FAKULTÄT FÜR !NFORMATIK

Faculty of Informatics

dbai Database and Artificial Intelligence Group

Shift Design with Answer Set Programming

LPNMR 2015

Michael Abseher

TU Wien, Vienna, Austria

Martin Gebser

University of Potsdam, Potsdam, Germany

Nysret Musliu

TU Wien, Vienna, Austria

Torsten Schaub

University of Potsdam, Potsdam, Germany

Stefan Woltran

TU Wien, Vienna, Austria

September 29, 2015

Relevance

Finding appropriate staff schedules is of great importance because ...

- work schedules influence health, social life, and motivation of employees at work.
- workforce requirements must be met to ensure the quality of services and operations.
- the required number of employees fluctuates throughout time periods, while operations dealing with critical tasks are often performed around the clock.
    - Examples include ...
        - emergency services
        - air traffic control
        - call centers
        - etc.

Relevance

Finding appropriate staff schedules is of great importance because ...

- work schedules influence health, social life, and motivation of employees at work.
- workforce requirements must be met to ensure the quality of services and operations.
- the required number of employees fluctuates throughout time periods, while operations dealing with critical tasks are often performed around the clock.
  - Examples include ...
    - emergency services
    - air traffic control
    - call centers
    - etc.
- Related Work:
  Team-Building at the Gioia-Tauro seaport [Grasso et al., 2010]

The Shift Design Problem

Input:

- A set of consecutive *time slots* of equal length (*planning horizon*)
- For each time slot, the number of required employees
- A set of *shift types* with associated parameters
    - *min-start* and *max-start*
    - *min-length* and *max-length*

shift type	*min-start*	*max-start*	*min-length*	*max-length*
M	07:00	08:00	07:00	09:00
D	10:30	11:30	07:00	09:00
A	14:00	16:00	07:00	08:00
N	22:00	24:00	07:00	09:00

Table 1: Example of possible shift types

The Shift Design Problem

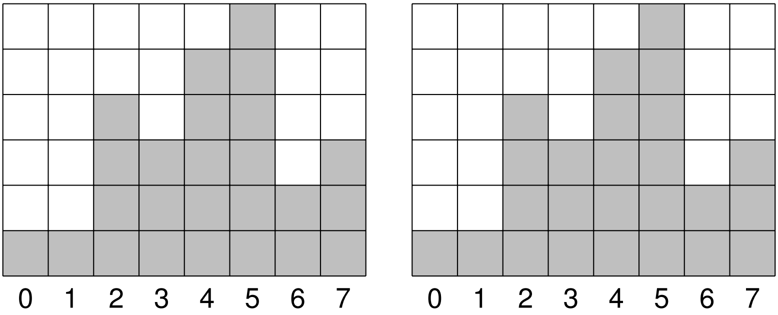
Solution:

- A set of *shifts* with associated parameters
  - The starting time *start*
  - The duration of the shift *length*
  - The number employees assigned to the respective shift
- Each of the generated shifts must belong to some shift type!

Optimization Criteria:

- Minimize the number of shifts
- Minimize understaffing
- Minimize overstaffing

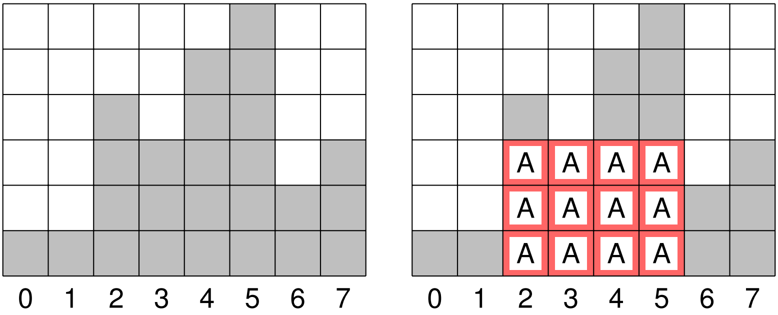
A Small Example



shift type	*min-start*	*max-start*	*min-length*	*max-length*
A	2	2	2	4
B	4	5	2	4
C	6	7	2	4

