimal schedule for the minimization problem that $P(S \cup T)$ is its solution, implying that $P(S \cup T) \leq P(S) + P(T)$. □

The next two examples show that the PMS game is neither monotone nor concave.

Example 2. Consider the instance $N = \{1, 2\}, v(\{1\}) = 2$, $v(\{2\}) = 1$, $J(\{1\}) = \{1\}$, $J(\{2\}) = \{2\}$, $p_1 = 1, p_2 = 2$. Thus, $P(\{1\}) = \frac{1}{2}, P(\{1, 2\}) = \frac{4}{3}, P(\{2\}) = 2$, implying that $P(\{1\}) < P(\{1, 2\}) < P(\{2\})$, and therefore, the PMS game is not monotone.

Example 3. Consider the instance $N = \{1, 2, 3\}$, $v(\{i\}) = 1$, and $J(\{i\}) = \{i\}$ for $i \in N$, $p_1 = p_2 = 1$, and $p_3 = 2$. Let $S = \{3\}, T = \{2, 3\}$, and $i = 1$. Then, $P(S \cup \{i\}) - P(S) = \frac{4}{2}$

$-2 = 0 < \frac{1}{3} = \frac{7}{3} - \frac{4}{2} = P(T \cup \{i\}) - P(T)$, proving that the PMS game (N, P) is not concave.

There exist several cooperative games that were proved to be totally balanced by identifying a core allocation where each player is assigned a simple nonnegative function of its parameters. Such a cost allocation, by definition, is symmetric (i.e., any two identical players will be assigned exactly the same cost). For example, in Anily and Haviv (2010), each server is associated with an $M/M/1$ queueing system defined by its service rate and its arrival rate. The servers can form coalitions by pooling their service capacities to serve the union of the respective individual streams of customers. The characteristic function value of any coalition is its steady-state mean number of customers in the pooled system. The first observation of the paper is that a simple symmetric cost allocation vector that assigns each server the system's steady-state number of customers of that server when all servers cooperate and form the grand coalition is in the core. We note that the core of the queueing game may also contain allocation vectors with some (but not all) negative entries, but its nonnegative core (i.e., the part of the core that consists of nonnegative vectors) is nonempty. Similarly, the nonnegative *Bird* cost allocation for the minimum spanning tree game (see Bird 1976) and the allocation provided for pooling risk games in Alon and Haviv (2020) are simple nonnegative core allocations for the associated games. We emphasize that there is nothing special about nonnegative core allocations, as identifying any cost allocation in the core proves that the game is totally balanced. As shown by the next example, for the PMS game, the set of nonnegative core allocations, hereafter called the *nonnegative core*, might be empty, implying that there exist instances of the PMS game in which any core cost allocation contains negative entries, meaning that some players will be paid by other players. One of the reasons for this phenomenon is that a player that owns a speedy machine but her jobs are relatively short might be valuable as other players may be interested in cooperating with her in order to reduce the cost of their coalition. In such a case, some players may be ready to pay such "valuable players" in order to persuade them to join their coalition.

Example 4. Consider the following instance of the PMS game (N,P): $N = \{1,2,3\}, v(\{1\}) = 10, v(\{2\}) = v(\{3\}) = 1$, $J(\{i\}) = \{i\}$ for $i = 1, 2, 3$, and $p_1 = 1, p_2 = p_3 = 10$, implying the following characteristic function values: $P(\{1\}) = \frac{1}{10}$, $P(\{2\}) = P(\{3\}) = 10$, $P(\{1,2\}) = P(\{1,3\}) = \frac{12}{11}, P(\{2,3\}) = 15$, and $P(\{1, 2,3\}) = 2.75$. The vector $\vec{x} = (x_1, x_2, x_3) = (-1.25, 2, 2)$ satisfies the efficiency and all the coalitional rationality conditions, implying that the instance is balanced. We show that the nonnegative core allocation set of this instance is empty. Suppose by contradiction that $\vec{x} = (x_1, x_2, x_3)$ is a nonnegative core allocation. As $x_1 \geq 0$, the coalitional rationality constraint for coalition $\{1,2\}$, namely $x_1 + x_2 \leq \frac{12}{11}$, implies that $x_2 \leq \frac{12}{11}$. By symmetry, also $x_3 \leq \frac{12}{11}$. By summing up the last two inequalities with the stand-alone condition $x_1 \leq 0.1$, we obtain that $x_1 + x_2 + x_3 \leq \frac{251}{110} < 2.282 < P(N)$, contradicting the efficiency constraint. Thus, the nonnegative part of the core of this instance is empty.

4.2. The PMS Game (N, P) and the Constrained PMS Game $(J(N), \mathcal{C}, F)$

The well-known cost set function $P : 2^N \to \Re$ of the scheduling problem $Q|split|\sum_j C_j$ proposed in Smith (1956) is given by

$$P(S) = \frac{1}{v(S)} \sum_{k=1}^{\psi(S)} (\psi(S) - k + 1) p_k^S$$
$$= \frac{1}{v(S)} \sum_{k=1}^{\psi(S)} (\psi(S) - k + 1) p_{j_k^S}. \qquad (1)$$

The PMS game (N, P) whose characteristic function is given in (1) can also be considered as a constrained PMS game $(J(N), \mathcal{C}, F)$ whose players are the jobs of $J(N)$, now called *job players*. As described, the players of N are split (see Definition 1) into job players in the set $J(N)$. The set $\mathcal{C}$ contains all *feasible coalitions* $J(S) \subseteq J(N)$, namely coalitions of job players that are descendants of a coalition of players $S \subseteq N$. More specifically, each job player is an offspring of a certain player $i \in N$; the feasible coalitions contain all the offspring of the players in some coalition $S \subseteq N$. Each job player $j \in J(N)$ is associated with its processing requirement $p_j \geq 0$. As mentioned at the beginning of this section, players that do not have jobs are assumed to have a single empty job. In order to complete the definition of the constrained PMS game, we need to associate a speed with the (virtual) machine of each job player. Recall that the speed of the machine of any player $i \in N$ is a positive rational number. We assume w.l.o.g. that in the constrained PMS game $(J(N), \mathcal{C}, F)$, each of the $\psi(\{i\})$ job players in $J(\{i\})$, $i \in N$, owns a machine whose speed is $v(\{i\})/\psi(\{i\}) \in \mathcal{Q}_{++}$, namely the speed $v(\{i\})$ of the machine of player i, $i \in N$, is equally allocated among its $\psi(\{i\})$ job players in $J(\{i\})$, also implying that the speeds of the machines of the job players are rational numbers.

The characteristic function satisfies the following equation for any coalition $S \subseteq N$:

$$F(J(S)) = P(S).$$

Note that all properties of the PMS game (N, P) mentioned in Section 4.1 continue to hold for the constrained PMS game $(J(N), \mathcal{C}, F)$. According to Observation 1, if the

PMS game $(J(N), F)$ is totally balanced, then the constrained PMS game $(J(N), \mathcal{C}, F)$ is also totally balanced.

For the rest of the analysis, we assume that the speeds of the machines of all the job players are natural numbers. This assumption is w.l.o.g. as it can be achieved by rescaling the time unit.

Assumption 1. *The speeds of the machines of all job players in* $J(N)$, *that is, the ratios* $v(\{i\})/\psi(\{i\})$, *for* $i \in N$, *are natural numbers.*

Next, we present an alternative formulation of the characteristic function F for any feasible coalition of job players as a function of the nonnegative marginal increment vector of the processing requirement vector $(p_1, \ldots, p_{\psi(N)})$. For this sake, let $\Delta_j = p_j - p_{j-1} \geq 0$ be the marginal increment of the processing requirement of job player $j \in J(N)$, implying that $p_j = \sum_{\ell=1}^{j} \Delta_\ell$. Recall that for a given feasible coalition $J(S) \in \mathcal{C}$ and the ℓth job player in $J(S)$, $1 \leq \ell \leq \psi(S)$, the index j_ℓ^S returns the index of that job player in $J(N)$. In addition, let Δ_ℓ^S be the marginal increment of the processing requirement of job player j_ℓ^S with respect to coalition $J(S)$, where $j_0^S = 0$, implying that

$$\Delta_\ell^S = p_{j_\ell^S} - p_{j_{\ell-1}^S} = \sum_{t=j_{\ell-1}^S+1}^{j_\ell^S} \Delta_t. \tag{2}$$

