Cooperative project scheduling with controllable processing times: a game theory framework

Cyril Briand, Jean-Charles Billaut

Cooperative project scheduling with controllable processing times: a game theory framework

Cyril Briand [1,2] and Jean-Charles Billaut [1,2,3]
[1] LAAS-CNRS ; Université de Toulouse ; 7, avenue du Colonel Roche,
F-31077 Toulouse Cedex 4, France.
[2] Université de Toulouse ; UPS, INSA, INP, ISAE ; UT1, UTM, LAAS ;
F-31077 Toulouse Cedex 4, France
[3] Universite Francois Rabelais Tours ; Laboratoire d'Informatique ;
64 avenue Jean Portalis, F-37200 Tours, France.
{cyril.briand, jean-charles.billaut}@laas.fr

Abstract

This paper considers a project-scheduling environment assuming that the activities of the project network are distributed among a set of actors (or agents). Activity durations are modeled as time intervals and are assumed controllable, meaning that every actor is allowed to shorten the duration of some activities by adding extra-money. For performing the project, actors have to collaborate with each other intending to satisfy the desired project duration, defined by the project customer as a time interval. In this work, every actor's payoff corresponds to a fixed percentage of the total customer's payment, which itself depends on the ability of the actors to achieve the project in time, provided daily penalty costs are applied in case of tardiness. This problem can be modeled as a game, where players (actors) have to select a strategy (a duration vector for their activities) intending to maximize their profit. In this paper, the focus is put on the modeling of this project game, and on the connections between various decision problems, arising either in decision or game theory. We also study the particular case where each activity is assigned to one specific agent.

1. Introduction

Many large-size projects are cooperative in the sense that they involve a set of actors (or organizations), each one being in charge of the execution of a part of a project. Actors have their own decisional autonomy, their specific competencies and have to collaborate together for supplying a product or a service to a customer. In such a context, project management becomes challenging since each actor is willing to collaborate with the others, but his outcome not only depends on his own decision strategies, but also on those of his partners [12].

The customer's payment is usually due by the end of the project (even if payments on account can be planned at various phases of the project execution). Its amount generally depends on the ability of the whole set of actors to fulfill a set of quality and efficiency requirements. In this paper, we assume that the customer's payment is shared among the actors accordingly to collective agreement that have been contracted during the design phase of the actors' network [6].

Such a framework appears in many realistic business context such that building trade [18], supply chain networks [10], or automotive industries [19]. Sometimes, a consortium of actors is collectively built up for facilitating the project operations, the customer relationship, as well as the actors' cost/profit sharing. Actually, actor's payoffs can depend on several agent's parameters, more or less objective, such as the place of each actor inside the project network, the agent's financial effort put into executing the project, the agent's economic or politic weight. The situation can be also very different depending on the consortium power. In aeronautics for instance, the consortium often plays both a financial, politic and economic role. It has a great decision power and can impose many constraints on the way actors have to work and interact together. In other context, such as the one of supply chain management, the decisional autonomy of actors can be much more strong, and the consortium just aims at putting some directions to promote efficient interaction rules and eventually favor fair cooperative behaviors.

For defining delivery and payment modalities, customer and actors negotiate together, possibly via the consortium. Here again the economic and politic power of each entities plays a major role in the decision-making process. Usually a fixed price is defined for the product or the service, together with penalties and/or rewards [11] that mostly depend on the ability of the actors to deliver the service or the product, in compliance with the initial requirements, at the right moment and the right place.

The framework depicted here can also be viewed as a particular project game where players, corresponding to actors, play together for performing a project. Their strategies correspond to the various possibilities they have for carrying out their activities. Their outcome depends on their collective ability to fulfill the customer's requirements. They play simultaneously and, given their strategy choices, the customer's payment can be determined in a deterministic way, as well as the agent's payoff. This link with game theory will be further expanded in the paper.

In the scope of this paper, the project is composed by a set of interdependent activities submitted to precedence relations and distributed among actors. Actors can control the activities durations: they are allowed to reduce the activity processing time by devoting additional resources to the activity, typically at a higher cost. In this setting, the actor's strategy is defined by the duration vector the actor chooses for his activities. Knowing every actor strategy, a simple longest path algorithm gives the completion times of the project, from which the customer's payment can be deduced. We assume that the agent's payoff is a fixed portion of this total payment.

The rest of the paper is organized as follows: Section 2 gives some insights into works connected to the scope of this paper. Sections 3 and 4 define formally the problem and propose a graphical model. Section 5 discusses some interesting decision or optimization problems. Section 6 analyzes the links with game theory and attached concepts thoroughly. The last section is devoted to the special case, where each activity is assigned to a specific agent. In that case, we show how stable actor's strategies can be characterized.

