SEQUENCING GREY GAMES

SERAP ERGÜN
Isparta University of Applied Sciences, Faculty of Technology
Department of Computer Engineering, Isparta, Turkey

OSMAN PALANCI
Süleyman Demirel University, Faculty of Economics and Business Administration
Department of Business Administration, Isparta, Turkey

SIRMA ZEYNEP ALPARSLAN GÖK* AND ŞULE NIZAMOĞLU
Süleyman Demirel University, Faculty of Arts and Sciences
Department of Mathematics, Isparta, Turkey

GERHARD WILHELM WEBER
Poznan University of Technology
Chair of Marketing and Economic Engineering, Poznan, Poland

(Communicated by Athanasios Yannacopoulos)

Abstract. The job scheduling problem is a notoriously difficult problem in combinatorial optimization and Operational Research. In this study, we handle the job scheduling problem by using a cooperative game theoretical approach. In the sequel, sequencing situations arising from grey uncertainty are considered. Cooperative grey game theory is applied to analyze these situations. Further, grey sequencing games are constructed and grey equal gain splitting (GEGS) rule is introduced. It is shown that cooperative grey games are convex. An application is given based on Priority Based Scheduling Algorithm. The paper ends with a conclusion.

1. Introduction. In a job scheduling situation or a sequencing problem, a number of jobs has to be processed in some order on one or more machines in such a way that a specific cost criterion is minimized. Job scheduling situations can be classified on the basis of many features. We mention the number of machines, the specific properties of the machines (e.g., parallel, serial), the chosen cost criterion (e.g., maximum completion time, weighted completion time), restrictions on the jobs (e.g., ready times, due dates) and possibly the specific order in which the jobs have to be processed on the machines (e.g., job-shop, flow-job) [4].

By associating jobs to clients, a sequencing problem gives rise to a multi-active decision making problem. Each client incurs costs, depending on the completion times of his jobs. By assuming an initial order on the jobs, the first problem which the clients jointly face is that of finding an optimal reordering of all jobs, i.e., a schedule maximizing joint cost savings. The subsequent problem is how to

2010 Mathematics Subject Classification. Primary: 91A12.
Key words and phrases. Sequencing games, grey data, grey games, cooperative game theory, scheduling algorithm.

* Corresponding author: zeynepalparslan@yahoo.com.
reallocate these cost savings in a fair way. This “fairness” problem can be analyzed from a game-theoretic point of view [4].

Uncertainty affects our decision making activities on a daily basis. There are many sources of uncertainty in the real world. The effect of uncertainty could be seen at noise in observation and experimental design, incomplete information and vagueness in preference structures. In the sequel, grey uncertainty is used in our results.

In this paper, we consider sequencing situations in which a certain number of customers has to be served by one server under grey uncertainty. Everyone of them has a grey cost function which depends on his completion time, i.e., the time which he or she has to wait plus the time it takes to serve him or her. Sequencing problems are well known in literature [1, 4, 5].

Several papers about cooperative games grey-valued appeared recently [9, 13, 14, 16, 19]. In this case a cooperative game is considered with a grey-valued characteristic function, i.e., the worth of a coalition is not a real number, but an element of the closed interval of real numbers. This means that one observes an element between a lower bound and an upper bound of the worth of the considered coalitions. This is very important, e.g., for example from a computational and algorithmic point of view.

In [5] the class of sequencing games are introduced. An updated survey on these games can be found in [6]. We also refer to the survey on Operations Research Games [2]. This paper extends the analysis of cooperative sequencing games to a setting with grey data.

The paper is organized as follows. In Section 2, we recall basic notions and results from one-machine sequencing situations and related games, grey calculus and the theory of cooperative grey games. In Section 3, we introduce sequencing grey situations and show that these games are convex. In Section 4, we extend the classical sequencing games to the grey setting. Finally, in Section 5, we consider an application related to sequencing grey situations.

2. Preliminaries and notations.

2.1. Sequencing situations and related games. In this section, some basic notions from one-machine sequencing problems and cooperative game theory are given [1, 5]. In one-machine sequencing problems, there is a queue of jobs in front of a machine waiting to be processed. A job can be interesting to more than one player and every player can be involved in more than one job. Formally, a one-machine sequencing situation is a 4-tuple $(N, \sigma_0, \alpha, p)$ where:

- $N = \{1, ..., n\}$ is the set of jobs;
- $\sigma_0 : N \rightarrow N$ is a permutation that defines the initial order of the jobs;
- $\alpha = (\alpha_i)_{i \in N} \in \mathbb{R}^n_+$ is a non-negative real vector, where $\alpha_i$ is the cost per unit of time of job $i$;
- $p = (p_i)_{i \in N} \in \mathbb{R}^n_+$ is a positive real vector, where $p_i$ is the processing time of job $i$.

Given a sequencing situation and an ordering $\sigma$ of the jobs, for each $i \in N$ we denote by $P(\sigma, i)$ the set of jobs preceding $i$, according to the order $\sigma$. The time spent in the system by job $i$ is the sum between the waiting time that jobs in $P(\sigma, i)$ need to be processed and the processing time of job $i$ yielding the related
The optimal order of the jobs $\sigma^*$ produces the minimum cost $C_{\sigma^*} = \sum_{i \in N} \alpha_i \left( \sum_{j \in P(\sigma^*,i)} p_j + p_i \right)$ or the maximum cost saving $C_{\sigma_0} - C_{\sigma^*}$. The authors of [17] proved that an optimal order can be obtained reordering the jobs according to decreasing urgency indices, where the urgency index of job $i \in N$ is defined as $u_i = \alpha_i/p_i$ (clearly, if this condition holds for the initial order, no reordering of the jobs is necessary).

