# Constraint programming for scheduling problems

Sütő Evelyne

December 22, 2019

## Contents

**1 Introduction** **1**

**2 Constructing university timetable using constraint satisfaction programming** **2**

**3 Fuzzy project scheduling using constraint programming** **3**
3.1 Example of project scheduling . . . . . 4

**4 Conclusion** **6**

**Abstract**

The aim of this research report is to present some real world applications of constraint programming in scheduling related problems. Since constraints are naturally occurring elements of real world scheduling tasks it is already simple enough to define the constraints. Another aspect that makes the scheduling issue well suited for the constraint programming approach is that these are usually large conbinatorial problems, so it reasonable to use the different consistency techniques and constraint propagations to reduce the search space for the generated solutions. In the beginning of this paper we will explore the general description of a constraint satisfaction and constraint programming. Afterwards we will consider some specific use cases where constraint programming was successfully apply to solve a scheduling and planning problem. These use cases are the timetable generation and fuzzy project scheduling.

# 1 Introduction

Constraint programming (CP) is used in declarative description and effective solving of large, particularly combinatorial, problems especially in areas of planning and scheduling. Not only it is based on a strong theoretical foundation but it is used to solve problems of commercial interest, in particular, in areas of modelling heterogeneous optimisation and satisfaction problems. Constraints are naturally occuring elements of our world, they can be described as a relation of different variables which can have a set of possible values in their domains. Constraints should be strictly declarative, by specifying the relationship of the constraints without giving specifications of the computational procedure that enforces them. By using constraint programming the aim is to solve problems based on the stated constraints by finding solutions which can satisfy them. [Barták, 1999]

The first ancestors of constraint programming can be considered the early Logic programming, which was focused on stating what has to be solved instead of how should it be solved.

A Constraint Satisfaction Problem (CSP) is defined as:

- a set of variables $X = x_1, ..., x_n$
- for each variable $x_i$, a finite set $D_i$ of possible values (its domain)
- a set of constraints restricting the values that the variables can simultaneously take.

A solution of the problem is an assignment of a value from its domain to every variable, in such a way that all constraints are satisfied at once.

These solutions are found by using systematic search algorithms such as backtracking. The simple backtracking algorithm is usually extended with consistency techniques and constraint propagation to avoid problems like: repeated failure due to the same reason, redundant work or late detection of conflict.

[Barták, 1999]

Constraint satisfaction can be implemented to any problem that can be described by a list of constraints. This means that the problem itself has to be modelled in a very specific way for us to use the constraint programming techniques on them. One of the most successful

areas where this approach is applied is related to scheduling problems. This is mainly because the problem itself is defined by its constraints.

In this paper we will be exploring some real life applications of constraint programming in scheduling and planning problems.

# 2 Constructing university timetable using constraint satisfaction programming

The timetable building problem is an NP complete one, which makes it an interesting research topic. It is difficult to find an optimal solution, because it naturally implies resource sharing and time constraints it is natural to use constraint programming to solve it.

As defined in [Zhang and Lau, 2005] "Timetabling is the allocation, subject to constraints, of given resources to objects being placed in space time, in such a way as to satisfy as nearly as possible a set of desirable objectives." The constraints of the timetabling problem: students attend one class at the same time, the room must be big enough for all the attending students, no core subject is scheduled at the same time, and only one class is scheduled in one room at any one timeslot.

Elements that are constrained in this case are: set of students, subjects, time slots and rooms. The model for a timetabling problem as a CSP is as follows: a timetable is a constrained variable the value of which is a function associating a value to each slot in time t. The timetable item is given by the set of subjects. Note that the subject can be offered as a lecture or a tutorial, which is considered as a timetable item. Basically our task consists ininstantiation of the set of three tuples CSP (timetable item, classroom, time), i.e., each lecture or tutorial of a subject has assigned its set of classroom and time. In solving the problem, we define each timetable item as an activity and the room as resource [Zhang and Lau, 2005]

In order to implement the solution for this problem the researchers have chosen to use the ILOG system, which decmposes the problem by separating the models from the search algorithms, so they could apply different algorithms to apply on the same model. The model was build with the ILOG scheduler and the resources represented with IloActivity and IloUnaryResources. Then they apply the ILOG Solver which generates a goal and searches for a solution. In the solution they use different search strategies defined in ILOG, such as ILORankForward, IloRankBackward etc. The result is analyzed from the perspectives of number of fails and number of choice points. Failure refers to the node which backtracks when the search cannot find the goal. The number of fails refers to the number of backtrack in the search process until a goal is found. Choice point refers to the node that has been explored or visited in the search process. Therefore the number of choice points refers to the number of nodes that has been visited in the search process.

