

Solving Shift Design Problems with Answer Set Programming

Masterstudium:
Software Engineering & Internet Computing

Michael Abseher

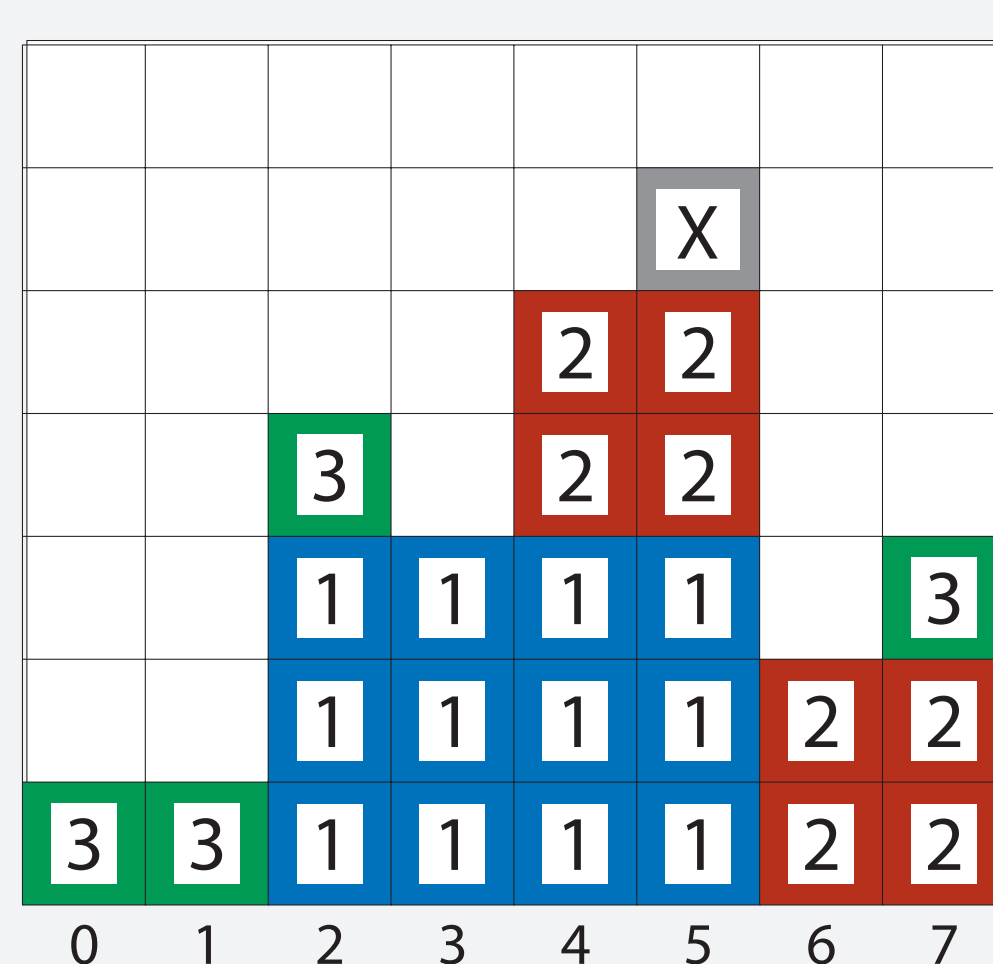
Motivation

- Workforce scheduling is an crucial task in many organizations.
- Starting in the 1950's, powerful heuristic and exact approaches were developed to automate the process of generating shifts.
- Software for designing shifts often contains thousands of lines of code, making the source code hard to maintain and even small extensions often lead to unreasonably high effort.
- ASP as a declarative programming paradigm could be a promising option to overcome the problem of rapidly decreasing maintainability:

Describe what the result for a given input should look like
... instead of specifying how the input has to be processed
... in order to save a lot of effort in the development process

Shift Design

- Shift Design is about
 - ... finding the minimal number of shifts
 - ... for a given problem instance (e.g. company needs)
 - ... so that the deviation from the requirements is minimized
- The problem instances contain
 - ... a set of templates for feasible shifts
 - ... the demand of workers at each point in time
- The expected result of Shift Design is
 - ... a subset of the set of feasible shifts
 - ... the number of assigned workers for each shift



- The image above shows a possible solution of the Shift Design Problem for an problem instance with eight time slots where every shift has a duration of four time slots:
 - Letter X denotes a shortage of one worker at time slot 5
 - Shift 1 starts at time slot 2 with three workers
 - Shift 2 starts at time slot 4 with two workers
 - Shift 3 starts at time slot 7 with one worker

Scientific Objective

- Modelling of real-world Shift Design Problems using Answer Set Programming
- Investigation of the performance characteristics of different ASP-environments
- Comparing our modelling approaches with state-of-the-art algorithms

Answer Set Programming (ASP)

