CentER

Discussion paper

No. 2001-22

TWO APPROACHES TO THE PROBLEM OF SHARING DELAY COSTS IN JOINT PROJECTS

By Rodica Brânzei, Giulio Ferrari, Vito Fragnelli and Stef Tijs

March 2001

ISSN 0924-7815

Two approaches to the problem of sharing delay costs in joint projects

Running title: Sharing delay costs

Rodica Brânzei[1]
Faculty of Computer Science, "Alexandru Ioan Cuza" University, 11, Carol I Bd., 6600 Iaşi, Romania, E–mail: branzeir@infoiasi.ro

Giulio Ferrari[2]
University of Genova, Department of Mathematics (DIMA), Via Dodecaneso 35, 16146 Genova, Italy, E-mail: ferrarig@dima.unige.it

Vito Fragnelli[2]
University of Eastern Piedmont, Department of Advanced Sciences and Technologies (DiSTA), Corso Borsalino 54, 15100 Alessandria, Italy, E-mail: fragnell@mfn.unipmn.it

Stef Tijs
CentER and Department of Econometrics and Operations Research, Tilburg University, P.O. Box 90153, 5000 LE Tilburg, The Netherlands, E-mail: S.H.Tijs@kub.nl

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 and complicated 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

[1]This author acknowledges financial support from CentER and Department of Econometrics and Operations Research, Tilburg University.

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

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 constrained equal contribution rule, the (truncated) proportional rule and the (truncated) constrained equal reduction 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 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 to each path are divided proportionally w.r.t. the individual delays among the activities in the path.
Key words: activity graph, bankruptcy problem, taxation problem, serial cost sharing method, delay cost.
AMS classification: 90D12, 90B35

1 Introduction

Many real-life projects consist of different activities that require specialized individuals or groups of individuals to be accomplished. We refer to such a project as a joint project and call the specialists (or the groups) involved agents. The successful execution of a joint project depends on good scheduling and problem anticipation. The output from the scheduling process is an activity graph showing the activities making up the project, the estimated durations of the activities and their dependencies, starting and finishing dates for each activity and the whole project.

Many joint projects are frequently delayed because of unanticipated problems, errors in estimating the duration of activities, the knock-on effect of unexpected delays when a specialist is involved in several scheduled projects, etc. If the project's delay generates a cost, a special cost sharing problem arises.

This paper presents two approaches to delay cost sharing problems, both of them rooted in cooperative game theory. 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 cost sharing among the agents.

The activity graph representing the project and some information obtained from the project scheduling are basic elements for our purpose. We pay attention to them in Section 2, introducing also the definitions of some kinds of delays and durations that play an important role in our approaches.

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 paths of the activity graph describing the joint project. The related cost allocation rule is inspired by the cost sharing methods (Moulin and Shenker [6], Tijs and Koster [16], Koster [4]). In the second stage the cost allocated to each path is divided over the activities in that path proportionally w.r.t. the absolute and relative delays of the corresponding activities.

In Section 5 some examples are given for illustrating the solutions proposed in Sections 3, 4 and the basic activity graph structures; they provide support for a comparison of the two approaches.

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 inter–dependencies are known.

Consider the joint project shown in Table 2.1.

Activity	Duration (time units)	Dependencies
A	5	
B	7	
C	4	A,B

Table 2.1

From Table 2.1 we see that activity C is dependent on 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 the finishing time for some activities (the entering activity arcs) and the starting time for some other activities (the 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 its estimated duration. Activity graphs provide insights 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 an earlier activity. The joint project described in Table 2.1 can be graphically represented by the activity graph given in Figure 2.1.

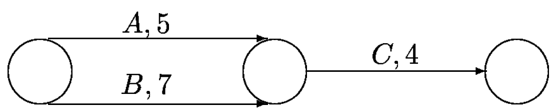

Figure 2.1

Activity graphs are used in project scheduling and in our approaches to delay cost sharing problems related to delayed joint projects.

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 moment for activity i. We call the interval $[b_i, e_i]$ the *planned processing interval of activity* i and denote it by PPI_i. The *duration of the project* is the minimum time required to complete the project. Supposing that the project starts at the moment zero, its duration gives the finishing moment of the project.