2. Literature review

This section aims at stressing the various connections of this work with some other existing research areas. First, in the field of project scheduling, many textbooks on production/operations management present the so-called crashing algorithm which is used in the Critical Path Method. Crashing refers to shortening the duration of an activity by devoting additional resources to the activity, typically at a higher cost. For each activity that can be crashed, a time/cost tradeoff function is specified , which is usually linear or piecewise linear. The overall objective is to determine the time/cost trade-off curve which specifies, for any possible total project duration, the minimum incurred crashing cost. From an algorithmic point of view, given a project duration upper-bound, the problem of determining the activity durations that minimize the crashing cost is equivalent to a minimum cost flow problem [16, 21], which can be solved in polynomial time. The literature is rich of papers that consider equivalent problem assuming more complex activity time/cost functions, *e.g.*, convex, discrete, that make the problem harder to solve [14]. As far as we know, multi-actor context has never been considered in any of these papers (the crashing costs being always supported by a unique decision-maker).

An important branch of the industrial engineering literature focuses on Supply-Chain Management (SCM), which precisely studies planning or scheduling problems arising up in organizations composed of several decision makers, members of the same supply-chain network, each with his own decisional autonomy, all being involved in a common production process. In this kind of organization, decision-maker can have different role, such as supplier, customer or subcontractor. The literature is rich of papers that propose approaches for enhancing the communication, the coordination and the cooperation among actors. In such works, a crucial element is the confidential aspect of the data handled by each actor: only a restricted set of information is shared amongst them. This set often contains the minimum amount of information needed to coordinate the actors. If we consider the case of a customer/supplier relationship, a commonly used coordination mode is the one in which actors exchange propositions and counter-propositions related to the completion times of their tasks, this negotiation being conducted as long as a compromise satisfying every agent is not found. In order to avoid frequent modifications of the planned decisions, agents can exchange intervals (instead of instant times), these intervals becoming more and more precise as the decision time is getting close to the execution of the tasks [10, 22, 5].

Other seminal works in the field of production scheduling focus on multi-agent scheduling paying attention in the multi-objective nature of the problem (brought by the agent concept). Most of these works propose centralized mathematical solving approach or exact algorithm for finding solutions that optimize the agent's objective. This kind of multi-agent scheduling problematic was introduced in the works of Agnetis, Mirchandani, Pacciarelli and Pacifici [1, 2] in which the authors consider a job shop problem where two agents, each being in charge of a set of jobs, are competing for the use of resources (the machines) while both trying to optimize their own objective function (the difference between both objective functions being ϵ-constrained). Several articles [3, 7, 8] focus on the case of multi-agent single machine problems with two or more competing agents and provide exact methods along some results on complexity, considering various criteria such as the makespan, the algebraic lateness or the average number of tardy tasks. Some other works are concerned about multi-agent scheduling in the field of the computing grid. In this case, each agent is associated to a computing cluster and negotiate with the others on which tasks it should undertake, these tasks possibly being affected to other agents (see [20] for a recent approach).

In scheduling, the concept of agent is also at the very heart of distributed solving methods, especially those based on the Multi-Agent Systems paradigm (MAS) [13]. In the field of job shop scheduling, heuristic approaches, based on MASs, have been put forward [23] in order to generate feasible schedules: the agents, with associated tasks, negotiate their processing intervals with resources agents, a supervisor agent taking care of potential conflicts. Moreover, we can find MAS based approaches in multi-project scheduling [9, 15] where agents, associated to the projects, share the resources and are guided by a manager agent. For a more thorough description of existing MAS approaches in planing and scheduling of manufacturing systems, the reader should refer to the survey by Shen, Wang and Hao [24].

In the field of cooperative game theory, some authors consider the project scheduling environment for exhibit some particular games. Most of them take an interest in the problem of sharing rewards and penalties between agents, assuming that activities can be disrupted or can last shorter than expected [11]. In multi-agent scheduling, several authors also consider the connection between the optimization of one global objective function, also called social objective, and game theory. Important concepts, linked to the non cooperative game theory, has been defined such as the *price of anarchy* [17] (ratio between the worst Nash equilibrium and the optimal value of the global objective function) or the *price of stability* [4] (ratio between the best Nash equilibrium and the optimal value of the global objective function).

In this paper, the focus is on multi-agent project-scheduling, which can be viewed as a special field of multi-agent scheduling, where resource constraints are not considered explicitly. Like in CPM method, the assumption that activity processing times are controllable is made, so that a time/cost tradeoff has to be achieved. The project-customer payment depends on the time performance of the agents since daily penalties are applied in case of lateness. The problem is to characterize the strategies that are of interest for the agents.

