

# Integer Programming: Using Branch and Bound to Solve the Nurse Scheduling Problem

Geetha Baskaran<sup>1</sup>, Andrzej Bargiela<sup>2,3</sup>, and Rong Qu<sup>2</sup>

<sup>1</sup>School of Computer Science, University of Nottingham Malaysia Campus, Semenyih, Malaysia

<sup>2</sup>School of Computer Science, University of Nottingham, Jubilee campus, Nottingham, NG8 1BB, UK

<sup>3</sup>School of Computer Science, Krakow Technical University, Krakow, Poland

**Abstract:** *The nurse scheduling problem (NSP) is a complex optimisation problem regarding the allocation of nurses to duty rosters in hospitals. The objective is to ensure that there are sufficient nurses on duty while considering individual preferences with respect to work patterns, requests for leave and financial restrictions, in such a way that all employees are treated equally. In this paper, we extend our novel approach to solving the NSP by transforming it through information granulation. The approach is general enough to be applied within a wide range of benchmark instances and the majority of these instances have real world applications. They have been collected from a variety of sources including industrial collaborators, other researchers and previous publications. Domain transformation is an approach to solving complex problems that relies on a simplification of the original problem. First, the solution to a problem and the refined solution favouring simplification are introduced. The approach we use involves information granulation of shift types to transform the problem into a smaller solution domain. Next, schedules derived from smaller problem domains are converted into the original problem domain. The conversion takes care of the constraints which were not represented in the smaller domain. The problem is then solved via integer programming (IP). IP is formulated to solve the transformed scheduling problem using the branch and bound (IP-BB) algorithm. We have used the GNU Octave, open source mathematical modelling and simulation software for Windows to solve this problem. We tested the incorporation of (IP-BB) within our proposed methodology to solve the NSP. The (IP-BB) outperforms in most cases other approaches when tested with the different demands and number of nurses. The results facilitated the development of a cost-benefit analysis across different levels of staffing.*

**Keywords:** *domain transformation, nurse scheduling, integer programming, branch and bound*

## 1. Introduction

The nurse scheduling problem (NSP) is a topic that has been discussed broadly over the years. This is due to the sophisticated and challenging real world circumstances that nurse management systems are located within, leading to ongoing nurse shortages. Scheduling nurses to meet the daily demand of hospitals and to satisfy staffing policies, such as those dictated by a contract or by regulations mandating specific nurse-to-patient ratios, is an extremely complex task to perform (Gino et al., 2012). Good work schedules must be generated to maximise recruitment and retention and it is imperative that nurses' work preferences are taken into account.

Nurse scheduling aims to create a more systematic approach to the assignment of nurses' shifts and to scheduling within fixed time periods. Nurse scheduling also aims to ensure that high quality services are consistently provided (Randhawa & Sitompul, 1990). As such, if a nurse can only handle a task for half of its required time, the scheduling must be done in such a way that the other remaining half of the time is covered by another available nurse. In a sense, this is therefore an assignment problem stacked with a scheduling problem. Further, if a nurse has already been assigned to one task, then that nurse becomes unavailable to be assigned to other tasks. This becomes another challenge in nurse scheduling because the algorithm must now consider the best choice of nurses to be assigned to each task. Systematic nurse scheduling requires a great deal of effort and

is time-consuming. Cost-efficient schedules are very important because nursing salaries typically constitute a significant portion of hospitals' budgets and because schedules may influence the quality of healthcare that hospitals are able to provide to patients (Alward & Monk 1994; Warner et al. 1991).

Early solutions to the NSP included a method known as self-scheduling (Hung, 2002). This involves a process whereby nurses created their own schedules, which led to collective schedule suggestions being compromised in the process of discussion or by managers. These processes were tedious and resulted in disagreements (Ronnberg & Larsson, 2010). The extensive literature on nurse scheduling (Ernst et al., 2004; Yi, 2005) identifies 28 different categories of methods which have been used in personnel scheduling problems. These methods include optimisation approaches (i.e. mathematical programming), constraint logic programming, constructive heuristic, expert systems, genetic algorithms, simple local search, simulated annealing, tabu search, knowledge based systems, artificial neural networks and hybrid systems.