Thus,

$$\begin{aligned} P(S) = F(J(S)) &= \sum_{k=1}^{\psi(S)} C_{j_k^S} = \frac{1}{v(S)} \sum_{k=1}^{\psi(S)} \sum_{\ell=1}^{k} p_{j_\ell^S} \\ &= \frac{1}{v(S)} \sum_{k=1}^{\psi(S)} \sum_{\ell=1}^{k} \sum_{t=1}^{\ell} \Delta_t^S \\ &= \frac{1}{v(S)} \sum_{k=1}^{\psi(S)} \sum_{\ell=1}^{k} (k-\ell+1)\Delta_\ell^S \\ &= \frac{1}{v(S)} \sum_{\ell=1}^{\psi(S)} \sum_{k=\ell}^{\psi(S)} (k-\ell+1)\Delta_\ell^S = \frac{1}{v(S)} \sum_{\ell=1}^{\psi(S)} \Delta_\ell^S \sum_{k=1}^{\psi(S)-\ell+1} k \\ &= \frac{1}{v(S)} \sum_{\ell=1}^{\psi(S)} \frac{(\psi(S)+1-\ell)(\psi(S)+2-\ell)}{2} \Delta_\ell^S. \end{aligned} \tag{3}$$

In order to gain some insight into (3), note that the ℓth marginal increment of the processing requirement of job player $\ell \in \{1, \ldots, \psi(S)\}$, namely Δ_ℓ^S, should be summed up while considering the completion time of the last $\psi(S) - \ell + 1$ job players of S: that is, job players $k \in \{j_\ell^S, \ldots, j_{\psi(S)}^S\} \subseteq J(N)$. Therefore, when considering the completion time of any job player $k \in \{j_\ell^S, \ldots, j_{\psi(S)}^S\}$, Δ_ℓ^S is taken into account $k - \ell + 1$ times. Thus, in total, Δ_ℓ^S should be counted $1 + 2 + \cdots + (\psi(S) - \ell + 1) = 0.5((\psi(S) - \ell + 2)(\psi(S) - \ell + 1))$ times in (3).

For any feasible coalition $J(S) = \{j_1^S, \ldots, j_{\psi(S)}^S\} \in \mathcal{C}$, and for any job player $j \in J(N)$, let $\ell^S(j) = \{\ell : j_{\ell-1}^S \leq j < j_\ell^S\}$ be the smallest indexed job in $J(S)$ whose index in $J(N)$ is at least as large as j. If $j > j_{\psi(S)}^S$, then $\ell^S(j) \overset{\text{def}}{=} \psi(S) + 1$. In addition, let

$$K_{J(S)}(j) = \psi(S) + 1 - \ell^S(j). \tag{4}$$

Note that $K_{J(S)}(j)$ is the number of job players of $J(S)$ whose index in $J(N)$ is at least as large as j.

Lemma 1 presents an alternative expression of the characteristic function value of any feasible coalition $J(S) \in \mathcal{C}$ given in (3) by using the values $K_{J(S)}(j)$, for $j = 1, \ldots, \psi(N)$, defined in (4). This presentation will turn out to be helpful later.

Lemma 1. *The characteristic function value of any feasible coalition* $J(S) \in \mathcal{C}$ *in the constrained PMS game* $(J(N), \mathcal{C}, F)$ *is equal to the following nonnegative linear combination of the marginal increments of the processing requirements of the jobs in* $J(N)$:

$$P(S) = F(J(S)) = \sum_{\ell=1}^{\psi(S)} C_{j_\ell^S} = \sum_{j=1}^{\psi(N)} \Delta_j \frac{K_{J(S)}(j)(K_{J(S)}(j)+1)}{2v(S)}. \tag{5}$$

Proof. Define $j_0^S = 0$. Using (2) and (4) (recall that for $j > j_{\psi(S)}^S$, $K_{J(S)}(j) = 0$), we have

$$\begin{aligned} &\sum_{\ell=1}^{\psi(S)} (\psi(S)+1-\ell)(\psi(S)+2-\ell)\Delta_\ell^S \\ &= \sum_{\ell=1}^{\psi(S)} (\psi(S)+1-\ell)(\psi(S)+2-\ell) \sum_{t=j_{\ell-1}^S+1}^{j_\ell^S} \Delta_t \\ &= \sum_{\ell=1}^{\psi(S)} \sum_{t=j_{\ell-1}^S+1}^{j_\ell^S} (\psi(S)+1-\ell)(\psi(S)+2-\ell)\Delta_t \\ &= \sum_{j=1}^{j_{\psi(S)}^S} (\psi(S)+1-\ell^S(j))(\psi(S)+2-\ell^S(j))\Delta_j \\ &= \sum_{j=1}^{\psi(N)} \Delta_j K_{J(S)}(j)(K_{J(S)}(j)+1), \end{aligned}$$

and the proof is completed by (3). □

5. The PMS Game (N, P) Is Totally Balanced

The total balancedness proof of the PMS game is a constructive one. The proof consists of a few steps that generate a line segment within the symmetric core of the game as described.

1. In Section 5.1.1, we define $\psi(N)$ *constrained basic* (CB) *PMS games*, denoted by $(J(N), \mathcal{C}, F^{(j)})$, $j = 1 \ldots \psi(N)$,

that share the same set of $\psi(N)$ job players and set of feasible coalitions as in the constrained PMS game. The processing requirement vector of the constrained basic PMS game $(J(N), \mathcal{C}, F^{(j)})$, $j = 1 \ldots \psi(N)$, is the nondecreasing zero-one vector whose first $j-1$ entries are zero. The characteristic function value of the constrained PMS game $(J(N), \mathcal{C}, F)$ for any feasible coalition $J(S)$, namely $F(J(S))$, is proved to be a weighted sum of the characteristic function values $F^{(j)}(J(S))$ for $j = 1 \ldots \psi(N)$, where the corresponding weights are the marginal processing requirements $\Delta_j = p_j - p_{j-1}$ of the job players in $J(N)$.

2. In Section 5.1.2, based on Assumption 1, each job player $\ell \in J(N)$ of a constrained basic PMS game, $(J(N), \mathcal{C}, F^{(j)})$, $j = 1 \ldots \psi(N)$, is split into unit-speed job players. Consequently, we obtain $v(N)$ unit-speed job players, where similarly as in the constrained basic PMS game $(J(N), \mathcal{C}, F^{(j)})$, $\psi(N) - j + 1$ of them are assigned a unit processing requirement, whereas the others are assigned a zero processing requirement. Hence, we obtain $\psi(N)$ *unit-speed constrained basic PMS games*, called for short UCB PMS games, each having $v(N)$ unit-speed job players whose processing requirement is in {0, 1}, and their set of feasible coalitions consists of all the unit-speed job players that are descendants of a certain feasible coalition $J(S) \subseteq J(N)$ of the constrained PMS game. Yet, we consider a subset of the core of each UCB PMS game by assuming that all the $2^{v(N)}$ coalitions of the $v(N)$ unit-speed job players are feasible and show that this subset of the core is nonempty. We call the UCB PMS games where all coalitions of unit-speed players are feasible *unit-speed* {0, 1} *PMS games*.

3. In Section 5.2 and Theorem 2, we fully characterize the symmetric core of unit-speed {0, 1} PMS games, namely PMS games in which all job players have a unit-speed machine and their processing requirement is zero or one.

4. In Section 5.3, we wrap up the proof. Theorem 4, the key theorem of the paper, proves that the PMS game is totally balanced. Thereafter, by combining the symmetric cores of the $\psi(N)$ unit-speed {0, 1} PMS games according to the nonnegative linear combination of the constrained basic PMS games, mentioned in the first item of this description, we derive a line segment in $\Re^{\psi(N)}$, which is an infinitely large subset of the symmetric core of the constrained PMS game.

Figure 1 presents a tree diagram that shows the relations among the various games.