From the best of our knowledge, it is the first time such a problem is studied in these terms. This paper proposes a formal model and stresses some decision problems connected both with decision and game theories.

3. Problem statement

The project is composed by a set $\mathcal{T} = \{0, \ldots, n+1\}$ of activities (or tasks), that are shared among a set $\mathcal{A} = \{A_0, \ldots, A_{m+1}\}$ of agents. Classically, activities are linked together by a set $\mathcal{P}$ of precedence constraints: $(i, j) \in \mathcal{P}$ means that i precedes j. The set of activities assigned to the activity-agent A_u is denoted $\mathcal{T}_u$. By convention, 0 and $n+1$ are dummy activities representing the beginning and the end of the project, respectively. We assume that they are assigned to two fictitious agents A_0 and A_{m+1}, such that $\mathcal{T}_0 = \{0\}$ and $\mathcal{T}_{m+1} = \{n+1\}$. A_0 and A_{m+1} can be viewed as the project launcher and the project customer, respectively. In the sequel of this paper, we consider that the project-customer agent defines a delivery interval for the project, further refers to as $[\underline{D}, \overline{D}]$. In a similar way, the project-launcher agent could choose a release date interval $[\underline{R}, \overline{R}]$ for the project. In this work, it is assumed w.l.o.g. that $\underline{R} = \overline{R} = 0$.

To each activity i is associated a minimum and a maximum processing time denoted by $\underline{p}_i$ and $\overline{p}_i$, respectively. Any agent A_u has to choose, for all his activities $i \in \mathcal{T}_u$, a duration p_i belonging to $[\underline{p}_i, \overline{p}_i]$. In this work, p_i is seen as a continuous variable. It is assumed that compressing the duration of activity i by one unit generates an extra-cost e_i. Therefore, any agent A_u has to pay a fixed cost $k_u = \sum_{i \in \mathcal{T}_u} \kappa_i$ corresponding to the execution of his activities at their maximum duration, plus an extra compression cost equal to $\sum_{i \in \mathcal{T}_u} (\overline{p}_i - p_i) \times e_i$. We further refer to as $\gamma_u(P_u)$ the total cost paid by A_u for performing her activities, P_u being the processing time vector corresponding to the activities $i \in \mathcal{T}_u$, *i.e.*:

$$\gamma_u(P_u) = k_u + \sum_{i \in \mathcal{T}_u} (\overline{p}_i - p_i) \times e_i.$$

The project customer is assumed to pay a given amount at the project delivery, from which daily penalties are deducted in case of tardiness. Then, if project ends at time D and if $[\underline{D}, \overline{D}]$ is the initial contracted delivery interval, the customer pays $K - \max(0, D - \underline{D}) \times \pi$, K and π being the initial arranged payment and the penalty cost per time unit, respectively. Let $\Pi(D)$ refers to this quantity.

A last assumption concerns the agents' payoff. We assume that the agents have agreed an arrangement for sharing the customer's payment (at a more strategic decision level), such that the payoff of Agent A_u is a fixed proportion w_u of the total payment. Therefore, the profit of agent A_u can be expressed as:

$$Z_u(P_u, D) = w_u \times \Pi(D) - \gamma_u(P_u).$$

In this paper, it is assumed that the agents are selfish and want to maximize their self profit.

4. Problem modeling

Such a project environment can be modeled by a colored graph $G = (V, A, C, \phi, l)$ in which:

- $V = \mathcal{T}$ is the set of vertices and corresponds to the set of activities (including the two dummy activities);
- $A = \mathcal{P} \cup \{(n+1, 0)\}$ is the set of arcs that depicts the precedence constraints, the supplementary arc $(n+1, 0)$ being added for convenience reason, as discussed below;
- $C = \{0, \ldots, m+1\}$ is the set of colors, each one being specific to an agent;
- ϕ is an application associating each vertex (activity) $i \in V$ with a color (agent) $c \in C$, such that $\phi(i) = u$ if and only if $i \in \mathcal{T}_u$;
- l is an application associating each arc $(i, j) \in A$ with a length $p_i \in [\underline{p}_i, \overline{p}_i]$.