Table 2: Allowed shift types

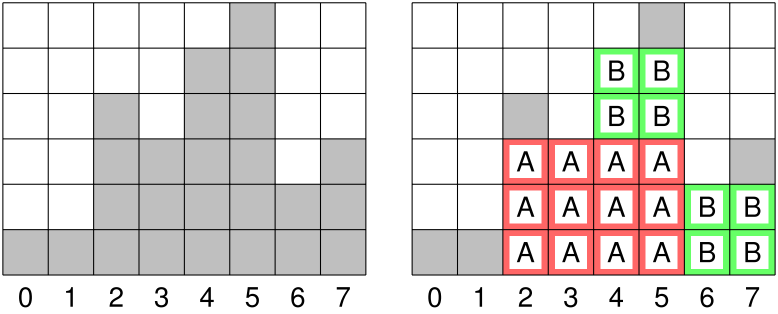
A Small Example



shift type	*min-start*	*max-start*	*min-length*	*max-length*
A	2	2	2	4
B	4	5	2	4
C	6	7	2	4

Table 2: Allowed shift types

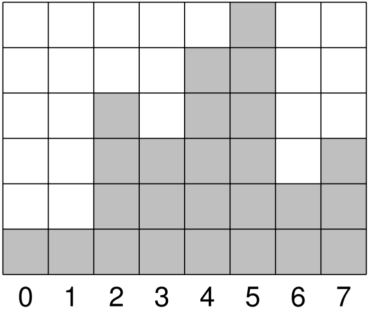
A Small Example

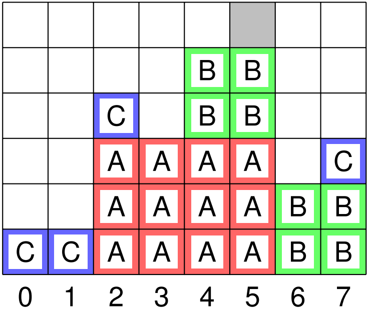


shift type	*min-start*	*max-start*	*min-length*	*max-length*
A	2	2	2	4
B	4	5	2	4
C	6	7	2	4

Table 2: Allowed shift types

A Small Example

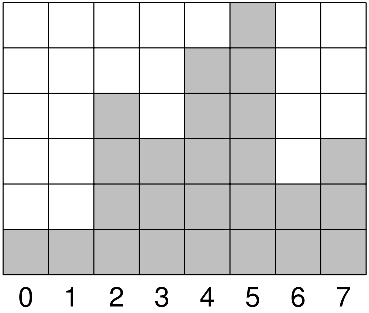


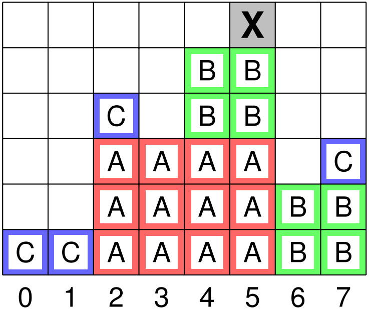


shift type	*min-start*	*max-start*	*min-length*	*max-length*
A	2	2	2	4
B	4	5	2	4
C	6	7	2	4

Table 2: Allowed shift types

A Small Example





shift type	*min-start*	*max-start*	*min-length*	*max-length*
A	2	2	2	4
B	4	5	2	4
C	6	7	2	4

Table 2: Allowed shift types

Shift Design in ASP

Main Challenge

The Shift Design Problem lacks helpful hard constraints.

- No obvious restrictions on overstaffing
- No obvious restrictions on understaffing
- No obvious restrictions on the number of shifts

Shift Design in ASP

Main Challenge

The Shift Design Problem lacks helpful hard constraints.

- No obvious restrictions on overstaffing
- No obvious restrictions on understaffing
- No obvious restrictions on the number of shifts

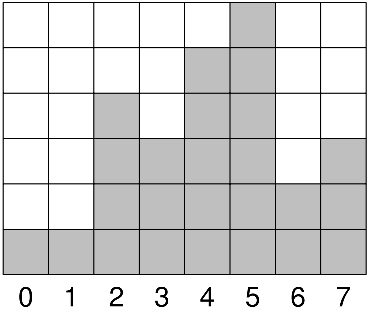
Important Ingredients of Our Encoding

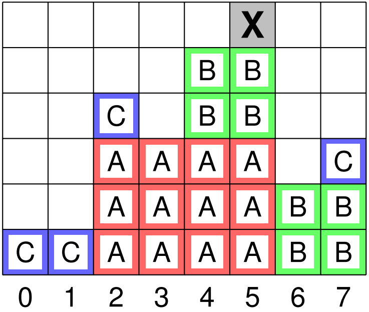
- Inverted decision strategy
  - Instead of guessing start, length and workers for a shift explicitly ...
  - Guess coverage of personnel requirements for a time slot and ...
  - Determine the shifts on this basis.
- Use closed intervals for quantitative values
  - Make implicit information explicit for the solver to speed up search.