They manage to solve the timetableing problem with constraint satisfaction programming and they also find that enforcing tight constraint level and ranking the constraints by positioning the constraints at the beginning of the activities can lead to better result.

# 3 Fuzzy project scheduling using constraint programming

Another possible scheduling problem can be the scheduling of projects. A possible solution for this using constraint programming can be seen in [Relich, 2013]. In order to model the imprecision of activity duration and cost they use fuzzy elements such as $\alpha$ cuts. This fuzzy aspect of the implementation also makes it possible to get to different solutions in terms of risk management, by modelling optimistic, pessimistic and intermediate scenarios.

While working on independent project their management can be straightforward since it only has the time and cost estimations are needed to be considered. However, most of the times when companies are developing different projects these can be very much dependent of each other and it is challenging to plan them in a way that they can be developed in the most agile way possible. Project management problems typically consist of resource planning and scheduling decisions. In the context of resource management, it is often required to know how much a particular project will cost, what resources are needed, what resource allocation procedure can ensure the completion of a project in target time, etc. [Relich, 2013]

A simple CSP problem can be defined as the set of $CSP = (V, D), C$ where V is a set of variables, D is the domain of the variables and C is a set of constraints that are applied on these variables. Fuzzy CSP is defined in a similar way with $CSP = (V', D), C$ where V' is a finite set of fuzzy variables, D is the set of domains for th fuzzy variables and C is the set of constraints linking and limiting these variables.

Given a set of projects $P = P_1, P_2, ..., P_I$, where the project P i consists of J activities: $P_i = A_{i,1}, ..., A_{i,j}, ..., A_{i,J}$. The j-th activity of i-th project that is specified by the starting time of the activity $s_{i,j,1}$ (i.e. the time counted from the beginning of the time horizon H), the completion time of the activity $s_{i,j,2}$ , and the duration of the activity $t_{i,j}$ . The project P i is described as an activity-on- node network, where nodes represent the activities and the arcs determine the precedence constraints between activities. According to this the precedence constraints are as follows: $s_{i,j} + t_{i,j} <= s_{i,n}$ (for the n-th activity follows the i-th one), $s_{i,j} + t_{i,j} <= s_{i,n}; s_{i,j+1} + t_{i,j+1} <= s_{i,n}; ...; s_{i,j+n} + t_{i,j+m} <= s_{i,n}$ (for the n-th activity follows other activities), and $s_{i,n} + t_{i,n} <= s_{i,j}; s_{i,n} + t_{i,n} <= s_{i,j+1}; ...; s_{i,n} + t_{i,n} <= s_{i,j+m}$ (for the n-th activity is followed by other activities). The imprecise variables are defined by $\alpha$ cuts, where the $\alpha$ -cut is a crisp set consisting of elements of A which belong to the fuzzy set at least to a degree of $\alpha$ ( 0 ¡ $\alpha$ ¡= 1 ). An $\alpha$ -cut is a method of defuzzifying a fuzzy set to a crisp set at desired $\alpha$ -levels that correspond to the perceived risk ($\alpha$=1 meaning no risk, $\alpha$=0+ meaning the highest risk). The main objective of fuzzy project scheduling is to apply fuzzy set theory concepts to the scheduling of real world projects where task duration can be specified as fuzzy numbers instead of crisp numbers. With the help of fuzzy project scheduling we can define fuzzy start and completion dates for each projects. The fuzzy start indicates the uncertainty of completion of proceeding projects, activities, whereas the completion is the sum of the fuzzy start and activity duration. Calculated using the fuzzy + operator:

$s + t =< a, b, c > + < d, e, f >=< a + d, b + e, c + f >$

The cost for unit of time calculated by dividing the cost of every activity by its duration. Since these can be defines for optimistic and pessimistic scenarios we can calculate $minD_\alpha$

(best case scenerio, activity starts as soon as possible) and $maxD_\alpha$ (activity starts at latest possible time).

