Australian Journal of Basic and Applied Sciences, x (x) December 2014, Pages: x-x

AENSI Journals

Australian Journal of Basic and Applied Sciences

Journal home page: www.ajbasweb.com

Seeing the Bigger Picture: Domain Transformation Approach to the Nurse Scheduling Problem

Geetha Baskaran *[1], Andrzej Bargiela [2,3], Rong Qu [2]

[1] *School of Computer Science, Faculty of Science, The University of Nottingham Malaysia Campus, Jalan Broga, 43500 Semenyih, Selangor, Malaysia.*
[2] *School of Computer Science, University of Nottingham, Jubilee Campus, Nottingham NG8 1BB, United Kingdom.*
[3] *School of Computer Science, Krakow Technical University. Krakow, Poland.*

ARTICLE INFO

Article history:
Received xx October 2014
Accepted xx December 2014
Available online xx xx 2014

Keywords:
Domain Transformation, Nurse Scheduling Problem, Scheduling, Granulation, Simulation

ABSTRACT

Background: Nurse scheduling is a complex combinatorial optimization problem. **Objective:** This paper presents a novel approach to solving the nurse scheduling problem by simplifying it through information granulation. The complexity of the problem is due to a large solution space and the many constraints that need to be satisfied. **Results:** In general, the problem consists of assigning one of ***S*** different shifts to each nurse (who has one of ***C*** possible contracts) over the scheduling period of ***W*** weeks. The study selected to depict the scheduling problem is staffed by 12 full time nurses working a contracted 36 hours per week, one part-time nurse working 32 hours per week and 3 other part-time nurses working 20 hours per week, each being assigned to one of the five possible shift types over a period of five weeks. The solution space for such a specific problem involves $\mathbf{16*5^{35}}$, which are approx. $\mathbf{4.6*10^{25}}$ possible schedules. The novel approach proposed here involves a simplification of the original problem by a judicious grouping of shift types and a grouping of individual shifts into weekly sequences that lead to a re-formulation of the original problem in a much reduced solution space of $\mathbf{8*10^{17}}$schedules defined through weekly patterns. Subsequently, the schedules from the reduced problem space are translated into the original problem space by taking into account the constraints that could not be represented in the reduced space. The grouping of the shift types into a prototype shift and the grouping of shift sequences into weekly patterns used in scheduling are collectively referred to as information granulation. The proposed method has been evaluated and has shown a capable of finding high quality schedules. **Conclusion:** Domain transformation represents departure from a conventional one-shift-at-a-time scheduling approach. It offers deterministic reproducibility of the results. We note that however it does not guarantee the global optimum results.

To Cite This Article: Geetha Baskaran, Andrzej Bargiela and Rong Qu, Paper title. ***Aust. J. Basic & Appl. Sci.,*** *X(XX): x-x, 2014*

INTRODUCTION

Efficient scheduling of healthcare personnel leads to effective utilization of valuable resources. The most general form of the nurse scheduling problem could be described as follows: subject to a set of constraints and given a set of shifts, nurses and a time frame, every nurse is assigned to a shift. The constraints are usually defined by regulations, working practices and the preferences of the nurses (Brucker et al., 2005). In general, the scheduling problem involves allocating suitably qualified staffs to meet a time dependent demand for different services. Creating rosters, however, is a difficult and challenging search problem which requires the satisfaction of many constraints and the balancing of a variety of requirements. As a consequence, more systematic approaches have been developed (Hadi and Jonathan, 2007). Most nurse

Corresponding Author: Geetha Baskaran, School of Computer Science, Faculty of Science, The University of Nottingham Malaysia Campus,Jalan Broga, 43500 Semenyih, Selangor, Malaysia. Telephone: +603-89248129. Email: Geetha.Baskaran@nottingham.edu.my

rostering problems in the real world are NP-hard (Karp, 1972). Nurse scheduling problems have been a research subject for a decade. A comprehensive discussion of a wide variety of methodologies and models developed to deal with different problem circumstances during the years in the literature is provided in the survey papers by Sitompul and Randhawa (1990), Cheang (2003) and Burke et al. (2004b).

Fundamentally, the approaches are range from traditional mathematical programming methods, linear programming to heuristic methods which guarantee to find an optimal solution. It is also to prove its optimality for every instance of a problem. However, computational difficulties exist with these methods due to the huge size of the search spaces that are generated. To reduce complexity, some researchers have restricted the problem dimensions and developed simplified models. However, this leads to solutions that are not applicable to real hospital situations. A major drawback of these meta-heuristics is they neither can provably produce optimal solutions nor provably reduce the search space. Also, they usually do not have well defined stopping criteria. Moreover, as most nurse scheduling problems are highly constrained problems, the feasible regions of their solution space can be disconnected (i.e. separated by the infeasible area), meta-heuristics generally have difficulty in dealing with such situation (Burke et.al, 2009).

