CONSTRAINT PROGRAMMING FOR SCHEDULING PROBLEMS

Sütő Evelyne [1]

[1]Babeş-Bolyai University, Computer Science, Cluj-Napoca

MOTIVATIONS TO US CONSTRAINT PROGRAMMING

WHY FOCUS ON SCHEDULING PROBLEMS

- Resources are very expensive (time or money measures) and we need optimal solutions for scheduling
- Need for intelligent optimal systems
- Similar solutions can be implemented for many problems such as personnel, machine , project scheduling etc

WHY USE CONSTRAINT PROGRAMMING?

- Scheduling problems naturally defined with their constraints in mind, thus it is easy to model with CSP
- Depending on resources and needed constraints to satisfy it can become a combinatorial problem
- Concentrates on the WHAT of the problem instead of the HOW
- Many effective ways already implemented that can boost up the search for the solutions

EXAMPLES

EXAMPLES

- Generating university timetables [Zhang and Lau, 2005]
- Project scheduling [Relich, 2013]

UNIVERSITY TIMETABLES

- NP complete problem, difficult to find optimal solution
- Constraints based on time restrictions and resource sharing (class rooms, teachers)
- 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.

UNIVERSITY TIMETABLES

- Each timetable item is an activity and rooms are resources
- Solution: Defined model in ILOG
- ILOG is a constraint optimization software developed by IBM for CSP (mainly scheduling problems)
- https://www.ibm.com/analytics/cplex-cp-optimizer

UNIVERSITY TIMETABLES

- ILOG decompesses programs by separating the model from the search algorithm itself
- Model built with ILOG scheduler
- Resources represented in IloActivity
- The ILOG Solver generates a goal and goes thorough the search space to find the solution

- Results analyze the number of backtracked nodes (fails) and number of choice point (nr of visited nodes)

Figure 1: Results [Zhang and Lau, 2005]

Table 1. Comparisons of results for different scenarios

Scenarios	Number of fails	Number of choice points	CPU time (seconds)
IloRankForward	373	761	4.91
IloRankBackward	380	766	4.96
IloSetTimesForward	2377	2767	5.86
IloSetTimesBackward	6460	6856	8.72
IloLow+IloRankForward	373	761	5.23
IloLow+IloRankBackward	380	766	5.3
IloLow+IloSetTimesForward	2377	2767	5.87
IloLow+IloSetTimesBackward	6460	6856	8.72
IloHigh+IloRankForward	3	391	3.91
IloHigh+IloRankBackward	10	396	3.93
loHigh+IloSetTimesForward	2007	2397	5.25
IloHigh+IloSetTimesBackforwad	6090	6486	9.33
IloExtended+IloRankForward	3	391	20.3
IloExtended +IloRankBackward	10	396	20.75
IloExtended +IloSetTimesForward	2007	2397	31.34
IloExtended+IloSetTimesBackward	6090	6486	68.25

FUZZY PROJECT SCHEDULING

- When developing multiple project at the same time sometimes we need to share some resources between then
- Constraint: project/task dependencies, human resource management, financial resource management
- Need for tools to help in scheduling the projects

FUZZY PROJECT SCHEDULING

- Estimating needed time and money for development is very challenging
- Need to include imprecise estimation for time/money
- Solution: consider 3 levels of risk management -> optimistic, pessimistic, intermediate by implementing α cuts

FUZZY PROJECT SCHEDULING

- Set of projects $P = P_1, P_2, ..., P_I$, where the project P i consists of J activities: $P_i = A_{i,1}, ..., A_{i,j}, ..., A_{i,J}$
- Each activity is defined by their start time and duration
- The imprecise variables are defined by α cuts, where the α -cut is a crisp set consisting of elements of A which belong to the fuzzy set at least to a degree of α (0 < α <= 1). An α -cut is a method of defuzzifying a fuzzy set to a crisp set at desired α -levels that correspond to the perceived risk (α=1 meaning no risk, α=0+ meaning the highest risk)

- Cost for unit of time calculated by dividing the cost of every activity by its duration.

- Financial cost allocating by taking into account all α-levels of a fuzzy number

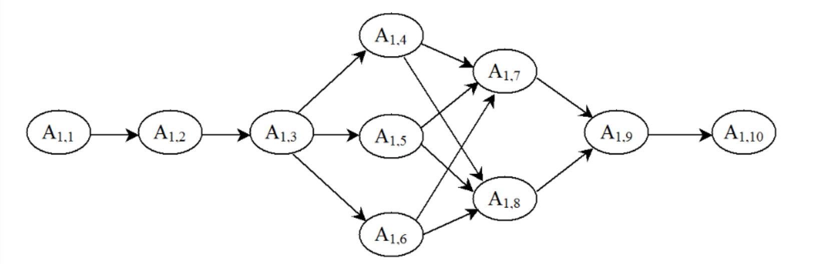
- $minD_\alpha$ (best case scenerio, activity starts as soon as possible)
- $maxD_\alpha$ (activity starts at latest possible time).

$$minD_\alpha = [\alpha(ba) + a, \alpha(ed) + d]$$

$$maxD_\alpha = [\alpha(bc) + c, \alpha(ef) + f]$$

FUZZY PROJECT SCHEDULING

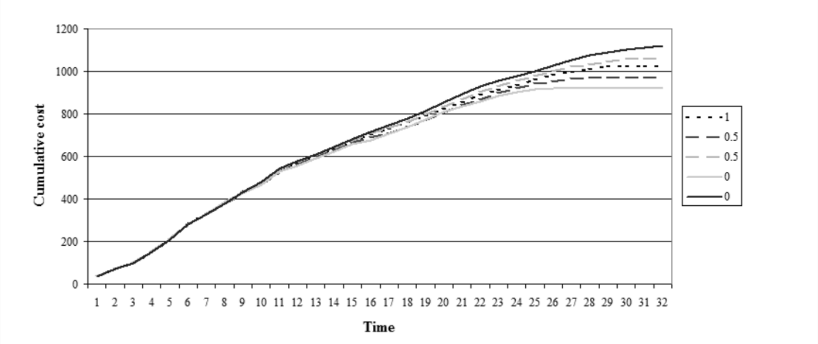
Figure 2: Example of activities and dependencies of project 1. Source: [Relich, 2013]



FUZZY PROJECT SCHEDULING

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

Figure 3: Example of fuzzy project cash flow with 5 different levels. e.g. At μ =0.5, there is an optimistic scenario below and a pessimistic one above (dashed line). Source: [Relich, 2013]



CONCLUSION

CONCLUSION

- The interest in scheduling is big, because it allows good resource allocation and time optimization
- Many other use cases other than the two mentioned in this paper: nurse scheduling, doctor scheduling, virual machine scheduling etc
- Constraint programming is successfully applied to these problems by using well defined elements for consistency technique and constraint propagation

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

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.