A DECOMPOSITION, CONSTRUCTION AND POST-PROCESSING APPROACH FOR NURSE ROSTERING

Peter Brucker*, Rong Qu†, Edmund Burke†, Gerhard Post‡

**University of Osnabrück, Faculty of Mathematic/Informatics, Albrechtstr. 28, 49069 Osnabrück, Germany, peter@mathematik.uni-osnabrueck.de †School of Computer Science & Information Technology, University of Nottingham, Jubilee Campus, Nottingham NG8 1BB, UK, {ekb, rxq}@cs.nott.ac.uk ‡ ORTEC bv, Groningenweg 6-33, 2803 PV Gouda, The Netherlands*

Abstract: This paper presents our work on decomposing a specific nurse rostering problem by cyclically assigning blocks of shifts, which are designed considering both hard and soft constraints, to groups of nurses. The rest of the shifts are then assigned to the nurses to construct a schedule based on the one cyclically generated by blocks. The schedules obtained by decomposition and construction can be further improved by a variable neighborhood search. Significant results are obtained and compared with a genetic algorithm and a variable neighborhood search approach on a problem that was presented to us by our collaborator, ORTEC bv, The Netherlands. We believe that the approach has the potential to be further extended to solve a wider range of nurse rostering problems.

Key words: nurse rostering problems, decomposition, variable neighborhood search

1. INTRODUCTION

Nurse rostering is concerned with highly constrained resource allocation problems. In hospitals and other medical institutes, the workload needs to be assigned to nurses periodically taking into account a number of constraints and requirements. *Hard* constraints are those that must be satisfied in order to have a *feasible* schedule due to physical resource restrictions and legislation. When requirements are desirable but not obligatory they are often referred to as *soft* constraints. Soft constraints are often used to evaluate the quality of feasible schedules.

Nurse rostering problems have a large number of variations on legal regulations and individual preferences, depending on countries and institutions. Typical issues concerned with nurse rostering include coverage demand, day-off requirements, weekend-off requirements, minimum and maximum workforce, etc.

A wide variety of different methodologies and models have been developed to deal with different problem circumstances during the years (see [13, 15, 19, 29]). These include mathematical programming (e.g. [4, 6, 26, 30, 31]), meta-heuristic methods (e.g. [2, 3, 9, 12, 18]) and constraint satisfaction techniques (e.g. [1, 8, 24, 25]). Meta-heuristic approaches which are particular appropriate for this paper because they employ variable neighborhood search for nurse rostering are [10, 11]. Column generation was employed in nurse scheduling (e.g. [5, 22]) using set covering-type models and integer programming. Knowledge based techniques have also been explored for solving nurse rostering problems (e.g. [7, 16, 23, 28]).

Over the years decomposition techniques have been applied successfully over a range of timetabling problems (e.g. [14, 21]). In [2] the authors developed a successful genetic algorithm where indirect representation was used. Different heuristic decoders were employed to construct the schedule, taking care of coverage and nurses' preferences in different stages. In [21] the constraints in the nurse scheduling problems in Japan were grouped into shift constraints and nurse constraints, based on which the problems were decomposed into sub-problems and which were solved repeatedly. In some of the staff scheduling problems, only predefined sequences of shifts, often referred as "stints" [19], are allowed to be used. Cyclical patterns were also studied in nurse rostering problems where each nurse receives cyclic scheduling each week for *n* weeks [13, 19, 20]. Pre and post processing methodologies for nurse rostering were presented and discussed in [17].

Our approach cyclically constructs partial solutions for groups of nurses based on pre-defined sequences of shifts in a specific nurse rostering problem taken from our commercial collaborator, ORTEC bv. The rest of the shifts are then assigned and the schedule is further improved by a post-processing procedure. It combines both the cyclic and non-cyclic schedules and aims at building near optimal sub-solutions while still being able to take care of individual requirements.

