Annals of Operations Research 109, 359–374, 2002

Two Approaches to the Problem of Sharing Delay Costs in Joint Projects

RODICA BRÂNZEI * branzeir@infoiasi.ro
Faculty of Computer Science, "Al.I Cuza" University, 11, Carol I Bd., 6600 Iaşi, Romania

GIULIO FERRARI ** ferrarig@dima.unige.it
University of Genova, Department of Mathematics, Via Dodecaneso 35, 16146 Genova, Italy

VITO FRAGNELLI ** fragnell@mfn.unipmn.it
University of Eastern Piedmont, Department of Advanced Sciences and Technologies, Corso Borsalino 54, 15100 Alessandria, Italy

STEF TIJS S.H.Tijs@kub.nl
CentER and Department of Econometrics and Operation Research, Tilburg University, PO Box 90153, 5000 Tilburg, The Netherlands

Abstract. This paper concentrates on cost sharing situations which arise when delayed joint projects involve joint delay costs. The problem here is to determine "fair" shares for each of the agents who contribute to the delay of the project such that the total delay cost is cleared. We focus on the evaluation of the responsibility of each agent in delaying the project based on the activity graph representation of the project and then on solving the important problem of the delay cost sharing among the agents involved. Two approaches, both rooted in cooperative game theory methods are presented as possible solutions. In the first approach delay cost sharing rules are introduced which are based on the delay of the project and on the individual delays of the agents who perform activities. This approach is inspired by the bankruptcy and taxation literature and leads to five rules: the (truncated) proportional rule, the (truncated) constrained equal reduction rule and the constrained equal contribution rule. By introducing two coalitional games related to delay cost sharing problems, which we call the pessimistic delay game and the optimistic delay game, also game theoretical solutions as the Shapley value, the nucleolus and the τ-value generate delay cost sharing rules. In the second approach the delays of the relevant paths in the activity graph together with the delay of the project play a role. A two-stage solution is proposed. The first stage can be seen as a game between paths, where the delay cost of the project has to be allocated to the paths. Here serial cost sharing methods play a role. In the second stage the allocated costs of each path are divided proportionally to the individual delays among the activities in the path.

Keywords: activity graph, delay cost, bankruptcy, taxation, serial cost sharing

AMS subject classification: 90D12, 90B35

* This author acknowledges financial support from CentER and Department of Econometrics and Operation Research, Tilburg University.

** These authors acknowledge financial support from MURST research project "Nuovi problemi di organizzazione e regolamentazione dei servizi pubblici: teoria economica e applicazioni".

1. Introduction

A joint project corresponds to a situation where an objective has to be achieved before an established due date by joint efforts of several firms (agents) with different specialization to carry out specific tasks, according to a certain order induced by technological considerations. If the project is finished after the established date, then a cost is generated according to the contract clauses concerning potential delays.

We deal with the problem: How to divide the delay cost among the agents involved? This paper proposes some delay cost sharing rules based on cooperative game theory methods. Cost sharing problems represent one of the most important applications of cooperative game theory to economic situations. We mention here some important contributions: the Shapley–Shubik method (Shubik [14]), the serial cost sharing method (Moulin and Shenker [6]) and the surveys by Young [19] and Tijs and Driessen [17].

The problem in delay cost sharing situations is to determine "fair" shares for each of the agents involved in causing the delay of the whole project, such that the total delay cost is cleared. We are interested in the evaluation of the responsibility of each agent in delaying the project, and then in the difficult problem of the delay cost sharing among the agents.

The outline of the paper is as follows. In section 2 we introduce definitions of different kinds of delays.

Section 3 presents our first approach to delay cost sharing problems. Five interesting division rules inspired by bankruptcy problems (O'Neill [8], Aumann and Maschler [1], Curiel et al. [2]) and taxation problems (Young [18]) are introduced, namely: the proportional rule PROP, the truncated proportional rule TPROP, the constrained equal reduction rule CER, the truncated constrained equal reduction rule TCER and the constrained equal contribution rule CEC. Two dual coalitional games related to delay problems are constructed: the pessimistic and optimistic delay game. It is proved that the pessimistic delay game is concave and that the solutions provided by all the delay division rules (PROP, TPROP, CER, TCER, CEC) are located in the large core. Also game theoretical solutions as the Shapley value, the nucleolus and the τ-value generate cost sharing rules.

Section 4 deals with our second approach. Here a two-stage procedure is developed to allocate the project delay cost. In the first stage we consider a game whose players are the relevant paths of the activity graph describing the joint project. The related cost allocation rule coincides with the weighted Shapley value and it is inspired by the serial cost sharing methods (Moulin and Shenker [6], Tijs and Koster [16], Koster [4]). In the second stage the cost allocated to each relevant path is divided over the activities in that path proportionally to the delays of the corresponding activities.

Examples in section 5 allow us to illustrate and compare the different delay cost allocation rules. Finally in section 6 some concluding remarks are made.

2. Basic elements

A joint project consists of a set of activities for which the estimated durations and the precedence relations are known. The most simple description of a joint project is given by a table.

Consider the project shown in table 1.

From table 1 we see that activity C is preceded by the activities A and B, that is A and B must be completed before C starts.

