

Minimizing a Makespan Under Uncertainty

Jérôme Fortin

UPS, IRIT,
118 route de Narbonne,
31062 Toulouse, France
fortin@irit.fr

Paweł Zieliński

Institute of Mathematics,
University of Technology
Wyb. Wyspiańskiego 27,
50-370 Wrocław, Poland
pziel@im.pwr.wroc.pl

Didier Dubois

UPS, IRIT,
118 route de Narbonne,
31062 Toulouse, France
dubois@irit.fr

Hélène Fargier

UPS, IRIT,
118 route de Narbonne,
31062 Toulouse, France
fargier@irit.fr

Abstract

This paper reconsiders the most basic scheduling problem, that of minimizing the makespan of a partially ordered set of activities, in the context of incomplete knowledge. After positioning this paper in the scope of temporal networks under uncertainty, we provide a complete solution to the problem of finding floats of activities, and of locating surely critical ones, as they are often isolated. The minimal float problem is NP-hard while the maximal float problem is polynomial. New complexity results and efficient algorithms are provided for the interval-valued makespan minimization problem.

1 Introduction and Motivation

Temporal Constraint Networks (TCN) represent relations between dates of events and also allow to express constraints on the possible durations of activities from intervals of values [Dechter *et al.*, 1991]. TCN have been extended to take into account uncertainty of durations of some tasks in realistic applications. A distinction is made between so-called contingent constraints (for example, when the duration of a task cannot be known before its execution) and controllable ones (for example a time interval to be chosen between starting times of two tasks). The resulting network (Simple Temporal Network with Uncertainty) becomes a decision-making problem under uncertainty, and notions of consistency must be refined so as to ensure controllability, that is, ensured consistency despite uncertainty [Morris *et al.*, 2001; Khatib *et al.*, 2001]. As far as we know, the TCN community has extensively worked on the controllability of a network, but the question of optimizing the total duration of set of tasks described by a STPU has not been studied. Nevertheless, not all solutions to an STPU are equally valuable, and solutions minimizing the makespan are of obvious practical interest.

This paper provides a full picture of the complexity of the makespan minimization problem under interval-based uncertainty. It is shown that the only NP-hard problem is the one of finding the greatest lower bound of the float, which is closely related to asserting the possible criticality of a task. All other problems turn out to be polynomial. The fact that the two problems of asserting if an arc is necessary critical or possibly critical do not have the same complexity is rather unexpected.

2 Preliminaries

An activity network is classically defined as a set of activities (or tasks) with given duration times, related to each other by means of precedence constraints. When there are no resource constraints, it can be represented by a directed, connected and acyclic graph. Of major concern, is to minimize the ending time of the last task, also called the makespan of the network. For each task, three quantities have practical importance for the management of the activity network: The *earliest starting time* est_{ij} of an activity (i, j) is the date before which the activity cannot be started without violation of a precedence constraint. The *latest starting time* lst_{ij} of an activity (i, j) is the date after which the activity cannot be started without delaying the end of the project. The *float* f_{ij} of an activity (i, j) is the difference between the latest starting time lst_{ij} and the earliest starting time est_{ij} . An activity is *critical* if and only if its float is equal to zero.

Now, activity durations (weights of the arcs) $(i, j) \in A$ are only known to belong to time intervals $D_{ij} = [d_{ij}^-, d_{ij}^+]$, $d_{ij}^- \geq 0$. We define a configuration as a precise instantiation of the duration of each task $(i, j) \in A$. Ω denotes a configuration, while $d_{ij}(\Omega)$ denotes the duration of activity (i, j) in configuration Ω . $G(i, j)$ is the subgraph of G composed of nodes succeeding i and preceding j .

Computing earliest starting dates is not a difficult issue. Here we solve four problems, stated in [Chanas *et al.*, 2002]: 1) determining the widest intervals LST_{kl} (bounds) of possible values of the latest starting times lst_{kl} of a given activity $(k, l) \in A$, i.e. the interval $LST_{kl} = [lst_{kl}^-, lst_{kl}^+]$ defined by $lst_{kl}^- = \min lst_{kl}(\Omega)$ and $lst_{kl}^+ = \max lst_{kl}(\Omega)$; the problem of computing lst_{kl}^- (resp. lst_{kl}^+) is denoted GBLBST (resp. LUBLST); 2) determining the widest intervals F_{kl} of possible values of floats (total floats) f_{kl} of a given activity $(k, l) \in A$, i.e. the interval $F_{kl} = [f_{kl}^-, f_{kl}^+]$ bounded by $f_{kl}^- = \min f_{kl}(\Omega)$ and $f_{kl}^+ = \max f_{kl}(\Omega)$. The problem of computing f_{kl}^- (resp. f_{kl}^+) is denoted GLBF (resp. LUBF). In both problems minimization and maximization are taken over all possible configurations. The solutions to problems GBLBST, LUBLST, LUBF and GLBF come down to finding an extreme configuration [Dubois *et al.*, 2005] where such bounds are attained. As there are 2^n extreme configurations, it explains the potentially exponential nature of the problem. GBLBST, LUBLST and GLBF have recently been solved in [Chanas

and Zieliński, 2002; Zieliński, 2005], and only GLBF is NP-hard. In this paper, we recall the solutions of these three problems, and present the solution to the last one LUBF, thus providing a full picture of the makespan minimization problem under incomplete information.