In the NSP, there are two types of constraints: hard constraints and soft constraints. Hard constraints must be encountered at all times because they may render the schedule unworkable. Soft constraints operate to estimate the quality of the solution. In this sense, soft constraints are not necessary, but should be fulfilled as often as possible. Nevertheless, to achieve a schedule that addresses all of the hard constraints, breaking some of the soft rules is necessary. A weight is allocated for each soft constraint which reflects its worth. The objective of nurse scheduling is to find a schedule that satisfies all of the hard constraints and minimises the degree to which the soft constraints are violated.

In this study, we present an alternative method for tackling a large, real world NSP through using integer programming (IP) across a large collection of diverse scheduling benchmark instances. In this approach, the hospital is supplied with detailed information about the schedule which they can use to make an objective selection. We used the domain transformation method to approach the problem of cost effectiveness in scheduling nurses. This approach was introduced practically by Baskaran et al. (2013) to demonstrate the information granulation methodology (Bargiela et al., 2002, 2008) for the generation of multiple feasible low-cost rosters. These rosters were then evaluated.

## 2. Nurse Scheduling Problem Using Domain Transformation

The NSP is defined as a problem in the assignment of nurses to a specific shift within a pre-defined scheduling horizon. It is subject to hard constraints originating from contractual agreements, legal requirements and local good practice. Any schedule that satisfies these constraints is referred to as a feasible schedule. Satisfying hard constraints is only a starting point for the construction of a good quality schedule. The degree of satisfaction of additional constraints reflecting staff preferences for allocation of specific shifts also provides a measure of the quality of the schedule.

We produce an efficient scheduling system by generating pattern sequences (Baskaran, et al. 2009). The scheduling problem is then changed to a problem of scheduling patterns. The number of patterns that need to be considered in domain transformation determines the computational gain.

Our domain transformation approach can be summarised as a three-stage process:

1. conversion of the problem from the original edINR domain into a problem in the *DNR* domain
2. solution of the problem in the *DNR* domain
3. conversion of the *DNR* solution into a solution in the original edINR domain.

We achieve performance gains in the patterns of the N-shifts and reduce the number of nurses that need to be allocated into D-shifts.

The approach relies on a well-justified simplification of the original problem. The problem is divided systematically into smaller sub-problems that can then be reproduced. This approach avoids random search. Other methods have failed to reproduce results, perform consistently and work on selected data sets while failing in others. The previous state-of-the-art approach did not use information granulation (Domain Transformation Approach [DTA]) and was confronted with data-checking and cross-referencing issues. We have investigated the optimum balance between ward staffing levels and achieving good quality schedules.

## 2.1. Benchmark Instances

To validate our algorithms and encourage more competition and collaboration between researchers addressing scheduling, we have built a collection of diverse and challenging benchmark instances. The collection has grown over several years, have been sourced from thirteen different countries and the majority of the data sets are based on real world scheduling scenarios. Table 1 lists these instances. The instances vary in the length of the planning horizon, the number of employees, the number of shift types and the number of skills required. Each instance also varies in the number, priority and type of constraints as well as the objectives present. The objectives were set by the organisation that provided the data. For example, some prefer to minimise overstaffing whereas others prefer to maximise staff satisfaction by setting a higher importance weighting for those objectives instead.

TABLE I: Benchmark Instances

Instance	Staff	Shift types	Length (days)	Skill types	Best known	Ref
Musa	11	1	14	3	175	[5]
GPost	8	2	28	1	5	
GPost-B	8	2	28	1	3	
Ozkarahan	14	2	7	2	0	[16]
Millar-2Shift-Data1	8	2	14	1	0	[4]
Millar-2Shift-Data1.1	8	2	14	1	0	[4]
Azaiez	13	2	28	2	0	[19]
WHPP	30	3	14	1	5	[14]
Valouxis-1	16	3	28	1	20	[6]
Ikegami-2Shift-Data1	28	2	30	9	0	[4]
Ikegami-3Shift-Data1	25	3	30	8	2	[4]
Ikegami-3Shift-Data1.1	25	3	30	8	3	[4]
Ikegami-3Shift-Data1.2	25	3	30	8	3	[4]
ORTEC01	16	4	31	1	270	[8]
ORTEC02	16	4	31	1	270	[8]
QMC-1	19	8	28	1	13	
QMC-2	19	3	28	3	29	
SINTEF	24	5	21	1	0	

The instances are available for download from <http://www.cs.nott.ac.uk/~tec/NRP/>.