Various software tools are now widely available on personal computers which automate the task of activity graph generation and project scheduling. The Critical Path Method (CPM) estimates the 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].

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.

For a delayed joint project the actual processing interval for each activity is known, determining also the actual finishing moment of the project. The difference between the actual finishing moment of the project and the planned one gives the *project delay* D. We denote by $API_i = [b_i^*, e_i^*]$ the *actual processing interval for activity* i, where b_i^* and e_i^* denote the actual beginning and ending moment, respectively.

The activity graph of the joint project, the planned and actual processing intervals for each activity represent all the information we need for the two approaches to delay cost sharing problem.

For each activity i, the *planned (estimated) duration* du_i and the actual duration du_i^* are given by

$$du_i = e_i - b_i; \; du_i^* = e_i^* - b_i^*.$$

Unforeseen delays of activities should be evaluated and taken into account for defining the *responsible duration* rdu_i, that is the duration for which activity i is responsible. Then the *responsible delay of activity* i, denoted by d_i, is defined as

$$d_i = \max\{0, rdu_i - du_i\} = (rdu_i - du_i)_+.$$

We distinguish the following cases:

Case 1. The activity i starts not later than its planned beginning time, that is $b_i^* \leq b_i$. Then

$$rdu_i = e_i^* - b_i, \text{ and } d_i = (e_i^* - e_i)_+.$$

Case 2. The activity i has a starting delay sd_i, that is $b_i^* > b_i$. Two subcases arise depending on the activity guilty for the starting delay:

- 2a) The starting delay $sd_i = b_i^* - b_i$ is due only to the activity i. Then $rdu_i = du_i^* + sd_i = e_i^* - b_i^* + b_i^* - b_i = e_i^* - b_i$, and $d_i = (e_i^* - e_i)_+$;

- 2b) The whole starting delay or part of it is due to some of the preceding activities.

  Let $\bar{b}_i$ be the earliest moment when the activity i could start, i.e. the moment when all the preceding activities are terminated. So

  $$\bar{b}_i = \max\{e_j^* \mid j < i\},$$

  where $j < i$ means that activity j precedes i in the activity graph. The responsible starting delay for i is $rsd_i = b_i^* - \bar{b}_i$, which implies that the responsible duration of i is $rdu_i = du_i^* + rsd_i = e_i^* - b_i^* + b_i^* - \bar{b}_i = e_i^* - \bar{b}_i$. Then the delay for which i is responsible is given by

$$d_i = (rdu_i - du_i)_+ = ((e_i^* - \bar{b}_i) - (e_i - b_i))_+ = ((e_i^* - e_i)_+ - (\bar{b}_i - b_i)_+)_+.$$

The following example illustrates the previous definitions.

Example 2.1. Consider the joint project in Figure 2.1 together with the planned and actual processing intervals for activities as follows:

$$PPI_A = [0,5]; \quad PPI_B = [0,7]; \quad PPI_C = [7,11];$$
$$API_A = [3,10]; \quad API_B = [0,9]; \quad API_C = [12,15].$$

The responsible durations and delays of the activities can be computed as follows:

- activity A is in case 2a), so $rdu_A = 10-0 = 10$ and $d_A = (10-5)_+ = 5$;

- activity B is in case 1, so $rdu_B = 9$ and $d_B = (9-7)_+ = 2$;

- activity C is in case 2b); we compute $\bar{b}_C = \max\{10, 9\} = 10$. Then $rdu_C = 15-10 = 5$ and $d_C = ((15-11)_+ - (10-7)_+)_+ = (4-3)_+ = 1$.

The project had to be finished at 11, but it lasted till 15, so its delay is $D = 4$.

In our second approach the durations and delays of paths in the activity graph play an important role, too.

A *path* (from α to ω) in the activity graph is a sequence of activities for which the first node of the first activity is α and the second node of the last activity is ω.

The (responsible) duration of a path is the sum of the responsible durations of the activities in that path.

The delay of a path is defined as the difference between its (responsible) duration and the minimum time required for the execution of the project, if it is positive; otherwise the delay of a path is zero.