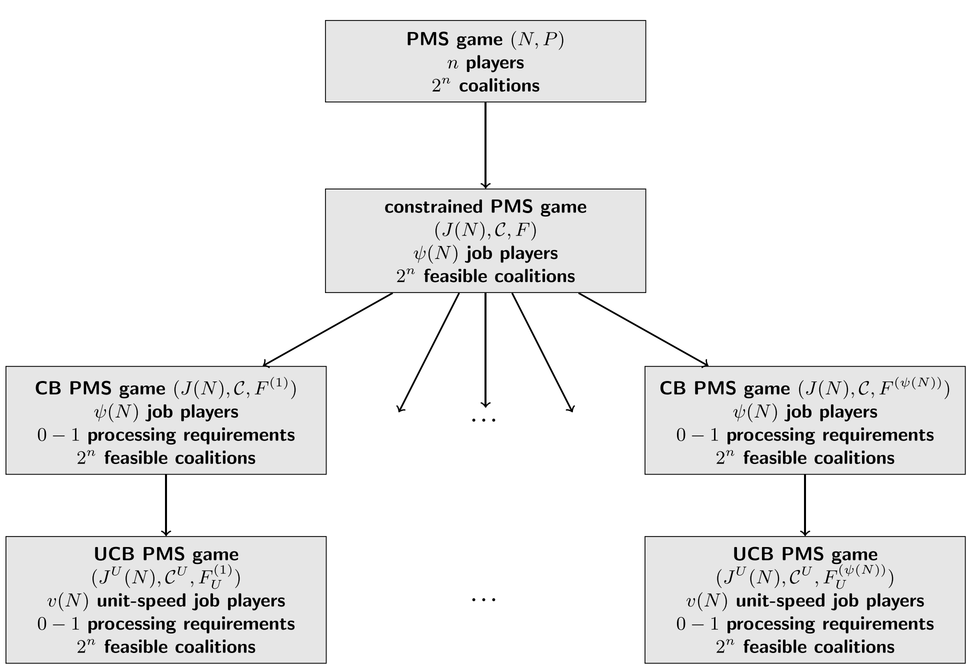
5.1. The Constrained Basic PMS Games

In this subsection, we present auxiliary PMS games that will enable us to prove the total balancedness of the PMS game.

5.1.1. The Constrained Basic PMS Games.

We start by presenting $\psi(N)$ games of a special form called *constrained basic (CB) PMS games*. The constrained PMS game $(J(N), \mathcal{C}, F)$ shares with the CB PMS games the same set of job players $J(N)$, the same speed of the machines, and the same set of feasible coalitions $\mathcal{C}$. In other words, the constrained PMS game and the $\psi(N)$ CB PMS games differ only in their processing requirement vectors. Let $(J(N), \mathcal{C}, F^{(j)})$, $j = 1 \ldots \psi(N)$, be the jth

Figure 1. A Tree Diagram That Represents the Relations Among the Various Games



CB game. Its processing requirement vector is the nondecreasing $0-1$ vector whose first $j-1$ entries are equal to zero, implying that its marginal processing requirement vector $\vec{u}^j$ is the jth unit vector in $\Re^{\psi(N)}$, namely the vector $\vec{u}^j_j = 1$, and $\vec{u}^j_k = 0$ for $k \neq j$. By using Lemma 1, we conclude that the characteristic function value of the CB PMS game $(J(N), \mathcal{C}, F^{(j)})$ for any feasible coalition, $J(S) \subset J(N)$, is given by

$$F^{(j)}(J(S)) = \sum_{k=1}^{\psi(N)} \vec{u}^j_k \frac{\mathrm{K}_{J(S)}(k)(\mathrm{K}_{J(S)}(k)+1)}{2v(S)} = \frac{\mathrm{K}_{J(S)}(j)(\mathrm{K}_{J(S)}(j)+1)}{2v(S)}. \quad (6)$$

By combining Equations (5) and (6), we get the following theorem.

Theorem 1. *The constrained PMS game* $(J(N), \mathcal{C}, F)$, *satisfies*

$$F(J(S)) = \sum_{j=1}^{\psi(N)} \Delta_j F^{(j)}(J(S)). \quad (7)$$

In view of Theorem 1, in order to prove the total balancedness of the constrained PMS game $(J(N), \mathcal{C}, F)$, it is sufficient to prove the total balancedness of the CB PMS games. This observation follows immediately from two properties that the core of a cooperative game satisfies; see Peleg and Sudhölter (2007, pp. 19–21). In order to simplify, we present the properties in their least general form that fits the needs of our proof. For this sake, let $\mathbb{C}(M, G)$ be the (possibly empty) core of a cooperative game (M, G). Then, (i) the core satisfies the *covariant under strategic equivalence* property; that is, if $G_2 = \alpha G_1$, for $\alpha > 0$, then $\mathbb{C}(M, G_1) = \alpha \mathbb{C}(M, G_2)$. (ii) The core satisfies the *superadditivity* property; that is, $\mathbb{C}(M, G_1) + \mathbb{C}(M, G_2) \subseteq \mathbb{C}(M, G_1 + G_2)$.

As it turns out, the CB PMS games $(J(N), \mathcal{C}, F^{(j)})$, $j = 1 \ldots \psi(N)$, are still too tricky to analyze. Yet, as the speed of the machine of each job player in $J(N)$ is a natural number, we further simplify the CB PMS games in Section 5.1.2 by splitting each job player into *unit-speed job players*.

5.1.2. The UCB PMS Games. In this subsection, we associate with each CB PMS game $(J(N), \mathcal{C}, F^{(j)})$, $j = 1 \ldots \psi(N)$, a *unit-speed constrained basic PMS game*, which is a CB PMS game where each job player has a unit-speed machine. For this sake, consider a certain CB PMS game $(J(N), \mathcal{C}, F^{(j)})$, $j = 1 \ldots \psi(N)$, and a job player $\ell \in J(N)$ whose father is player $i \in N$, and that is, $f(\ell) = i$. Recall that the speed of the machine of job player ℓ, denoted by $v'(\{\ell\}) = v(\{i\})/\psi(\{i\})$, is assumed to be a natural number; see Assumption 1. In order to get a UCB PMS game, we split each job player $\ell \in J(N)$ into $v'(\{\ell\})$ unit-speed job players, where one of them is associated with the same processing requirement (zero or one) as its father (i.e., job player $\ell \in J(N)$,) and the others are associated with a zero processing requirement. Let $J^U(N) \stackrel{\text{def}}{=} \{1, \ldots, v(N)\}$ be the resulting set of $v(N)$ unit-speed descendants of $J(N)$. The set of feasible coalitions of each of the $\psi(N)$ UCB PMS games is denoted by $\mathcal{C}^U$, where $|\mathcal{C}^U| = 2^n$, exactly as the number of coalitions in the original PMS game (N, P). In fact, any coalition $S \subseteq N$ is associated in each of the $\psi(N)$ CB PMS games with the coalition $J(S) \subseteq J(N)$, and $J(S)$, in turn, is associated with the coalition of unit-speed job players $J^U(S) \subseteq J^U(N)$ in the UCB PMS games. Applying this procedure on each CB PMS game $(J(N), \mathcal{C}, F^{(j)})$, $j = 1 \ldots \psi(N)$, generates $\psi(N)$ UCB PMS games, each of which is defined on the set $J^U(N)$ of unit-speed job players under the set of feasible coalitions $\mathcal{C}^U$. Let $(J^U(N), \mathcal{C}^U, F_U^{(j)})$, $j = 1 \ldots \psi(N)$, be the UCB PMS games that are associated with the PMS game (N, P). Note that the $\psi(N)$ UCB PMS games $(J^U(N), \mathcal{C}^U, F_U^{(j)})$ are identical to each other, except for their zero-one processing requirement vectors. More precisely, in the UCB PMS game $(J^U(N), \mathcal{C}^U, F_U^{(j)})$, $\psi(N) - j + 1$ job players are associated with a unit processing requirement, where the others have a zero processing requirement.

In Section 5.2, we consider UCB PMS games in which all coalitions are allowed. Each job player of such a game owns a unit-speed machine and a job whose processing requirement is in {0, 1}. We call such games *unit-speed* {0, 1} *PMS games*. We fully characterize the symmetric core of these games, proving that they are totally balanced.

5.2. The Symmetric Core of Unit-Speed {0, 1} PMS Games (N^U, F_U)

The whole symmetric core of a *unit-speed* {0, 1} *PMS game* is derived in this subsection.

Definition 5. A unit-speed {0, 1} PMS game (N^U, F_U) is a PMS game in which each job player owns a machine of unit speed and a job whose processing requirement is either zero or one; the job players whose processing requirement is zero are called unit-speed 0 job players, and the others are called unit-speed 1 job players.

In what follows, we fully characterize the nonempty symmetric core of a unit-speed {0, 1} PMS game (N^U, F_U) with $n \geq 2$ job players, where $z \geq 0$ is the number of unit-speed 0 job players and $u = n - z \geq 0$ is the number of unit-speed 1 job players. We first consider the two trivial cases where $zu = 0$, and that is, the unit-speed job players are all of the same type, implying a single symmetric core allocation; if $z = n$, all coalitions of N^U have a zero cost, and therefore, assigning a cost 0 to all job players is the only symmetric cost allocation in the core. If $u = n$, all job players are unit-speed 1 job players, implying that the cost of the grand coalition is $n(n+1)/2n$, and therefore, the only symmetric core allocation assigns a cost

$$\frac{n+1}{2n} = 0.5\left(1+\frac{1}{n}\right)$$

to each job player.