## 3. Domain Transformation with Integer Programming

In this paper, we use IP which is an extension of linear programming. IP seeks to solve problems requiring integer solutions. We have implemented branch and bound (IP-BB). To specify the problem, the objective is to minimise the value of individual variables. Referring to Baskaran et al. (2014) on the pattern constructions, Figure 1 provides some examples of zero-cost patterns and Figure 2 provides examples of non-zero-cost patterns.

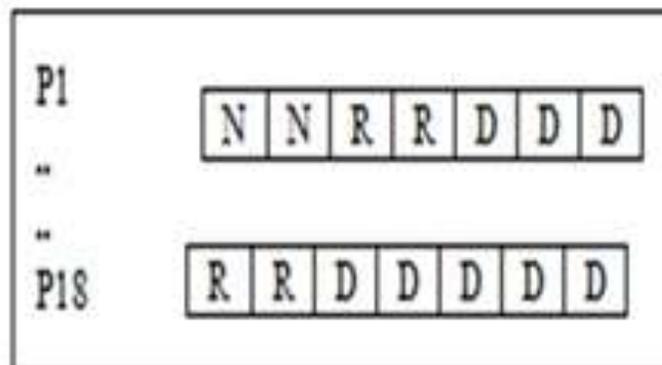


Fig. 1: No violation of soft constraints (called 'zero-cost patterns')

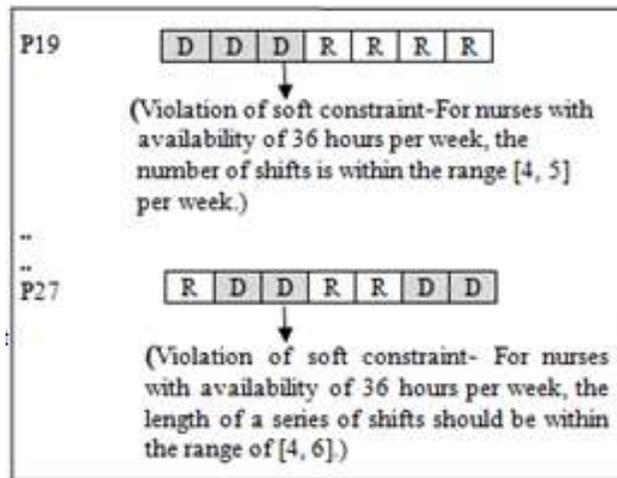


Fig. 2: Violation of soft constraints with cost 10 (called ‘non-zero-cost patterns’)

We formulate the problem of a two-week scheduling period with the following IP model. This model can be altered to adapt to any other problem with different constraints. The above patterns have three states: D, N and R. Therefore, if we want to use binary representation of patterns we need to separate the day and night components of the patterns, as shown in Figure 3. This will allow for the representation of three states.

Day				Night			
1	1	0	0	0	0	0	0
0	0	0	0	1	1	1	0
0	0	0	0	1	1	1	1
1	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0
0	0	0	1	0	0	0	0
1	0	0	1	0	0	0	0
0	0	1	1	0	0	0	0
0	0	1	1	0	0	0	0
1	1	1	1	0	0	0	0

Fig. 3: Binary pattern matrix

This binary pattern matrix will be given the title ‘B’. This matrix is replicated for each nurse and the combined pattern matrix, ‘C’, is represented in Figure 4.

C = [

1	1	0	0	0	0	0	0	0	0	1	1	0	0	0
0	0	0	0	0	1	1	1	1	0	0	0	0	0	0
0	0	0	0	1	1	1	1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	1	1	1	1
1	1	0	0	0	0	0	0	0	0	0	1	1	1	1
0	0	0	1	0	0	0	0	0	0	0	1	1	1	1
1	0	0	1	0	0	0	0	0	0	0	1	1	1	1
0	0	1	1	0	0	0	0	0	0	0	1	1	1	1
1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
....														
....														
1	1	0	0	0	0	0	0	0	0	1	1	0	0	0
0	0	0	0	0	1	1	1	1	0	0	0	0	0	0
0	0	0	0	1	1	1	1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	1	1	1	1
1	1	0	0	0	0	0	0	0	0	0	1	1	1	1
0	0	0	1	0	0	0	0	0	0	0	1	1	1	1
1	0	0	1	0	0	0	0	0	0	0	1	1	1	1
0	0	1	1	0	0	0	0	0	0	0	1	1	1	1
1	1	1	1	0	0	0	0	0	0	0	0	0	0	0