In this paper, we propose an alternative granular formulation of the problem that reduces the size of the problem space with optimal solutions. In addition, the formulation of domain transformation allows the replacement of the extended-time scheduling with the recursive application of a week-at-a-time scheduling process. The nested nature of sets of feasible schedules for consecutive weeks gives rise to a natural hierarchical algorithm for nurse scheduling. The problem is introduced in the next section. This is followed by the proposed solution for this nurse scheduling problem, computational results from a number of experiments using this algorithm, and finally, discuss conclusions and future work, respectively.

THE NURSE ROSTERING PROBLEM

In this study, we interpret the nurse scheduling problem as the problem of constructing appropriate information granules and using these granules to design an optimized roster. The roster must satisfy a variety of hard constraints relating to work regulations and as many soft constraints as possible relating to employee requests and personal preferences. Without the loss of generality, we discuss our contribution in the context of a specific nurse scheduling problem as encountered by ORTEC, the Netherlands, in intensive care units in Dutch hospitals. Over the years, this problem has been tested by a range of meta-heuristic algorithms (Burke et al., 2005; Brucker et al., 2005; Burke, et al., 2008, Burke, et al. 2011; Li et al., 2012), and has become a benchmark data in the literature. The characteristics of this problem have been discussed in Baskaran et. al., 2009.

We focus on creating weekly schedules for a ward with 16 nurses. The problem is to assign a certain number of different types of shifts to 16 nurses in a ward within a scheduling period of 5 weeks. 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 20 hours per week. The required number of nurses on individual shifts for different days of the week is summarized in Table 1. The hard and soft constraints that need to be satisfied are described below, respectively.

Table 1 Shift Types and Demand during a Week

Shift Type	Start Time	End Time	Demand						
			M	T	W	T	F	S	S
Early (E)	07.00	16.00	3	3	3	3	3	2	2
Day (D)	08.00	17.00	3	3	3	3	3	2	2
Late (L)	14.00	23.00	3	3	3	3	3	2	2
Night (N)	23.00	07.00	1	1	1	1	1	1	1
Rest (R)	Denotes any of the above if the nurse is not required to work during this shifts								

Constraints

The nurse scheduling problem involves allocating the required workload to nurses subject to a number of constraints. Constraints are categorized into two groups, hard and soft constraints, which vary with legal regulations and individual preferences.

Hard Constraints

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

1. Demands need to be fulfilled
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 number of night shifts is 3 per period of 5 consecutive weeks.
5. 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.
6. Following a series of at least 2 consecutive night shifts, a 42 hours rest period is required.
7. 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.
8. The number of consecutive night shifts is at most 3.
9. The number of consecutive shifts (workdays) is at most 6.
10. One of the full-time nurses requires not receiving any late shifts.
11. The maximum labour time averages 36 hours per week over a period of 13 consecutive weeks if this period does not include work during night shifts.

Soft Constraints

Soft constraints in the problem we are dealing with are listed in Figure 1. 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 highest weight is 1000 and this can be said as the more strongly desired the constraint is. On the other hand, the lowest weight is 1 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.

Soft Constraint 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

Figure 1: Soft Constraints and their weights

Definition of Shifts, Sequence of Shifts and Schedule

There are 5 shift types in the problem as presented in Table 1. Feasible sequences of shifts are those that must satisfy all the hard constraints. For the above problem we have identified 16768 feasible sequences of shifts for a one-week period. An example of a feasible sequence of shift is *EDLLRRR* and an example of an infeasible sequence of shift is *EDLNNNN.* The latter is infeasible because the sequence of night shifts violates hard constraint 9 by having 4 nights instead of just 3 nights.

These feasible sequences can also be classified as zero cost and non-zero cost sequences of shifts. Here, zero cost means sequences of shifts that do not violate any soft constraints and non-zero cost means sequences of shifts that violate one or more soft constraints. The cost here associated with soft constraints where it can be from cost 1 to cost 1000 or even more when the sequences are connected in weekly basis due to the different number of violations. Briefly, sequences of shifts mean satisfying constraints. An example of a feasible sequence of shifts that do not violate any soft constraints is *ELLLRRR*. We will refer to such sequences as "zero-cost sequences". Similarly, an example of a feasible sequence of shifts that violates soft constraint 2 by having sequence of shift with length of 1 and gives a cost of 1000 is *ELLLRRE*. Among all the feasible sequences there are 193zero cost sequences for the 36/32hours nurses and 66zero cost sequences for the 20hours nurses. The remaining 16510 feasible shift sequences have non-zero cost.