If all the jobs to be processed belong to a unique agent, he or she will accept to reorder them optimally, according to Smith’s result [17]. A different situation has to be faced when each job belongs to a different agent. In this case, a reordering requires that at least the agents who change their positions agree on the new order. So, we can say that a switch between two jobs is possible whenever they are consecutive in the current order or all the agents who own one of the jobs in between.

The following question arises: Is it possible to share this cost savings $C_{\sigma_0} - C_{\sigma^*}$ among the agents in such a way that no agent recedes from the agreement on an optimal order? In other words, we want to find fair shares of the overall cost savings to be given to the different agents, which motivate all of them in agreeing on the optimal order and have no incentive to recede from the agreement. This question can be answered by a direct analysis of sequencing situations or by using related cooperative games.

A cooperative game with transferable utility (TU-game) in characteristic function form is a pair $< N, v >$, where $N$ is the set of players and $v : 2^N \to \mathbb{R}$ is the characteristic function with $v(\emptyset) = 0$. This function assigns to each group of players (coalition), $S \subset N$, the value $v(S)$ which represents what the members in $S$ obtain when they jointly cooperate. A classic issue in cooperative game theory is how to distribute the profit generated by the cooperating players. An important role is played by the allocations in the core of the game. The core of a TU-game $< N, v >$ is the subset of vectors $x \in \mathbb{R}^N$ satisfying $\sum_{i \in N} x_i = v(N)$ (efficiency) and $\sum_{i \in S} x_i \geq v(S)$, $S \subset N$ (coalitional rationality).

A sequencing game is a pair $< N, v >$ where $N$ is the set of players, that coincides with the set of jobs, and the characteristic function $v$ assigns to each coalition $S$ the maximal cost savings which the members of $S$ can obtain by reordering only their jobs. We say that a set of jobs $T$ is connected according to an order $\sigma$ if for all $i, j \in T$ and $k \in N, \sigma(i) < \sigma(k) < \sigma(j)$ implies $k \in T$.

A switch of two connected jobs $i$ and $j$, where $i$ precedes $j$, generates a change in cost equal to $\alpha_j p_i - \alpha_i p_j$. This amount is positive if and only if the urgency indices verify $u_i < u_j$. Clearly, if $\alpha_j p_i - \alpha_i p_j$ is negative it is not beneficial for $i$ and $j$ to switch their positions. We denote the gain of the switch as

$$g_{ij} = (\alpha_j p_i - \alpha_i p_j)_+ = \max\{0, \alpha_j p_i - \alpha_i p_j\}$$

and, consequently, the gain of a connected coalition $T$ according to an order $\sigma$ is defined by $v(T) = \sum_{j \in T} \sum_{i \in P(\sigma,j) \cap T} g_{ij}$.

If $S$ is not a connected coalition, the order $\sigma$ induces a partition in connected components, denoted by $S \setminus \sigma$. In view of this, the characteristic function $v$ of the sequencing game can be defined as $v(S) = \sum_{T \in S \setminus \sigma} v(T)$ for each $S \subset N$, or
equivalently as $v = \sum_{i,j \in N: i < j} g_{ij}u_{[i,j]}$, where $u_{[i,j]}$ is the unanimity game defined as:

$$u_{[i,j]}(S) = \begin{cases} 
1, & \text{if } \{i, i+1, \ldots, j-1, j\} \subset S, \\
0, & \text{otherwise.}
\end{cases}$$

The paper [5] shows that sequencing games are convex games and, consequently, their core is nonempty. Moreover, it is possible to determine a core allocation without computing the characteristic function of the game. The authors propose to share equally between the players $i, j$ the gain $g_{ij}$ produced by the switch and call this rule the Equal Gain Splitting rule (EGS-rule). It can be computed by $EGS_i = \frac{1}{2}\sum_{k \in P(\sigma, i)} g_{ki} + \frac{1}{2}\sum_{j: i \in P(\sigma, j)} g_{ij}$ for each $i \in N$.

### 2.2. Grey calculus and cooperative grey games.

This section provides some preliminaries from grey calculus and the theory of cooperative grey games [7, 14].

A grey number is such a number whose exact value is known but a range within that the value lies is known. In applications, a grey number in general is an interval or a general set of numbers.

In this paper, the interval grey numbers are considered. A grey number with both a lower limit $\underline{a}$ and an upper limit $\overline{a}$ is called an interval grey number, denoted as $\underline{a} \otimes \overline{a}$.

For example, the weight of a seal is between 20 and 25 kg. A specific person’s height is between 1.8 and 1.9 meters. These two grey numbers can be respectively written as $\underline{a} \otimes \overline{a}$.

Assume that we have grey numbers $\underline{a} \otimes \overline{a}, a < b$, and $\underline{b} \otimes \overline{b}, b < c$. The sum of $\underline{a} \otimes \overline{a}$ and $\underline{b} \otimes \overline{b}$, written by $\underline{a} \otimes \overline{b} + \underline{a} \otimes \overline{b}$, is defined as $\underline{a} \otimes \overline{b} + \underline{a} \otimes \overline{b}$.

**Example 1.** Given $\underline{a} \otimes \overline{a} = [2, 3]$ and $\underline{b} \otimes \overline{b} = [4, 5]$, then $\underline{a} \otimes \overline{b} + \underline{a} \otimes \overline{b} = [6, 8]$.

**Example 2.** Given $\underline{a} \otimes \overline{a} = [2, 5]$ and $\underline{b} \otimes \overline{b} = [6, 8]$, $\underline{a} \otimes \overline{b}$ is defined since $|\underline{a} \otimes \overline{a}| \geq |\underline{b} \otimes \overline{b}|$, but $\underline{b} \otimes \overline{b}$ is not defined since $|\underline{b} \otimes \overline{b}| \not\geq |\underline{a} \otimes \overline{a}|$, then we have $\underline{a} \otimes \overline{b} - \underline{b} \otimes \overline{b} = [2 - 6, 5 - 8] = [-4, -3]$.