The problem being tested in this work has been addressed by the commercialized software HARMONY™[1] [27], which has been developed by ORTEC, The Netherlands and is in use in hospitals, and other organizations in different countries. In this paper we propose a decomposition, construction and post-processing approach to solve this specific problem. The interface of HARMONY™ is used as a tool to visualize the schedules, to indicate infeasibilities, and to calculate penalties for violations of soft constraints.

2. THE NURSE ROSTERING PROBLEM

The nurse rostering problem we consider here is based on the situation encountered by ORTEC in intensive care units in Dutch hospitals. We outline the following characteristics.

1. We have to adhere to Dutch national legislation, and the collective labor agreements in force in Dutch hospitals. They are translated to a shorter period, to be meaningful for a situation without (much) history.
2. The requests of the personnel are very important, and should be met as much as possible; the soft constraints we use are those that, in our experience, represent the situation in Dutch hospitals.
3. It is not necessary to consider qualifications, as all personnel are highly qualified. Nurses still in training are disregarded in the original planning (and are added by hand afterwards).
4. The size of the department is rather small, but convenient for testing purposes.

The problem consists of assigning a certain number of different types of shifts (which are described in Section 2.1) within a scheduling period of 4 weeks to 16 nurses in a ward. Twelve of the nurses are full-time and have a contract of 36 hours per week. One part-time nurse works 32 hours per week and the other three part-time nurses work maximally 20 hours per week. The hard and soft constraints that need to be satisfied are described below in Sections 2.2 and 2.3, respectively. This problem is also presented in [10].

2.1 Shift Types and Demand

There are 4 different shift types in the problem. Table 1 gives the definitions and the daily coverage requirements for these shift types. Each of the shift types cover 9 hours including one hour of resting time, except the night shift, which contains no rest. So there are 8 actual working hours for each of these shift types.

2.2 Hard Constraints

The hard constraints listed below must be met in any circumstances otherwise the schedule is considered to be infeasible and unacceptable.

1. Demand needs to be fulfilled[2] (i.e. all the requested shifts in Table 1 must be covered).
2. For each day, 1 nurse may start only one shift.
3. Within a scheduling period, a nurse is allowed to exceed the number of hours for which he/she is available for his/her department by at most 4 hours.
4. The maximum labor time per week is on average 36 hours over a period of 13 consecutive weeks if this period does not include work during night shifts.
5. The maximum number of night shifts is 3 per period of 5 consecutive weeks.
6. A nurse must receive at least 2 weekends off duty per 5 week period. A weekend off duty lasts 60 hours including Saturday 00:00 to Monday 04:00.
7. Following a series of at least 2 consecutive night shifts, a 42 hours rest is required.

[1] http://www.ortec.com/us/area.php?id=ORTEC_harmony

[2] It would be possible to consider this as the main objective in the problem rather than as a hard constraint. However, we treat it as one of the hard constraints, as do many other papers in the nurse scheduling.

8. During any period of 24 consecutive hours, at least 11 hours of rest is required. A night shift has to be followed by at least 14 hours rest. An exception is that once in a period of 21 days for 24 consecutive hours, the resting time may be reduced to 8 hours.

9 The number of consecutive night shifts is at most 3.

10 The number of consecutive shifts (workdays) is at most 6.

11 One of the full-time nurses requires not receiving any late shifts.

2.3 Soft Constraints

Requirements, or soft constraints, and their weights in the problem we are dealing with are listed in Table 2. Ideally these constraints should be satisfied as much as possible. However, in real world circumstances, it is usually necessary to violate some of these soft constraints. Depending on how strongly these soft constraints are desired (especially in comparison to other soft constraints), a weight is assigned to each of them. The higher the weight, the more strongly desired the constraint is. For example, soft constraints 9 and 10 in Table 2 have very low weights as they are not so important and may be violated at low cost. The penalty of a feasible schedule is the sum of the weights of all the violations of soft constraints in the schedule.