Consider again the delayed joint project given in Example 2.1. The minimum required time of the project is 11. There are two paths: A-C and B-C. According to the previous definitions and Example 2.1 we have

$$\begin{aligned} du_{A-C} &= rdu_A + rdu_C = 10 + 5 = 15, \\ du_{B-C} &= rdu_B + rdu_C = 9 + 5 = 14; \\ d_{A-C} &= (du_{A-C} - 11)_+ = (15 - 11)_+ = 4, \\ d_{B-C} &= (du_{B-C} - 11)_+ = (14 - 11)_+ = 3. \end{aligned}$$

3 Cost sharing rules based on individual delays

Our starting point here is a joint project, as described earlier, where a set of players $N = \{1, ..., n\}$ 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 $d_1, ..., d_n$ and the aggregate delay D. Note that

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

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

$$0 \leq f_i(D, d) \leq d_i, \text{ for each } i \in N = \{1, 2, ..., n\}, \tag{3.2}$$

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

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 w.r.t.

f. The delay cost allocation $(x_1, ..., x_n)$ of the problem (D, d), corresponding to the rule f, is then given by

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

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

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 is a vector with d_i as the claim of customer i, with $0 < E \leq \sum_{i=1}^{n} d_i$. 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 is a vector with m_i as the income of agent i, with $0 < T \leq \sum_{i=1}^{n} m_i$.

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 i-th coordinate of $\text{PROP}(D, d)$ is given by

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

According to the proportional rule, player i has to contribute $\text{PROP}_i(D, d)k(D)$ in the delay problem (D, d). So, the (aggregate) 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

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

(iii) The i-th coordinate of $\text{CER}(D,d)$ is given by

$$\text{CER}_i(D,d) = \max\{d_i - \beta, 0\},$$

where β is the unique real number so that

$$\sum_{i=1}^{n} \text{CER}_i(D,d) = D.$$

So, the constrained equal reduction rule assigns to the players with $d_i \geq \beta$ 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

$$\text{TCER}_i(D,d) = \text{CER}_i(D,d^T),\ i = 1, ..., n.$$

(v) The i-th coordinate of $\text{CEC}(D,d)$ is given by $\text{CEC}_i(D,d) = \min\{d_i, \alpha\}$, where α is the unique real number so that

$$\sum_{i=1}^{n} \text{CEC}_i(D,d) = D.$$

So, the constrained equal contribution rule assigns to the players with $d_i \geq \alpha$ 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 $< N, c_{(D,d)} >$ and the optimistic delay game $< N, c^*_{(D,d)} >$.

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

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

and the *optimistic delay game* by

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

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 aggregate delay D if $\sum_{i\in S} d_i \geq D$ 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

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

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

$$(3.5) \qquad c_{(D,d)}(S \cup \{j\}) - c_{(D,d)}(S) \geq c_{(D,d)}(T \cup \{j\}) - c_{(D,d)}(T).$$

This implies that the *core*, defined as

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

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

Theorem 3.1. *Given a delay problem* (D,d)*, the coalitional game* $< N, c_{(D,d)} >$ *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

$$(3.6) \qquad c_{(D,d)}(S \cup \{j\}) + c_{(D,d)}(T) \geq c_{(D,d)}(T \cup \{j\}) + c_{(D,d)}(S).$$

Note that

$$c_{(D,d)}(S \cup \{j\}) + c_{(D,d)}(T) =$$

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

$$c_{(D,d)}(T \cup \{j\}) + c_{(D,d)}(S) =$$

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

Since $\sum_{i \in S \cup \{j\}} d_i \geq \sum_{i \in S} d_i, \; \sum_{i \in T} d_i \geq \sum_{i \in S} d_i$, 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

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

So, $\sum_{i \in S} f_i(D, d) \leq \min \left\{ \sum_{i \in S} d_i, D \right\} = c_{(D,d)}(S)$. 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 $< N, c >$ a non–negative vector $\psi(N, c)$ in the *imputation set* $I(N, c)$ of the game, where

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

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 i-th 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

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

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 the 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 sizes of demands is important.

These methods can be applied to the penalty cost sharing problem related to delayed joint projects by considering delays of paths as demands and by using a two–stage procedure.

In the first stage, we consider a TU-game whose players are the paths in the activity graph; delay cost shares for paths are provided. The second stage allocates the cost share of each path to its activities and then aggregates the induced costs for each activity according to its inclusion in different paths.