In general, the cost of the grand coalition is

$$F_U(N^U) = \frac{u(u+1)}{2(u+z)} = \frac{u(u+1)}{2n}.$$

The cost of any other coalition is computed in a similar way. In the rest of this subsection, we assume that $zu \geq 1$, and that is, the grand coalition contains both types of unit-speed job players, implying that the symmetric core of the game, if it is nonempty, consists of a collection of pairs $(\alpha, \beta) \in \Re^2$, where α is the cost allocated to unit-speed 0 job players and β is the cost allocated to unit-speed 1 job players.

In unit-speed {0, 1} PMS games, unit-speed 0 job players are the most valuable players as they do not have jobs to process, but they own machines that speed up the completion time of the jobs of the other job players. Thus, we expect that if the core is nonempty, in any symmetric core cost allocation, the unit-speed 0 job players will be compensated for by the unit-speed 1 job players in order to persuade them to join the grand coalition. This observation is justified by the fact that $F_U(S) = 0$ for any coalition S of unit-speed 0 job players, implying that in any symmetric core cost allocation, the unit-speed 0 job players will be compensated for by the unit-speed 1 job players or they will pay nothing for joining the grand coalition; that is, any core allocation (α, β) satisfies $\alpha \leq 0$ and $\beta \geq 0$. From now on, we refer to the compensation $-\alpha \geq 0$ that unit-speed 0 job players get for joining the grand coalition rather than to the corresponding cost.

In Observation 2, we provide an upper bound on the compensation $-\alpha \geq 0$ paid to the unit-speed 0 job players and a lower bound $\beta \geq 0$ on the cost imposed on unit-speed 1 job players. The upper bound on $-\alpha$ is based on the fact that in any cooperative game, a player cannot expect to get a compensation that is higher than the marginal reduction in the total cost when this player is the last to join the grand coalition. Similarly, the lower bound on β is based on the fact that a player cannot expect to pay less than the marginal increase in the cost of the grand coalition when this player is the last to join the grand coalition.

Let $c(z,u)$ for $0 \leq z \leq n$, and $u = n - z$, be the cost of a coalition that consists of z (u) unit-speed 0 (1) job players in a unit-speed {0, 1} PMS game. Observation 2 follows directly from the discussion.

Observation 2. Suppose that the symmetric core of a unit-speed {0, 1} PMS game with z (u) unit-speed 0 (1) job players is nonempty. Then, any symmetric core allocation (α, β) of the game satisfies

1. $-\alpha \leq -(c(z,u) - c(z-1,u))$ and
2. $\beta \geq c(z,u) - c(z,u-1)$.

Let $-\tilde{\alpha}_1 \stackrel{\text{def}}{=} -(c(z,u) - c(z-1,u))$ and $\tilde{\beta}_2 \stackrel{\text{def}}{=} c(z,u) - c(z,u-1)$. In the next theorem, we prove that the symmetric core of unit-speed {0, 1} PMS games is nonempty and that the bounds specified in Observation 2 are tight; that is, there exist core allocations in which (i) $-\tilde{\alpha}_1$ assigns the maximum compensation to the unit-speed 0 job players and thus, the maximum cost to unit-speed 1 job players and (ii) $\tilde{\beta}_2$ assigns the minimum cost to the unit-speed 1 job players and thus, the minimum compensation to the unit-speed 0 job players. For given $z \geq 1$ and $u \geq 1$, let

$$\alpha_1 = -\frac{u(u+1)}{2(u+z)(u+z-1)}$$
$$\beta_1 = \frac{(u+1)(u+2z-1)}{2(u+z)(u+z-1)} \tag{8}$$

$$\alpha_2 = -\frac{u(u-1)(z-1)}{2z(u+z)(u+z-1)}$$
$$\beta_2 = \frac{u(u+2z-1)}{2(u+z)(u+z-1)}. \tag{9}$$

It is easy to verify that $\tilde{\alpha}_1$ and $\tilde{\beta}_2$, as defined, are equal to α_1 and β_2, respectively. The values of α_2 and β_1 are derived by the efficiency property of the core. We prove now that unit-speed {0, 1} PMS games are totally balanced by fully characterizing their symmetric core. In Theorem 2, (8) and (9) will be proven to be the two extreme symmetric core allocations. The theorem will also imply that the convex hull of these two cost allocations is the whole symmetric core of a unit-speed {0, 1} PMS game with z (u) unit-speed 0 (1) job players. We defer the technical proof of Theorem 2 to the appendix.

Theorem 2. *Any symmetric core allocation of a unit-speed* {0, 1} *PMS game* (N^U, F_U)*, where* $|N^U| = n = z + u$, $zu > 0$*, assigns a cost* α (β) *to each unit-speed* 0 (1) *job player, such that* $\alpha = \rho\alpha_1 + (1-\rho)\alpha_2$, $\beta = \rho\beta_1 + (1-\rho)\beta_2$ *for* $\rho \in [0,1]$*, for* α_1 *and* β_1 *defined in* (8) *and* α_2 *and* β_2 *defined in* (9). *If* $zu = 0$*, the core is a singleton; if* $u = 0$*, each unit-speed* 0 *job player is assigned a cost* $\alpha = 0$*, and if* $z = 0$*, each unit-speed* 1 *job player is assigned a cost* $\beta = 0.5 + 1/2n$.

As stated in Theorem 2, for any given unit-speed {0, 1} PMS game with $z > 0$ unit-speed 0 job players and $u > 0$ unit-speed 1 job players, all cost allocations in the convex hull of (α_1, β_1) and (α_2, β_2) are within the symmetric core of the game. Yet, not all cost allocations are identical in terms of fairness. As mentioned, the cost allocation (α_1, β_1) is the best to unit-speed 0 job players but the worst for unit-speed 1 job players, where the opposite holds for the cost allocation (α_2, β_2), which is the worst for the unit-speed 0 job players and the best for unit-speed 1 job players. In this sense, the core cost allocation that may reduce to the minimum the gap between the unhappiness of one type of job players versus the happiness of the second type of job players is the average of the two extreme cost allocations, namely the cost allocation $0.5(\alpha_1+\alpha_2, \beta_1+\beta_2)$. The idea behind this cost allocation is similar to the idea behind the well-known EGS rule proposed as a core allocation for sequencing games; see Curiel et al. (1989). Later, Hamers et al. (1996) generalize the EGS rule and propose the *split core* of a sequencing game by dividing the gain generated by the switch of two players not necessarily equally. We call the average of the two extreme core costs allocations for the unit-speed {0, 1} PMS game the *equal cost splitting* rule or for short, the *ECS* rule. Thus, let

$$\alpha_{ECS}=0.5(\alpha_1+\alpha_2)$$
$$\beta_{ECS}=0.5(\beta_1+\beta_2), \tag{10}$$

implying Definition 6.

Definition 6. The equal cost splitting rule for unit-speed {0, 1} PMS games generates the cost allocation $(\alpha_{ECS}, \beta_{ECS})$.

As a side remark, note that the core of a unit-speed {0, 1} PMS game also contains nonsymmetric cost allocations. For example, consider a unit-speed {0, 1} PMS game with three unit-speed job players; two of them are zero job players, and the third is a one job player. The nonsymmetric cost allocation that allocates one of the zero job players a cost 0, the second a cost $-\frac{1}{6}$, and the one job player a cost 0.5 is within the core.

5.3. The Basic Core of the PMS Game (*N*, *P*)

In this subsection, we wrap up the results of this section and prove that the PMS game (N, P) is totally balanced, and then, we show how to derive a line segment within the symmetric core of the game.

The next theorem follows from Theorem 1, the discussion following the theorem, and Observation 1.

Theorem 3. *The PMS game (N, P) is totally balanced if and only if the UCB PMS games $(J^U(N), \mathcal{C}^U, F_U^{(j)})$, $j=1\ldots\psi(N)$, are totally balanced.*

The next theorem is the key theorem of the paper.

Theorem 4. *The PMS game (N, P) is totally balanced.*

Proof. According to Theorem 2, unit-speed {0, 1} PMS games are totally balanced, and therefore, in view of Observation 1, the UCB PMS games $(J^U(N), \mathcal{C}^U, F_U^{(j)})$, $j=1\ldots\psi(N)$, which are constrained unit-speed {0, 1} PMS games, are also totally balanced, implying by Theorem 3 that the PMS game (N, P) is totally balanced. □

Definition 7. We call the symmetric part of the core of the PMS game, which is generated by the procedure described in this paper and in particular, by splitting players into unit-speed players whose processing requirement is zero or one, the basic core of the PMS game.