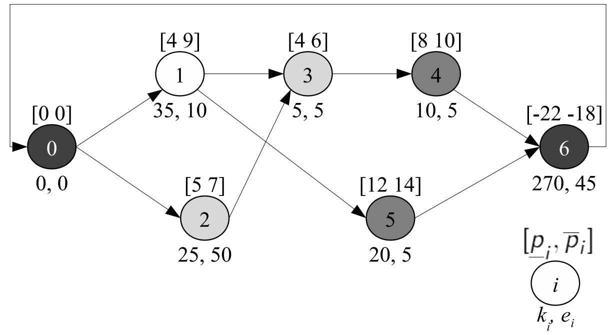

Figure 1. A multi-agent project scheduling example

The graph of Figure 1 depicts an example with five agents: black agents correspond to fictitious agents A_0 and A_4, activity agents are A_1, A_2 and A_3 whose associated colors are white, light gray and dark gray, respectively. There are seven activities that are allocated to the agents such that $\mathcal{T}_0 = \{0\}$, $\mathcal{T}_1 = \{1\}$, $\mathcal{T}_2 = \{2, 3\}$, $\mathcal{T}_3 = \{4, 5\}$ and $\mathcal{T}_4 = \{6\}$.

As indicated by the legend of the figure, the duration interval of each task i, as well as its fixed cost and crashing cost (k_i, e_i), are given at the top and the bottom of each vertex, respectively. In the case of activity $n+1$, allocated to agent A_{m+1}, things are a bit different: the duration interval is $[-\overline{D}, -\underline{D}]$ since the length of the arc $(n+1, 0)$ has to be negative for avoiding positive length circuits in G. In the example, Activity 6 belonging to A_4 has a processing interval equals to [-22 -18]. Moreover, the pair (k_i, e_i) at the bottom of the activity $n+1$ is set to (K, π) (the fixed price and the daily penalty cost, respectively). In the example, $K = 270$ and $\pi = 45$.

As said before, each activity agent has to choose a strategy, *i.e.*, a duration vector P_u for his tasks, provided that each vector component p_i of P_u belongs to $[\underline{p}_i, \overline{p}_i]$. Graphically, the duration p_i of every activity is reported on the arcs outgoing from i and are referred to as the length of the arc. It is not hard to see that, once these duration vectors set, a simply longest-path algorithm allows to know the smallest project completion time D. Consequently the total crashing cost and the penalty cost can be calculated, as well as the agents' payoff $w_u \times \Pi(D)$. Note that for matter of clarity, the length of the arc $(n+1, 0)$ is set to $-D$ (so that the longest circuit in G has a length equals to 0).

We define the agents' strategy S as the concatenation of the individual strategies P_u, *i.e.*, $S = (P_1, \ldots, P_m)$. The strategy S induces the project completion time denoted as $D(S)$. A strategy S has to be both *time-consistent* and *cost-consistent*.

A *time-consistent* strategy S is such that the duration vectors chosen by the activity agents involve a total project duration $D(S)$ that belongs to the interval $[\underline{D}, \overline{D}]$. Let $\mathcal{S}_{TC}$ be the set of time-consistent strategies.

To any strategy S, we can associate a profit vector $Z(S) = (Z_1(P_1, D(S)), \ldots, Z_m(P_m, D(S)))$. A strategy S is *cost-consistent* if and only if the total compression cost $\gamma_u(P_u)$ paid by every activity agent A_u does not exceed its payoff $w_u \times \Pi(D(S))$, that is if every agent's profit vector $Z_u(P_u, D(S))$ is positive. Let $\mathcal{S}_{CC}$ be the set of cost-consistent strategies.

For illustration, considering the problem of Figure 1, let us discuss the strategy $S = (9, 7, 4, 9, 13)$, which leads to the project completion $D(S) = 22$. This strategy can be modeled graphically as displayed in Figure 2 (every arc is labeled with the duration of its source activity). We assume that the revenue is equally shared among agents (*i.e.*, $w_u = \frac{1}{3}$). This strategy is time consistent (since $22 \in [18, 22]$) and leads to the total revenue 90 (the agent's payoff equals 30), but it is not cost-consistent since $Z(S) = (-5, -10, -10)$.

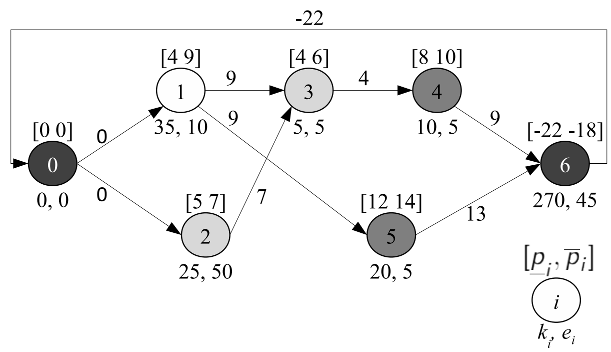

Figure 2. A cost-inconsistent agents' strategy

We further refer $\mathcal{S} = \mathcal{S}_{TC} \cap \mathcal{S}_{CC}$ to as the set of valid strategies, *i.e.*, the strategies being both time-consistent and cost-consistent. We also denote by $\mathcal{S}(D)$ the set of strategies $S \in \mathcal{S}$ such that $D(S) = D$.