$minD_\alpha = [\alpha(b–a) + a, \alpha(e–d) + d]$

$maxD_\alpha = [\alpha(b–c) + c, \alpha(e–f) + f]$

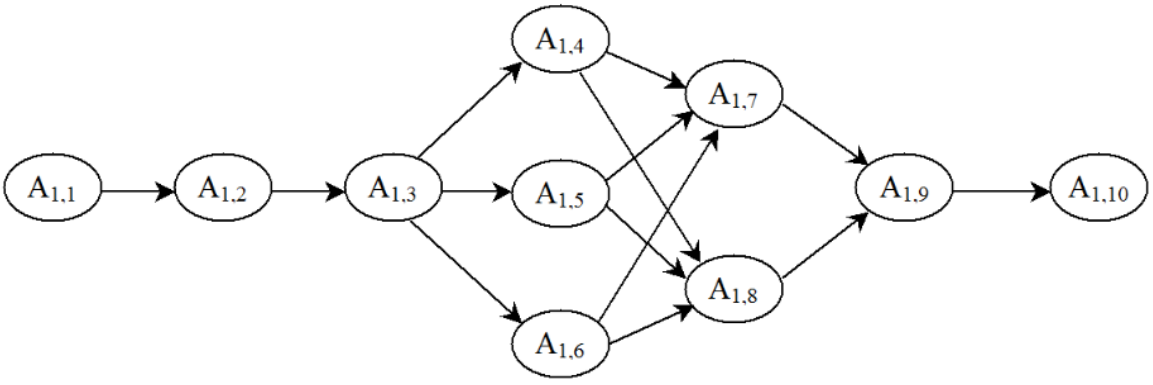
The financial means are allocated to the activity $A_{i,j}$ . taking into account all $\alpha$-levels of a fuzzy number.

## 3.1 Example of project scheduling

Consider having 3 dependent projects $P = P_1, P_2, P_3$ which need to be done in 32 months and the budget is 1159 money unit. The duration of each activity as estimated by using past experiences or estimated by experts in this field. Some of the activities have constant durations, while some activities are defined by fuzzy duration e.g "about 6 weeks" which would be between 5-7 weeks in reality.

The aim of the Fuzzy scheduling is to create a fuzzy cash flow for different risk levels, which would help the project managers to determine which schedule would be the best option, by considering the different levels in terms of duration and cost efficiency. The answer to the questions is connected with the determination of the starting time of project portfolio activities s i, j and the allocation of financial means to the activities by different $\alpha$ -level $dp_{i,j,\alpha}$ .

Figure 1: Example of activities and dependencies of project 1. Source: [Relich, 2013]




As a result, using these methodologies one can generate different cash flow graph for the different risk levels. In the best case, the project portfolio will be completed in 26 months with the total cost of 921 m.u., whereas in the worst in 32 months with the total cost of 1,119 m.u. This allows the decision-maker to consider a wide range of analyses, including the requirement of the cost allocation in the horizon of project portfolio. [Relich, 2013]

Since the problem of project scheduling always holds some uncertainty as well, because of problematic activity estimations and unseen problems using fuzzy methods to solve this problem seems to be a good approach. It is also important to mention that having constraints which are naturally defined for each projects in terms of resource management it is fairly straightforward to use constraint programming to model this problem. Having a generated

Figure 2: Example of activities and dependencies of project 2. Source: [Relich, 2013]