3. A SOLUTION APPROACH

Our decomposition approach has two stages of solution construction and one stage of solution improvement. The first two stages are currently manually carried out and the 3rd stage is an automatic process implemented in [10]. They are described as below:

1. Firstly, sequences of shifts (we call them *blocks*) are designed taking into account both the soft and hard constraints. These blocks are then used to construct cyclic schedules for groups of nurses (we call the schedule for a group a *cyclic pattern*).
2. After the assignment by cyclic patterns, the rest of the shifts are assigned to nurses who are not involved in the cyclic patterns. In connection with this, some shifts assigned in the cyclic patterns may need to be re-assigned to accommodate the remaining shifts.
3. A variable neighborhood search approach developed in [10] is then applied to further improve the schedule obtained from the first two stages. It reassigns the shifts to improve the quality of the schedule.

3.1 Cyclic Pattern Design

In the first stage, cyclic patterns are constructed. Each cyclic pattern defines a schedule for 5 nurses over 5 weeks. Here a duration of 5 weeks is considered to be the length of the scheduling period due to hard constraints 5 and 6 (presented in Section 2.2). This duration can be adjusted to any other number of x weeks and the number of nurses can also be adjusted. The building elements of a cyclic pattern are called blocks (see below).

3.1.1 Construction of Blocks

When designing blocks, we first consider shifts which are the most important or most difficult to be scheduled. In this problem, night shifts are the most important ones with a cost of 1000 if the length is not within the range of [2, 3] (soft constraint 3). Also shifts at weekends are important as a non-complete weekend will cause a cost of 1000 (soft constraint 1).

We therefore consider the night shifts first in designing the blocks. This will result in three basic shift patterns (presented in the lefthand-side of the table in Table 3). They are the only three valid assignments for night shifts satisfying soft constraints 1 and 3. An example of invalid basic patterns is shown in the righthand-side of the table in Table 3, from which we can see that in *Pattern e* a night shift on Friday ends at 7:00 am (as defined in Table 1) on Saturday, making the weekend not complete. This leads to the fact that the other two patterns *Pattern d* and *f* cannot be considered in designing the blocks as they have to be used in conjunction with *Pattern e*.

It can be observed from soft constraints 3 and 5-9 in Table 2 that good schedules will consist of blocks within certain length ranges. When designing the blocks based on the basic patterns presented in Figure 1, both the hard and soft constraints listed in Sections 2.2 and 2.3 are considered with. The aim is to avoid violations of hard constraints, and to minimize the costs of violations of soft constraints with high penalties (soft constraints 1-3 in Table 2) associated with each block.

Sequencing penalties within blocks are formulated and presented in Table 4, and are checked when designing the blocks. In the table the values are the penalties of the violations (weights of soft constraints). Here “n/f” indicates a non-feasible schedule if the shift type on the corresponding row is followed by the shift type on the corresponding column. For example, a night shift followed by an early shift (“n/f” at the 2nd column and the 1st row in Table 4) indicates a non-feasible schedule (hard constraint 8), and a day shift followed by an early shift (soft constraint 9a) causes a cost of 5 (at the 2nd column and the 3rd row in Table 4) in the schedule.

3.1.2 Building of Cyclic Patterns

Based on the blocks built by the above process, a set of cyclic patterns can be designed concerning both the hard and soft constraints. For example, the total number of hours during 5 weeks needs to be within the maximum working hours allowed, and the resting time after blocks needs to be not shorter than required, etc. The cyclic patterns with the least penalties (below a certain threshold) are used to construct the schedules for groups of 5 nurses in the first stage of the approach.

Besides trying to maximize the number of shifts within blocks (to leave as few as possible shifts to be assigned later), the balance of the coverage of D, E and L shift types are also concerned. They should be equally distributed so that the rest of shifts can also be distributed evenly within the schedule in the 2nd stage of approach. The total numbers of D, E and L shifts on the corresponding weekdays within a week in the upper table in Table 5 are presented in the lower table in Table 5. The block covers one E shift type each day except Wednesday, Saturday and Sunday, one D shift type almost equally each day except Friday, and one L shift type each day except Wednesday and Thursday, shown in the lower table in Table 5. The maximum number of N shift types within 5 weeks needs to be 3 (see hard constraint 5), thus there are only 3 N shifts in this cyclic pattern, as also shown in the lower table in Table 5.