A schedule is a set of sequences allocated to each nurses such that they add up to the requirement of cover described in Table 1. In our specific case study, schedules are constructed so as to minimize the cost of sequences of shifts over the period of 5 weeks. Figure 2 shows an example of schedule with specific number of nurses and cover requirement for a period of 5 weeks.

Sequences of Shifts					
	Week 1	Week 2	Week3	Week 4	Week 5
Nurses	MTWTFSS	M…….S	M…….S	M…….S	M…….S
1 . 12 . 16	ELLRRLL ……….. LRRRDDD ……….. ………..	LDNNRRR … DLLRRLL … …	RLLLDRR LDNNRRR	EEELLRR RDDLLRR	LLLRRLL DDDDLRR
Cover Early (E) Day (D) Late (L)	3333322	3E3D3L1N	3E3D3L1N	3E3D3L1N	3E3D3L1N

Figure 2. Example of schedule in 5 weeks with 16 nurses

Scheduling

Scheduling is a process of allocating shifts to nurses over a pre-defined period of time and subject to various constraints. Scheduling that satisfies only the hard constraints on sequences of shifts and the cover requirement generates a feasible schedule (as illustrated in Figure 2). This may need to be refined to lower the cost of the schedule by ensuring satisfaction of as many as possible soft constraints. The scheduling problem in the above scenario presents a combinatorial optimization in a space of$\mathbf{16*5^{35}=4.6*10^{25}}$possible schedules, clearly a computational prohibitive task. Most of the methods highlighted in section 1 perform optimization on feasible schedule by adjusting individual shifts. This can be a replacement of one shift type with another and subsequent balancing of the required cover. Alternatively optimization may involve swapping of shifts allocated to two nurses in the same day which, by definition, does not alter the staff cover.

Figure 3 shows the swapping of the shift which can produce a lower-cost schedule. Both, a simple change of a single shift and swapping of two shifts imply non-monotonic changes in the cost of a schedule. In other words, the decrease of the number of violated soft constraints does not necessarily imply the decrease of the cost function. So, the process of optimization of the non-monotonic cost may converge to local optima rather than global optima. For example, if we have 15 constraints with different cost value; it in itself a challenge to make the right choice which constraint to violate and not to violate. In other words, if the schedule violates one constraint that is more expensive (i.e. say it has associated cost of 1000), it can be replaced with a schedule that violates several less expensive constraints. We may have a larger

Australian Journal of Basic and Applied Sciences, 7(13) Nov 2014, Pages: x-x

number of violations but it comes at a lower cost. Unfortunately local optima evaluated in this way do not provide any guidance with regards of the required adjustment of the independent variables that would facilitate convergence to global optima. This means that the previous methods had to perform combinatorial search in a large problem domain space.

Sequences of Shifts					
	Week 1	Week 2	Week3	Week 4	Week 5
Nurses	MTWTFSS	M…….S	M…….S	M…….S	M…….S
1	ELLRRLL	LDNNRRR	RLLLDRR	EEELLRR	LLLRRLL
.	………..	**Swapping**			
12	LRRRDDD	DLLRRLL	LDNNRRR	RDDLLRR	DDDDLRR
.	………..	….			
16	………..	….			
Cover	3E3D3L1N	3E3D3L1N	3E3D3L1N	3E3D3L1N	3E3D3L1N

Figure 3: Shift swapping for a schedule with lower cost

PROPOSED SOLUTION

The main challenge in nurse scheduling is to allocate specific shifts to nurses while ensuring minimum cost (penalty). In this study we adopt a novel information granulation approach to nurse scheduling. A basic notion of granular computing is processing aggregated information that represents semantically meaningful entities in the context of a specific application. Like sets theory, granular computing explores the composition of information items into information granules (analogous to forming set-theoretic classes from set elements), their interrelationships, and the semantic transformation of the data (Bargiela and Pedrycz, 2008). The model build on information granules represents a simplified representation of the actual scheduling problem but it can potentially have enhanced generality because of the degree of abstraction from non-critical information that is inherent to the process of data granulation (Bargiela and Pedrycz, 2003). In this context, the challenge of granular computing is to design and validate appropriate information granules based on multilevel and multi-view representation of the problem (Yao, 2006). Generally speaking, information granules are collections of entities that usually originate at the numeric level and are arranged together due to their similarity, functional or physical adjacency; in distinguish ability, coherency, or the like. The information granules produced in this study are aggregated shifts types and patterns representing shift sequences with soft constraints taken into consideration. This data process creates a significant methodological development of nurse scheduling practice. The aggregation was inspired by the insights from our previous studies (Bargiela, 1985; Peytchev, et. al., 1996) and has been formalized as a Granular Computing Methodology (Bargiela, et. al., 2003, 2004, 2008).