Moreover, for $w_1^t, w_2^t \in \mathbb{G}G^N$ we say that $w_1^t \preceq w_2^t$ if $w_i^t(S) \preceq w_j^t(S)$, for each $S \in 2^N$.

For $w_1^t, w_2^t \in \mathbb{G}G^N$ and $\lambda \in \mathbb{R}_+$ we define $w_1^t + w_2^t$ and $w_1^t + \lambda w_2^t$ by $(w_1^t + w_2^t)(S) = w_1^t(S) + w_2^t(S)$ and $(\lambda w_2^t)(S) = \lambda \cdot w_2^t(S)$ for each $S \in 2^N$. So, we conclude that $\mathbb{G}G^N$ endowed with “$\preceq$” being a partially ordered set and having a
cone structure with respect to addition and multiplication by non-negative scalars described above.

For \( w'_1, w'_2 \in \mathcal{G}^{G^N} \) with \( |w'_1(S)| \geq |w'_2(S)| \) for each \( S \in 2^N, < N, w'_1 - w'_2 > \) is defined by \( (w'_1 - w'_2)(S) = w'_1(S) - w'_2(S) \).

We call a game \( < N, w' > \) size monotonic if \( < N, |w'| > \) is monotonic, i.e., \( |w'(S)| \leq |w'(T)| \) for all \( S, T \in 2^N \) with \( S \subset T \). For further use we denote by \( \text{SMGG}^N \) the class of size monotonic grey games with player set \( N \).

The grey Shapley value \( \Phi' : \text{SMGG}^N \to \mathcal{G}(\mathbb{R})^N \) is defined by

\[
\Phi'(w') := \frac{1}{n!} \sum_{\sigma \in \Pi(N)} m^\sigma(w') \in \left[ \frac{1}{n!} \sum_{\sigma \in \Pi(N)} m^\sigma(\mathcal{A}), \frac{1}{n!} \sum_{\sigma \in \Pi(N)} m^\sigma(\mathcal{A}) \right].
\]

3. Sequencing situations with grey data. In this section, we introduce a one-machine sequencing grey situation described by a 4-tuple \( (N, \sigma_0, \alpha', p') \), where \( N \) and \( \sigma_0 \) are as usual, whereas \( \alpha' \in (\bar{\alpha}_i', \underline{\alpha}_i')_{i \in N} \in \mathcal{G}(\mathbb{R}_+)^N \) and \( p' \in (p'_i, \overline{p}_i)_{i \in N} \in \mathcal{G}(\mathbb{R}_+)^N \) are vectors of grey numbers with \( \bar{\alpha}_i' \) and \( \underline{\alpha}_i' \), here representing the minimal and maximal unitary cost of job \( i \), respectively, then \( p'_i \) and \( \overline{p}_i \), representing the minimal and maximal processing time of job \( i \), respectively.

To handle sequencing situations in which all parameters are given by grey numbers, we propose an grey urgency approach and an grey relaxation approach. The grey urgency index of job \( i \in N \) is defined as \( u'_i = \frac{\alpha'_i}{p'_i} = \frac{\alpha'_i}{\overline{p}_i} = \frac{\bar{\alpha}_i'}{\underline{\alpha}_i'} \) (\( i \in N \)).

To extend Smith’s result for finding the optimal order we need not only to compare \( u_i \) and \( u_j \) to check whether \( u_i \leq u_j \) for any two possible candidates \( i \) and \( j \) to a neighbor switches, but also that these grey numbers are disjoint, i.e., \( \bar{\tau}_i \geq \underline{\tau}_j \). Our setting corresponds to the maximal risk aversion of the agents that agree on a switch of their jobs only if it is surely profitable. Notice that when the uncertainty of costs per unit of time and of processing time is resolved, any of their realizations will belong to the forecasted grey numbers.

We notice that in the classical case the relaxation index is the inverse of the urgency index, so we may reformulate the rule of Smith saying that to obtain an optimal order, the jobs have to be ordered according to increasing grey relaxation indices. Two jobs \( i, j \in N \) may be switched only if \( r_i > r_j \) and the grey numbers are disjoint, i.e., \( \tau_i \geq \tau_j \). Our setting corresponds to the maximal risk aversion of the agents that agree on a switch of their jobs only if it is surely profitable. Notice that when the uncertainty of costs per unit of time and of processing time is resolved, any of their realizations will belong to the forecasted grey numbers.

We notice that the domain of grey sequencing situations under consideration is quite small, containing only situations where all grey urgency indices exist and are pair-wise disjoint (for the grey urgency approach) and situations where all grey relaxation indices exist and are pair-wise disjoint (for the grey relaxation approach). Examples 3, 4 and 5 are inspired by [1] and illustrate situations which cannot be handled by our approaches, whereas Example 6 allows the application of our grey urgency approach.

**Example 3.** Consider the two-agent situation with \( p_1 \in [1, 3], p_2 \in [5, 8], \alpha_1 \in [5, 6], \alpha_2 \in [10, 30] \). Now, \( r_1 \) is defined but \( r_2 \) is undefined; on the other hand \( u_1 \) is undefined and \( u_2 \) is defined. Hence no comparison is possible and, consequently, the reordering cannot take place.