Given the precedence order and estimated duration of activities, several different graphical representations of the project can be generated. One example is an *activity graph* (Malcom et al. [5]) where the arcs correspond to the activities and the nodes represent stages corresponding to the finishing time for some activities (entering activity arcs) and the starting time for some other activities (outgoing activity arcs); two special nodes α and ω (representing the starting and the finishing of the project) are included, and each arc is labelled with the name of the activity and estimated duration. Activity graphs provide insight into the precedence relationships of the activities which are not intuitively obvious. An activity graph shows which activities can be carried out in parallel and which must be executed in sequence because of a dependency on a previous activity. The joint project described in table 1 can be graphically represented by the activity graph given in figure 1.

From now on we use activity graphs to describe joint projects.

In this paper we study how to distribute the cost generated by a delay in a joint project among the agents involved. This study concerns different categories of joint projects such as: the construction of buildings, the manufacture of sophisticated goods, the delivery of software systems, etc.

When someone is interested in achieving such an objective he usually submits his request to a contractor for an opportunity study that can result in a contract. Among other things the contract contains information related to the starting time, the delivery date and clauses concerning the delay penalties. Supposing that the project starts at instant 0, let $[0, E]$ be the allowed execution time interval for the project and k the delay cost function defined on $[E, +\infty)$, with $k(E) = 0$, that contains all the information of the generated penalties.

Table 1

Activity	Duration (time units)	Previous activities
A	5	–
B	7	–
C	4	A, B

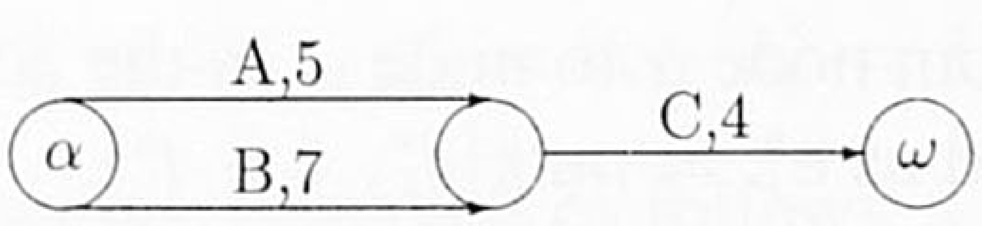

Figure 1.

To carry out the project the contractor has to contact some executing firms according to the necessities which emerge from the description of the project and then to solve the project scheduling problem.

The project scheduling determines when each activity is scheduled to begin and finish and assesses when the project will be optimally completed. Let us denote by b_i and e_i the planned beginning and ending times of activity i.

Various software tools are now widely available on personal computers which automate the tasks of activity graph generation and project scheduling. The Critical Path Method (CPM) estimates the minimal duration of the project by considering one of the longest paths in the activity graph (a critical path). For details we refer to the book of Gondran and Minoux [3].

Our approach to the project scheduling problem results in a feasible plan together with the agreement made with the agents regarding the execution of the project and the penalties in case of delay.

A plan is called feasible when for each activity i the estimated duration a_i is equal to $e_i - b_i$ and the planned execution time interval of the project $[S, T]$ is included in the allowed execution time interval $[0, E]$, where

$$S = \min\{b_i,\ i \in N\}, \quad T = \max\{e_i,\ i \in N\} \quad \text{and} \quad N = \{1, 2, \dots, n\}$$

is the set of activities.

When no availability constraints for the firms executing the project exist, the optimal feasible plan corresponds to the critical path in the CPM method and it lies in the interval $[0, T]$. Our setting is more general in order to model more realistic situations in which agents may have availability constraints (for example because they are involved in other projects).

The agreement between the contractor and each agent i with respect to the execution of his task and his punishment in case of delay can be stated informally as follows:

- agent i is not allowed to start his activity before all the preceding activities (according to the activity graph) are completed; he is not obliged to start before his planned starting time;
- if agent i finishes not later than the planned beginning time of his followers then agent i will be not punished;
- if agent i finishes later than the planned beginning time of his followers but he does not increase the tardiness that the preceding activities caused to him, he will not be punished.

After the completion of the project the actual ending time of the activities, denoted by e_i^*, $i \in N$, and, consequently, also the actual ending time of the project, denoted by E^* are known, where $E^* = \max\{e_i^*,\ i \in N\}$.

Also for each path P from node α to node ω in the activity graph the actual finishing time e_P^* can be determined as $e_P^* = \max\{e_i^*,\ i \in P\}$.

Many joint projects are frequently delayed, that is they terminate after the planned completion time, because of unanticipated problems, errors in estimating durations of activities, the knock-on effect of unexpected delays when an agent is involved in other projects, etc., that result in tardiness in the starting time or slowness in the execution of the activities.

For a delayed project the deviations from the plan have to be evaluated and used in order to obtain "fair" cost shares for each of the agents who contribute to the delay of the project so that the total delay cost is cleared.

In the following we use for any $x \in R$ the notation x_+ for $\max\{0, x\}$. Further we use the notation $i < j$ to indicate that activity i has to precede activity j.

The *delay of the project*, denoted by D, measures the deviation from the due date E and it is given by $D = (E^* - E)_+$.

For each $i \in N$ the *delay of activity* i, denoted by d_i, measures the responsibility of i in delaying the followers, by taking into account the tardiness caused by i and its predecessors to the followers and the tardiness caused to i by its predecessors, according to the formula

$$d_i = \left((e_i^* - b_{+i})_+ - (e_{-i}^* - b_i)_+\right)_+.$$

Here $b_{+i} = \min\{b_j \mid j > i\}$ if activity i has at least one follower and $b_{+i} = T$ otherwise; $e_{-i}^* = \max\{e_j^* \mid j < i\}$ if activity i has at least one predecessor and $e_{-i}^* = S$ otherwise. So, $(e_i^* - b_{+i})_+$ refers to the disturbance caused to the activities following i, while $(e_{-i}^* - b_i)_+$ expresses the disturbance caused to i by the previous activities.