In the rest of this subsection, we derive the basic core of the PMS game (N, P), which is a line segment in $\Re^n$.

The following corollary follows from Theorem 2, (8), and (9).

Corollary 1. *The core of any UCB PMS game $(J^U(N), \mathcal{C}^U, F_U^{(j)})$, $1\le j\le\psi(N)$, contains the core of the respective unit-speed {0, 1} PMS game with $v(N)$ unit-speed job players, where $u^j=\psi(N)-j+1$ is the number of the unit-speed 1 job players and the rest (i.e., $z^j=v(N)-u^j$ players) are unit-speed 0 job players. Except for the case $j=1$ and $v(N)=\psi(N)$, its set of symmetric core allocations is a line segment in $\Re^{v(N)}$, which covers the following nondegenerate line segment defined by the following two extreme points. The first (second) point assigns a cost α_1^j (α_2^j) to each unit-speed 0 job player in $J^U(N)$ and a cost β_1^j (β_2^j) to each unit-speed 1 job player in $J^U(N)$, where $\alpha_1^j<\alpha_2^j<0$ and $\beta_2^j<\beta_1^j$; that is,*

$$\alpha_1^j=-\frac{(\psi(N)-j+1)(\psi(N)-j+2)}{2v(N)(v(N)-1)},$$

$$\beta_1^j=\frac{(\psi(N)-j+2)(2v(N)-\psi(N)+j-2)}{2v(N)(v(N)-1)}, \tag{11}$$

$$\alpha_2^j=-\frac{(\psi(N)-j+1)(\psi(N)-j)(v(N)-\psi(N)+j-2)}{2v(N)(v(N)-1)(v(N)-\psi(N)+j-1)},$$

$$\beta_2^j=\frac{(\psi(N)-j+1)(2v(N)-\psi(N)+j-2)}{2v(N)(v(N)-1)}. \tag{12}$$

If $j=1$ and $v(N)=\psi(N)$, then the UCB PMS game $(J^U(N), \mathcal{C}^U, F_U^{(1)})$ coincides with the constrained basic PMS game $(J(N), \mathcal{C}, F^{(1)})$. In this game, all job players are unit job players, and therefore, a single symmetric core allocation exists, where each job player of $J^U(N)$, and of $J(N)$, is allocated a cost of $(\psi(N)+1)/2\psi(N)$.

Recall that each job player $k\in J(N)$ is associated with its father, namely player $i\in N$, denoted by $f(k)=i$. In the CB PMS games, job player k is assumed to have a machine whose speed is $v'(\{k\})=v(\{f(k)\})/\psi(\{f(k)\})$. In Lemma 2, except for the case considered in Corollary 1, where $j=1$ and $v(N)=\psi(N)$, we present, for any $j=1,\ldots,\psi(N)$, a line segment in $\Re^{\psi(N)}$, which is a subset of the symmetric core of the CB PMS game $(J(N), \mathcal{C}, F^{(j)})$.

Lemma 2. *The symmetric core of any CB PMS game $(J(N), \mathcal{C}, F^{(j)})$ for $j=1,\ldots,\psi(N)$, except for the case $j=1$ and $v(N)=\psi(N)$, considered in Corollary 1, contains the convex hull of the following two nonidentical symmetric cost allocation vectors $(\ddot{\alpha}_t^j(k), \ddot{\beta}_t^j(k))$ for $t=1,2$:*

$$\ddot{\alpha}_t^j(k)=v'(\{k\})\alpha_t^j \qquad 1\le k\le j-1,$$
$$\ddot{\beta}_t^j(k)=\beta_t^j+(v'(\{k\})-1)\alpha_t^j \quad j\le k\le\psi(N). \tag{13}$$

Proof. Recall that the job players of any CB PMS game $(J(N), \mathcal{C}, F^{(j)})$ for $j=1,\ldots,\psi(N)$, are split into unit-speed

{0, 1} job players, resulting in the corresponding UCB PMS game $(J^U(N), \mathcal{C}^U, F_U^{(j)})$, for which we identified in Corollary 1 two nonidentical symmetric core allocations. In the proof here, we fold back these two symmetric core allocations into two symmetric core allocations of the corresponding CB PMS game. The proof is based on the fact that in order to generate the UCB PMS game $(J^U(N), \mathcal{C}^U, F_U^{(j)})$, any job player $k \in J(N)$ whose speed is $v'(k)$ in the CB PMS game is split into $v'(\{k\})$ unit-speed job players in the $(J^U(N), \mathcal{C}^U, F_U^{(j)})$ game, where one of them has the same processing requirement as that of its father, namely job player k, and the others have zero processing requirements. By summing up the corresponding first (second) symmetric core allocations of the unit-speed job players in the feasible coalition of job player k in the UCB PMS game $(J^U(N), \mathcal{C}^U, F_U^{(j)})$, we obtain two core allocations for job player k in the corresponding CB PMS game $(J(N), \mathcal{C}, F^{(j)})$. Repeating this process for any job player in $J(N)$ results in two nonidentical core allocations of the CB PMS game $(J(N), \mathcal{C}, F^{(j)})$ whose convex hull $\lambda(\ddot{\alpha}_1^j(1), \dots, \ddot{\alpha}_1^j(j-1), \ddot{\beta}_1^j(j), \dots, \ddot{\beta}_1^j(\psi(N))) + (1-\lambda)(\ddot{\alpha}_2^j(1), \dots, \ddot{\alpha}_2^j(j-1), \ddot{\beta}_2^j(j), \dots, \ddot{\beta}_2^j(\psi(N)))$, for $\lambda \in [0,1]$, preserves the symmetry, efficiency, and coalitional rationality conditions of the feasible coalitions in $\mathcal{C}^U$ satisfied by the game $(J^U(N), \mathcal{C}^U, F_U^{(j)})$. □

Similarly to Definition 6, we define the equal cost splitting allocation for the CB PMS game $(J(N), \mathcal{C}, F^{(j)})$, for $j = 1, \dots, \psi(N)$, by using Equations (10) and (13):

$$\ddot{\alpha}_{ECS}^j(k) = v'(\{k\})\alpha_{ECS}^j \qquad 1 \le k \le j-1,$$
$$\ddot{\beta}_{ECS}^j(k) = \beta_{ECS}^j + (v'(\{k\})-1)\alpha_{ECS}^j \quad j \le k \le \psi(N). \tag{14}$$

The cost allocation specified in (14) is the average of the two core cost allocation vectors of the CB PMS game $(J(N), \mathcal{C}, F^{(j)})$, for $j = 1, \dots, \psi(N)$, given in (13).

The following theorem is the main result of the paper as it identifies the basic core (see Definition 7) of the PMS game (N, P).

Theorem 5. *The basic core of the PMS game* (N, P) *is the nondegenerate line segment in* $\Re^n$ *that connects the following core allocation* $(f_1, \dots, f_n)$ *and* $(g_1, \dots, g_n)$, *where*

$$f_i = \sum_{k \in J(\{i\})} \left(\sum_{j=1}^{k} \Delta_j \ddot{\beta}_1^j(k) + \sum_{j=k+1}^{\psi(N)} \Delta_j \ddot{\alpha}_1^j(k) \right)$$

$$g_i = \sum_{k \in J(\{i\})} \left(\sum_{j=1}^{k} \Delta_j \ddot{\beta}_2^j(k) + \sum_{j=k+1}^{\psi(N)} \Delta_j \ddot{\alpha}_2^j(k) \right). \tag{15}$$

The basic core coincides with the symmetric core of the PMS game (N, P) *only if each player of N has a single job and the speed of the machine of each player is a positive rational number.*

Proof. According to Theorem 1, $P(S) = F(J(S)) = \sum_{j=1}^{\psi(N)} \Delta_j F^{(j)}(J(S))$, for any coalition $S \subseteq N$. By combining Lemma 2 and the constructive method of generating the cost allocation vectors in (15), we conclude that $(f_1, \dots, f_n)$ and $(g_1, \dots, g_n)$, are two symmetric core allocations of the PMS game (N, P), and their convex hull is a line segment in $\Re^n$, which according to Definition 7, is the basic core of the PMS game (N, P).