**Example 4.** Consider the two-agent situation with \( p_1 \in [1, 3], p_2 \in [4, 6], \alpha_1 \in [5, 6], \alpha_2 \in [11, 12] \). Here, we can compute \( r_1 \in \left[ \frac{1}{3}, \frac{1}{2} \right] \) and \( r_2 \in \left[ \frac{1}{4}, \frac{1}{2} \right] \), but we cannot reorder the jobs as the grey numbers are not disjoint.
and + 1

\[ G = \max \{ \sigma_0 = \{1, 2, 3, 4\}, \alpha \in ([1, 6], [8, 15], [2, 3], [6, 8]) \} \text{ and } p \in ([1, 3], [2, 3], [6, 12], [2, 4]). \]

We may compute \( u_1 \in [1, 2], u_2 \in [4, 5], r_3 \in [3, 4] \text{ and } r_4 \in [\frac{1}{2}, \frac{1}{2}], \) while the other indices are undefined. Jobs 1 and 2 may be switched and also jobs 3 and 4 may be switched, but we can say nothing about jobs 1 and 4 that become adjacent after the first two switches, as there is no common index.

Example 6. Consider the two-agent situation with \( p_1 \in [1, 4], p_2 \in [6, 8], \alpha_1 \in [5, 25], \alpha_2 \in [10, 30]. \) We can compute \( u_1 \in [5, \frac{25}{4}] \text{ and } u_2 \in [\frac{5}{4}, \frac{15}{4}] \) and use them to reorder the jobs as the grey numbers are disjoint.

Remark 1. The cases where only the processing times or only the costs per unit of time are affected by grey uncertainty are special cases of the general model, since deterministic data can be described as degenerate grey numbers \( a \in [a, a] \in I(\mathbb{R}^+). \)

4. Sequencing situations with grey data. On the domain of sequencing grey situations considered in Section 3, we introduce the grey equal gain splitting rule by extending the equal gain splitting rule for deterministic sequencing situations.

Let \((N, \sigma_0, \alpha', p')\) and \((N, \tau_0, \alpha', p')\) be sequencing grey situations where either all grey urgency indices exist and are pair-wise disjoint or all grey relaxation indices exist and are pair-wise disjoint.

Let \(i, j \in N.\) We define the grey gain of the switch of jobs \(i\) and \(j\) by

\[ G'_{ij} \in \left[ G'_{ij}, \overline{G'_{ij}} \right]. \]

Here, \(G'_{ij}\) and \(\overline{G'_{ij}}\) are defined as follows:

\[ G'_{ij} = \left\{ \begin{array}{ll} \alpha_j p'_i - \alpha_i p'_j, & \text{if jobs } i \text{ and } j \text{ switch,} \\ 0,0, & \text{otherwise.} \end{array} \right. \]

and

\[ \overline{G'_{ij}} = \left\{ \begin{array}{ll} \alpha_j p'_i - \alpha_i p'_j, & \text{if jobs } i \text{ and } j \text{ switch,} \\ 0,0, & \text{otherwise.} \end{array} \right. \]

The grey equal gain splitting (GEGS) rule is defined by

\[ \text{GEGS}(N, \sigma_0, \alpha', p') \]

\[ \in \left[ \frac{1}{2} \sum_{j \in N, j > i} G'_{ij} + \frac{1}{2} \sum_{j \in N, j < i} G'_{ij}, \frac{1}{2} \sum_{j \in N, j > i} \overline{G'_{ij}} + \frac{1}{2} \sum_{j \in N, j < i} \overline{G'_{ij}} \right] \]

for each \(i \in N.\)

Example 7. Consider the sequencing grey situation with \( N = \{1, 2, 3\}, \sigma_0 = \{1, 2, 3\}, p' = ([1, 2], [3, 5], [4, 8]) \text{ and } \alpha' = ([2, 6], [9, 20], [16, 40]). \) The grey urgency indices are \( u'_1 = [2, 3], u'_2 = [3, 4] \text{ and } u'_3 = [4, 5], \) so all jobs may be switched. The grey gains are obtained by:

\[ G'_{12} = \max \left\{ 0, \alpha'_2 p'_1 - \alpha'_1 p'_2 \right\} = \max \left\{ 0, 9 \cdot 1 - 2 \cdot 3 \right\} = \max \{0, 3\} = 3, \]

\[ G'_{12} = \max \left\{ 0, \alpha'_2 p'_1 - \alpha'_1 p'_2 \right\} = \max \left\{ 0, 20 \cdot 2 - 6 \cdot 5 \right\} = \max \{0, 10\} = 10, \]

\[ G'_{12} \in \left[ G'_{12}, \overline{G'_{12}} \right] \implies G'_{12} \in [3, 10]. \]
The other grey gains are calculated similarly as follows:

\[ G_{21}' \in [0, 0], G_{13}' \in [8, 32], G_{31}' \in [0, 0], G_{23}' \in [12, 40], G_{32}' \in [0, 0]. \]

Then, the grey equal gain splitting (\( \mathcal{EGS} \)) rule is

\[
\begin{align*}
\mathcal{EGS}_1 & \in \frac{1}{2} \left[ G_{12}' + G_{13}' \right], \quad \frac{11}{2}, 21, \\
\mathcal{EGS}_2 & \in \frac{1}{2} \left[ G_{12}' + G_{23}' \right], \quad \frac{15}{2}, 25, \\
\mathcal{EGS}_3 & \in \frac{1}{2} \left[ G_{13}' + G_{23}' \right], \quad [10, 36].
\end{align*}
\]

Consequently, the grey equal gain splitting (\( \mathcal{EGS} \)) rule is obtained by

\[ \mathcal{EGS}_i(N, \sigma_0, \alpha', p') \in (\frac{11}{2}, 21, \frac{15}{2}, 25, [10, 36]). \]

5. Cooperative sequencing grey games. In this section, we introduce the class of cooperative sequencing grey games, and prove that the corresponding \( \mathcal{EGS} \) allocation belongs to the grey core of the related sequencing grey game.