5. Some relevant decision problems

Among the relevant decision problems that can arise in such a multi-agent scheduling context, the first trivial one is to know if there exist any valid strategy? In other words, is $\mathcal{S}$ empty? Actually, answering this question is easy since we only need to solve the following linear program. The variables are c_i, p_i, and Π the completion time of activity i, its duration and the customer's payment, respectively.

$$
\begin{array}{ll}
\min c_{n+1} & \\
\text{s.t} & \\
c_j \geq c_i + p_j, \forall (i,j) \in \mathcal{P} & (1) \\
w_u \times \Pi - \gamma_u(P_u) \geq 0, \forall A_u \in \mathcal{A} & (2) \\
\Pi \leq K - (c_{n+1} - \underline{D}) \times \pi & (3) \\
c_i \geq 0, c_{n+1} \leq \overline{D}, p_i \geq \underline{p}_i, \Pi \leq K &
\end{array}
$$

This program aims at minimizing $D = c_{n+1}$. Constraint (1) ensures that precedence constraints are satisfied. Constraint (2) imposes that any solution has to be cost-consistent, where Π corresponds to the customer's payment. Constraint (3) establishes the relation between the variables c_{n+1} and Π.

Clearly, $\mathcal{S}$ is empty if and only if the previous program is infeasible. In the other case, we can get the best project duration $D^{\min}$ that can be achieved by the agents. Let us highlight that this solution is also the one that ensures the maximum value of the overall revenue Π (which has to be distinguished from the agent's profit). If we move back to our example assuming that the revenue is equally shared among agents (*i.e.*, $w_u = \frac{1}{3}$), the optimum value is $D^{\min} = 18$, which is obtained with the strategy $S = (6, 6, 4, 8, 12)$ depicted in Figure 3, with $\Pi = 270$ and $Z(S) = (25, 0, 40)$.

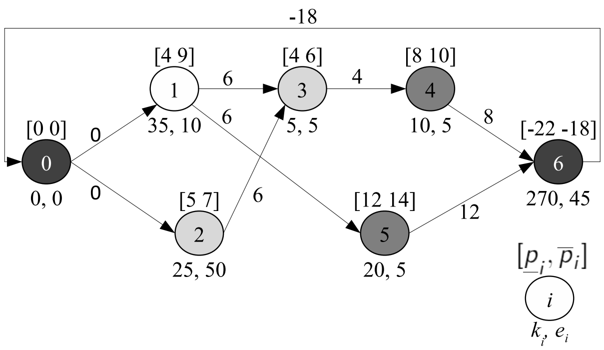

Figure 3. A valid strategy minimizing D

Now we can also solve the next linear program that aims at finding the less costing strategy for the whole set of agents (it is actually a compression cost problem that can be solved in polynomial time [16]). Constraints (1)-(3) are unchanged. Then, assuming the program feasibility (or equivalently the non-emptiness of $\mathcal{S}$), we also get the maximum value $D^{\max}$ that is reachable (hence the minimum value of Π). Indeed, since increasing D implies the reduction of the compression cost, there cannot exist any other valid strategy having a D greater than $D^{\max}$. For our example with $w_u = \frac{1}{3}$, a less costing strategy is $S = (8, 7, 4, 9, 13)$ displayed in Figure 4, which gives $D^{\max} = 21$, $\Pi = 225$ and $Z(S) = (0, 5, 5)$.

$$
\begin{array}{lr}
\min \ \sum_{A_u \in \mathcal{A}} (\gamma_u(P_u)) & \\
\text{s.t} & \\
c_i - c_j - p_i \geq 0, \forall (j,i) \in \mathcal{P} & (1) \\
w_u \times \Pi - \gamma_u(P_u) \geq 0, \forall A_u \in \mathcal{A} & (2) \\
\Pi \leq K - (c_{n+1} - \underline{D}) \times \pi & (3) \\
c_i \geq 0, c_{n+1} \leq \overline{D}, P \geq \underline{P}, \Pi \leq K &
\end{array}
$$

Clearly, if $\mathcal{S}$ is not empty then $\underline{D} \leq D^{\min} \leq D^{\max} \leq \underline{D}$, which gives an interesting knowledge of the best and worst temporal performance that the agents could achieve, respecting the cost-consistency constraints.

