MISTA 2013

Solving a cooperative project scheduling with controllable processing times, self-interested agents and equal profit sharing

Cyril Briand, Sandra Ulrich Ngueveu and Přemysl Šůcha

Abstract This paper considers a cooperative project that involves a set of self-interested agents, each in charge of a part of the project. An agent can reduce the duration of the activities he is responsible for, by gathering extra-resources, at a given cost. If the overall project ends earlier than expected, the client offers a reward to be shared equally among agents. The financial outcome for each agent therefore depends on all agents decisions and on the satisfaction of the customer, which is function of the project makespan. Under these assumptions, we address the problem of finding a Nash equilibrium, ensuring the stability of the schedule, that minimizes the project makespan. We explain how this problem, which is NP-hard, can be efficiently modeled and solved with mixed integer linear programming.

1 Introduction

Large-size projects in the building and construction sector or automotive industries usually require cooperation among groups of self-interested actors, often contractors, each being in charge of a part of the project, as described in [6] or [7]. These actors, often referred as *agents* in the literature, control the duration of activities they are responsible for. They can reduce durations by gathering extra-resources, but at a given cost. The client of the project is interested in minimizing the total duration of his project, and as an incentive, offers a daily reward to be shared among agents and

Cyril Briand
CNRS, LAAS, 7 avenue du colonel Roche, F-31400 Toulouse, France and
Univ de Toulouse, UPS, F-31400 Toulouse, France
E-mail: briand@laas.fr

Sandra Ulrich Ngueveu
CNRS, LAAS, 7 avenue du colonel Roche, F-31400 Toulouse, France and
Univ de Toulouse, INP, F-31400 Toulouse, France
E-mail: ngueveu@laas.fr

Přemysl Šůcha
CNRS, LAAS, 7 avenue du colonel Roche, F-31400 Toulouse, France and
Czech Technical University in Prague; Faculty of Electrical Engineering; Karlovo namesti 13, 121 35, Prague 2, Czech Republic
E-mail: suchap@fel.cvut.cz

proportional to the earliness of the realization [4]. Let us note that this notion of reward is symmetric to the one of penalty that agents must pay proportionally to the tardiness. Therefore, the payoff for each agent depends not only on his own decision, but also on the decisions of all other agents, which translate into a given project makespan, and thus a given reward.

Under these assumptions, we address the problem of finding a Nash equilibrium that minimizes the project makespan, assuming that the reward is equally shared among the agents involved into the project. In other words, we aim at determining a strategy that favor the customer objective with the constraint that the strategy remains stable (no agent will have interest in changing his decisions). This problem has been already considered in the literature by Agnetis et al. [1] and proved to be NP-Hard. This paper shows how such a problem can be modeled and solved efficiently with mixed integer linear programming (MILP).

The paper is structured as follows. Section 2-4 give a formal definition of our problem and recall some basic concepts and properties. Section 5 proposes an original MILP formulation for that problem. Section 6 reports some computational experiments.

2 Problem statement

The multi-agent project scheduling problem considered in this paper can be modeled with an activity-on-arc graph $\mathcal{G} = (X, U)$ that defines the project activity network. The set of nodes $X = \{0, 1, \ldots, n-1\}$ represents the project events, and the set of arcs U represents project activities. Nodes 0 and $(n-1)$ correspond to the project beginning and project end. The set of arcs U results from the union of two subsets $U_R \cup U_D$, where U_R is the set of real project activities and U_D is the set of dummy activities. Dummy activities are related to precedence constraints. U_R is divided among a set $\mathcal{A} = \{A_1, \ldots, A_m\}$ of m agents.

The set of m_u activities assigned to agent A_u is denoted $\mathcal{T}_u$. $\mathcal{T}_u \cap \mathcal{T}_v = \emptyset$ for any pair of agents $(A_u, A_v) \in \mathcal{A}^2$ such that $u \neq v$. Each activity $(i,j) \in U$ is associated with a minimal duration $\underline{p}_{i,j}$ and a normal duration $\overline{p}_{i,j}$.