The sequencing grey game associated to a one-machine sequencing situation \((N, \sigma_0, \alpha', p')\), with \(\alpha', p' \in \mathcal{G}(\mathbb{R}_+)\), is defined by

\[
w \in \left\{ \sum_{i,j \in N; i < j} G'_{ij} u_{[i,j]}, \sum_{i,j \in N; i < j} \overline{G}'_{ij} u_{[i,j]} \right\},
\]

provided that \(G'_{ij} \in \mathcal{G}(\mathbb{R})\) for all switching jobs \(i, j \in N\). Here, \(u_{[i,j]}\) is the unanimity game defined as:

\[
u_{[i,j]} = \begin{cases} 
1, & \text{if } \{i, i+1, ..., j-1, j\} \subseteq S, \\
0, & \text{otherwise}.
\end{cases}
\]

We notice that the condition \(G'_{ij} \in \mathcal{G}(\mathbb{R})\) is equivalent to \(G'_{ij} \leq \overline{G}_{ij}\). Note that such condition may be not satisfied. Consider for example the sequencing grey situation with \(N = \{1, 2\}, \sigma_0 = \{1, 2\}, p \in (2, 2, 3, 3)\) and \(\alpha \in (2, 4, [12, 13])\). The grey urgency indices are \(u_1 \in [1, 2]\) and \(u_2 \in [4, 13]\), so the switch is profitable, as \(u_2\) is larger than \(u_1\); moreover, the greys are disjoint but \(G_{12} \in [18, 14]\) is not a grey number.

In the following we show that each sequencing grey game is convex.

**Proposition 1.** Let \(< N, w' >\) be a sequencing grey game. Then, \(< N, w' >\) is convex.

**Proof.** By definition \(G_{ij} \succ [0, 0]\). So, \(G_{ij} \geq 0\) and \(|G_{ij}| \geq 0\) for all \((i, j)\). It is well known that classical unanimity games are convex. Then, \(w' = \sum_{i,j \in N; i < j} G_{ij} u_{[i,j]}\) and \(|w'| = \sum_{i,j \in N; i < j} |G_{ij}| u_{[i,j]}\) are convex games, in the classical sense. So, according to Proposition 3.1 in [1],

\[
w' \in \left\{ \sum_{i,j \in N; i < j} G_{ij} u_{[i,j]}, \sum_{i,j \in N; i < j} \overline{G}_{ij} u_{[i,j]} \right\}
\]

is convex. \(\square\)
6. **An Application.** In this section, we apply the Priority Based Scheduling Algorithm which is a non-preemptive algorithm and one of the most common scheduling algorithms in batch systems. Each process is assigned a priority. Process with highest priority is to be executed first and so on. Some systems use low numbers to represent low priority, others use low numbers for high priority. In this scenario, we assume that low numbers represent high priority. Memory requirements, time requirements or any other resource requirement can be parameters of priority.

Consider that we have three departments as Network and Systems Management (D1), Database Management (D2), and Energy Management (D3) in the factory. All departments are connected to Management Unit of Information Technology (see Figure 1).

![Figure 1. An illustration of our application.](image)

In each department, three basic jobs (process) as Network I/O (J1), Disk IO (J2), and CPU (J3) are being run. Jobs are nonpreemptiable, which means that their execution on a processor cannot be suspended until completion. The properties of each jobs of departments are different and can be seen from Table 1, 2, and 3. Consider the following set of jobs, assumed to have arrived at arrival time, in the order of J1, J2, J3 with the length of the execution time and service time in milliseconds. The running times of the processes may be uncertain depending on the variables like work intensity, capacity etc. These uncertainties can be determined by intervals.

**Table 1. The properties of each jobs of D1**

| Job | Arrival Time | Execute Time | Priority | Service Time   |
|-----|--------------|--------------|----------|---------------|
| J1  | [0, 1]       | [2, 2]       | 1        | [95, 101]     |
| J2  | [1, 3]       | [3, 3]       | 2        | [191, 198]    |
| J3  | [3, 4]       | [5, 5]       | 3        | [288, 294]    |
Figure 2. Gantt charts of D1.

Table 2. The properties of each jobs of D2

| Job | Arrival Time | Execute Time | Priority | Service Time |
|-----|--------------|--------------|----------|--------------|
| J1  | 3, 5         | 3, 5         | 2        | 153, 160     |
| J2  | 0, 2         | 4, 6         | 1        | 120, 127     |
| J3  | 6, 8         | 7, 9         | 3        | 186, 193     |

Figure 3. Gantt charts of D2.

Table 3. The properties of each jobs of D3

| Job | Arrival Time | Execute Time | Priority | Service Time |
|-----|--------------|--------------|----------|--------------|
| J1  | 2, 4         | 2, 2         | 2        | 124, 132     |
| J2  | 4, 7         | 3, 3         | 3        | 152, 160     |
| J3  | 0, 3         | 4, 4         | 1        | 90, 98       |

Using priority scheduling, we schedule these jobs according to the following Gantt charts (cf. Figs. 2, 3, 4).

Wait time (service time-arrival time) $t$ of each job of D1, D2, and D3 as follows:
Figure 4. Gantt charts of D3.