The upper table in Table 5 presents an example of a schedule constructed by a block based on the basic pattern, *Pattern c*, presented in Table 3. The schedule of a group of 5 nurses for the 1st week is shown. For the 2nd week, for the same group of nurses, a cyclic shift is applied to their first week schedule (i.e. the schedule of nurse i is replaced by the schedule of nurse $i + 1$ for $i = 1, 2, 3, 4$ and the schedule of nurse 5 is replaced by the schedule of nurse 1). These cyclic exchanges are repeated four times.

Figure 1 presents a HARMONY^{TM} screenshot of a schedule with two cyclic patterns in a nurse-day view for the problem we described in Section 2. As the number of full-time nurses in this problem is 12, two cyclic patterns are designed for 10 nurses (as shown in the figure). There are only three part-time nurses who work maximally 20 hours a week, and one part-time nurse who work 32 hours a week, so no cyclic patterns are designed for part-time nurses in this problem.

We can see that for nurse “Banar” (the 2nd nurse in the schedule in Figure 3), in the last week’s schedule there is one L (on Friday 28th February) less than the other 4 nurses with the same cyclic pattern. The same situation also applies for nurse “Navel” (the last nurse in the schedule in Figure 3), whose schedule in the last week (27th and 28th Thursday and Friday) is two L less than the others with the same cyclic pattern. The reason is that in HARMONY^{TM}, the average maximum working hours for each nurse is calculated within a calendar month instead of 5 weeks in the problem presented. This adds an extra restriction to the problem so three shifts have to be removed from the schedule.

3.2 Construction of a Schedule

The shifts not covered by the cyclic patterns are then assigned into the schedule taking into account both the hard and soft constraints in the second stage of the approach. This may involve re-assigning some of the shifts already in the schedule.

Figure 2 presents a complete schedule by using the first two stages of (decomposition and construction) the overall approach proposed. The last column in the schedule gives the penalties for each of the nurses on the corresponding rows. The quality of the schedule is evaluated by summing up all the penalties on the last column, calculating the penalty of the overall schedule in Figure 2 for the problem to be 340.

At this stage we are only investigating the construction of the cyclic patterns and the construction of the schedules manually to have a good understanding of the potential and the limitations of the approach proposed.

3.3 Improving the Complete Schedule

The last stage of the approach employs a variable neighborhood search to further improve the schedules constructed by the first two stages (decomposition and construction). Please note that after the improvements made in the 3rd stage, the part of the schedule generated by the cyclic patterns may not be cyclic any more. The aim of the 1st stage of the approach is to build good sub-solutions when decomposing the problem and doing this with much less effort.

4. COMPARISONS AND DISCUSSION

Due to the large number of variants and different techniques investigated for nurse rostering problems, it is difficult to carry out fair and meaningful comparisons upon the computational effort taken and the solution quality obtained. However, it is possible to compare against the methods that are used to solve the same problem as that presented in this paper.

In the current version of HARMONY^{TM}, a hybrid genetic algorithm is employed. A local search is carried out after each generation of the genetic algorithm to further improve the individual schedules to their local optima before the next generation. The moves taken in the local search include moving a shift assigned from one nurse to another, and swapping two single shifts assigned to two nurses.

A hybrid variable neighborhood search (VNS) has also been developed and applied to the same problem [10], where the same types of moves as in the above genetic algorithm are used within a descent local search on single schedules rather than a population of schedules at a time. When no further improvement can be made by using both moves, parts of the schedule which cause high penalties are re-assigned and the descent local search is started again. See [10] for more details.

Using the decomposition, construction and post-processing approach, we obtained a number of different schedules on the problem presented in Section 2. The best results out of 5 runs on each of the approaches, namely the hybrid genetic algorithm, the variable neighborhood search and our approach with and without the variable neighborhood search approach as the 3rd stage of post-processing, are presented in Table 6. The values in parentheses give the computational time of the corresponding approaches.