The Small Example Revisited

Inversion of Decision Strategy

- Naive guessing of concrete values for start, length and the number of workers for each shift involves three dimensions and leads to an unnecessary complex decision strategy.

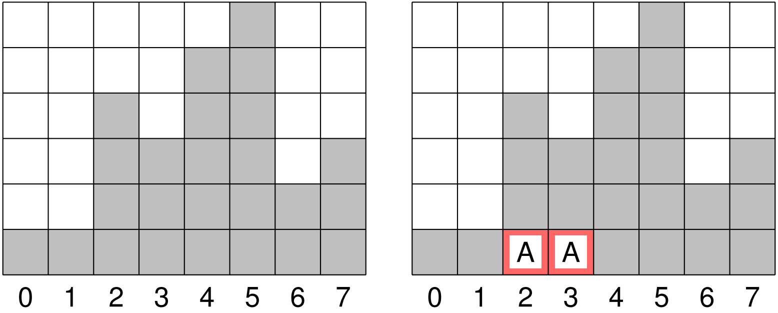




The Small Example Revisited

Inversion of Decision Strategy

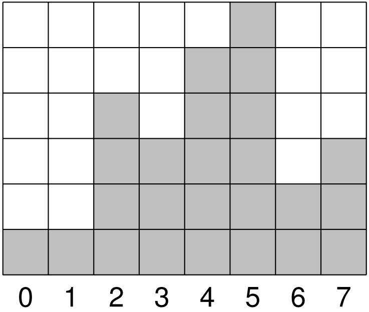
- Naive guessing of concrete values for start, length and the number of workers for each shift involves three dimensions and leads to an unnecessary complex decision strategy.

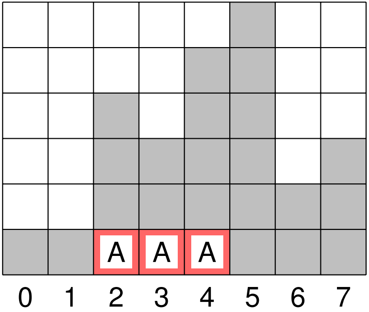


The Small Example Revisited

Inversion of Decision Strategy

- Naive guessing of concrete values for start, length and the number of workers for each shift involves three dimensions and leads to an unnecessary complex decision strategy.

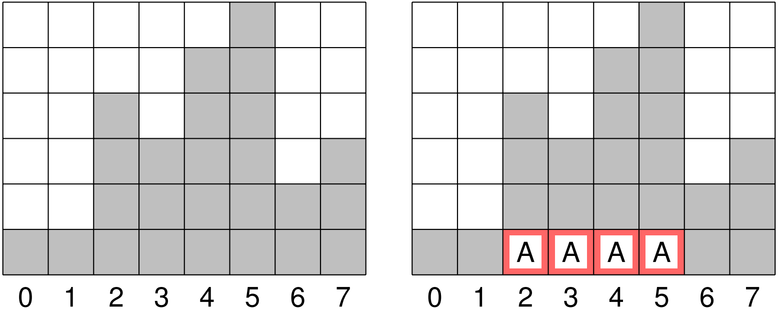




The Small Example Revisited

Inversion of Decision Strategy

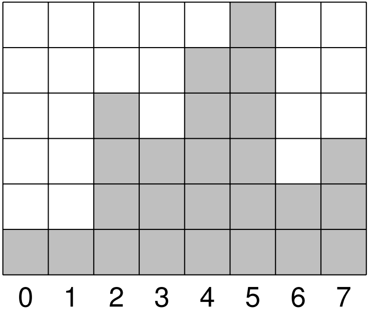
- Naive guessing of concrete values for start, length and the number of workers for each shift involves three dimensions and leads to an unnecessary complex decision strategy.