Table 4. The wait time $t$ of each jobs of D1, D2 and D3

| Job (Process) | Wait Time |
|---------------|-----------|
| J1 of D1      | $t_{11} = [95, 100]$ |
| J2 of D1      | $t_{12} = [180, 195]$ |
| J3 of D1      | $t_{13} = [285, 290]$ |
| J1 of D2      | $t_{21} = [150, 155]$ |
| J2 of D2      | $t_{22} = [120, 125]$ |
| J3 of D2      | $t_{23} = [180, 185]$ |
| J1 of D3      | $t_{31} = [120, 125]$ |
| J2 of D3      | $t_{32} = [150, 155]$ |
| J3 of D3      | $t_{33} = [90, 95]$ |

Let $c_{i,j} = (C, D, N)$ be the cost vector of running job $i$ on department $j$, with $C$, $D$, $N$ representing the cost of CPU, Disk I/O, and Network I/O, respectively. Let $c_{i,j} = c \cdot C + d \cdot D + n \cdot N$, where $c, d, n$ are the weights of $C, D, N$, respectively. These weights will vary because of the type of the jobs. If the job
is compute-intensive, then $c$ will be 3, $d$ will be 2, and $n$ will be 1. If it is a data parsing job, then $d$ will be 3, $c$ will be 2, and $n$ will be 1. If it is about network, then $n$ will be 3, $d$ will be 2, and $c$ will be 1. Hereby, we can say for D1, the priority is $n > d > c$. This situation is different for D2 and D3. It can be seen from Table 5.

Table 5. The weights of $c, d, n$ of J1 for D1, D2, D3

| Property of job | Compute Intensity | Data parsing | Network |
|-----------------|------------------|--------------|---------|
|                 | cost             | $c$          | $d$     | $n$     |
| J1D1            | 3                | 2            | 1       |
| J2D1            | 2                | 3            | 1       |
| J3D1            | 1                | 2            | 3       |
| J1D2            | 3                | 2            | 1       |
| J2D2            | 1                | 3            | 2       |
| J3D2            | 1                | 2            | 3       |
| J1D3            | 3                | 1            | 2       |
| J2D3            | 2                | 3            | 1       |
| J3D3            | 1                | 1            | 1       |

Let us take $C = [200, 210]$ (MHz), $D = [100, 110]$ (TB), and $N = [50, 60]$ (Mbit). Then, the cost calculations of D1 can be as follows:

- $c_{11} = 1 \cdot [200, 210] + 2 \cdot [100, 110] + 3 \cdot [50, 60] = [550, 610],$
- $c_{12} = 2 \cdot [200, 210] + 3 \cdot [100, 110] + 1 \cdot [50, 60] = [750, 810],$
- $c_{13} = 3 \cdot [200, 210] + 2 \cdot [200, 220] + 1 \cdot [50, 60] = [850, 910].$

The urgency index of job can be obtained by dividing the cost by the wait time:

- $u_{11} = \frac{c_{11}}{t_{11}} = [5.78, 6.1]$, $u_{12} = \frac{c_{12}}{t_{12}} = [3.94, 4.15]$, and $u_{13} = \frac{c_{13}}{t_{13}} = [2.98, 3.13].$

For D1, we see that $u_{11} > u_{12} > u_{13}$. We can calculate the costs of D2.

- $c_{21} = 1 \cdot [200, 210] + 2 \cdot [100, 110] + 3 \cdot [50, 60] = [550, 610],$
- $c_{22} = 1 \cdot [200, 210] + 3 \cdot [100, 110] + 2 \cdot [50, 60] = [600, 660],$
- $c_{23} = 3 \cdot [200, 210] + 2 \cdot [200, 220] + 1 \cdot [50, 60] = [850, 910].$

Now, we can calculate and then examine the priorities of each jobs of D2:

- $u_{21} = \frac{c_{21}}{t_{21}} = [3.66, 3.93]$, $u_{22} = \frac{c_{22}}{t_{22}} = [5, 5.28]$, and $u_{23} = \frac{c_{23}}{t_{23}} = [4.72, 4.91].$

For D2, we see that $u_{22} > u_{23} > u_{21}$. We can calculate the costs of D3.

- $c_{31} = 2 \cdot [200, 210] + 1 \cdot [100, 110] + 3 \cdot [50, 60] = [650, 710],$
- $c_{32} = 2 \cdot [200, 210] + 3 \cdot [100, 110] + 1 \cdot [50, 60] = [750, 810],$
- $c_{33} = 3 \cdot [200, 210] + 2 \cdot [200, 220] = [800, 860].$

Now, we can calculate and then examine the priorities of each jobs of D3:

- $u_{31} = \frac{c_{31}}{t_{31}} = [5.41, 5.68]$, $u_{32} = \frac{c_{32}}{t_{32}} = [5, 5.22]$, and $u_{33} = \frac{c_{33}}{t_{33}} = [8.88, 9.05].$

For D3, we see that $u_{32} > u_{31} > u_{33}$. 

Let us take $C = [200, 210]$ (MHz), $D = [100, 110]$ (TB), and $N = [50, 60]$ (Mbit). Then, the cost calculations of D1 can be as follows:

- $c_{11} = 1 \cdot [200, 210] + 2 \cdot [100, 110] + 3 \cdot [50, 60] = [550, 610],$
- $c_{12} = 2 \cdot [200, 210] + 3 \cdot [100, 110] + 1 \cdot [50, 60] = [750, 810],$
- $c_{13} = 3 \cdot [200, 210] + 2 \cdot [200, 220] + 1 \cdot [50, 60] = [850, 910].$

The urgency index of job can be obtained by dividing the cost by the wait time:

- $u_{11} = \frac{c_{11}}{t_{11}} = [5.78, 6.1]$, $u_{12} = \frac{c_{12}}{t_{12}} = [3.94, 4.15]$, and $u_{13} = \frac{c_{13}}{t_{13}} = [2.98, 3.13].$

Now, we can calculate and then examine the priorities of each jobs of D1.