For each activity $(i,j) \in \mathcal{T}_u$, the responsible agent A_u can choose a processing time $p_{i,j} \in [\underline{p}_{i,j}, \overline{p}_{i,j}]$. For each dummy activity $(i,j) \in U_D$, $\underline{p}_{i,j} = \overline{p}_{i,j} = 0$. A crashing unitary-cost $c_{i,j}$ is associated to each activity $(i,j) \in U$ (provided that $c_{i,j} = 0, \forall (i,j) \in U_D$). The cost incurred by an agent A_u when shortening an activity $(i,j) \in \mathcal{T}_u$ from $\overline{p}_{i,j}$ to $p_{i,j}$ is therefore $(\overline{p}_{i,j} - p_{i,j})c_{i,j}$.

The strategy of an agent A_u is denoted P_u and is the duration vector of all his activities $\mathcal{T}_u$. All individual strategies can be gathered into a vector $S = (P_1, \ldots, P_m)$ called strategy profile. For any given strategy profile S, the resulting project makespan $D(S)$ can be computed using a classical longest path algorithm on the graph $\mathcal{G}(S)$, which results from $\mathcal{G}$ by setting the length of every arc to its corresponding value in strategy profile S. We further refer to $\overline{D} = D(\overline{S})$ as the project makespan obtained when every activity duration is set to its normal (and thus maximal) value $\overline{p}_{i,j}$.

Let parameter π be the daily reward offered by the client, therefore the value $Z_u(S) = \frac{\pi}{m}(\overline{D} - D(S)) - \sum_{(i,j)\in\mathcal{T}_u} c_{i,j}(\overline{p}_{i,j} - p_{i,j})$ corresponds to the profit of A_u for the strategy profile S, where $(\overline{D} - D(S))$ is the project-makespan reduction. This profit is equal to the difference between the reward received from the client and the expenses incurred to shorten tasks durations.

3 Efficiency vs. stability

A strategy profile is said *efficient* if it corresponds to a Pareto-optimal solution. In other word, if no other strategy profile S' exists that dominates S with respect to the agents' profit. On the other hand, a strategy profile is *stable* if there is no incentive for any agent to modify it in order to improve his profit. The stability of a strategy profile is important since it ensures that agents can trust each other. It is totally connected to the notion of a Nash equilibrium.

Let S_{-u} denote the strategies played by the $(m-1)$ agents except A_u, (*i.e.*, $S_{-u} = (P_1, \ldots, P_{u-1}, P_{u+1}, \ldots P_m)$). Then, focusing on a particular agent A_u, any S can be also written as a couple (P_u, S_{-u}). A strategy profile vector $S = (P_u, S_{-u})$ is a Nash equilibrium if for all agents A_u and any local alternate strategy P'_u, we have $Z_u(P_u, S_{-u}) \geq Z_u(P'_u, S_{-u})$.

For illustration, let us consider the activity network of Figure 1 with 5 activities distributed among the two agents A_1 and A_2 such that $\mathcal{T}_1 = \{a, c\}$ and $\mathcal{T}_2 = \{b, d, e\}$. We assume that the daily reward π equals 120 to be shared fairly (*i.e.*, $w_1 = w_2 = 1/2$). When the durations of all activities are set to their maximal value, *i.e.*, $S = (\overline{p}_a, \overline{p}_b, \overline{p}_c, \overline{p}_d, \overline{p}_e) = (7, 9, 3, 8, 5)$, the project makespan is 15 and no agent makes any profit ($Z_1(S) = Z_2(S) = 0$). With the strategy profile $S' = (7, 9, \mathbf{2}, \mathbf{7}, 5)$, the makespan becomes 14 and profits are $Z_1(S') = Z_2(S') = 40$. The makespan can be further reduced to 13 with the strategy profile $S'' = (\mathbf{6}, 9, \mathbf{3}, 7, \mathbf{4})$, leading to $Z_1(S'') = 50$ and $Z_2(S'') = 50$. S'' is efficient since it corresponds to a Pareto optimum that maximizes the profit of both agents. Nevertheless, it is not stable: agent A_2 can improve his profit, to the detriment of A_1, by simply increasing back the duration of d and e, which leads to the strategy profile $S''' = (6, 9, 3, \mathbf{8}, \mathbf{5})$, with makespan 14 and profits $Z_2(S''') = 60$ and $Z_1(S''') = -10$. On the other hand, the strategy profile S', which is not a Pareto optimum, is stable since no agent is able to improve his profit by himself, *i.e.*, it is a Nash equilibrium.