From the table we can observe that for the problem presented in this paper, the schedules generated by the hybrid GA have lower penalty than that of VNS but requiring much more computational time (6 hours as opposed to 1 minute). The decomposition, construction and post-processing approach obtained better results both with and without VNS. It is difficult to compare the computational time among the approaches as currently the design of blocks and cyclic patterns are made manually. The post-processing by variable neighborhood search usually stops within 1 minute.

It can be observed here that the decomposition and construction approach can serve as a good initialization method for searching algorithms such as VNS. The reason may be that pre-designed blocks and cyclic patterns can provide a good backbone in the schedule, based on which searching algorithms can start from promising areas in the search space, which is usually very huge and complex for nurse rostering problems. This observation leads us to hypothesize that this approach may produce good results on other problems and with other search algorithms.

5. CONCLUDING REMARKS

A nurse rostering problem presented by ORTEC has been solved by our decomposition, construction and post-processing approach. Blocks of shifts are designed to cyclically assign sequences of shifts, rather than single shifts, to groups of nurses. This will generate good partial solutions within the schedule, based on which the rest of the shifts are assigned afterwards. Both stages consider the hard and soft constraints and aim at minimizing the penalty of the schedule constructed. Extra constraints such as personal requests can be taken care of by the 2nd and 3rd stages of the approach, which are flexible enough to take care of other possible constraints or personal requests.

Our decomposition and construction approach is able to construct near optimal sub-solutions for sub-problems, which are obviously less complex than the overall problem. It combines the advantages of cyclical patterns and the non-cyclic schedules, which greatly reduce the computational time and take care of individual constraints/requests.

In our future work, the approach can be extended by employing heuristic approaches on automatically designing cyclic patterns, assigning the rest of the shifts, and improving the schedules generated. Based on the observations and experience collected in this work, the combinations between the patterns (not just in cyclic but rather at a general manner) will be further investigated. Column generation techniques will also be investigated in our future research.

Although our approach is tested on a specific problem, the idea of decomposition can be extended to solve a wider range of nurse rostering problems. Our future work will also include testing this approach on more problem instances and a wider range of problems to carry out more comparisons among different nurse rostering techniques.

ACKNOWLEDGEMENTS

The work is supported by the UK Engineering & Physical Sciences Research Council (EPSRC) (grant number GR/T23374/01). Professor Brucker's contribution is supported by a EPSRC visiting fellowship (grant number GR/T23374/01). We would like to thank the anonymous referees' for their valuable comments. We would also like to acknowledge the valuable support from ORTEC bv, The Netherlands.

REFERENCES

1. Adbennadher S. and Schlenker H.: Nurse Scheduling using Constraint Logic Programming, Eleventh Annual Conference on Innovative Applications of Artificial Intelligence, IAAI-99 (1999), 838-843.
2. Aickelin U. and Dowsland K.: An Indirect Genetic Algorithm for a Nurse Scheduling Problem. JORS, (2003) 31(5): 761-778.
3. Aickelin U. and White P.: Building Better Nurse Scheduling Algorithms. Annals of Operations Research, (2004), 128:159-177.
4. Bailey J. and Field J.: Personnel Scheduling with Flexshift Models. Journal of Operations Management, (1985) 5(3): 327-338.
5. Bard J. and Purnomo H.W.: Preference Scheduling for Nurses using Column Generation, EJOR, (2005), 164: 510-534.
6. Beaumont N.: Scheduling Staff using Mixed Integer Programming. EJOR, (1997) 98: 473-484.
7. Beddoe G. and Petrovic S.: Determining Feature Weights Using a Genetic Algorithm in a Case-Based Reasoning Approach to Personnel Rostering. Accepted for publication in EJOR, (2005).
8. Berrada I., Ferland J.A. and Michelon P.: A Multi-objective Approach to Nurse Scheduling with both Hard and Soft Constraints. Socio-Economic Planning Science, (1996), 30(3): 183-193.

9. Burke E., Cowling P., De Causmaecker P. and Vanden Berghe G.: A Memetic Approach to the Nurse Rostering Problem. Applied Intelligence, (2001) 15: 119-214.