Remark 2.1. Note that $d_i = 0$ if and only if $(e_i^* - b_{+i})_+ = 0$ (i.e., the agent i finishes not later than the planned beginning time of his followers) or if $(e_i^* - b_{+i})_+ > 0$ and $(e_i^* - b_{+i})_+ \leqslant (e_{-i}^* - b_i)_+$ (i.e., the agent i finishes later than the planned beginning time of his followers but he does not increase the tardiness that the preceding activities caused to him). This matches the requests for punishment in the agreement.

The *delay of path* P, denoted by d_P, measures the deviation of P from the due date E of the project and it is given by $d_P = (e_P^* - E)_+$.

The *aggregated delay of path* P, denoted by D_P is given by $D_P = \sum_{i \in P} d_i$. It plays a role in our second approach.

In the following example we illustrate the previous definitions of delays.

Example 2.1. Consider the joint project in figure 1 together with the planned and actual processing information for activities as follows:

$$E = 13, \quad b_A = 1, \quad e_A = 6, \quad b_B = 0, \quad e_B = 7, \quad b_C = 7, \quad e_C = 11$$

(so, $S = 0$ and $T = 11$);

$$e_A^* = 12, \quad e_B^* = 9, \quad e_C^* = 17.$$

The delays of the activities can be computed as follows:

- $d_A = ((12 - 7)_+ - (0 - 1)_+)_+ = 5;$

- $d_B = ((9-7)_+ - (0-0)_+)_+ = 2$;
- $d_C = ((17-11)_+ - (12-7)_+)_+ = 1$.

The project had to be finished at 13, but it lasted till 17, so its delay is $D = 4$.

There are two paths: A–C and B–C. According to the previous definitions we have

$$d_{A-C} = (17-13)_+ = 4, \qquad D_{A-C} = 5+1 = 6;$$
$$d_{B-C} = (17-13)_+ = 4, \qquad D_{B-C} = 2+1 = 3.$$

Remark 2.2. In this example the condition $D \leqslant \sum_{i \in N} d_i$ holds; this condition is verified for all delayed joint projects; the proof is a bit technical but not difficult, so we leave it to the reader.

3. Cost sharing rules based on individual delays

Our starting point here is a joint project, as described earlier, where a set of agents $N = \{1, \ldots, n\}$ is involved and a delay of the project $D > 0$ is generated, resulting in a cost $k(D)$, according to a cost function k. The question how to share these costs is tackled in this section by taking into consideration only the individual delays $d_1, \ldots, d_n$ and the delay D. Note that, according to our previous remark,

$$0 < D \leqslant \sum_{i=1}^{n} d_i. \tag{3.1}$$

We will introduce some interesting *delay division rules* f, which assign to each *delay problem* (D, d) a vector $f(D, d)$ in $\mathbb{R}^n$ such that

$$0 \leqslant f_i(D, d) \leqslant d_i \quad \text{for each } i \in N, \tag{3.2}$$

$$\sum_{i=1}^{n} f_i(D, d) = D. \tag{3.3}$$

Given the rule f, $f_i(D, d)$ can be interpreted as the part of the delay D for which i is made responsible. We call $f_i(D, d)$ the *delay share of player* i with respect to f. The delay cost allocation $(x_1, \ldots, x_n)$ of the problem (D, d), corresponding to the rule f, is then given by

$$x_i = D^{-1} k(D) f_i(D, d) \quad \text{for each } i \in N. \tag{3.4}$$

According to (3.4), the delay cost $k(D)$ is divided among the agents proportionally to their delay shares $f_i(D, d)$, $i = 1, \ldots, n$.

Note that our approach here to delay problems is similar to treatments of bankruptcy problems (O'Neill [8], Aumann and Maschler [1], Curiel et al. [2]) and taxation problems (Young [18]).

A bankruptcy problem is described by a pair (E, d) where E is the estate and $d = (d_i)_{i \in N}$, where d_i as the claim of customer i, with $0 < E \leqslant \sum_{i=1}^{n} d_i$. Here the problem is how to divide the estate E among the claimants.

A taxation problem is described by a pair (T, m), where T is the amount of tax to be generated and $m = (m_i)_{i \in N}$, where m_i as the income of agent i, with $0 < T \leqslant \sum_{i=1}^{n} m_i$. Because of the similarities between the three problems, we can and will profit for our delay division problem from the other two problems. So, the following is really inspired by the bankruptcy and taxation literature.

Let us start by introducing five interesting and somewhat familiar rules, namely: the proportional rule PROP, the truncated proportional rule TPROP, the constrained equal reduction rule CER, the truncated constrained equal reduction rule TCER and the constrained equal contribution rule CEC.

(i) The ith coordinate of PROP(D, d) is given by

$$\mathrm{PROP}_i(D, d) = \left(\sum_{i=1}^{n} d_i \right)^{-1} d_i D, \quad i = 1, \ldots, n.$$

According to the proportional rule, player i has to contribute $D^{-1}\mathrm{PROP}_i(D, d)k(D)$ in the delay problem (D, d). So, the delay D is divided among the players proportionally to their individual delays.