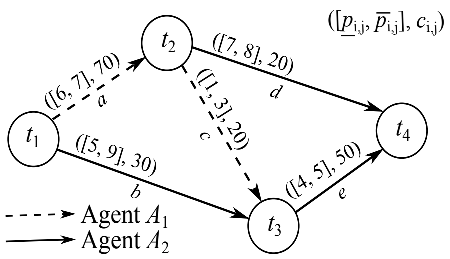

Fig. 1 A multiagent activity network with two agents and five activities

To sum up, the notion of Pareto strategy is important in regards of the agents profits. The notion of Nash equilibrium is complementary to the previous one and ensures organizational stability. Ideally, agents should choose a strategy that is both a Pareto optimum and a Nash equilibrium, but such a strategy does not always exist (as shown for the example in Figure 1).

In addition, from the customer viewpoint, the problem of finding a Nash equilibrium that minimizes the project makespan is also of interest: its solution gives the lower makespan that can be reached for the customer, provided that the organization remains stable. It can be viewed as the problem of minimizing the price of stability from the customer viewpoint [3]. We recall that such problem has been proved NP-hard in [1] using a reduction from problem 3-PARTITION.

Let us also point out that, when only a single agent is considered (instead of multiple agents), our problem reduces to the well-known discrete time-cost trade-off problem in project scheduling (that can be solved in polynomial time [8]).

4 Characterization of a Nash equilibrium

As proposed by Agnetis et al., a Nash equilibrium can be characterized using the notion of residual cuts in critical activity networks. This notion of residual cut is also used in [8] for solving the time-cost trade-off in project scheduling problem (single agent case), and in [5] for increasing or decreasing the tension in a graph at minimum cost.

Given a cut $\omega = (\Omega, \overline{\Omega})$ (with $0 \in \Omega$ and $n-1 \in \overline{\Omega}$), let us refer to ω^+ (resp. to ω^-) as the set of critical activities (i,j) such that $i \in \Omega$ and $j \in \overline{\Omega}$ (resp such that $i \in \overline{\Omega}$ and $j \in \Omega$). A cut ω is said residual with respect to a strategy profile S if there exists $\delta \in \mathbb{R}$ such that:

$$\begin{cases} p_{i,j} + \delta \in [\underline{p}_{i,j}, \overline{p}_{i,j}], \; \forall (i,j) \in \omega^+ \\ p_{i,j} - \delta \in [\underline{p}_{i,j}, \overline{p}_{i,j}], \; \forall (i,j) \in \omega^- \end{cases} \tag{1}$$

An increasing cut identifies a way to increase by $\delta > 0$ the project makespan with the unitary profit $Profit(\omega) = +(\sum_{(i,j)\in\omega^+} c_{i,j} - \sum_{(i,j)\in\omega^-} c_{i,j})$. Similarly, a decreasing-cut allows to decrease by $\delta < 0$ the project makespan with the unitary profit $Profit(\omega) = -(\sum_{(i,j)\in\omega^+} c_{i,j} - \sum_{(i,j)\in\omega^-} c_{i,j})$. When only the activities belonging to one agent A_u are taken into account in the previous formula, we obtain the projected cost, $Profit_u(\omega)$, which gives the unitary profit for agent A_u with respect to cut ω. It corresponds to the increase or decrease of the profit of agent A_u when modifying the duration of the activities involved in ω.

We also need to introduce the notion of a *poor* strategy profile. A strategy profile S with project duration $D(S)$ is said poor if there exists an alternative strategy profile $S' = (P'_u, S_{-u})$ with the same makespan $D(S') = D(S)$, only differing from S by the strategy taken by A_u, such that $Z_u(S) < Z_u(S')$ and $Z_v(S) \leq Z_v(S')$, for all $A_v \neq A_u$. In other words, given the strategies chosen by all agent except A_u, a poor strategy profile will not maximize the profit of agent A_u. Obviously, any poor strategy profile can be neither a Pareto or a Nash one.

With that in mind, the following proposition holds, aiming at characterizing a Nash equilibrium.