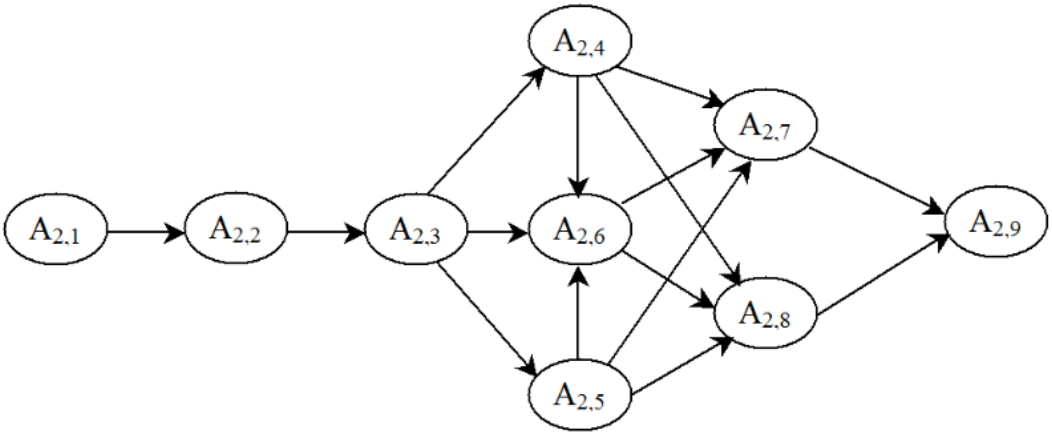
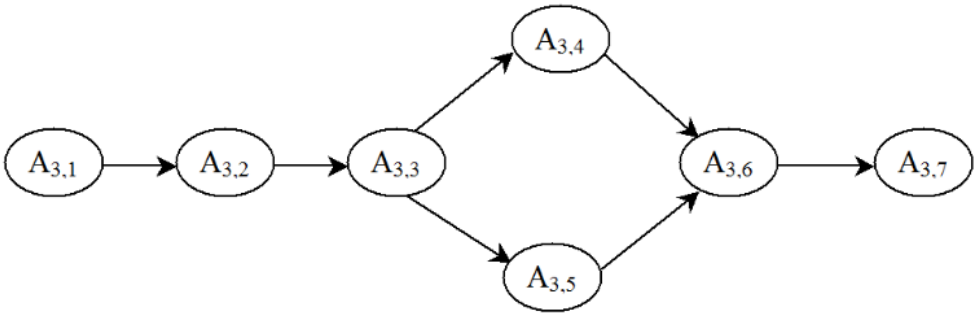
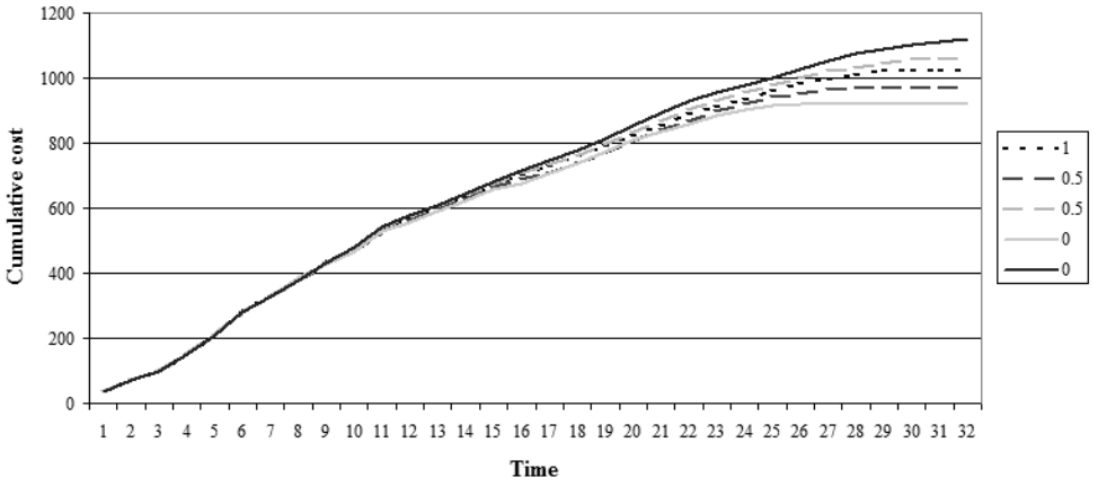



Figure 3: Example of activities and dependencies of project 3. Source: [Relich, 2013]




fuzzy cash flow could be very helpful for project managers for decision making and efficient cost/time estimation as well.

Figure 4: Example of fuzzy project cash flow with 5 different levels. e.g. At $\mu$ =0.5, there is an optimistic scenario below and a pessimistic one above (dashed line). Source: [Relich, 2013]




# 4 Conclusion

In this paper we have examined some applications of constraint programming in scheduling problems. We have mainly considered only two use cases for this: project scheduling and timetable generation, however there are many other examples in the literature for other similarly defined cases as well. For example applications in medical resident scheduling [Topaloglu and Ozkarahan, 2011], nurse scheduling [Abdennadher and Schlenker, 1999], airport runway scheduling [Bennell et al., 2011] are just a few of these real life uses.

As we've seen, since constraint programming in general is mostly interested in what is the problem that needs to be solved, rather than how should the problem be solved. This means that the main challenge in these cases is to create the right model for our tasks that we want to solve, because if the specified variables or domains are not defined correctly this could mean that the solution that we find is not a solution for our desired problem.

Overall I think that constraint programming as a solution for scheduling problem is very promising and already successful, but we need to be sure that the optimization steps of consistency technique and constraint propagation are used in this solution to ensure that we get results in optimal time for these combinatorial problems.

# References

[Abdennadher and Schlenker, 1999] Abdennadher, S. and Schlenker, H. (1999). Nurse scheduling using constraint logic programming. In *AAAI/IAAI*, pages 838–843.

[Barták, 1999] Barták, R. (1999). Constraint programming: In pursuit of the holy grail. In *Proceedings of the Week of Doctoral Students (WDS99)*, volume 4, pages 555–564. MatFyzPress Prague.

[Bennell et al., 2011] Bennell, J. A., Mesgarpour, M., and Potts, C. N. (2011). Airport runway scheduling. *4OR*, 9(2):115.

[Relich, 2013] Relich, M. (2013). Fuzzy project scheduling using constraint programming. *Applied Computer Science*, 9(1).

[Topaloglu and Ozkarahan, 2011] Topaloglu, S. and Ozkarahan, I. (2011). A constraint programming-based solution approach for medical resident scheduling problems. *Computers & Operations Research*, 38(1):246–255.

[Zhang and Lau, 2005] Zhang, L. and Lau, S. (2005). Constructing university timetable using constraint satisfaction programming approach. In *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)*, volume 2, pages 55–60. IEEE.