10. Burke E., Curtois T., Post G., Qu R. and Veltman B. A Hybrid Heuristic Ordering and Variable Neighbourhood Search for the Nurse Rostering Problem. Technical Report NOTTCS-TR-2005-3, School of CSiT, University of Nottingham 2005.

11. Burke E., De Causmaecker P., Petrovic S. and Vanden Berghe G.: Variable Neighborhood Search for Nurse Rostering Problems, Ch. 7 in Metaheuristics: Computer Decision-Making (Combinatorial Optimization Book Series), (eds. Resende M.G.C. and de Sousa J.P.), Kluwer, (2003), 153-172.

12. Burke E., De Causmaecker P. and Vanden Berghe G.: A Hybrid Tabu Search Algorithm for the Nurse Rostering Problem. Selected Papers from the 2nd Asia Pacific Conference on Simulated Evolution and Learning (1998), LNAI 1585, 187-194.

13. Burke E., De Causmaecker P., Vanden Berghe G. and Van Landeghem H.: The State of the Art of Nurse Rostering. Journal of Scheduling, (2004) 7(6): 441-499.

14. Burke E. and Newall J.: A Multi-Stage Evolutionary Algorithm for the Timetable Problem, IEEE Transactions on Evolutionary Computation, (1999) 3(1): 63-74.

15. Cheang B., Li H., Lim A. and Rodrigues B.: Nurse Rostering Problems - a Bibliographic Survey. EJOR, (2003) 151: 447-460.

16. Chen J. and Yeung T.: Hybrid Expert System Approach to Nurse Rostering. Computers in Nursing, (1993) 11: 183-190.

17. De Causmaecker P. and Vanden Berghe G.: Relaxation of Coverage Constraints in Hospital Personnel Rostering. In: Proceedings of the 4th International Conference on the Practice and Automated Timetabling (2002), 187-206.

18. Dowsland K.: Nurse Scheduling with Tabu Search and Strategic Oscillation. EJOR, (1998) 106: 393-407.

19. Ernst A.T., Jiang H., Krishnamoorthy M. and Sier D.: Staff Scheduling and Rostering: A Review of Applications, Methods and Models. EJOR, (2004) 153: 3-27.

20. Howell J.P.: Cyclical Scheduling of Nursing Personnel. Hospital JAHA, (1998) 40: 77-85.

21. Ikegami A. and Niwa A: A Subproblem-centric Model and Approach to the Nurse Scheduling Problem, Mathematical Programming, (2003), 97(3): 517-541.

22. Jaumard B., Semet F. and Vovor T.: A Generalized Linear Programming Model for Nurse Scheduling, EJOR, (1998), 107: 1-8.

23. Lukman D., May J. Shuman L. and Wolfe H.: Knowledge-Based Schedule Formulation and Maintenance Under Uncertainty. Journal of the Society of Health Systems, (1991) 2: 42-64.

24. Meyer H auf’m Hofe H.: Solving Rostering Tasks as Constraint Optimisation. In: Selected Papers from the 3rd International Conference on Practice and Theory of Automated Timetabling (PATAT’00), LNCS 2079, (2000) 280-297.

25. Nonobe K. and Ibaraki T.: A Tabu Search Approach to the Constraint Satisfaction Problem as a General Problem Solver. EJOR, (1998) 106: 599-623.

26. Pierskalla W. and Rath G.: Nurse Scheduling using Mathematical Programming. OR, (1976) 24(5): 857-870.

27. Post G. and Veltman B.: Harmonious Personnel Scheduling. In: Proceedings of the 5th International Conference on the Practice and Automated Timetabling (PATAT’04), 557-559.

28. Scoot S. and Simpson R.: Case-Bases Incorporating Scheduling Constraint Dimensions – Experiences in Nurse Rostering. In: Advances in Case-Based Reasoning, European Workshop on Case Based Reasoning 1998, LNAI 1488, (1998) 392-401.