Proposition 1 *A non-poor strategy profile S is a Nash equilibrium if and only if there is no residual cut $\omega(\Omega, \overline{\Omega})$ satisfying one of the two following conditions: (1) $\exists \delta > 0$ and $\exists A_u \in \mathcal{A}$ with $Profit_u(\omega) \geq \frac{\pi}{m}$ or (2) $\exists \delta < 0$ and $\forall (i,j) \in \omega^+$, $(i,j) \in \mathcal{T}_u$ and $-Profit_u(\omega) > \frac{\pi}{m}$.*

The proof of this proposition is summarized below. First, S being a non-poor strategy profile, the only way for an agent to increase his profit is to increase or decrease the project makespan, *i.e.*, to identify either an increasing or decreasing cut being profitable for him. Condition (1) ensures that there cannot exist any increasing cut profitable for any agent, *i.e.*, such that the agent's profit is greater than his part of daily reward. Similarly, Condition (2) states that there cannot exist any decreasing cut profitable for one agent, *i.e.*, such that the daily reward part is greater than the agent's profit. If both conditions (1-2) stand, then no agent can individually make profit and therefore strategy profile S is a Nash equilibrium.

5 Finding a Nash equilibrium S minimizing $D(S)$

According to Proposition 1, the problem of finding a Nash equilibrium that minimizes the makespan can be formulated as a mathematical program as follows.

$$
\begin{array}{lllr}
\min & t_{n-1} - t_0 & & (1)\\
\text{s.t} & t_j - t_i - p_{i,j} \geq 0 & ,\forall (i,j) \in U & (2)\\
 & \underline{p}_{i,j} \leq p_{i,j} \leq \overline{p}_{i,j} & ,\forall (i,j) \in U & (3)\\
 & Profit_u(\omega) < \frac{\pi}{m} & \forall \text{ cut } \omega, \forall A_u \in \mathcal{A} & (4)\\
 & t_i \in \mathbb{R} & ,\forall i \in X & (5)
\end{array}
$$

Variables t_i, with $i \in [0, n-1]$, are associated with occurrence times of the project events of X. The objective function (1) aims at minimizing the project makespan. Let us note that this objective ensures that the optimal strategy profile cannot admit any decreasing cut, so that the condition (2) of Proposition 1 is enforced. Constraints (2) model precedence relations between project activities, constraints (3) specify the bounds on the processing times values. With respect to condition (1) of Proposition 1, the high-level constraints (4) impose that any existing residual cut must not have a projected profit greater or equal to $w_u\pi$. Because the number of cuts in $\mathcal{G}$ is exponential, the resulting problem remains hard to handle. However, as explained in the sequel, we were able to model constraints (4) efficiently inside the MILP, using a finite number of primal-dual constraints.

First, it was noted that, for each agent, determining a cut that maximizes $Profit_u(\omega)$ facilitates the verification of constraints (4). Such problem is a max-cut problem that can be formulated as the linear program (6)-(10). Variable $\gamma_i^u = 1$ if i is in Ω, and 0 otherwise. Variable $\alpha_{i,j}^u = 1$ if $(i,j) \in \omega^+$, and 0 otherwise. Similarly, $\beta_{i,j}^u = 1$ if $(i,j) \in \omega^-$, and 0 otherwise. The parameters $[\underline{c}_{i,j}^u, \overline{c}_{i,j}^u]$ are the minimum and maximum capacity of each arc (i,j). If (i,j) is in $\mathcal{T}_u$ and is critical and $p_{i,j} > \underline{p}_{i,j}$, then $\overline{c}_{i,j}^u = c_{i,j}$. If $(i,j) \notin \mathcal{T}_u$ and is not critical, then $\overline{c}_{i,j}^u = 0$. For any other case, $\overline{c}_{i,j}^u = \infty$. If (i,j) is in $\mathcal{T}_u$ and is critical and $p_{i,j} < \overline{p}_{i,j}$, then $\underline{c}_{i,j}^u = c_{i,j}$. For any other case, $\underline{c}_{i,j}^u = 0$.