3 Evaluating Necessary Criticality

This section presents a new method which can decide if a given task (k, l) is necessary critical.

First, under the assumption that the durations of the predecessors of task (k, l) are precisely known, we already know an algorithm that asserts if (k, l) is necessary critical. It constitutes the basis for computing the LUB and GLB of the latest starting dates in polynomial time in [Zieliński, 2005]. We have extended these results and can provide a general algorithm which asserts if (k, l) is necessary critical in a network G in polynomial time without any consideration of the durations of tasks preceding (k, l) .

Suppose that (k, l) is necessary critical in the sub-network $G(i, n)$, then we proved that we can fix the duration of tasks preceding node i at their lower bounds, without changing the necessary criticality of (k, l) in G . On the contrary, if (k, l) is not necessary critical in $G(i, n)$, then we proved that we can fix the duration of tasks preceding node i at their upper bounds. Applying this idea on networks $G(i, n)$ for i from k down to 1, for each network, tasks preceding (k, l) have always precise duration in $G(i, n)$, and so we can easily assert the necessary criticality in polynomial time. The resulting algorithm requires $O(mn)$ steps to assert if (k, l) is necessary critical in G .

4 Computing the Least Upper Bound on the Floats

To compute the LUB of the floats, we are going to increase step by step the duration of (k, l) from $D_{kl} = d_{kl}^-$ until (k, l) becomes necessary critical with a duration $d_{kl}^- + f^*$. We proved that the value f^* is the LUB of the float of (k, l) .

If we assume that activities preceding (k, l) have precise durations, then $f_{kl}^+ = l_{st_{kl}}^+ - e_{st_{kl}}^+$. So at each step of the computation, we can easily compute f_i^+ representing the float of (k, l) in network $G(i, n)$. After initialization of durations in the network by the algorithm which asserts if (k, l) is necessary critical, if (k, l) is not necessary critical, we have to increase the duration of (k, l) by the minimal non-null f_i^+ for i preceding k , then go back to the test of asserting necessary criticality. When (k, l) becomes necessary critical, we have found f_{kl}^+ . At each iteration, (k, l) becomes necessary critical in at least one new network $G(i, n)$, and so the resulting algorithm is polynomial and takes $O(n^3m)$.

The following table summarizes the complexity of the different problems of the PERT/CPM on intervals:

Earliest starting date (all tasks)	GLB	P	$O(n+m)$
	LUB	P	$O(n+m)$
Latest starting date (one task)	GLB	P	$O(mn)$
	LUB	P	$O(mn)$
Float (all tasks)	GLB	NP-Hard	$O((n+m) P)$
Float (one task)	LUB	P	$O(n^3m)$

In this table we give the complexity of the theoretically best known algorithm which computes the quantity of interest. In particular, the path algorithm [Dubois *et al.*, 2005] which computes the GLB and LUB of floats and latest starting dates runs in $O((n+m)|P|)$ time, where $|P|$ is the number of path of the graph G . The real complexity thus depends of the topology of the network.

5 Conclusion

This paper proposes a complete solution to the criticality analysis of a network of activities when durations of tasks are ill-known. It is shown that moving from precise to imprecise durations radically changes the complexity of the problem, ruining the traditional critical path method. Part of the problem, pertaining to the GLB of the float, becomes NP-hard, the other questions remaining polynomial, although not straightforwardly so. These complexity results shed light on reasons why the more familiar stochastic counterparts to this basic scheduling problem are so difficult to handle, part of the difficulty being already present in the interval analysis. The proposed algorithms can be of obvious relevance to practical predictive project scheduling problems where durations of tasks are not known in advance. Clearly, due to the basic nature of the addressed problem, several lines of research can be envisaged for future research. For instance one may assume that part of the tasks durations are controllable and additional constraints relating durations and starting or ending times may be available. Then one obtains a makespan minimization problem in the more general setting of STPU's.

References

- [Chanas and Zieliński, 2002] S. Chanas and P. Zieliński. The computational complexity of the criticality problems in a network with interval activity times. *EJOR*, 136:541–550, 2002.
- [Chanas *et al.*, 2002] S. Chanas, D. Dubois, and P. Zieliński. On the sure criticality of tasks in activity networks with imprecise durations. *IEEE Transactions on Systems, Man, and Cybernetics*, 34:393–407, 2002.
- [Dechter *et al.*, 1991] R. Dechter, I. Meiri, and J. Pearl. Temporal constraint networks. *AI*, 49:61–95, 1991.
- [Dubois *et al.*, 2005] D. Dubois, H. Fargier, and J. Fortin. Computational methods for determining the latest starting times and floats of tasks in interval-valued activity networks. *J. of Intelligent Manufacturing*, 2005. To appear.
- [Khatib *et al.*, 2001] L. Khatib, P. Morris, R. Morris, and F. Rossi. Temporal constraint reasoning with preferences. In *IJCAI*, pages 322–327, 2001.
- [Morris *et al.*, 2001] Paul Morris, Nicola Muscettola, and Thierry Vidal. Dynamic control of plans with temporal uncertainty. In *IJCAI*, pages 494–502, 2001.
- [Zieliński, 2005] P. Zieliński. On computing the latest starting times and floats of activities in a network with imprecise durations. *Fuzzy Sets and Systems*, 150:53–76, 2005.