In the case that all players have a single job and a machine of the same speed $v \in \mathcal{Q}_{++}$, then $|N| = n = \psi(N) = v(N)$; in particular, the n UCB PMS games coincide with the respective n CB PMS games, and the constrained PMS game coincides with the PMS game (N, P), implying that the symmetric core of the PMS game (N, P) is fully characterized. □

We conclude this subsection by presenting the core cost allocation for the PMS game (N, P) generated by ECS rule by using the ECS cost allocation vectors of the CB PMS games (see (14)):

$$ECS_i = \sum_{k \in J(\{i\})} \left(\sum_{j=1}^{k} \Delta_j \ddot{\beta}_{ECS}^j(k) + \sum_{j=k+1}^{\psi(N)} \Delta_j \ddot{\alpha}_{ECS}^j(k) \right).$$

6. Conclusions

In this paper, we analyze the cooperative game of the $Q|split|\sum C_j$ problem, namely the PMS under job splitting for a set N of n players and a set $J(N)$ of $\psi(N)$ jobs. Each player is assumed to own several machines whose speeds are positive rational numbers and several jobs, where each job is associated with its processing requirement. If a player has no jobs to process, we assume that the player has an empty job with a zero processing requirement. The characteristic function of the game is the minimum sum of completion times of the jobs of any coalition of players $S \subseteq N$. We prove that the game is totally balanced by generating an infinitely large subset of its symmetric core. More precisely, we specify a line segment in $\Re^n$, such that each of its points is a cost allocation vector that assigns a cost to each player of N. We call the line segment that we have identified the basic core of the game as its derivation involves the use of $\psi(N)$ CB PMS games, namely PMS games that are associated with independent zero-one processing requirement vectors that form a basis for $\Re^{\psi(N)}$. The $\psi(N)$ CB PMS games are shown to linearly span the PMS game. The complexity of the proposed algorithm is linear in the number of jobs. This is most remarkable as the core of a cooperative game is defined by an exponential number of constraints. In addition, in general, symmetric core cost allocations are attractive in terms of fairness, as any two players with exactly the same characteristics are assigned the same cost. In fact, symmetry is one of the four attributes that the Shapley value satisfies. However, the Shapley value is a single-cost allocation, which is not necessarily a member of the core of the game.

We view the main contribution of the paper in the new methodology that we developed for analyzing the cooperative PMS game under job splitting. The methodology is based on the unique presentation of the PMS game as a nonnegative linear combination of the CB PMS games. The original game is totally balanced if and only if the CB PMS games are totally balanced. This methodology has the potential to be helpful in the analysis of other sequencing and PMS games, where the order of the players plays a central role.

Appendix

We first introduce a proposition that presents properties satisfied by the cost allocations (8) and (9). These properties are used in the proof of Theorem 2.

Proposition A.1. *Consider a unit-speed* {0, 1} *PMS game* (N^U, F_U), $|N^U| = n^U = z + u$, *where* z (u) *is the number of unit-speed* 0 (1) *players. If* $zu \geq 1$, *then the symmetric cost allocations* $(\alpha_i(z,u), \beta_i(z,u))$ *for* $i = 1,2$, *defined in* (8) *and* (9), *respectively, satisfy the following properties.*

1. *For any given* $u \geq 1$, $\beta_1(z,u)$ *is decreasing in* z *to zero.*
2. *For any* u, $\beta_2(1,u) = 0.5$, *and for any fixed* $z \geq 2$, $\beta_2(z,u)$ *is strictly increasing in* u *and* $\lim_{u\to\infty}\beta_2(z,u) = 0.5$.

Proof.

1. The proof that $\beta_1(z,u)$ is decreasing in z follows by verifying that the partial derivative of $\beta_1(z,u)$ with respect to z is negative. The convergence of $\beta_1(z,u)$ to zero as z grows to infinity follows by applying the L'Hopital rule on $\lim_{z\to\infty}\beta_1(z,u)$.
2. The proof for $z = 1$ is obtained by substitution of z by one in (9). The proof for $z \geq 2$ is obtained by showing that $\frac{\partial\beta_2(z,u)}{\partial u} > 0$, implying that $\beta_2(z,u)$ is increasing in u. By using the L'Hopital rule, we obtain that $\lim_{u\to\infty}\beta_2(z,u) = 0.5$ for any $z \geq 2$, concluding the proof. □

Proof of Theorem 2. Consider a unit-speed {0, 1} PMS game (N^U, F_U), where $|N^U| = u + z$, and $zu \geq 1$. For any given pair (z,u), the symmetric cost allocations (α_i, β_i), for $i = 1,2$, are distinct. Let $\omega(\ell,k)$ denote the cost of a coalition with ℓ unit-speed 0 players and k unit-speed 1 players, where $0 \leq \ell \leq z$, and $0 \leq k \leq u$, and that is, $\omega(\ell,k) = k(k+1)/2(k+\ell)$. It is sufficient to show that both cost allocations (α_i, β_i), for $i = 1,2$, are the only extreme symmetric core allocations of the unit-speed {0, 1} PMS game (N^U, F_U) in the sense that any symmetric core allocation (α, β) satisfies $\alpha \in [\alpha_1, \alpha_2]$ and $\beta \in [\beta_2, \beta_1]$, and the rest of the proof follows by the convexity of the core.

The efficiency property of the cost allocations (α_i, β_i) for $i = 1,2$, defined in (8) and (9), follows by verifying that the equations

$$z\alpha_i + u\beta_i = F_U(N^U) = \frac{u(u+1)}{2(z+u)},$$

hold. It remains to prove that for any coalition $S \subset N^U$ with ℓ, $0 \leq \ell \leq z$, unit-speed 0 players and k, $0 \leq k \leq u$, unit-speed 1 players, where $\ell + k < z + u$, the coalitional rationality condition holds; that is, $\alpha_i\ell + \beta_i k \leq \omega(\ell,k)$ for $i = 1,2$. For this sake, let $\Delta_i(\ell,k) = \omega(\ell,k) - \alpha_i\ell - \beta_i k$, for $i = 1,2$.

Consider first the symmetric cost allocation (α_1, β_1). In view of the first item of Proposition A.1, $\beta_1 = \beta_1(z,u) \leq \beta_1(1,u) = 0.5\left(1 + \frac{1}{u}\right)$ for any z and u satisfying $zu \geq 1$. We need to prove that the function $\Delta_1(\ell,k)$ is nonnegative for all proper coalitions of N^U. Coalitions that have no unit-speed 0 players, that is, $\ell = 0$, satisfy

$$\omega(0,k) = 0.5(1+k) = 0.5\left(1 + \frac{1}{k}\right)k \geq 0.5\left(1 + \frac{1}{u}\right) \geq \beta_1 k.$$

This proves that $\Delta_1(0,k) \geq 0$. Also, coalitions that contain just one unit-speed 0 player, that is, $\ell = 1$, satisfy the coalitional rationality conditions as $\omega(1,k) = 0.5k$, and the allocated cost is $\alpha_1 + \beta_1 k$, implying that it is sufficient to prove that $-\alpha_1 \geq (\beta_1 - 0.5)k$. By using (8), this inequality boils down to the inequality $u(u+1) \geq k(u + 3z - z^2 - 1)$. The quadratic function $3z - z^2 - 1 \leq 1$ for all natural numbers, implying that it is sufficient to prove that $u(u+1) \geq k(u+1)$, which trivially holds.

By simple algebra, one can verify that the coalitional rationality constraint for $z-1$ unit-speed 0 players and u unit-speed 1 players is tight; that is, $(z-1)\alpha_1 + u\beta_1 = \omega(z-1,u)$.