$$
\begin{array}{lllr}
\max & \sum_{(i,j)\in\mathcal{T}_u} \alpha_{i,j}^u \underline{c}_{i,j}^u - \sum_{(i,j)\in\mathcal{T}_u} \beta_{i,j}^u \overline{c}_{i,j}^u & & (6)\\
\text{s.t} & \alpha_{i,j}^u - \beta_{i,j}^u - \gamma_i^u + \gamma_j^u \leq 0 & ,\forall (i,j) \in U & (7)\\
 & \gamma_0^u = 1, \gamma_{n-1}^u = 0 & & (8)\\
 & \gamma_i^u \geq 0 & ,\forall i \in X & (9)\\
 & \alpha_{i,j}^u, \beta_{i,j}^u \geq 0 & ,\forall (i,j) \in U & (10)
\end{array}
$$

The dual of this problem is a min-flow problem, that can also be expressed as the linear program (11)-(14) where $\phi_{i,j}^u$ is the flow assigned to A_u and circulating on $(i,j) \in U$:

$$\begin{array}{llr} \min & \sum_{k \in X \setminus \{0\}} \phi^u_{0,k} & (11) \\ \text{s.t} & \sum_{(k,i) \in U} \phi^u_{k,i} = \sum_{(i,k) \in U} \phi^u_{i,k} \quad ,\forall i \in X & (12) \\ & \underline{c}^u_{i,j} \le \phi^u_{i,j} \le \overline{c}^u_{i,j} \quad ,\forall (i,j) \in U & (13) \\ & \phi^u_{i,j} \ge 0 \quad ,\forall (i,j) \in U & (14) \end{array}$$

From the duality theory, we know that only an optimal solution can satisfy both sets of constraints (7)-(10) and (12)-(14) as well as additional constraint: $\sum_{k \in X \setminus \{0\}} \phi^u_{0,k} \le \sum_{(i,j) \in \mathcal{T}_u} \alpha^u_{i,j} \underline{c}^u_{i,j} - \sum_{(i,j) \in \mathcal{T}_u} \beta^u_{i,j} \overline{c}^u_{i,j}$ (15) (which imposes that primal and dual objective functions are equal).

To summarize, in order to find a minimum makespan Nash equilibrium, we propose the MILP obtained from the model (1)-(5) after replacing constraints (4) by constraints (7)-(10) and (11)-(15), with the addition of constraints: $\sum_{k \in [1,n-1]} \phi^u_{0,k} < \frac{\pi}{m}$ (16).

Note that $\underline{c}^u_{i,j}$ and $\overline{c}^u_{i,j}$ depend on the $p_{i,j}$ variables, therefore the constraint (15) is not linear because it involves multiplication of variables. Nevertheless, it can be linearized efficiently because each multiplication involves one binary and one real variable.

6 Preliminary numerical results

The resulting MILP model was implemented in IBM ILOG CPLEX Optimization Studio 12.3 and solved with the CPLEX solver on a PC with 8 GB of RAM and 3.0 GHz CPU. It was applied on the 4 largest set of instances $|U_R| = \{20, 40, 60, 80, 100\}$ available from authors upon request. Each benchmark set consists of 100 problem instances considering 5 agents + 1 virtual agent for dummy activities. The daily reward π is determined with respect to the maximal cut cost $C(\omega_{max})$, which represents the theoretical most expensive cut that can be used to decrease project makespan when affiliations to agents are ignored.

The results in terms of computing times for $\pi = 0.05 \cdot C(\omega_{max})$ are available in Table 1. It shows the average, the minimal and the maximal CPU time needed to solve the instances for each $|U_R|$. In addition, the table also provides the total number of activities U and nodes X from the network graphs corresponding to the instances.

$\lvert U_R \rvert$	avg $\lvert U \rvert$	avg $\lvert X \rvert$	CPU time		
			avg	min	max
$[-]$	$[-]$	$[-]$	$[s]$	$[s]$	$[s]$
20	50.6	20.3	0.27	0.02	0.92
40	162.4	47.2	0.41	0.05	1.56
60	327.4	76.4	1.17	0.09	3.49
80	538.4	106.2	2.00	0.78	4.60
100	790.9	135.0	3.16	1.30	16.35

Table 1 Resolution time for our MILP depending on the number of activities

Analysis of this table shows that (i) the total number of activities is significantly larger than the number of real activities, and (ii) the solving phase remains very fast since we are able to solve instances with 100 real activities (almost 800 total activities!) in less than few seconds on average.