29. Sitompul D. and Randhawa S.: Nurse Rostering Models: A State-of-the-Art Review. Journal of the Society of Health Systems. (1990) 2: 62-72.

30. Valouxis C. and Housos E.: Hybrid Optimisation Techniques for the Workshift and Rest Assignment of Nursing Personnel. Artificial Intelligence in Medicine, (2000), 20: 155-175.

31. Warner M.: Scheduling Nursing Personnel According to Nurse Preference: a Mathematical Programming Approach. OR, (1976) 24: 842-856.

Table 1. Shift types and demand during a week

Shift type	Start time	End time	Demand						
			Mon	Tue	Wed	Thu	Fri	Sat	Sun
Early	07:00	16:00	3	3	3	3	3	2	2
Day	08:00	17:00	3	3	3	3	3	2	2
Late	14:00	23:00	3	3	3	3	3	2	2
Night	23:00	07:00	1	1	1	1	1	1	1

Table 2. Soft constraints and their weights

	Soft Constraints	Weights
1	For the period of Friday 23:00 to Monday 0:00, a nurse should have either no shifts or at least 2 shifts (Complete Weekend).	1000
2	Avoid sequence of shifts with length of 1 for all nurses.	1000
3a	For nurses with availability of 30-36 hours per week, the length of a series of *night* shifts should be within the range [2, 3]. It could be part of, but not before, another sequence of shifts.	1000
3b	For nurses with availability of 0-30 hours per week, the length of a series of *night* shifts should be within the range [2, 3]. It could be part of, but not before, another sequence of shifts.	1000
4	The rest after a series of *day*, *early* or *late* shifts is at least 2 days.	100
5a	For nurses with availability of 30-36 hours per week, the number of shifts is within the range [4, 5] per week.	10
5b	For nurses with availability of 0-30 hours per week, the number of shifts is within the range [2, 3] per week.	10
6a	For nurses with availability of 30-36 hours per week, the length of a series of shifts should be within the range of [4, 6].	10
6b	For nurses with availability of 0-30 hours per week, the length of a series of shifts should be within the range [2, 3].	10
7	For all nurse, the length of a series of *early* shifts should be within the range [2, 3]. It could be within another series of shifts.	10
8	For all nurse the length of a series of *late* shifts should be within the range of [2, 3]. It could be within another series of shifts.	10
9a	An *early* shift after a *day* shift should be avoided.	5
9b	An *early* shift after a *late* shift should be avoided.	5
9c	A *day* shift after a *late* shift should be avoided.	5
10	A *night* shift after an *early* shift should be avoided.	1

Table 3. Basic night shift patterns

Valid	M	T	W	T	F	S	S	Invalid	M	T	W	T	F	S	S
Pattern a	N	N	-	-	-	-	-	Pattern d	N	N	N	-	-	-	-
Pattern b	-	-	N	N	-	-	-	Pattern e	-	-	-	N	N	-	-
Pattern c	-	-	-	-	N	N	N	Pattern f	-	-	-	-	-	N	N

Table 4. Penalties of the violations within blocks

		Succeeding shifts			
		N	*E*	*D*	*L*
Preceding shift	*N*	ok	n/f	n/f	n/f
	E	ok	ok	ok	ok
	D	ok	5	ok	ok
	L	ok	n/f or 5*	n/f or 5*	ok

*: depends on the scheduling period where the shifts are (hard constraint 8, or soft constraints 9b and 9c)

Table 5. A cyclic pattern of 5 weeks (upper table) and the coverage of different shift types (lower table).

	Week 1							Week 2							
	M	T	W	T	F	S	S	M	T	W	T	F	S	S	
Nurse1	D	D	L	L	L					D	D	**N**	**N**	**N**	
Nurse2			D	D	**N**	**N**	**N**				E	E	D	D	… etc.
Nurse3				E	E	D	D	L	L				L	L	
Nurse4	L	L				L	L	E	E	L	L				
Nurse5	E	E	L	L				D	D	L	L	L			

Cost for each nurse = 5

Violation: L is followed by E (soft constraint 9b in Table 2)

Number of shifts covered in one week

	M	T	W	T	F	S	S
N	0	0	0	0	1	1	1
E	1	1	0	1	1	0	0
D	1	1	1	1	0	1	1
L	1	1	2	2	1	1	1

Table 6. Comparisons of the approaches (Hybrid GA: genetic algorithm + local search; Hybrid VNS: iterated variable neighborhood search, and our approaches) on the problem.