(ii) Related to the proportional rule is TPROP, where the individual delay of each player is reduced to D, if it is larger, and the other individual delays are kept. So the new delay of player i is given by $d_i^T = \min\{d_i, D\}$, called the *truncated delay of player i*. Then

$$\mathrm{TPROP}_i(D, d) = \mathrm{PROP}_i\big(D, d^T\big) = l\left(\sum_{i=1}^{n} d_i^T \right)^{-1} d_i^T D, \quad i = 1, \ldots, n.$$

(iii) The ith coordinate of CER(D, d) is given by

$$\mathrm{CER}_i(D, d) = \max\{d_i - \beta, 0\}, \quad i = 1, \ldots, n,$$

where β is the unique real number so that $\sum_{i=1}^{n} \mathrm{CER}_i(D, d) = D$.
So, the constrained equal reduction rule assigns to the players with $d_i \geqslant \beta$ a delay share obtained by reducing the individual delay with β, while for the other players, with $d_i < \beta$, the delay share is 0.

(iv) The truncated constrained equal reduction rule is defined by

$$\mathrm{TCER}_i(D, d) = \mathrm{CER}_i\big(D, d^T\big), \quad i = 1, \ldots, n.$$

(v) The ith coordinate of CEC(D, d) is given by

$$\mathrm{CEC}_i(D, d) = \min\{d_i, \alpha\},$$

where α is the unique real number so that $\sum_{i=1}^{n} \mathrm{CEC}_i(D, d) = D$.

So, the constrained equal contribution rule assigns to the players with $d_i \geqslant \alpha$ a delay share of α, while for the other players, with $d_i < \alpha$, the delay share is equal to their individual delay.

Another way to obtain interesting solutions is to introduce cooperative games related to delay problems and consider game related solutions.

We introduce two coalitional games: the pessimistic delay game $\langle N, c_{(D,d)}\rangle$, and the optimistic delay game $\langle N, c^*_{(D,d)}\rangle$.

The *pessimistic delay game* corresponding to the delay problem (D, d) is defined by

$$c_{(D,d)}(S) = \min\left\{\sum_{i\in S} d_i,\ D\right\} \quad \text{for each coalition } S \subset N,$$

and the *optimistic delay game* by

$$c^*_{(D,d)}(S) = \max\left\{D - \sum_{i\in N\setminus S} d_i,\ 0\right\} \quad \text{for each coalition } S \subset N.$$

The first game is called pessimistic because $c_{(D,d)}(S)$ assigns a maximal responsibility of the delay D to the members of S, where they become responsible for the whole delay D if $\sum_{i\in S} d_i \geqslant D$ and otherwise each $i \in S$ contributes for d_i.

For the second game, the players outside S are maximally responsible (for at most d_i) and the rest is the responsibility of S.

The two games are dual to each other because

$$c^*_{(D,d)}(S) = c_{(D,d)}(N) - c_{(D,d)}(N \setminus S) \quad \text{for each } S \subset N.$$

In theorem 3.1 we prove that $\langle N, c_{(D,d)}\rangle$ is a *concave game*, i.e., for all $S \subset T$ and $j \in N \setminus T$ we have

$$c_{(D,d)}\big(S \cup \{j\}\big) - c_{(D,d)}(S) \geqslant c_{(D,d)}\big(T \cup \{j\}\big) - c_{(D,d)}(T). \tag{3.5}$$

This implies that the *core*, defined as

$$\mathrm{Core}(c_{(D,d)}) = \left\{x \in \mathbb{R}^n \;\middle|\; \sum_{i=1}^n x_i = c_{(D,d)}(N) \text{ and } \sum_{i\in S} x_i \leqslant c_{(D,d)}(S),\ \forall S \subset N\right\}$$

is large (Shapley [11]). It turns out that all the delay division rules, satisfying (3.2) and (3.3), assign to the delay problem (D, d) a core element of $\langle N, c_{(D,d)}\rangle$ as theorem 3.2 shows.

Theorem 3.1. Given a delay problem (D, d), the coalitional game $\langle N, c_{(D,d)}\rangle$ is a concave game.

Proof. Take S, T, j such that $S \subset T \subset N \setminus \{j\}$. We have to prove (3.5) or equivalently

$$c_{(D,d)}\big(S \cup \{j\}\big) + c_{(D,d)}(T) \geqslant c_{(D,d)}\big(T \cup \{j\}\big) + c_{(D,d)}(S). \tag{3.6}$$

Note that

$$c_{(D,d)}(S \cup \{j\}) + c_{(D,d)}(T) = \min\left\{\sum_{i\in S} d_i + d_j + \sum_{i\in T} d_i, \sum_{i\in S\cup\{j\}} d_i + D, D + \sum_{i\in T} d_i, 2D\right\}, \tag{3.7}$$

$$c_{(D,d)}(T \cup \{j\}) + c_{(D,d)}(S) = \min\left\{\sum_{i\in T} d_i + d_j + \sum_{i\in S} d_i, D + \sum_{i\in T\cup\{j\}} d_i, D + \sum_{i\in S} d_i, 2D\right\}. \tag{3.8}$$

Since

$$\sum_{i\in S\cup\{j\}} d_i \geqslant \sum_{i\in S} d_i \quad \text{and} \quad \sum_{i\in T} d_i \geqslant \sum_{i\in S} d_i$$

we conclude that (3.7) and (3.8) imply (3.6). □

Theorem 3.2. For each delay division rule f and each delay problem (D, d), we have that $f(D, d) \in \mathrm{Core}(c_{(D,d)})$.