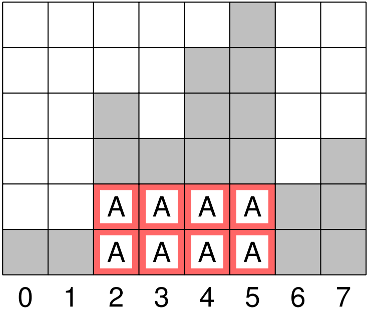


The Small Example Revisited

Inversion of Decision Strategy

- Naive guessing of concrete values for start, length and the number of workers for each shift involves three dimensions and leads to an unnecessary complex decision strategy.

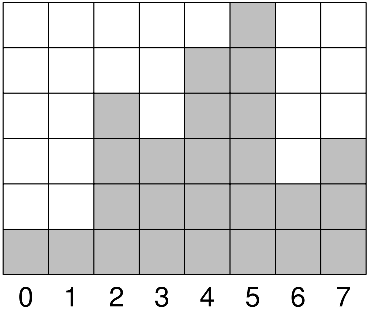


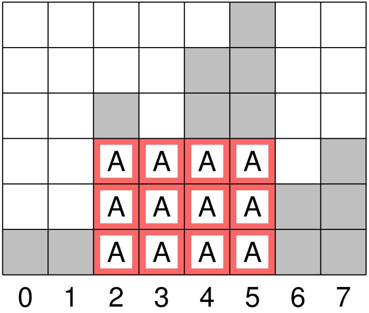


The Small Example Revisited

Inversion of Decision Strategy

- Naive guessing of concrete values for start, length and the number of workers for each shift involves three dimensions and leads to an unnecessary complex decision strategy.

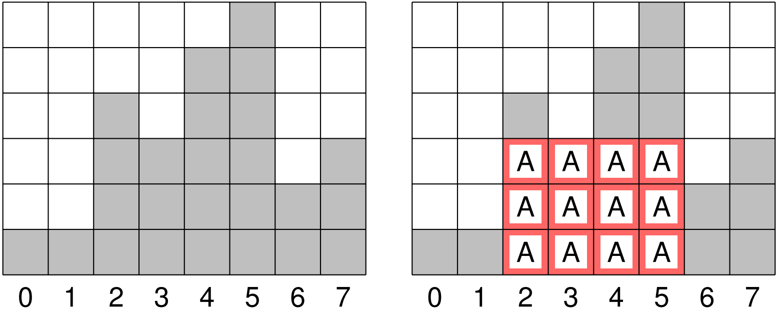




The Small Example Revisited

Inversion of Decision Strategy

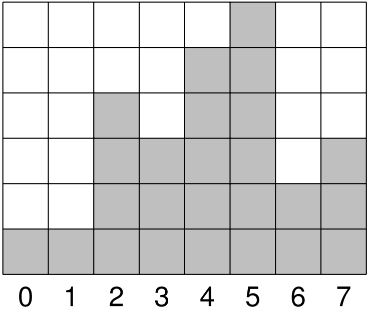
- Naive guessing of concrete values for start, length and the number of workers for each shift involves three dimensions and leads to an unnecessary complex decision strategy.
- Isn't it easier to just guess the coverage of requirements?

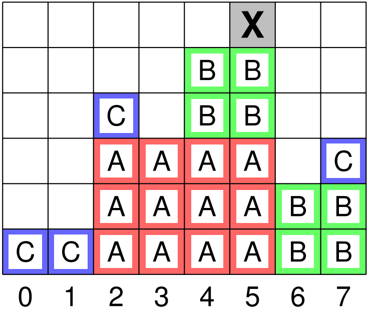


The Small Example Revisited

Inversion of Decision Strategy

- Naive guessing of concrete values for start, length and the number of workers for each shift involves three dimensions and leads to an unnecessary complex decision strategy.
- Isn’t it easier to just guess the coverage of requirements?
  In fact, this is indeed the case.