- $u_{11} = \frac{c_{11}}{t_{11}} = [5.78, 6.1]$, $u_{12} = \frac{c_{12}}{t_{12}} = [3.94, 4.15]$, and $u_{13} = \frac{c_{13}}{t_{13}} = [2.98, 3.13].$
In this way, it is possible to determine which job is priority for each department. Let us construct the game now. We choose J1 job for D1, J2 for D2 and J3 for D3. Firstly, we need to grey gains. Then,

\[
\begin{align*}
c_{11} &= \alpha_1' \in [\alpha_1', \bar{\alpha}_1'] = [550, 610], \\
t_{11} &= p_1' \in [p_1', \bar{p}_1'] = [95, 100], \\
u_{11} &= u_1' \in [u_1', \bar{u}_1'] = [5.78, 6.1], \\
c_{22} &= \alpha_2' \in [\alpha_2', \bar{\alpha}_2'] = [600, 660], \\
t_{22} &= p_2' \in [p_2', \bar{p}_2'] = [120, 125], \\
u_{22} &= u_2' \in [u_2', \bar{u}_2'] = [5, 5.28], \\
c_{33} &= \alpha_3' \in [\alpha_3', \bar{\alpha}_3'] = [800, 860], \\
t_{33} &= p_3' \in [p_3', \bar{p}_3'] = [90, 95], \\
u_{33} &= u_3' \in [u_3', \bar{u}_3'] = [8.88, 9.05].
\end{align*}
\]

The grey gains as follows:

\[
\begin{align*}
G'_{12} &= \left[ G'_{12}, \bar{G}'_{12} \right] ; \\
G'_{12} &= \max \left\{ \alpha_2'p_1' - \alpha_1'p_2', 0 \right\} = \max \left\{ 600 \cdot 95 - 550 \cdot 120, 0 \right\} \\
&= \max \{-9000, 0\} = 0, \\
\bar{G}'_{12} &= \max \left\{ \bar{\alpha}_2'\bar{p}_1' - \bar{\alpha}_1'\bar{p}_2', 0 \right\} = \max \left\{ 660 \cdot 100 - 610 \cdot 125, 0 \right\} \\
&= \max \{-10250, 0\} = 0.
\end{align*}
\]

so, \( G'_{12} \in [0, 0] \).

The other grey gains can be calculated as follows:

\[
\begin{align*}
G'_{21} &= [9000, 10250], G'_{13} = [26500, 28050], G'_{31} \in [0, 0], \\
G'_{13} \in [42000, 44800] \quad \text{and} \quad G'_{31} \in [0, 0].
\end{align*}
\]

**Example 8.** Let \( N = \{1, 2, 3\} \) and the grey coalitional values are

\[
\begin{align*}
w'(1) &= w'(2) = w'(3) \in [0, 0], \\
w'(12) &= \sum_{i \in 2} \sum_{k \in 1} G'_{ki} = G'_{12} \in [0, 0], \\
w'(23) &= \sum_{i \in 3} \sum_{k \in 2} G'_{ki} = G'_{23} \in [42000, 44800], \\
w'(13) &= \sum_{i \in 1} \sum_{k \in 3} G'_{ki} = G'_{31} \in [0, 0], \\
w'(N) &= \sum_{i \in 1} \sum_{k \in 3} G'_{ki} = G'_{31} \in [0, 0].
\end{align*}
\]

Now, we compute the grey equal splitting rule without constructing the game and the grey Shapley value by using cooperative game theory. Firstly, we find the grey
equal splitting rule as follows:
\[
\mathcal{EGS}_1(N, \sigma_0, \alpha', p') \in \left[ \frac{1}{2} (G'_{12} + G'_{13}), \frac{1}{2} (G'_{12} + G'_{13}) \right] = [13250, 14025],
\]
\[
\mathcal{EGS}_2(N, \sigma_0, \alpha', p') \in \left[ \frac{1}{2} (G'_{12} + G'_{23}), \frac{1}{2} (G'_{12} + G'_{23}) \right] = [21000, 22440],
\]
\[
\mathcal{EGS}_3(N, \sigma_0, \alpha', p') \in \left[ \frac{1}{2} (G'_{13} + G'_{23}), \frac{1}{2} (G'_{13} + G'_{23}) \right] = [34250, 36425].
\]
Thus, the grey equal splitting rule is
\[
\mathcal{EGS}_s(N, \sigma_0, \alpha', p') \in ([13250, 14025], [21000, 22440], [34250, 36425]).
\]
Now, we want to compute the grey Shapley value. The grey marginal vectors are given in Table 6, where \( \sigma : N \to N \) is identified with \( (\sigma(1), \sigma(2), \sigma(3)) \). For \( \sigma_1 = (1, 2, 3) \), we calculate the grey marginal vectors. Then,
\[
m_{1}^{\sigma_1}(w') = w'(1) \in [0, 0],
\]
\[
m_{2}^{\sigma_1}(w') = w'(12) - w'(1) \in [0, 0] - [0, 0] = [0, 0],
\]
\[
m_{3}^{\sigma_1}(w') = w'(123) - w'(12) \in [68500, 72850] - [0, 0] = [68500, 72850].
\]
The others can be calculated similarly, which is shown in Table 6.

| \( \sigma \)  | \( m_1^\sigma (w') \)          | \( m_2^\sigma (w') \)          | \( m_3^\sigma (w') \)          |
|----------------|--------------------------------|--------------------------------|--------------------------------|
| \( \sigma_1 = (1, 2, 3) \) | \text{not calculated} | \text{not calculated} | \text{not calculated} |
| \( \sigma_2 = (1, 3, 2) \)  | \text{not calculated} | \text{not calculated} | \text{not calculated} |
| \( \sigma_3 = (2, 1, 3) \)  | \text{not calculated} | \text{not calculated} | \text{not calculated} |
| \( \sigma_4 = (2, 3, 1) \)  | \text{not calculated} | \text{not calculated} | \text{not calculated} |
| \( \sigma_5 = (3, 1, 2) \)  | \text{not calculated} | \text{not calculated} | \text{not calculated} |
| \( \sigma_6 = (3, 2, 1) \)  | \text{not calculated} | \text{not calculated} | \text{not calculated} |

The average of the six grey marginal vectors is the grey Shapley value of this game which can be shown as:
\[
\Phi'(w') \in ([53000, 56100], [179000, 190500], [179000, 190500]).
\]

7. **Conclusion.** In this paper, we study one-machine sequencing situations under grey uncertainty. Under the assumptions of one-machine sequencing situations the \( \mathcal{EGS} \)-rule is introduced as a solution concept. Further, the cooperative game under grey uncertainty is constructed. On the other hand, the grey Shapley value is proposed as a solution concept. It is shown that cooperative grey games are convex. Finally, in addition to our theoretical results, an application based on Priority Based Scheduling Algorithm is given.