Proof. From (3.1) and (3.3) it follows that

$$\sum_{i=1}^{n} f_i(D, d) = D = \min\left\{\sum_{i\in N} d_i, D\right\} = c_{(D,d)}(N).$$

Furthermore, for each $S \subset N$ it follows from (3.2) and (3.3) that

$$\sum_{i\in S} f_i(D, d) \leqslant \sum_{i\in S} d_i, \qquad \sum_{i\in S} f_i(D, d) \leqslant \sum_{i\in N} f_i(D, d) = D.$$

So,

$$\sum_{i\in S} f_i(D, d) \leqslant \min\left\{\sum_{i\in S} d_i, D\right\} = c_{(D,d)}(S).$$

Hence, $f(D, d) \in \mathrm{Core}(c_{(D,d)})$. □

Now we concentrate on game theory related delay division rules, which we can obtain as follows. Let ψ be a one-point solution concept, which assigns to each non-negative monotonic concave game $\langle N, c\rangle$ a non-negative vector $\psi(N, c)$ in the *imputation set* $I(N, c)$ of the game, where

$$I(N, c) = \left\{x \in \mathbb{R}^n \;\middle|\; \sum_{i=1}^{n} x_i = c(N) \text{ and } x_i \leqslant c(\{i\}),\ \forall i \in N\right\}.$$

Consider the map f^{ψ} which assigns to the delay problem (D, d) the vector $\psi(N, c_{(D,d)})$; this means that f^{ψ} assigns to (D, d) the ψ-value of the corresponding pessimistic delay game.

It is easy to show that f^{ψ} satisfies (3.2) and (3.3). So, f^{ψ} is a delay division rule generated by ψ. We conclude the section by mentioning some results without proof. The Shapley value (Shapley [10]) and the nucleolus (Schmeidler [9]) generate delay division rules. The τ-value for cost games (Tijs and Driessen [17]) generates also a delay division rule f^{τ}. The ith coordinate of $f^{\tau}(D,d)$ is given by

$$f_i^{\tau}(D,d) = \gamma c_{(D,d)}(\{i\}) + (1-\gamma)c^*_{(D,d)}(\{i\}),$$

where γ is the unique real number such that $\sum_{i=1}^{n} f_i^{\tau}(D,d) = D$.

So, f^{τ} is the feasible compromise between the minimal duty vector $\{c^*_{(D,d)}(\{i\})\}_{i\in N}$ and the maximal contribution vector $\{c_{(D,d)}(\{i\})\}_{i\in N}$.

Note that a game theory related rule f^{ψ} has the *truncation property*, that is

$$f^{\psi}(D,d) = f^{\psi}(D,d^T) \quad \text{for all } (D,d).$$

This follows from

$$c_{(D,d)}(S) = \min\left\{\sum_{i\in S} d_i,\, D\right\} = \min\left\{\sum_{i\in S} \min\{d_i, D\},\, D\right\} = c_{(D,d^T)}(S).$$

Also TPROP, TCER and CEC have the truncation property, but PROP and CER have not.

In a similar way as above, also game-theoretical solutions arising from a class of non-negative convex games containing the optimistic delay games can generate delay division rules.

4. Cost sharing rules based on delays of paths

The approach to delay cost sharing problem presented in this section is inspired by serial methods developed for cost sharing problems for machine use for joint production (Shenker [12,13], Moulin and Shenker [6,7], Tijs and Koster [16], Koster [4]).

In these methods the ordering of the demands by size is important.

Here we suggest a two stage procedure to allocate delay cost shares to the activities responsible for the tardiness of a joint project.

In the first stage a weighted serial cost sharing method is applied for allocating cost shares to the relevant paths by considering the delays of the paths as demands and the aggregated activity delays of the paths as weights. In the second stage we allocate the delay cost share of each path among its activities proportionally to their delays. If an activity is included in more than one path we aggregate its cost shares resulting from the different paths.

We denote by $\mathcal{P}$ the set of all *relevant paths*, i.e., $\mathcal{P} = \{P \mid d_P > 0 \text{ and } D_P > 0\}$.

Note that a path is relevant only if it is finished after the due date E and it contains at least one activity with positive delay.

Let p be the number of relevant paths. For each path $P \in \mathcal{P}$ we compute the actual completion time e^*_P, its delay d_P and its aggregated activity delay D_P.

Let $t_1, t_2, \ldots, t_q \in (E, E^*]$ be the different completion times of the relevant paths. Note that $q \leqslant p$ as different paths may have the same completion time (for example all those that have the same last activity).

In the time interval $[E, E^*]$ we consider the intervals $i_j = [t_{j-i}, t_j]$, $j = 1, \ldots, q$, where $t_0 = E$ and $t_q = E^*$ and compute the corresponding delay cost $k(t_j) - k(t_{j-i})$.

Stage 1.

Step 1. Each path $P \in \mathcal{P}$ contributes to the delay cost of all the intervals i_j such that $t_j \leqslant e_P^*$. As a result the delay cost of each interval i_j is divided among the corresponding paths proportionally to their weights D_P.

Step 2. For each path we aggregate its delay cost shares obtained from the different time intervals i_j.

Stage 2.

Step 1. For each path j, with $j = 1, \ldots, p$ allocate its delay cost share to the activities in the path proportionally to their delay.

Step 2. For each activity aggregate its delay cost shares resulting from the different paths to which the activity belongs.