];

Fig. 4: Combined pattern matrix

The selection of patterns from C represents the schedule that satisfies the equality constraints such as the cover requirement. This can be expressed as:

$$C' * x = c' \tag{1}$$

where  $\mathbf{x}$  is the unknown binary vector, representing a solution to the scheduling problem and  $\mathbf{c}$  is the staff cover requirement. The requirement that each nurse is assigned to one pattern at most represents a constraint that can be written as:

$$\mathbf{A}' * \mathbf{x} \leq \mathbf{b}' \tag{2}$$

where  $\mathbf{A}$  is a matrix with the number of columns corresponding to the number of nurses and the number of rows equal to the product of the number of nurses( $n$ ) and the number of patterns ( $p$ ), represented as :

$$\mathbf{m} = \mathbf{n} * \mathbf{p} \tag{3}$$

Figure 5 represents matrix  $\mathbf{A}$  for  $n=15$  and  $p=18$ .

row1	1	0	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0
row2	1	0	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0
	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
row18	1	0	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0
row19	0	1	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0
row20	0	1	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0
	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
row36	0	1	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0
	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
row253	0	0	.	.	.	.	.	.	.	.	.	.	.	.	.	.	1
row254	0	0	.	.	.	.	.	.	.	.	.	.	.	.	.	.	1
	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
row270	0	0	.	.	.	.	.	.	.	.	.	.	.	.	.	.	1

Fig. 5: Example of rows = number of nurses (say 15) \* number of patterns (say 18)

The vector ' $\mathbf{b}$ ' is a vector of 1s corresponding to the number of nurses. Subsequent weeks will need to use different sets of patterns for each nurse. This will depend on what has been assigned in Week 1. The objective of the optimisation of the scheduling could be defined as trying to satisfy the cover requirement with the minimum number of nurses. This is expressed simply as:

$$\text{Min NP} * \mathbf{x} \tag{4}$$

where  $\mathbf{NP}$  is a vector of 1s of size ' $\mathbf{m}$ '. The cost function is defined as a sum of penalties representing a nurse working a given shift on a day. Therefore, our aim is to minimise the penalty subject to constraints (1) and (2).

### 3.1. Branch and Bound

B&B methods implicitly enumerate all possible solutions to an IP. The basic concept underlying the B&B technique is to 'divide and conquer'. Since the original 'large' problem is difficult to solve directly, it is divided into smaller sub-problems until these sub-problems can be 'conquered'. The dividing (branching) is done by partitioning the entire set of feasible solutions into smaller and smaller subsets. The conquering (fathoming) is carried out by providing a bound for the best solution in the subset or discarding the subset if the bound indicates that it does not contain an optimal solution. The steps used for each iteration were:

1. *Branching*: This was used among the unfathomed sub-problems ( $F_i$ ) and the one that was created most recently was selected.
2. *Fathoming*: If the sub-problems were not feasible and then they were discarded.
3. *Bounding*: The new sub-problems were solved and a lower bound  $b(F_i)$  for the sub-problem was computed.
4. *Fathoming*: For each new sub-problem, if  $b(F_i) \geq U$ , then the current best upper was bound and the fathomed sub-problem was discarded.
5. *Optimality test/partitioning*: If there are no unfathomed sub-problems left, then they were either obtained at an optimal solution to the sub-problem (stop), or the corresponding problem was broken into further sub-problems to perform another iteration.

## 4. Computational Result

Within our approach, the IP process was solved by using the latest GNU Octave's GLPK (4.45). The resting was carried out using the Intel Pentium Dual Core T4500 2.3 GHz x64 PC with 4 GB RAM and Windows 8 Pro

x64 was used. We have also used the database engine SQL Server Compact 3.5. The results obtained through solving the BBIP are presented in Table 2. The problem is a minimisation problem and the results in bold indicate optimal solutions. Solving within a practical time limit will be dependent on the performance of the IP solver.