The daily reward π is an important parameter for the project client. Of course, it can be expected that the larger π, the shorter the project makespan accessible.

However, no client wants to waste money by paying more than what is necessary to reach a given makespan, or worse, by paying more without any makespan improvement. Figure 2 shows the project makespan gain $D^*/\overline{D}$ in function of the customer reward π obtained on the benchmark set with $|U_R| = 60$.

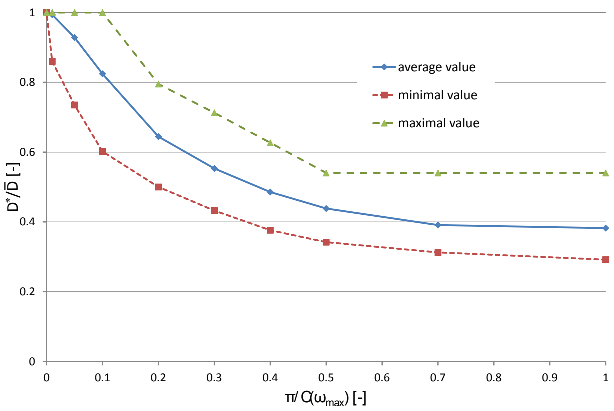

Fig. 2 Project makespan gain in function of the customer reward

As expected, the makespan decreases when the reward increases. However, it is important to note that (i) the correlation is not linear, (ii) until approximately $0.3C(\omega_{max})$ every additional reward results into significant makespan decrease, but after $0.3C(\omega_{max})$, each additional reward leads to less and less improvement to the makespan until $0.7C(\omega_{max})$, after which almost no improvement is obtained, and of course, (iii) there is a makespan lower bound that cannot be overpassed.

A project customer should therefore apply our model on the specific instance corresponding to his own project, in order to identify the best way to set its reward that would lead to the makespan reduction he aims at.

7 Conclusion

This paper presents a MILP formulation based on the characterization of a Nash equilibrium with residual cuts in a critical activity networks, to model a multi-agent cooperative project with self-interested agents and equal profit-sharing. The model obtained has been implemented and tested on several problem instances. Preliminary results are encouraging since optimal solutions are found in a few seconds for problems having up to 100 real activities (800 total activities) and 6 agents. Large scale experiments are in progress and a deeper analysis of the performance of our approach will be provided during the conference. Another advantage of the MILP formulation introduced here lies in the fact that the profit share of each agent may be replaced by a parameter so that any other predefined sharing policy may be implementable, without changing the nature of the MILP. Such a study is of interest, especially if the customer is able to influence the rewards sharing policy, because it could help designing more efficient sharing policies for the problem considered.

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

References

1. Agnetis, A., Briand, C., and Billaut, J.-C, Přemysl Š. 2012. Nash equilibria for the multi-agent project scheduling problem with controllable processing times, Research Report, LAAS-CNRS, http://homepages.laas.fr/briand/mapsp3.pdf.
2. Cachon, G. P. and Lariviere, M. A. 2005. Supply chain coordination with revenue-sharing contracts: Strengths and limitations. *Management Science* 51(1): 30–44.
3. Christodoulou, G. and Koutsoupias, E. 2005. The price of anarchy of finite congestion games, *37th Annual ACM STOC*, 67–73.
4. Fernandez, E. 2008. A game theoretical approach to sharing penalties and rewards in projects, European Journal of Operational Research 01; 216:647-657.
5. Hadjiat, M. and Maurras, J. F. 1997. A strongly polynomial algorithm for the minimum cost tension problem. *Discrete Mathematics* 165-166: 377–394.
6. Leufkens , A. S. and Noorderhaven, N. G. 2011. Learning to collaborate in multi-organizational projects. *International Journal of Project Management* 29(4): 432–441.
7. Martin Diaz, L. 2006. Evaluation of cooperative planning in supply chains : an empirical approach of the european automotive industry. In 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.
8. Philips S.J. and Dessouky, M. I. 1977. Solving the project time/cost tradeoff problem using the minimal cut concept. *Management Science* 24(4): 393–400.