Remark 4.1. Our proposal takes the sum as the aggregation rule for obtaining the delay cost shares of the paths involved in several time intervals i_j and for computing the delay cost shares of activities belonging to several paths.

Remark 4.2. In step 1 of stage 2 we propose to divide the delay cost shares of each relevant path among its activities according to their delays d_i; another possibility is to divide it according to their relative delays, i.e., the ratio among the delay d_i and the duration a_i. This second way can be interesting if the project includes activities with very short and very long durations.

Remark 4.3. Stage 1 can be seen as a cooperative game between relevant paths. The weighted serial cost allocation rule coincides with the related weighted Shapley value for this game. To be more concrete, we consider a TU-game $\langle \mathcal{P}, c\rangle$ whose players are the relevant paths in the activity graph and for each non-empty subset S of $\mathcal{P}$ we have $c(S) = k(d_S)$, where d_S is the maximum of the delays of the paths in S. See also Potters and Sudhölter [15]. The game $\langle P, c\rangle$ is concave, so its core is large, and the weighted Shapley value is a core element.

Although the two-stage approach will be illustrated by examples in section 5, we consider it is useful to illustrate only stage 1 now.

Example 4.1. Consider the joint project represented by the activity graph in figure 2. There are the following five paths: A–D, A–E–H, B–H, C–G, C–F–H. Let $t^* = e_D^*$ be the completion time of A–D, $t_1 = e_G^*$ the completion time of C–G and $t_2 = e_H^*$ the

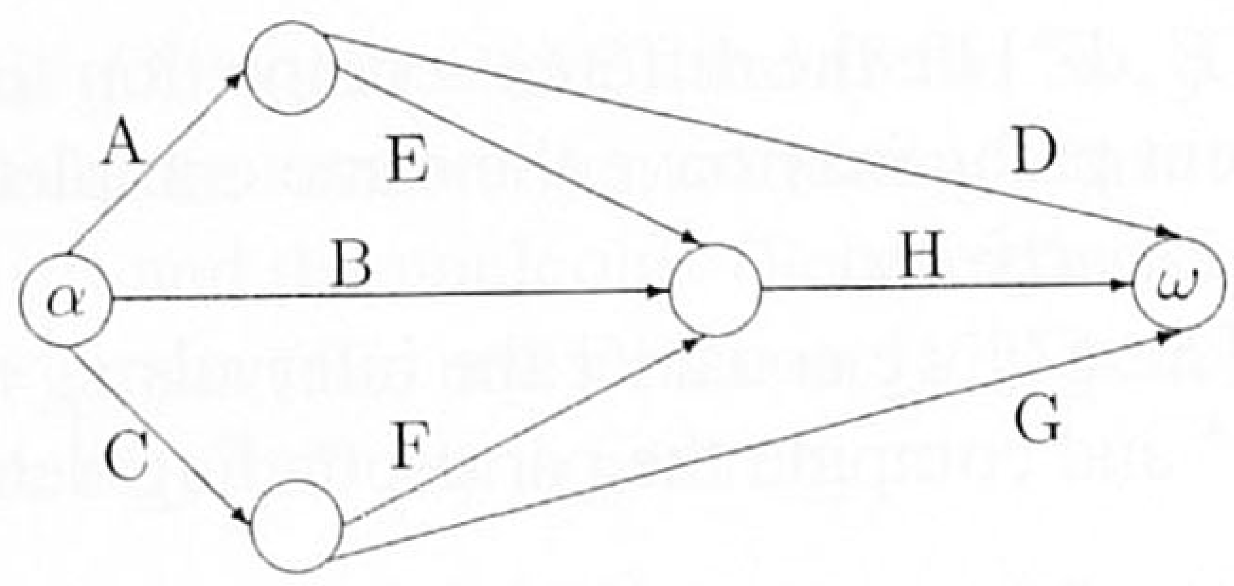

Figure 2.