TABLE II: Results for Branch and Bound on Benchmark Instances

Instance	Best known cost	BUR 14 [8] Cost time(s)	MET 09 Cost time(s)	BUR 09b (SS2) Cost time(s)	BUR 09b (MEH) Cost time(s)	Our approach Cost time(s)
Musa	<b>175</b>	175 <0.1	175 39			175 1
GPost	<b>5</b>	5 2	8 234	9 4305	915 605	5 20
GPost-B	<b>3</b>	3 29.3		5 3955	789 475	<b>2 15</b>
Ozkarahan	<b>0</b>	0 <0.1	0 1			0 1
Millar-2Shift-Data1	<b>0</b>	0 <0.1	0 1	0 910		0 43
Millar-2Shift-Data1.1	<b>0</b>	0 <0.1		0 20		0 30
Azaiez	<b>0</b>	0 0.3	0 233			0 30
WHPP	<b>5</b>	5 17.6				<b>0 17</b>
Valouxis-1	<b>20</b>	80 909.6	160 3780	100 4000		20 42
Ikegami-2Shift-Data1	<b>0</b>	0 41.7				0 40
Ikegami-3Shift-Data1	<b>2</b>	2 597.8	63 671			2 68
Ikegami-3Shift-Data1.1	<b>3</b>	4 995.2				3 88
Ikegami-3Shift-Data1.2	<b>3</b>	5 5411.9				3 95
ORTEC01	<b>270</b>	270 69.3		365 3400	535 7580	<b>120 135</b>
ORTEC02	<b>270</b>	270 105.1				<b>120 155</b>
QMC-1	<b>13</b>	13 57.6		20 4435	39 3160	13 62
QMC-2	<b>29</b>	29 1.9				29 3
SINTEF	<b>0</b>	0 10.5		4 4105		0 48

The B&B method was able to solve most of the instances to optimality, but the computation time varied from one second to 2.6 minutes in the case of the hardest instance. In comparison with the best known result, we have achieved a new result for GPost-B with cost, two in 15 seconds and WHPP zero-cost in 17 seconds. One result for large instances outperformed the other examples, ORTEC, which appears significantly better than the best results achieved in the existing literature. Our approach achieved 120 costs within 135 seconds for ORTEC01 and 155 seconds for ORTEC02. Overall, our results are equal to the best known cost. However, 61 per cent of the result has a slower computation score when compared to Burke et al.’s results (2014).

## 5. Conclusion

We have presented novel, information granulation based formulation of the nurse scheduling problem and have solved it using Integer Programming with Branch and Bound search. The results show that the B&B method can solve some instances very effectively. For other instances, the time and resource requirements may be restrictive. However, with more advanced branching schemes in the B&B tree and through the development of new ideas it may be possible to further improve the performance. The domain transformation approach uses a number of novel ideas which we believe are general enough to be adapted to other problem domains. All instances tested were modelled using a generic model.

## 6. References

- [1] A. Bargiela and W. Pedrycz, “Granular Computing –An Introduction”, Kluwer Academic Publishers. 2002. <http://dx.doi.org/10.1007/978-1-4615-1033-8>.
- [2] A. Bargiela, and W. Pedrycz, “Toward a theory of Granular Computing for human-centred information processing”, IEEE Trans. On Fuzzy Systems. 16(2): 320-330. <http://dx.doi.org/10.1109/TFUZZ.2007.905912>. 2008.
- [3] A.T. Ernst, H. Jiang, M. Krishnamoorthy and D. Sier, “Staff Scheduling and Rostering: A Review of Applications, Methods and Models”. *EJOR*. 153: 3-27. 2004 [http://dx.doi.org/10.1016/S0377-2217\(03\)00095-X](http://dx.doi.org/10.1016/S0377-2217(03)00095-X)