Hybrid GA	*Hybrid VNS*	*Decomposition + construction*	*Decomposition + construction + VNS*
630 (5 minutes)	466 (1 minute)	340	170 (< 1 minute)
505 (40 minutes)	-	-	-
411 (6 hours)	-	-	-

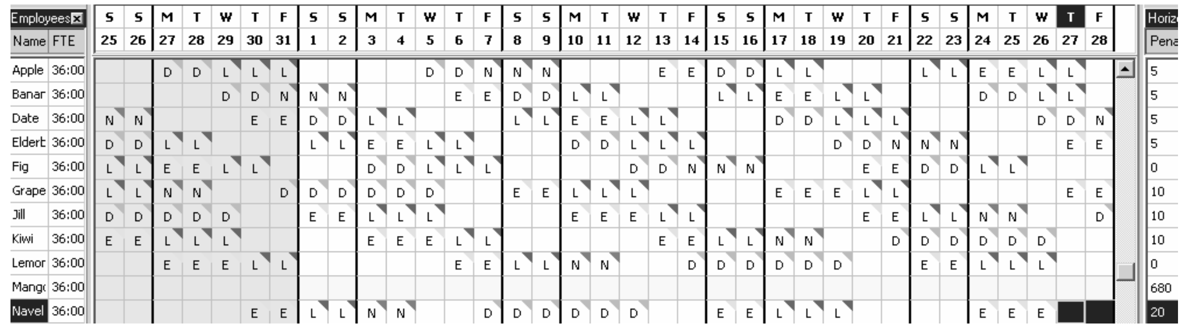

Figure 1. Blocks assigned to 2 groups of 5 nurses, respectively

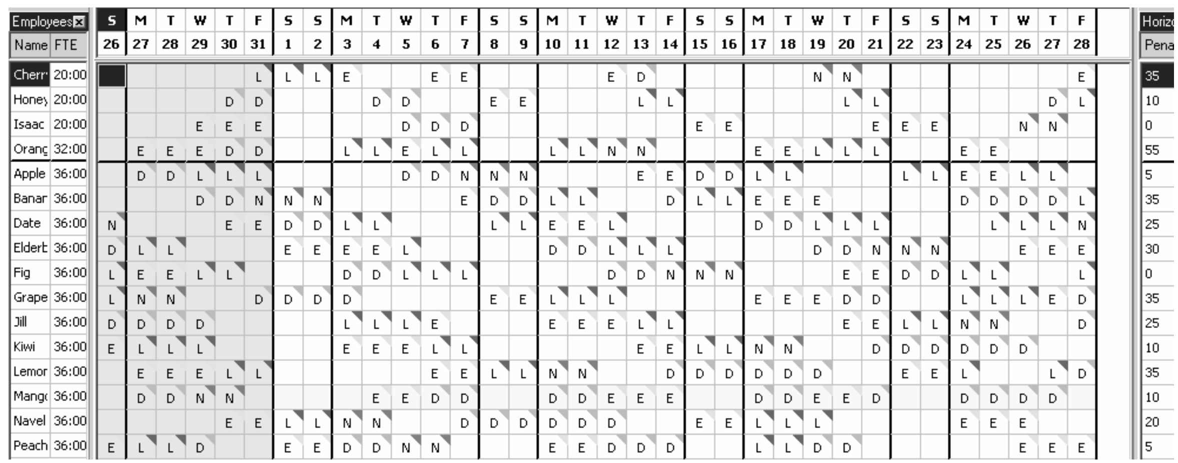

Figure 2. A complete schedule using the decomposition approach