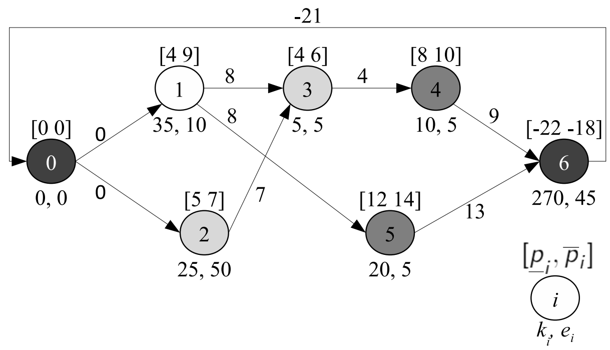

Figure 4. A valid strategy maximizing D

Beyond these problems, an other significant one is to determine the set of strategies $\mathcal{S}^*$ that are Pareto optimal for the problem of maximizing $Z(S)$, for all $S \in \mathcal{S}$. Indeed, determining all the non-dominated strategies is a multi-objective problem that allow to characterize all the strategies that are non-dominated. A strategy vector S is a Pareto-optimal solution, if for each agent A_u, it does not exist any alternate strategy vector S' such that $Z(S') \geq Z(S)$. Among the Pareto-optimal solutions, one can distinguish the solutions that maximize the profit of one agent. These solutions can be obtained by using the previous linear program, simply replacing the objective function by $\max Z_u$. Let $Z_u^{\max}$ refers to as the maximum profit of A_u.

For our example, a manual enumeration of the solution allows to determine three Pareto optimal strategies $\mathcal{S}^* = \{S_1^* = (6, 7, 4, 8, 13), S_2^* = (7, 7, 4, 8, 12), S_3^* = (6, 6, 4, 8, 12)\}$. The corresponding profit vector are $Z(S_1^*) = (10, 35, 30)$, $Z(S_2^*) = (20, 35, 25)$ and $Z(S_3^*) = (25, 0, 40)$ leading to the project durations $D(S_1^*) = D(S_2^*) = 19$ and $D(S_3^*) = 18$, respectively. So $Z_1^{\max} = 25$, $Z_2^{\max} = 35$ and $Z_3^{\max} = 40$,

6. The price of cooperation

In this section, a game theory point of view is taken. Indeed, another problem that makes sense is to determine the set $\mathcal{S}^{\text{NE}}$ of Nash-equilibrium strategies, if there any. Before defining what a Nash equilibrium is in our context, let us introduce the notation S_{-u} to refer as the strategies played by the $(m-1)$ agents but not A_u. A strategy vector is said to be a Nash equilibrium if for all agents A_u and each alternate strategy P'_u, we have:

$$
Z_u(P_u, S_{-u}) \quad \geq \quad Z_u(P'_u, S_{-u}) \tag{4}
$$

Equation (4) expresses that no agent A_u can change his chosen duration vector from P_u to P'_u and thereby improve his profit, assuming that all other agents keep their own strategies unchanged. We remark that in the context of this work, an agent strategy move, as defined above, can lead either to shorten the project makespan or, in the contrary, to its increase it. For instance, if we move back to our example, we observe that the Pareto optimum strategy S_3^* is not a Nash equilibrium since $Z_2(S_3^*) < Z_2(S_2^*)$, the strategy S_3^* only differing from S_2^* by the choices of A_2 (*i.e.*, $P_2 = (6, 4)$ and $P_2' = (7, 4)$). Hence, there is no reason for A_2 to accept to shorten the duration of the project from 19 to 18 since it decreases his profit. Actually for our example, there are only two strategies which are both Nash equilibria and Pareto optimum: S_1^* and S_2^*, both having the project duration $D = 19$. Of course, the strategies belonging to $\mathcal{S}^{\text{NE}}$ are all of interest for at least one agent and it is not possible to say which one will be preferred at the end.

An interesting problem is to determine the Nash equilibria that induce a minimal project duration. The complexity of this problem will be discussed during the conference. Nonetheless, it can be stated that it exists a particular project duration D^{NE}, with $D^{\min} \leq D^{\text{NE}} \leq D^{\max}$, so that any strategy S with $D(S) < D^{\text{NE}}$ cannot be a Nash equilibrium. By illustration in our example, we have $D^{\text{NE}} = 19$ which is obtained for the Pareto strategies S_1^* and S_2^*. D^{NE} can be viewed as a threshold which cannot be crossed due to the selfishness of agents (none agent wants to reduce its profit for increasing the profit of other ones). Therefore the ratio:

$$\frac{\max_{S \in \mathcal{S}^{\text{NE}}} Z_u(S)}{Z_u^{\max}}$$

measures for every agent A_u the minimum distance between the maximum agent's profit in the set of Nash-equilibrium strategies, and his best profit. It expresses in one sense the price of the cooperation.