Given the activity graph representing the joint project, the responsible duration rdu_i for each activity i, the actual project completion time t_0 and the cost function k, our approach can be described as follows:

Stage 1.

Step 1. Generate the list of paths based on the activity graph: the activites are aggregated w.r.t. the paths from α to ω. Let p be the number of paths;

Step 2. For each path j, $j = 1, ..., p$, compute the (responsible) duration and delay of j;

Step 3. Sort the paths according to the increasing order of path duration;

Step 4. Let $t_1, t_2, ..., t_q$, with $t_1 < t_2 < ... < t_q$, be the different path completion times. Note that $q \leq p$ because different paths may require the same completion time. Divide the set of paths in subsets depending on the different completion times of the paths;

Step 5. Generate the intervals $i_j = [t_{j-1}, t_j], j = 1, ..., q$, and for each interval i_j:

- compute the corresponding delay cost contribution $k(t_j)-k(t_{j-1})$;
- assign the paths ℓ with $t_\ell \geq t_j$ to i_j;
- divide equally the delay cost contribution of i_j over the assigned paths.

Stage 2.

Step 1. For each path j, with $j = 1, ..., p$ allocate its delay cost share to the activities in the path proportionally w.r.t. their (absolute) delay d_i or the relative delay rd_i of i, defined as $rd_i = d_i/du_i$, where du_i is the planned duration of activity i;

Step 2. For each activity i 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 the different activities in a path, as the definition of path duration shows. The sum is also used for aggregating the delay cost shares allocated to an activity belonging to several paths.

Remark 4.2. In Step 1 of Stage 2 we propose a proportional allocation of path delay cost among the activities in a path. The proportional division can be done either w.r.t. the delay d_i or the relative delay rd_i. The former variant is better for short activities, which could be too much charged by using the other variant, while the second variant provides no incentives for longer activities to recover small delays.

Although this approach will be entirely illustrated by examples in Section 5, we consider it useful to partially illustrate Stage 1 now.

Example 4.1. Consider the joint project represented by the activity graph in Figure 4.1.

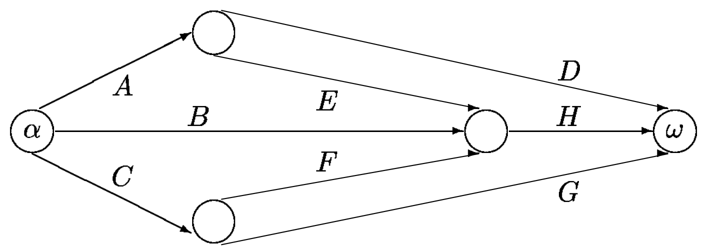

Figure 4.1.

Remark 4.3. The stage illustrated in Example 4.1 can be seen as a cooperative game between paths. The serial cost allocation rule coincides with the Shapley value in this game. To be more concrete, the involved game $< P, c >$ has the set P of paths as player set and for each non–empty subset Q of paths $c(Q) = k(d_Q)$, where d_Q is the maximum of the delays of the paths in Q. See also Potters and Sudhölter [15].

There are the following five paths: A-D, A-E-H, C-G, C-F-H, B-H. Let t_1 be the completion time of A-E-H, t_2 the completion time of C-G and C-F-H, t_3 the completion time of B-H, with $t_0 < t_1 < t_2 < t_3$, and suppose that the path A-D terminates earlier than the minimum required time t_0 for the project. This situation is graphically represented in Figure 4.2.

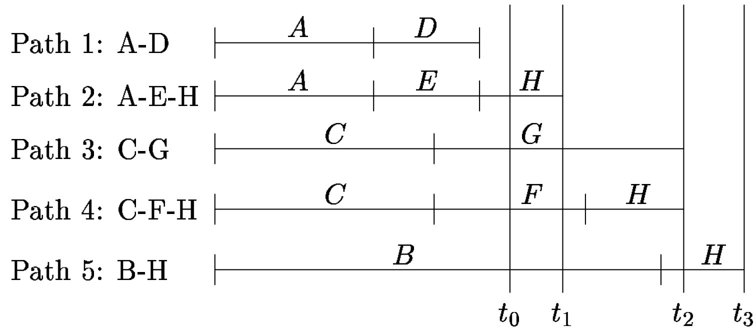

Figure 4.2

According to our approach the cost corresponding to the delay $t_1 - t_0$ is equally divided among paths 2, 3, 4, 5; the cost induced by the delay $t_2 - t_1$ is equally split among paths 3, 4, 5, and the cost corresponding to the delay $t_3 - t_2$ is charged to path 5.