- ASP is a declarative programming technique.
- Logic programs in ASP consist of a set of rules:

```

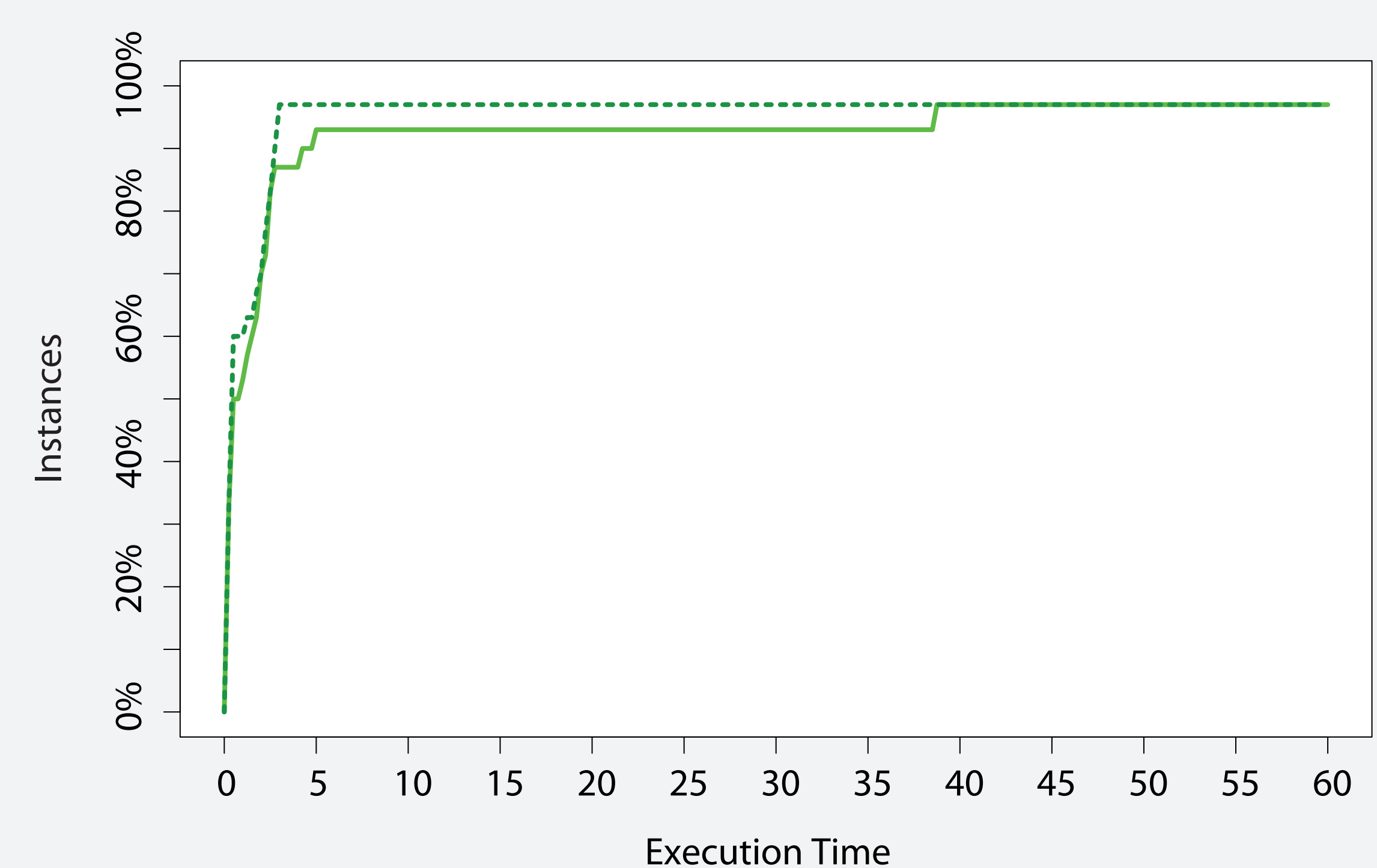
1 Requirement [
2     assigned(Start, Length, Amount) = Amount :
3     min_requirement(Start, Length, Requirement1) :
4     amount(Amount) :
5     Start <= Time : Start + Length - 1 >= Time :
6     Amount <= Requirement1,
7     assigned(Start, Length, Amount) = Amount :
8     min_requirement(Start, Length, Requirement1) :
9     amount(Amount) :
10    Start + Length - timeslots - 1 >= Time :
11    Amount <= Requirement1
12 ] Requirement :- required(Time, Requirement).

13 :- assigned(Start, Length, Amount1),
14    assigned(Start, Length, Amount2),
15    Amount1 > Amount2.
```

- The code fragment above shows the actual generation of shifts relying on defined requirements. The rule in lines 13 to 15 is used to ensure that the number of assigned workers is unambiguous.

Obtained Results

- We propose different modelling approaches for the Shift Design Problem using Answer Set Programming, each with its own strengths and weaknesses.
- Different solvers were used for our experiments to investigate differences between the implementations and to obtain more representative results.
- For the general case of the Shift Design Problem, where deviation from the actual staff demand is allowed, ASP is currently not competitive with existing solvers from previous literature.
- Encouraging results have been obtained for instances of the Shift Design Problem where a exact solution without any deviation from the requirements exists.



- The diagram above illustrates the number of solved instances for two solvers using the same input program developed by us in relation to the execution time (in minutes) for a benchmark set of 30 instances where a zero-deviation solution exists.