7. The special case $\mathcal{T}_u = \{u\}$

In this section, we consider the special case where each agent manages exactly one activity (*i.e.*, $n = m$). In this case, we show how the value D^{NE} can be easily computed and how the set $\mathcal{S}^{\text{NE}}$ of Nash-equilibria can be characterized. We highlight that, since the notions of activity and agent are equivalent in this section, the index u is systematically used instead of i in the sequel.

Let us recall that reducing the project duration by one time unit increases the collective revenue by the value π exactly (the daily penalty). Moreover, the exact profit part of every agent A_u on this amount is $w_u \times \pi$. Since A_u only manages a single activity u, this quantity has to be compared with the unitary crashing cost e_u of u. Then the set $\mathcal{A}$ of agents can be divided into two subsets $\mathcal{A}^+$ and $\mathcal{A}^-$, with $\mathcal{A} = \mathcal{A}^+ \cup \mathcal{A}^-$:

- every agent A_u, such that $e_u < w_u \times \pi$, is put inside $\mathcal{A}^+$ since shortening the activity u by one unit of time is always of interest for A_u, provided it also shortens the project duration by one unit (the profit, $w_u \times \pi - e_u$ is positive) ;
- every agent A_u, such that $e_u \geq w_u \times \pi$, is put inside $\mathcal{A}^-$ since it is systematically uninteresting for A_u to crash the duration of his activity as his profit can never increase.

Now, let us consider the particular extreme strategy $\hat{S}$ where agents belonging to $\mathcal{A}^+$ has set their activity duration p_u to $\underline{p}_u$, while those of $\mathcal{A}^-$ has keep the activity durations to their maximum value (*i.e.*, $p_u = \overline{p}_u$). Then it is easy to see that the length of the longest path in G for this strategy $\hat{S}$ actually equals D^{NE}. Indeed the project duration cannot be shortened without crashing one activity of $\mathcal{A}^-$, that is without reducing the profit of one agent. We assume here that the D^{NE} value is time consistent *i.e.*, $D^{\text{NE}} \in [\underline{D}, \overline{D}]$

Let $\mathcal{C}^{\hat{S}}$ refers the set of agents belonging to the critical path in G under strategy $\hat{S}$. We also refer to $L_{u,v}(S)$ as the length of the longest path from u to v in G under strategy S, and to $L^{\max}(u, v)$ as the length of the longest path from u to v under the strategy where each agent set their p_u to $\overline{p}_u$ for all u.

Then any strategy S such that:

- $p_u = \underline{p}_u$ if A_u in $\mathcal{A}^+ \cap \mathcal{C}^{\hat{S}}$,
- $p_u = \overline{p}_u$ if A_u in $\mathcal{A}^-$,
- p_u is chosen in $[\underline{p}_u, \overline{p}_u]$ such that $L_{0,u}(S) + L_{u,n+1}(S) = \min(D^{\text{NE}}, L_{0,u}^{\max} + L_{u,n+1}^{\max})$ else, *i.e.*, if $A_u \in \mathcal{A}^+$ and $A_u \notin \mathcal{C}^{\hat{S}}$,

is a Nash equilibrium. We highlight that the third condition only ensures that the longest path traversing every A_u has a length that never exceeds $(D^{\text{NE}}$, nor its maximum value *i.e.*, $L_{0,u}^{\max} + L_{u,n+1}^{\max}$).

Concluding remarks

This paper presents a new game theory framework for multi-agent project scheduling problem, which can be used to cope with numerous cooperation situations that occur in real industrial contexts (supply-chain management, huge project management, etc.). We particularly propose a decision model and stress the connections between Pareto-optimum strategies and Nash-equilibrium ones. The last section also shows how to characterize the best Nash-equilibrium strategies in the case where each activity is managed by a specific agent. Further works are ongoing to characterize Nash-equilibria in the general case or to consider more complex profit sharing functions, as well as discrete activity duration strategies.

Acknowledgements

This work was supported by the ANR project no. 08-BLAN-0331-02 named "ROBOCOOP".

References

[1] A. Agnetis, P. B. Mirchandani, D. Pacciarelli, and A. Pacifici. Nondominated schedules for a job-shop with two competing users. *Comput. Math. Organ. Theory*, 6:191–217, July 2000.

[2] A. Agnetis, P. B. Mirchandani, D. Pacciarelli, and A. Pacifici. Scheduling problems with two competing agents. *Oper. Res.*, 52:229–242, March 2004.