5 Examples

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

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 (a line–graph and a 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 $k(t) = ((t-11)_+)^2$, 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 5.1 and the delay cost shares are given in Table 5.2.

Method \ Player	1	2	3
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 5.1

Method \ Player	1	2	3
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 5.2

Now, the second approach uses the path durations and delays already computed in Example 2.1. One can see that the two paths are equally responsible for the first three units of delay, and the path A-C is the only responsible for the last delay unit. The first three units of delay generate a cost of $(14-11)_+)^2 = 9$, which is equally split between A-C and B-C, while the last unit causes a cost of $((15-11)_+)^2 - 9 = 7$, which is charged to A-C. As a result, a cost of $9/2 + 7 = 11.50$ is allocated to A-C and $9/2 = 4.50$ is allocated to B-C.

The cost of each path is proportionally divided over the corresponding activities both w.r.t. the (absolute) delay and relative delay. Activity A as part of path A-C gets a share of 9.58 and 9.20, respectively. Activity B as part of path B-C receives an allocation of 3 and 2.40, respectively. Activity C is part of the both paths. As part of path A-C it receives the allocations of 1.92 w.r.t. its (absolute) delay and 2.30 w.r.t. its relative delay. As part of B-C it receives the shares of 1.50 and 2.10, respectively. By aggregating these delay cost shares, activity C gets allocations of 3.42 and 4.40, respectively. All these delay cost allocations are synthesized in Table 5.3.

Method \ Player	1	2	3
Absolute Delay	9.58	3.00	3.42
Relative Delay	9.20	2.40	4.40

Table 5.3

Example 5.2. Consider the joint project with activity graph in Figure 5.1 and the following planned and actual processing intervals:

$$PPI_A = [0,3]; \quad PPI_B = [3,6]; \quad PPI_C = [6,9];$$
$$API_A = [1,6]; \quad API_B = [6,8]; \quad API_C = [8,13];$$

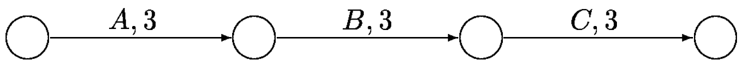

Figure 5.1

The minimum required time to complete the project is 9, the completion time 13, so the project delay is 4. The responsible durations and delays of the activities are 6, 2, 5 and 3, 0, 2, respectively. 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 5.4 shows the corresponding delay shares.

Method \ Player	1	2	3
PROP=TPROP	2.40	0.00	1.60
CER=TCER	2.50	0.00	1.50
CEC	2.00	0.00	2.00

Table 5.4

However, the approach based on path delays is trivial (there is one path!).

Example 5.3. Consider a joint project consisting of two parallel activities as Figure 5.2 shows.

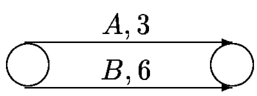

Figure 5.2

The planned and actual processing intervals of the activities are:

$$PPI_A = [0,3];\; PPI_B = [0,6];\; API_A = [0,5];\; API_B = [0,7].$$

The responsible durations and delays of the activities are 5,7 and 2,1, respectively; the project delay is 1. In this case the approach based on individual delays divides the project delay as it is shown in Table 5.5.

Player / Method	1	2
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

Table 5.5

It is easy to see that the first activity has a larger delay and, consequently, more responsibility in the delay of the project.

In the serial approach there are two one–activity paths. The path consisting of activity A has no delay (it terminates before the minimum time required), so all the cost is charged to the activity B (this is a critical activity!).

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 paths. 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 represent the joint project and some information provided by project scheduling with the Critical Path Method (CPM). 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.

Acknowledgements. The authors gratefully acknowledge 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*, PhD 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, Op. Res. 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. Econ. Th. 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, Int. 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, USA, 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 assignment 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 OR 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, Math. Op. Res. 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.