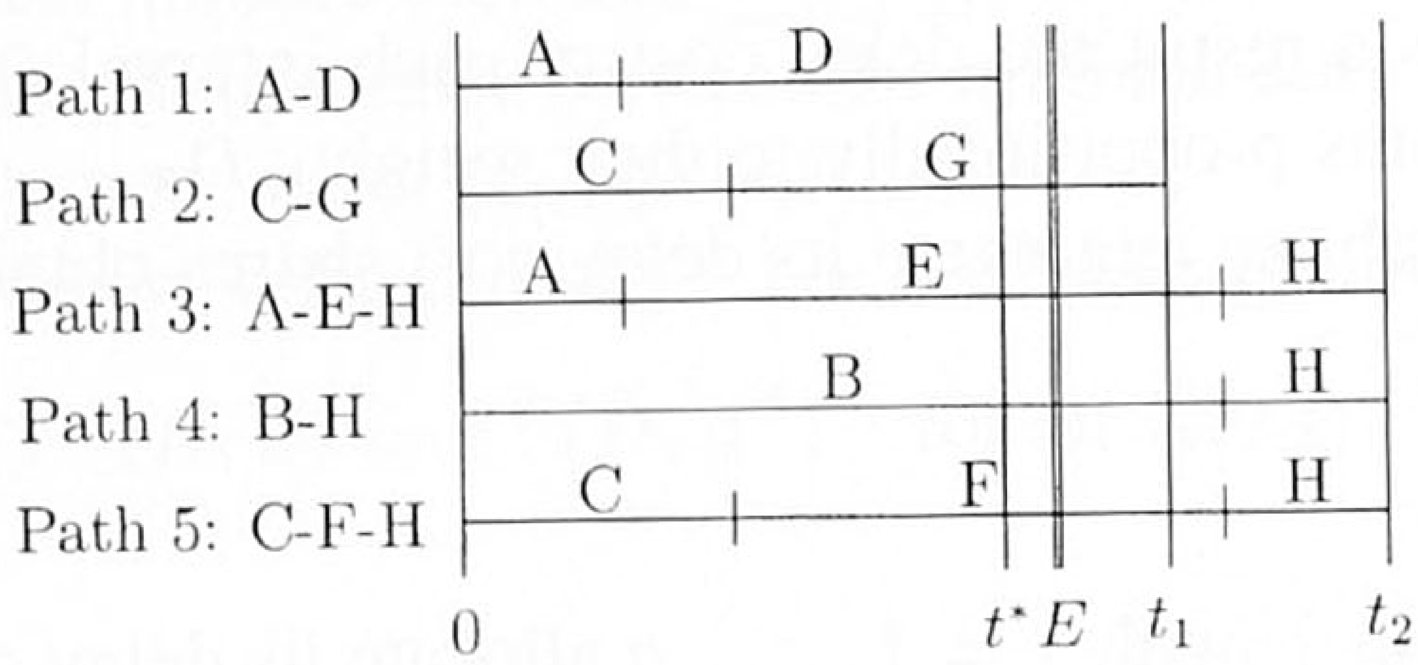

Figure 3.

completion time of A–E–H, B–H, C–F–H, with $t^* < E < t_1 < t_2$. This situation is graphically represented in figure 3. According to our approach the cost corresponding to the delay $t_1 - E$ is divided among paths 2, 3, 4, 5, proportionally to their weights (aggregated delays) and the cost induced by the delay $t_2 - t_1$ is split among paths 3, 4, 5, again proportionally to their weights.

5. Examples

This section includes three examples which enlighten some characteristics of the two approaches, proposed solutions and activity graph structures.

Example 5.1 reconsiders the joint project from example 2.1 and computes all the proposed solutions. In examples 5.2 and 5.3 projects with basic structures of activity graphs (line-graph and graph consisting of parallel activities, respectively) are considered and the usefulness of the two approaches is discussed.

Example 5.1. We reconsider the joint project from example 2.1 and compute all the delay cost allocation solutions in the two approaches when the cost function is $k(t) = ((t - E)_+)^2$, where t is the project completion time. First we apply cost sharing rules based on individual delays. Using the delays computed in example 2.1 one gets the delay problem (4;5,2,1). The corresponding delay allocations are shown in table 2. The cost to be divided is $k(17) = ((17 - 13)_+)^2 = 16$ and the corresponding delay cost shares are given in table 3. Now, the second approach uses the path delays and weights already computed in example 2.1. One can see that the two paths A–C and B–C are both responsible for the delay so the cost of 16 is divided proportionally to their weights of 6 and 3, respectively. The path A–C is charged of 10.67 and the path B–C of 5.33; the cost of A–C is divided among A and C proportionally to their delay of 5 and 1, respectively,

Table 2

Method	Player		
	A	B	C
PROP	2.50	1.00	0.50
TPROP	2.29	1.14	0.57
CER	3.50	0.50	0.00
TCER	3.00	1.00	0.00
CEC	1.50	1.50	1.00

Table 3

Method	Player		
	A	B	C
PROP	10.00	4.00	2.00
TPROP	9.14	4.57	2.29
CER	14.00	2.00	0.00
TCER	12.00	4.00	0.00
CEC	6.00	6.00	4.00

Table 4

Player	A	B	C
Path method	8.89	3.55	3.55

so A is charged of 8.89 and C of 1.78; analogously the cost of B–C is divided among B and C proportionally to their delays of 3 and 1, respectively, so B is charged of 3.55 and C of 1.78. Finally we obtain the aggregated cost penalties as shown in table 4.

Example 5.2. Consider the joint project with the activity graph in figure 4 and the following planned and actual processing information:

$$E = 9, \quad b_A = 0, \quad e_A = 3, \quad b_B = 3, \quad e_B = 6, \quad b_C = 6, \quad e_C = 9,$$
$$e_A^* = 6, \quad e_B^* = 8, \quad e_C^* = 13.$$

The delays of the activities are 3, 0, 2, respectively and the delay of the project is 4. Despite the simple structure of the activity graph, the approach inspired by bankruptcy and taxation problems is not trivial. It generates the delay problem (4;3,0,2). Table 5 shows the corresponding delay shares. However, the two-stage approach is trivial (there is one path!).

Example 5.3. Consider a project consisting of two parallel activities as figure 5 shows and the following planned and actual processing information:

$$E = 7, \quad b_A = 0, \quad e_A = 3, \quad b_B = 0, \quad e_B = 6, \quad e_A^* = 8, \quad e{*}_B = 7.$$

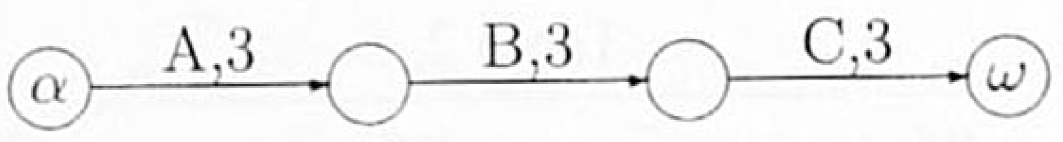

Figure 4.