Job scheduling is the process of allocating system resources to many different tasks by an operating system (OS). The system handles prioritized job queues that are awaiting CPU time, and it should determine which job to be taken from which queue and the amount of time to be allocated for the job. This type of scheduling makes sure that all jobs are carried out fairly and on time. In scheduling, many different schemes are used to determine which specific job to run. Some parameters which may be considered are as follows: job priority, availability of computing resource, license key if the job is utilizing a licensed software, execution time assigned
to the user, number of parallel jobs permitted for a user, projected execution time, elapsed execution time, presence of peripheral devices, number of cases of prescribed events. Job scheduling problem is studied in various application [3, 4, 8, 10, 11, 15].

CPU scheduling deals with the problem of deciding which of the processes in the ready queue is to be allocated the CPU. There are many different CPU scheduling algorithms [18]. Before construction of the cooperative scheduling a game, we have specified the parameters to be used in the scenario, based on priority scheduling algorithm which is used for CPU Scheduling. A priority is associated with each process, and the CPU is allocated to the process with the highest priority. CPU scheduling is the task of selecting a waiting process from the ready queue and allocating the CPU to it. The CPU is allocated to the selected process by the dispatcher.

REFERENCES

[1] S. Z. Alparslan Gök, R. Branzei, V. Fragnelli and S. Tijs, Sequencing interval situations and related games, CEJOR Cent. Eur. J. Oper. Res., 21 (2013), 225–236.
[2] P. Borm, H. Hamers and R. Hendricks, Operations research games: A survey, Top, 9 (2001), 139–216.
[3] M. E. Bruni, L. D. P. Pugliese, P. Beraldi and F. Guerriero, An adjustable robust optimization model for the resource-constrained project scheduling problem with uncertain activity durations, Omega, (2016), in press.
[4] P. Calleja, M. A. Estevez-Fernandez, P. Borm and H. Hamers, Job scheduling, cooperation, and control, Operations Research Letters, 34 (2006), 22–28.
[5] I. Curiel, G. Pederzoli and S. Tijs, Sequencing games, European Journal of Operational Research, 40 (1989), 344–351.
[6] I. Curiel, H. Hamers and F. Klijn, Sequencing games: A survey, Chapters in Game Theory, Theory Decis. Lib. Ser. C Game Theory Math. Program. Oper. Res., Kluwer Acad. Publ., Boston, MA, 31 (2002), 27–50.
[7] J.-L. Deng, Control problems of grey systems, Systems and Control Letters, 1 (1981/82), 288-294.
[8] D. G. Feitelson, L. Rudolph, U. Schwiegelshohn, K. C. Sevcik and P. Wong, Theory and practice in parallel job scheduling, Workshop on Job Scheduling Strategies for Parallel Processing, Springer Berlin Heidelberg, (1997), 1–34.
[9] E. Kose and J. Y.-L. Forrest, N-person grey game, Kybernetes, 44 (2015), 271–282.
[10] E. L. Lawler, J. K. Lenstra, A. H. R. Kan and D. B. Shmoys, Sequencing and scheduling: Algorithms and complexity, Handbooks in Operations Research and Management Science, 4 (1993), 445–522.
[11] I. S. Lee and S. H. Yoon, Coordinated scheduling of production and delivery stages with stage-dependent inventory holding costs, Omega, 38 (2010), 509–521.
[12] R. E. Moore, Methods and Applications of Interval Analysis, SIAM Studies in Applied Mathematics, 2. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, Pa., 1979.
[13] M. O. Olgun, S. Z. Alparslan Gök and G. Özdemir, Cooperative grey games and an application on economic order quantity model, Kybernetes, 45 (2016), 828–838.
[14] O. Palanci, S. Z. Alparslan Gök, S. Ergün and G.-W. Weber, Cooperative grey games and grey Shapley value, Optimization, 64 (2015), 1657–1668.
[15] R. Ramasesh, Dynamic job shop scheduling: A survey of simulation research, Omega, 18 (1990), 43–57.
[16] S. K. Roy, G. Maity and G.-W. Weber, Multi-objective two-stage grey transportation problem using utility function with goals, CEJOR Cent. Eur. J. Oper. Res., 25 (2017), 417–439.
[17] W. E. Smith, Various optimizer for single-stage production, Naval Research Logistics Quarterly, 3 (1956), 59–66.
[18] W. Stallings and G. K. Paul, Operating systems: Internals and design principles, Upper Saddle River, NJ: Prentice Hall, 3 (1998).
[19] H. Wu and Z. Fang, The graphical solution of zero-sum two-person grey games, *Proceedings of 2007 IEEE International Conference on Grey Systems and Intelligent Services*, 1/2 (2007), 1617–1620.

Received for publication December 2018.

E-mail address: serapbakioglu@isparta.edu.tr
E-mail address: osmanpalanci@sdu.edu.tr
E-mail address: zeynepalparslan@yahoo.com
E-mail address: gerhard.weber@put.poznan.pl
E-mail address: sule_nizamoglu@hotmail.com