Note that coalitions with no unit-speed 1 players satisfy the coalitional rationality constraints as for any $\ell \in \{1,\ldots,z\}$, $\omega(\ell,0) = 0$, and $\Delta_1(\ell,0) = -\alpha_1\ell \geq 0$. Thus, consider pairs (ℓ,k), where $\ell \in \{2,\ldots,z\}$ and $k \in \{1,\ldots,u\}$. Let $\Delta_1^k(\ell) = \Delta_1(\ell,k)$. For the sake of the proof, we extend the function $\Delta_1^k(\ell)$ to be defined on the interval $[0,z]$. We prove the following properties for any $k \in \{1,\ldots,u\}$:. (i) The function $\Delta_1^k(\ell)$ is strictly convex in ℓ (ii) there exists a single real value $\ell_1(k)$, $0 < \ell_1(k) < z$, where $\Delta_1^k(\ell)$ decreases in $\ell \in (0, \ell_1(k))$ and increases in $\ell \in (\ell_1(k), z)$; (iii) the sequence $\ell_1(k)$ for $k \in \{1,\ldots,u\}$ is strictly increasing in k; and (iv) $\ell_1(u)$ satisfies $z-1 < \ell_1(u) < z$, and $\Delta_1^u(\ell_1(u)) < 0$, whereas for $\ell \in \{0,\ldots,z\}$, $\Delta_1^u(\ell) \geq 0$. We prove these items as follows. (i) The convexity of the function $\Delta_1^k(\ell)$ follows from its form, that is,

$$\Delta_1^k(\ell) = \frac{k(k+1)}{2(k+\ell)} - \alpha_1\ell - \beta_1 k,$$

or alternatively, from the fact that

$$\frac{\partial\Delta_1^k(\ell)}{\partial\ell} = -\frac{k(k+1)}{2(k+\ell)^2} - \alpha_1,$$

is increasing in ℓ; (ii) the unconstrained minimizer of $\Delta_1^k(\ell)$ is obtained by equating its derivative to zero, that is, $\frac{d\Delta_1^k(\ell)}{d\ell} = 0$, implying that

$$\ell_1(k) = \sqrt{\frac{k(k+1)}{2|\alpha_1|}} - k; \tag{A.1}$$

(iii) the sequence $\ell_1(k)$ is increasing in k, as $\frac{d\ell_1(k)}{dk} > 0$, and in view of (8), $|\alpha_1| < 0.5$ implying that the sequence $\ell_1(k)$ is strictly increasing in k; and (iv) by substituting $k = u$ and the expression for α_1, see (8), into (A.1), we get

$$\ell_1(u) = \sqrt{\frac{u(u+1)}{2|\alpha_1|}} - u = \sqrt{(u+z)(u+z-1)} - u \in (z-1, z).$$

As $\ell_1(u)$ is the unique minimizer of $\Delta_1^u(\ell)$, and $\Delta_1^u(z-1) = \Delta_1^u(z) = 0$, we conclude that $\Delta_1^u(\ell_1(u)) < 0$, but for any integer $\ell \in \{0,\ldots,z\}$, $\Delta_1^u(\ell) \geq 0$.

In order to terminate the proof, it is sufficient to show that for $k \in \{1,\ldots,u-1\}$, $\Delta_1^k(\ell_1(k)) > 0$, as it implies that $\Delta_1^k(\ell) > 0$

for $\ell \in \{0, \ldots, z\}$. By using (8) and (A.1),

$$\ell_1(k) = \sqrt{k(k+1)}\sqrt{\frac{(u+z)(u+z-1)}{u(u+1)}} - k,$$

and

$$\Delta_1(\ell_1(k), k) = \frac{k(k+1)}{2(k+\ell_1(k))} - \alpha_1 \ell_1(k) - \beta_1 k$$
$$= \sqrt{k(k+1)}\sqrt{\frac{u(u+1)}{(u+z)(u+z-1)}} - k\frac{(u+1)(2u+2z-1)}{2(u+z)(u+z-1)}.$$

In order to show that $\Delta_1(\ell_1(k), k) > 0$ for $k = 1, \ldots, u-1$, it remains to check that

$$2\sqrt{(k+1)u(u+z)(u+z-1)} \geq \sqrt{k(u+1)}(2u+2z-1).$$

By taking the square of both sides of the inequality, it is equivalent to showing that $4(k+1)(u^3 + u^2(2z-1) + uz(z-1)) \geq k(4u^3 + 4uz^2 - 3u + 8u^2z + 4zu + 4\ z^2 + 1 - 4z)$, which in turn, is equivalent to showing that $4(k+1)(u^3 + 2u^2z - u^2 + uz^2 - zu) \geq k(4u^3 + 4uz^2 - 3u + 8u^2z + 4zu + 4z^2 + 1 - 4z)$. As both sides of the last inequality are positive and as $4(k+1)/k$ is decreasing in k, the inequality gets tighter as k is larger. Thus, it suffices to check the inequality for $k = u-1$: By using some simple algebraic manipulations, we get that it is sufficient to show that $u(3u + 8z - 4) + 4z^2 - 4z + 1 \geq 0$, which clearly holds. This concludes the proof that the symmetric cost allocation (α_1, β_1) is in the core of the unit-speed $\{0,1\}$ PMS game (N^U, F_U).

According to Observation 2 and the discussion in between the observation and Theorem 2 in Section 5.2, there does not exist any symmetric core cost allocation (α, β) for the unit-speed $\{0,1\}$ PMS game (N^U, F_U) for which $\alpha > \alpha_1$, and that is, the unit-speed 0 players are compensated for by the maximum possible according to the symmetric cost allocation (α_1, β_1).

Similarly, we prove that the symmetric cost allocation (α_2, β_2), given in (9), is in the core of the unit-speed $\{0,1\}$ PMS game (N^U, F_U) and that it is extreme in the sense that there does not exist another symmetric core allocation (α, β) with $\beta < \beta_2$. Define the function $\Delta_2(\ell, k) = \omega(\ell, k) - \alpha_2 \ell - \beta_2 k$, and for any fixed $\ell \in \{0, \ldots, z\}$, let $\Delta_2^\ell(k) = \Delta_2(\ell, k)$. As the efficiency condition holds, it remains to prove the coalitional rationality conditions for all proper coalitions that consist of ℓ unit-speed 0 players and k unit-speed 1 players. If $k=0$, the coalition's cost is zero, whereas the cost allocated is $\alpha_2 \ell < 0$ for $1 \leq \ell \leq z$, proving the coalitional rationality conditions for such coalitions. We proceed to verifying the function $\Delta_2^\ell(k)$ for $\ell \in \{0, 1\}$, and $k \in \{1, \ldots, u\}$. Recall from the second item of Proposition A.1 that

$$\beta_2 = \frac{u(u+2z-1)}{2(u+z)(u+z-1)} \leq 0.5,$$

and for $z \geq 2$, $\beta_2 < 0.5$.

Next, we prove that $\Delta_2^0(k) \geq 0$ for $k \in \{1, \ldots, u\}$. These inequalities follow by the second item of Proposition A.1, which implies that $\beta_2 \leq 0.5$, and therefore, $\Delta_2^0(k) = (k+1)/2 - \beta_2 k \geq 0$. We consider now coalitions with a single unit-speed 0 player: that is, $\ell = 1$. Under this case, $\Delta_2^1(k) = 0.5k - \alpha_2 - \beta_2 k = (0.5 - \beta_2)k - \alpha_2$, which is positive as $\beta_2 \leq 0.5$ and $\alpha_2 < 0$. Thus, the coalitional rationality constraints for $\ell = 1$, hold too. In particular, the symmetric cost allocation (α_2, β_2) is in the core of unit-speed $\{0,1\}$ PMS games (N^U, F_U) with $z = 1$.

In order to conclude the proof of the coalitional rationality constraints for any $z \geq 2$, we consider the continuous extension of the function $\Delta_2^\ell(k)$ in the interval $k \in [1, u]$. We show that the function $\Delta_2^\ell(k)$ is strictly convex in k by verifying that its second derivative is positive. Calculations reveal that

$$\frac{d}{dk}\Delta_2^\ell(k) = 0.5\left(1 - \frac{\ell(\ell-1)}{(k+\ell)^2}\right) - \beta_2,$$

and

$$\frac{d^2}{dk^2}\Delta_2^\ell(k) = \frac{\ell(\ell-1)}{(k+\ell)^3} > 0.$$

The unconstrained minimizer $k_2(\ell)$ for any given ℓ is obtained by solving $\frac{d}{dk}\Delta_2^\ell(k) = 0$:

$$k_2(\ell) = \sqrt{\frac{\ell(\ell-1)}{1-2\beta_2}} - \ell. \tag{A.2}$$

Apparently, $k_2(\ell)$ is increasing in ℓ as the numerator of

$$\frac{d}{d\ell}k_2(\ell) = \frac{0.5(2\ell-1) - \sqrt{1-2\beta_2}\sqrt{\ell(\ell-1)}}{\sqrt{1-2\beta_2}\sqrt{\ell(\ell-1)}}$$

is positive. To see this, note that $0.5(2\ell-1)$ is the average of $\ell-1$ and ℓ, whereas $\sqrt{\ell(\ell-1)}$ is their geometric mean. The numerator is positive as the geometric mean is bounded from above by the average. In addition, it follows from the second item of Proposition A.1 that for $z \geq 2$, $\sqrt{1-2\beta_2} < 1$ as $\beta_2 < 0.5$. In order to show that $k_2(\ell) < u$ for any $\ell \in \{2, \ldots, z\}$, we substitute β_2 in (A.2) by (9), and we get that

$$k_2(z) = \sqrt{\frac{z(z-1)(u+z)(u+z-1)}{(u+z)(u+z-1) - u(u+2z-1)}} - z$$
$$= \sqrt{(u+z)(u+z-1)} - z,$$

thus $u - 1 < k_2(z) < u$. In fact, simple calculations reveal that in addition to the efficiency condition, the cost of the coalition that consists of z unit-speed 0 players and $u-1$ unit-speed 1 players also satisfies $\Delta_2^z(u-1) = 0$. As $\Delta_2^z(k)$ is a decreasing function of k, for $k \in [1, k_2(z)]$, we conclude that $\Delta_2^z(k) \geq 0$ for $k \in \{0, \ldots, u\}$, implying the coalitional rationality conditions for any number of unit-speed 1 players.