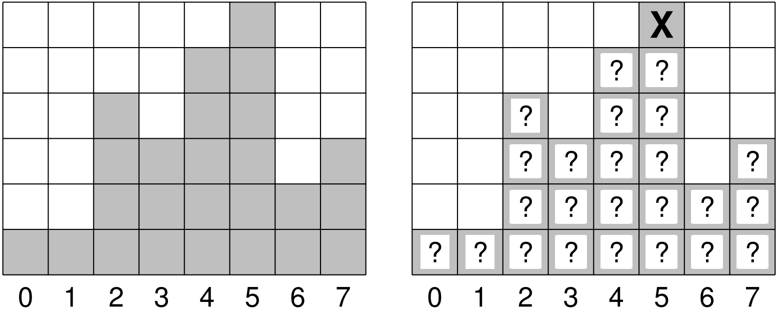




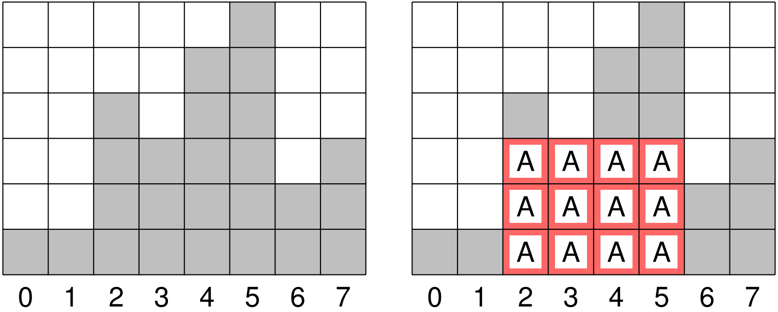
The Small Example Revisited

Inversion of Decision Strategy

- Naive guessing of concrete values for start, length and the number of workers for each shift involves three dimensions and leads to an unnecessary complex decision strategy.
- Isn't it easier to just guess the coverage of requirements? In fact, this is indeed the case.



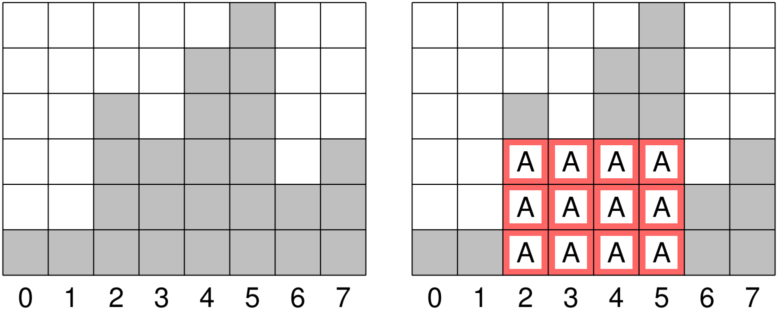
The Small Example Revisited



What we can see at one glance:

- One shift of type “A” is scheduled:
  - Start: 2
  - Length: 4
  - Workers: 3

The Small Example Revisited

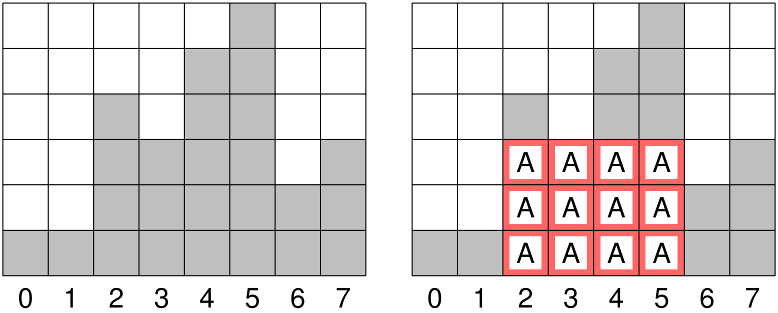


What we can see at one glance:

- One shift of type “A” is scheduled:
  - Start: 2
  - Length: 4 ... but also {3, 2, 1}
  - Workers: 3 ... but also {2, 1}

Represent Quantitative Values via Closed Intervals

- Similar Idea is used in [Crawford and Baker, 1994]
  Application of Satisfiability Algorithms to Scheduling Problems
  - “Coherence Criteria”



The Goal:

Make implicit information explicit for speeding up search!

Experiments

- Based on four datasets[1]:
  - DataSet1 (30 Instances, solvable without over- and understaffing)
    - Known optimum, small number of necessary shifts
  - DataSet2 (30 Instances, solvable without over- and understaffing)
    - Known optimum, higher number of necessary shifts
  - DataSet3 (30 Instances requiring over- and/or understaffing)
    - Unknown optimum
  - DataSet4 (3 Instances derived from a real-world setting)
    - Unknown optimum

[1] Introduced in [Musliu, 2001, Musliu et al., 2004].
Available at `http://www.dbai.tuwien.ac.at/proj/Rota/DataSetASP.zip`