[3] A. Agnetis, G. Pascale, and D. Pacciarelli. A lagrangian approach to single-machine scheduling problems with two competing agents. *J. of Scheduling*, 12:401–415, August 2009.

[4] E. Angel, E. Bampis, and F. Pascual. The price of approximate stability for scheduling tasks on two links. In *Proceedings of the European Conference on Parallel Computing (Euro-Par 2006)*, volume 4128 of *Lecture Notes in Computer Science*, pages 157–166, Allemagne, 2006.

[5] C. Briand, S. Ourari, and B. Bouzouiai. A cooperative approach for job shop scheduling under uncertainties. In *Proceeding of the 2008 conference on Collaborative Decision Making: Perspectives and Challenges*, pages 5–15, Amsterdam, The Netherlands, The Netherlands, 2008. IOS Press.

[6] G. P. Cachon and M. A. Lariviere. Supply chain coordination with revenue-sharing contracts: Strengths and limitations. *Management Science*, 51(1):pp. 30–44, 2005.

[7] T. Cheng, C. Ng, and J. Yuan. Multi-agent scheduling on a single machine to minimize total weighted number of tardy jobs. *Theoretical Computer Science*, 362(1-3):273 – 281, 2006.

[8] T. Cheng, C. Ng, and J. Yuan. Multi-agent scheduling on a single machine with max-form criteria. *European Journal of Operational Research*, 188(2):603 – 609, 2008.

[9] G. Confessore, S. Giordani, and S. Rismondo. A market-based multi-agent system model for decentralized multi-project scheduling. *Annals of Operations Research*, 150:115–135, 2007. 10.1007/s10479-006-0158-9.

[10] G. Dudek and H. Stadtler. Negotiation-based collaborative planning between supply chains partners. *European Journal of Operational Research*, 163(3):668 – 687, 2005. Supply Chain Management and Advanced Planning.

[11] M. Estevez Fernandez. A game theoretical approach to sharing penalties and rewards in projects. Discussion Paper 2008-84, Tilburg University, Center for Economic Research, 2008.

[12] R. Evaristo and P. C. van Fenema. A typology of project management: emergence and evolution of new forms. *International Journal of Project Management*, 17(5):275–281, 1999.

[13] J. Ferber. *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 1999.

[14] S. Foldes and F. Soumis. Pert and crashing revisited: Mathematical generalizations. *European Journal of Operational Research*, 64(2):286 – 294, 1993. Project Management anf Scheduling.

[15] J. Homberger. A multi-agent system for the decentralized resource-constrained multi-project scheduling problem. *International Transactions in Operational Research*, 14(6):565–589, 2007.

[16] J. Kamburowski. On the minimum cost project schedule. *Omega*, 23(4):463–465, August 1995.

[17] E. Koutsoupias and C. Papadimitriou. Worst-case equilibria. In *Proceedings of the 16th annual conference on Theoretical aspects of computer science*, STACS'99, pages 404–413, Berlin, Heidelberg, 1999. Springer-Verlag.

[18] A. S. Leufkens and N. G. Noorderhaven. Learning to collaborate in multi-organizational projects. *International Journal of Project Management*, 29(4):432 – 441, 2011. European Academy of Management (EURAM 2010) Conference.

[19] L. Martin Diaz. Evaluation of cooperative planning in supply chains : an empirical approach of the european automotive industry. Publications of Darmstadt Technical University, Institute for Business Studies (BWL) 25825, Darmstadt Technical University, Department of Business Administration, Economics and Law, Institute for Business Studies (BWL), Jan. 2006.

[20] F. Pascual, K. Rzadca, and D. Trystram. Cooperation in multi-organization scheduling. *Concurr. Comput. : Pract. Exper.*, 21:905–921, May 2009.

[21] J. Phillips, Steve and M. I. Dessouky. Solving the project time/cost tradeoff problem using the minimal cut concept. *Management Science*, 24(4):pp. 393–400, 1977.

[22] M.-C. Portmann and Z. Mouloua. A window time negotiation approach at the scheduling level inside supply chains. In *3rd Multidisciplinary International Scheduling Conference: Theory and Applications - MISTA 2007*, pages 410–417, Paris France, 2007.

[23] A. Saad, K. Kawamura, and G. Biswas. Performance evaluation of contract net-based heterarchical scheduling for flexible manufacturing systems. In *in proceedings of the 1995 International Joint Conference on Artificial Intelligence (IJCAI'95), Workshop on Intelligent Manufacturing*, pages 310–321, 1996.

[24] W. Shen, L. Wang, and Q. Hao. Agent-based distributed manufacturing process planning and scheduling: a state-of-the-art survey. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 36(4):563 –577, july 2006.