Table 5

Method	Player		
	A	B	C
PROP = TPROP	2.40	0.00	1.60
CER = TCER	2.50	0.00	1.50
CEC	2.00	0.00	2.00

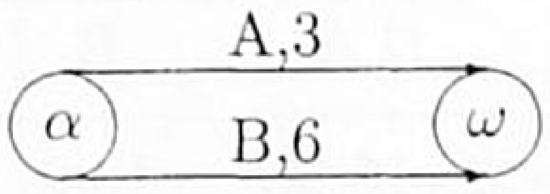

Figure 5.

Table 6

Method	Player	
	A	B
PROP	0.67	0.33
TPROP	0.50	0.50
CER	1.00	0.00
TCER	0.50	0.50
CEC	0.50	0.50

The delays of the activities are 2 and 1, respectively and the delay of the project is 1. In this case the approach based on individual delays divides the delay as it is shown in table 6. In the serial approach there are two one-activity paths. The path consisting of activity A has a delay $d_A = 1$ and an aggregated delay $D_A = 2$, while the path consisting of activity B has a delay $d_B = 0$ and an aggregated delay $D_B = 1$ so this second path is not relevant. Consequently the whole cost is charged to the activity (path) A. Notice that the CER rule of the first approach gives the same result.

Remark 5.1. Real-life situations imply large projects with many sequential and parallel activities. Complex activity graphs result by combining the basic structures considered in examples 5.2 and 5.3. Example 5.1 provides a standard structure with both sequential and parallel activities. See example 4.1 for a more interesting activity graph structure.

Remark 5.2. From the computational point of view the solutions based on individual delays behave better; if the activity graph contains many paths it can be very difficult and time consuming to identify and analyse all the relevant paths, even if in some situations their number can be quite small. On the other hand, solutions based on delays of paths allow us to take into account the different role played by each activity when it belongs to several paths; sometimes too much responsibility is probably assigned to some activities in this approach.

6. Concluding remarks

In this paper we presented two approaches to penalty cost sharing in delayed joint projects. Both of them use the activity graph to describe joint projects. The first approach is activity oriented, the second is path (in the activity graph) oriented. In both approaches corresponding cooperative games are constructed where the players are the activities in the first case and the paths in the (first stage of the) second case. Our implicit assumption was that different activities are carried out by different agents. The extension to the case when some multispecialized agents, that can execute more than one activity, are involved in the project can be easily obtained by charging each such an agent with the sum of the corresponding activity delay cost shares.

Acknowledgments

The authors gratefully acknowledge the anonymous referee for valuable comments on the earlier version of the paper and some useful discussions on relations between PERT graphs and Game Theory with Gustavo Bergantiños-Cid and Estela Sanchez-Rodriguez.

References

[1] R.J. Aumann and M. Maschler, Game theoretic analysis of a bankruptcy problem from the Talmud, J. Economic Theory 36 (1985) 195–213.
[2] I.J. Curiel, M. Maschler and S.H. Tijs, Bankruptcy games, Zeitschrift für OR 31 (1987) 143–159.
[3] M. Gondran and M. Minoux, *Graphs and Algorithms* (Wiley Interscience, New York, 1984).
[4] M. Koster, Cost sharing in production situations and network exploitation, Ph.D. Dissertation, Tilburg University (1999).
[5] D.G. Malcom, J.H. Roseboom, C.E. Clark and W. Fazar, Application of a technique for research and development program evaluation, Operations Research 7 (1959) 646–669.
[6] H. Moulin and S. Shenker, Serial cost sharing, Econometrica 60 (1992) 1009–1037.
[7] H. Moulin and S. Shenker, Average cost pricing versus serial cost sharing: An axiomatic comparison, J. Economic Theory 64 (1994) 178–201.
[8] B. O'Neill, A problem of rights arbitration from the Talmud, Math. Soc. Sci. 2 (1982) 345–371.
[9] D. Schmeidler, The nucleolus of a characteristic function game, SIAM J. Appl. Math. 17 (1969) 1163–1170.
[10] L.S. Shapley, A value for n-person games, Ann. Math. Studies 28 (1953) 307–317.
[11] L.S. Shapley, Cores of convex games, Internat. J. Game Theory 1 (1971) 11–26.
[12] S. Shenker, Efficient network allocations with selfish users, in: *Performance '90*, eds. P. King, I. Mitrani and R. Pooley (North-Holland, New York, 1990).
[13] S. Shenker, Making greed work in networks: a game-theoretic analysis of switch service disciplines, IEEE/ACM Transactions on Networking 3 (1995) 819–831.

[14] M. Shubik, Incentives, decentralized control, the assignement of joint cost, and internal pricing, Management Science 8 (1962) 325–343.

[15] J. Potters and P. Sudhölter, Airport problems and consistent allocation rules, Math. Soc. Sci. 38 (1999) 83–102.

[16] S. Tijs and M. Koster, General aggregation of demand and cost sharing, Annals of Operations Research 84 (1998) 137–164.

[17] S. Tijs and T. Driessen, Game theory and cost allocation problems, Management Science 32 (1986) 1015–1028.

[18] H.P. Young, On dividing an amount according to individual claims or liabilities, Mathematics of Operations Research 12 (1987) 398–414.

[19] H.P. Young, Cost allocation, in: *Handbook of Game Theory*, Vol. II, eds. R.J. Aumann and S. Hart (North-Holland, Amsterdam, The Netherlands, 1994) pp. 1193–1235.