Experiments

- Based on four datasets[1]:
  - DataSet1 (30 Instances, solvable without over- and understaffing)
    - Known optimum, small number of necessary shifts
    - **Optimum found within one hour in all but two cases.**
  - DataSet2 (30 Instances, solvable without over- and understaffing)
    - Known optimum, higher number of necessary shifts
    - **Optimum found within one hour in all but five cases.**
  - DataSet3 (30 Instances requiring over- and/or understaffing)
    - Unknown optimum
  - DataSet4 (3 Instances derived from a real-world setting)
    - Unknown optimum

[1] Introduced in [Musliu, 2001, Musliu et al., 2004].
Available at `http://www.dbai.tuwien.ac.at/proj/Rota/DataSetASP.zip`

Experiments

- Based on four datasets[1]:
  - DataSet1 (30 Instances, solvable without over- and understaffing)
    - Known optimum, small number of necessary shifts
    - **Optimum found within one hour in all but two cases.**
  - DataSet2 (30 Instances, solvable without over- and understaffing)
    - Known optimum, higher number of necessary shifts
    - **Optimum found within one hour in all but five cases.**
  - DataSet3 (30 Instances requiring over- and/or understaffing)
    - Unknown optimum
    - **Solutions found for 22 instances within one hour.**
  - DataSet4 (3 Instances derived from a real-world setting)
    - Unknown optimum
    - **Solutions found for all instances within one hour.**

[1] Introduced in [Musliu, 2001, Musliu et al., 2004].
Available at `http://www.dbai.tuwien.ac.at/proj/Rota/DataSetASP.zip`

Experiments

- Based on four datasets[1]:
  - DataSet1 (30 Instances, solvable without over- and understaffing)
    - Known optimum, small number of necessary shifts
    - **Optimum found within one hour in all but two cases.**
  - DataSet2 (30 Instances, solvable without over- and understaffing)
    - Known optimum, higher number of necessary shifts
    - **Optimum found within one hour in all but five cases.**
  - DataSet3 (30 Instances requiring over- and/or understaffing)
    - Unknown optimum
    - **Solutions found for 22 instances within one hour.**
    - **Optimum found within one hour in four cases.**
  - DataSet4 (3 Instances derived from a real-world setting)
    - Unknown optimum
    - **Solutions found for all instances within one hour.**

[1] Introduced in [Musliu, 2001, Musliu et al., 2004].
Available at `http://www.dbai.tuwien.ac.at/proj/Rota/DataSetASP.zip`

Conclusion and Future Work

- Our ASP encoding for the Shift Design Problem ...
  - is non-trivial.
  - is effective.
  - is efficient.

Conclusion and Future Work

- Our ASP encoding for the Shift Design Problem ...
  - is non-trivial.
  - is effective.
  - is efficient.

In fact, we showed that using our proposed approach we are able to provide global optima for four hard instances, not previously solved to the optimum.

Conclusion and Future Work

- Our ASP encoding for the Shift Design Problem ...
  - is non-trivial.
  - is effective.
  - is efficient.

In fact, we showed that using our proposed approach we are able to provide global optima for four hard instances, not previously solved to the optimum.

- Future Work:
  - Exploit Clingo's integrated features, i.e., use its domain heuristics.
  - Combine ASP with meta-heuristics or min-cost max-flow techniques.
  - As our presented results open up the area of workforce scheduling for ASP, investigate related problems.

Acknowledgments

This work was funded by

- AoF (251170), DFG (550/9), and
- FWF (P25607-N23, P24814-N23, Y698-N23).

References I

Crawford, J. and Baker, A. (1994).
Experimental results on the application of satisfiability algorithms to scheduling problems.
In *Proc. of AAAI'94*, pages 1092–1097. AAAI Press.

Grasso, G., Iiritano, S., Leone, N., Lio, V., Ricca, F., and Scalise, F. (2010).
An ASP-Based System for Team-Building in the Gioia-Tauro Seaport.
In *Practical Aspects of Declarative Languages*, volume 5937 of *Lecture Notes in Computer Science*, pages 40–42. Springer.

Musliu, N. (2001).
Intelligent Search Methods for Workforce Scheduling: New Ideas and Practical Applications.
PhD thesis, Technische Universität Wien.

Musliu, N., Schaerf, A., and Slany, W. (2004).
Local search for shift design.
European Journal of Operational Research, 153(1):51–64.