- [4] A. Ikegami and A. Niwa, A Subproblem-centric Model and Approach to the Nurse Scheduling Problem, *Mathematical Programming* 97(3) (2003) 517-541.  
<http://dx.doi.org/10.1007/s10107-003-0426-2>
- [5] A. Musa and U. Saxena, Scheduling nurses using goal-programming techniques, *IIE transactions* 16 (1984) 216-221.  
<http://dx.doi.org/10.1080/07408178408974687>
- [6] C. Valouxis and E. Housos, Hybrid optimization techniques for the workshift and rest assignment of nursing personnel, *Artificial Intelligence in Medicine* 20(2) (2000) 155-175.  
[http://dx.doi.org/10.1016/S0933-3657\(00\)00062-2](http://dx.doi.org/10.1016/S0933-3657(00)00062-2)
- [7] D. Sitompul and S. Randhawa, "Nurse Rostering Models: A State-of-the-Art Review". *Journal of the Society of Health Systems*. 2: 62-72. 1990
- [8] E.K. Burke, T. Curtois, G. Post, R. Qu, and B. Veltman, A Hybrid Heuristic Ordering and Variable Neighbourhood Search for the Nurse Rostering Problem, *European Journal of Operational Research* 188(2) (2008) 330-341.  
<http://dx.doi.org/10.1016/j.ejor.2007.04.030>
- [9] E.K. Burke., T. Curtois, R. Qu and G. Vanden Berge. A Scatter Search Methodology for the Nurse Rostering Problem. *Journal of the Operational Research Society*, Accepted for publication, 2009.
- [10] E.K Burke. and T. Curtois. New Approaches to Nurse Rostering Benchmark Instances. *European Journal of Operational Research*, Accepted for publication 2014.
- [11] G. Baskaran, A. Bargiela, R. Qu., "Hierarchical method for nurse rostering based on granular pre-processing of constraints", *Proc. 23rd European Conference on Modelling and Simulation, ECMS*, Madrid, Spain, pp.855-861, June 2009, (doi: 10.7148/2009-0855-0861).
- [12] G. Baskaran, , A. Bargiela, R. Qu, "A Study of Cost Effective Scheduling of Nurses Based on the Domain Transformation Method", *Proc. 27th European Conference on Modelling and Simulation, ECMS*, Alesund, Norway, pp.309-314, May 2013, (doi: 10.7148/2013-0309).
- [13] G. Weil, K. Heus, P. Francois, and M. Poujade, Constraint programming for nurse scheduling, *IEEE Engineering in Medicine and Biology Magazine* 14(4) (1995) 417-422.  
<http://dx.doi.org/10.1109/51.395324>
- [14] H. Li, A. Lim, and B. Rodrigues. A Hybrid AI Approach for Nurse Rostering Problem. in *Proceedings of the 2003 ACM symposium on Applied computing*. 2003. pp. 730-735.  
<http://dx.doi.org/10.1145/952532.952675>
- [15] I. Ozkarahan. The Zero-One Goal Programming Model of a Flexible Nurse Scheduling Support System, in *Proceedings of International Industrial Engineering Conference*. in *Proceedings of International Industrial Engineering Conference*. 1989. pp. 436-441.
- [16] J.L. Gino, A. Mobasher, , J.C. Murray, 2012. "Multi-objective Nurse Scheduling Models with Patient Workload and Nurse", *Preferences Management 2012*. 2(5): 149-160 -DOI: 10.5923/j.mm.20120205.03.  
<http://dx.doi.org/10.5923/j.mm.20120205.03>
- [17] J.-P. Métyvier, P. Boizumault, and S. Loudni, Solving Nurse Rostering Problems Using Soft Global Constraints, in *Principles and Practice of Constraint Programming - CP 2009*, I.P. Gent, Editor. 2009, Springer. pp. 73-87.  
[http://dx.doi.org/10.1007/978-3-642-04244-7\\_9](http://dx.doi.org/10.1007/978-3-642-04244-7_9)
- [18] M.N. Azaiez and S.S. Al Sharif, A 0-1 goal programming model for nurse scheduling, *Computers and Operations Research* 32(3) (2005) 491 - 507.  
[http://dx.doi.org/10.1016/S0305-0548\(03\)00249-1](http://dx.doi.org/10.1016/S0305-0548(03)00249-1)
- [19] M. Warner, "Scheduling Nursing Personnel According to Nurse Preference: a Mathematical Programming Approach." *OR*.24: 842-856. 1976.  
<http://dx.doi.org/10.1287/opre.24.5.842>
- [20] R. Alward & T. H. Monk, "Supporting shift workers", *JONA*, volume 24, number 5, pp. 53-59. 1994.  
<http://dx.doi.org/10.1097/00005110-199405000-00011>
- [21] R. Elina, L. Torbjörn, "Automating the self-scheduling process of nurses in Swedish healthcare: a pilot study", *Health Care Management Science*, 13(1): 35-53, 2010.  
<http://dx.doi.org/10.1007/s10729-009-9107-x>
- [22] R. Hung, (2002) A note on nurse self-scheduling. *Nursing Economics* 20 (1), 37-39.  
 Y. Yi, "Application of finite state machines in medical care scheduling", *Electrical and Computer Engineering*, Duke University, Durham. (2005).