In order to complete the proof, let $\Delta_2^k(\ell) = \Delta_2(\ell, k)$ be a continuous function of $\ell \in (1, z]$ for a fixed $k \in \{1, \ldots, u\}$. By definition, $\Delta_2^k(\ell)$ is convex in ℓ. For $k = u$, note that

$$\frac{d}{d\ell}\Delta_2^u(\ell) = -\frac{u(u+1)}{2(u+\ell)^2} - \alpha_2,$$

which is an increasing function of ℓ, and its maximum is obtained at $\ell = z$.Thus,

$$\frac{d}{d\ell}\Delta_2^u(\ell) \leq -\frac{u(u+1)}{2(u+z)^2} - \alpha_2 = -\frac{u(2z(z-1) + u(u+2z-1))}{2z(u+z)^2(u+z-1)} < 0,$$

proving that $\Delta_2^u(\ell)$ is decreasing in ℓ, and in view of the efficiency condition, we conclude that $\Delta_2^u(\ell) \geq 0$ for any $\ell \in \{0, \ldots, z\}$. It can be proven, by similar arguments, that $\Delta_2^{u-1}(\ell)$ is also decreasing in ℓ, and by using $\Delta_2^{u-1}(z) = 0$, the coalitional rationality conditions also hold for coalitions that consist of $u-1$ unit-speed 1 players and any number ℓ, $0 \le \ell \le z$, of unit-speed 0 players.

Similarly to the case of $\Delta_1^k(\ell)$, $\Delta_2^k(\ell)$ is convex, and its unconstrained minimizer is

$$\ell_2(k) = \sqrt{\frac{k(k+1)}{2|\alpha_2|}} - k.$$

Next, we show that the coalitional rationality conditions hold for any $k \in (0,u)$, for which $\ell_2(k) \ge z$, by using the following facts. First, the function $\Delta_2^k(\ell)$ is strictly convex and decreasing in $\ell \in (1, \ell_2(k)) \supseteq (1, z]$, which implies that it is decreasing in $\ell \in (1, z)$, and second, $\Delta_2^k(z) \ge 0$. By solving the equation $\ell_2(k) = z$, we get the lowest value of k for which $\ell_2(k) \ge z$, which we denote by $\hat{u}$, where

$$\hat{u} = \frac{4z|\alpha_2| - 1 + \sqrt{8z|\alpha_2|(z-1)+1}}{2(1-2|\alpha_2|)}.$$

Thus, it remains to prove the coalitional rationality conditions for any $k \in (0,\hat{u})$. For this sake, note that the function, $\Delta_2^k(\ell_2(k)) = \sqrt{2|\alpha_2|k(k+1)} + (\alpha_2 - \beta_2)k$, is concave in $k \in (0,\hat{u})$, where at the extreme points of the interval, namely at $k=0$ and $k=\hat{u}$, $\Delta_2^k(\ell) \ge 0$. In view of the concavity of the function $\Delta_2^k(\ell_2(k))$, the set $\{k : \Delta_2^k(\ell_2(k)) \ge 0\}$ is convex, completing the proof that the symmetric cost allocation (α_2, β_2) is in the core of the unit-speed {0, 1} PMS game (N^U, F_U).

According to Observation 2 and the discussion in between the observation and Theorem 2 in Section 5.2, there does not exist any symmetric core cost allocation (α, β) for unit-speed {0, 1} PMS game (N^U, F_U) for which $\beta < \beta_2$, and that is, the unit-speed 1 players pay the minimum possible cost according to the symmetric cost allocation (α_2, β_2). □

References

Alon T, Haviv M (2020) Pooling risk games. *Internat. Game Theory Rev.* 22(3):1950015.

Anily S (2018) Full characterization of the nonnegative core of some cooperative games. *Naval Res. Logist.* 65(4):303–316.

Anily S, Haviv M (2007) The cost allocation problem for the first order interaction joint replenishment model. *Oper. Res.* 55(2):292–302.

Anily S, Haviv M (2010) Cooperation in service systems. *Oper. Res.* 58(3):660–673.

Bird CG (1976) On cost allocation for a spanning tree: A game theoretic approach. *Networks* 6(4):335–350.

Curiel I (2010) *Cooperative Game Theory and Applications* (Kluwer Academic Publishers, New York).

Curiel I, Hamers H, Klijn F (2002) Sequencing games: A survey. Borm P, Peters H, eds. *Chapters in Game Theory*, Theory and Decision Library C, vol. 31 (Springer, Boston), 27–50.

Curiel I, Pederzoli G, Tijs S (1989) Sequencing games. *Eur. J. Oper. Res.* 40(3):344–351.

Curiel I, Potters J, Prasad R, Tijs S, Veltman B (1994) Sequencing and cooperation. *Oper. Res.* 42(3):566–568.

Gillies D (1953) Some theorems on n-person games. PhD dissertation, Princeton University, Princeton, NJ.

Graham RL, Lawler EL, Lenstra JK, Kan AR (1979) Optimization and approximation in deterministic sequencing and scheduling: A survey. *Ann. Discrete Math.* 5:287–326.

Hamers H, Klijn F, Suijs J (1999) On the balancedness of multiple machine sequencing games. *Eur. J. Oper. Res.* 119(3):678–691.

Hamers H, Suijs J, Borm P (1996) The split core for sequencing games. *Games Econom. Behav.* 15(2):165–176.

Kushwaha M, Gupta S (2015) Various schemes of load balancing in distributed systems—A review. *Internat. J. Sci. Res. Sci. Engrg. Tech.* 4(7):741–748.

Moulin H (2007) On scheduling fees to prevent merging, splitting, and transferring of jobs. *Math. Oper. Res.* 32(2):266–283.

Moulin H (2008) Proportional scheduling, split-proofness, and merge-proofness. *Games Econom. Behav.* 63(2):567–587.

Peleg B, Sudhölter P (2007) *Introduction to the Theory of Cooperative Games* (Springer Science & Business Media, New York).

Penmatsa S, Chronopoulos AT (2011) Game theoretic static load balancing for distributed systems. *J. Parallel Distributed Comput.* 71(4):537–555.

Pinedo M (2016) *Scheduling: Theory, Algorithms and Systems*, 4th ed. (Springer International Publishing, Cham, Switzerland).

Potts CN, Van Wassenhove LN (1992) Integrating scheduling with batching and lot-sizing: A review of algorithms and complexity. *J. Oper. Res. Soc.* 43(5):395–406.

Serafini P (1996) Scheduling jobs on several machines with the job splitting property. *Oper. Res.* 44(4):617–628.

Shapley LS (1971) Cores of convex games. *Internat. J. Game Theory* 1:11–26.

Shapley LS, Shubik M (1972) The assignment game I: The core. *Internat. J. Game Theory* 1(1):111–130.

Slikker M (2006) Balancedness of multiple machine sequencing games revisited. *Eur. J. Oper. Res.* 174(3):1944–1949.

Smith WE (1956) Various optimizers for single-stage production. *Naval Res. Logist. Quart.* 3(1–2):59–66.

Tahar DN, Yalaoui F, Chu C, Amodeo L (2006) A linear programming approach for identical parallel machine scheduling with job splitting and sequence-dependent setup times. *Internat. J. Production Econom.* 99(1–2):63–73.

Tijs SH, Parthasarathy T, Potters JAM, Rajendra Prasad V (1984) Permutation games: Another class of totally balanced games. *OR Spectrum* 6:119–123.

Xing W, Zhang J (2000) Parallel machine scheduling with splitting jobs. *Discrete Appl. Math.* 103(1–3):259–269.

Yu Y, Benjaafar S, Gerchak Y (2015) Capacity sharing and cost allocation among independent firms with congestion. *Production Oper. Management* 24(8):1285–1310.

Zhang J (2009) Cost allocations for joint replenishment models. *Oper. Res.* 57(1):146–156.