Our novel approach to the solution of the scheduling problem is referred here as the *domain transformation approach (in the context of Information granulation).* The domain transformation is a general methodological approach that has been used in other application domains such as control system design. In this case, the Laplace Transform converts a difficult problem of solving partial differential equations in the time-domain into a relatively easy problem of solving algebraic equations in the Laplace s-domain (Goodwin et. al, 2000). The combined computational effort of the domain transformation plus the solution of the transformed problem and the conversion from the transformed to the original domain is significantly smaller than the solution in the original problem domain. The same problem solving philosophy is proposed here in the context of nurse scheduling. The general view of our approach can be summarized as a 3-stage process:

I) conversion of the problem from the original EDLNR domain into a problem in the smaller **D**NR domain (the EDLNR and **D**NR terms are explained below)
II) solution of the problem in the **D**NR domain
III) conversion of the **D**NR solution into a solution in the original EDLNR domain

Granulation on shift types

The EDLNR domain is the problem domain with five types of shifts defined in Table 1. The logic of granulation of shifts into patterns can be applied directly to the EDLNR shifts.. From the description of the shifts, it is clear that the Early (E), Day (D) and Late(L) type shifts are similar in terms of working hours and also in terms of work regulations that are applicable to them. The similarities of EDL shifts justify the consideration of these three shifts as a shift of type "merged-Day" (**D**-shift) which simplifies the scheduling task. By contrast, the night shifts Night (N) have clearly distinct set of work regulations and are therefore retained as N-shift. Similarly we also retain the rest shift (R). We propose therefore, that the nurse-scheduling problem may be expressed at a more abstract level using just three types of shifts merged-Day (**D**) ,night (N) and rest (R) shift.

To assess the complexity of the scheduling problem, we can consider the following: the problem consists of S shift types and we are concerned with providing a schedule for N nurses over a period of W weeks of the solution search space is N*S^(7*W); which means that for every nurse there is a possibility of assigning one of S shifts in each day within the scheduling horizon of 7*W days. In the specific case of 16 nurses working over 35 days (5weeks) and involving 5 shift types we have (**16*5^35=4.6*10^25**) different schedules in the overall EDLNR solution space. By contrast in the **D**NR solution space for schedules for the same number of nurses and the same duration but involving 3 shift types the number of possible schedules is considerably smaller (**16*3^35=8*10^17**) although this is still prohibitive computational.

A further reduction of the cardinality of the solution space can be obtained by considering shorter scheduling periods. The reduction of the search space is probably best illustrated by making it explicit that since we adopt a week-at-a-time approach the problem space reduces to: W*(N*S^7) which is 5*(**16*5^7**) in the EDLNR and 5*(**16*3^7**) in the **D**NR. So, the reduction is from 80*5^7 to 80*3^7 (from 78125 in the EDLNR to 2187 in the **D**NR domain). Among those are 16768 out of 78125 refers to feasible EDLNR sequences and 160 out of 2187 refers to feasible **D**NR sequences. Table 2 summarizes the requirement for the staff cover for the corresponding shifts in the **D**NR domain during one week.

Table 2: Demand summarization

	Demand						
	M	T	W	T	F	S	S
Merged-shift (**D**)	9	9	9	9	9	6	6
Night (N)	1	1	1	1	1	1	1
Rest (R)	Denotes any of the above if the nurse is not required to work during this shifts						

Granulation of shift sequences into patterns

We note that the soft constraints are expressed in terms of penalties associated with specific shift sequences during one week. We can therefore produce sequences of shifts of one-week duration that do not have any penalties associated with them and sequences that have some arbitrary penalties. We will call those sequences "patterns" and we will use them as basic building blocks for the schedules. The distinct value added of patterns is that because of their prior assessment with regard to satisfaction of soft constraints they can be used in the scheduling process without the need for subsequent checking of the constraints. This is an advantage compared to the scheduling with sequences of shift where a change of a single shift implies the need for the evaluation of all constraints (hard and soft). Figure 4 provides examples of such zero-cost patterns and Figure 5 provides examples of non-zero-cost patterns.

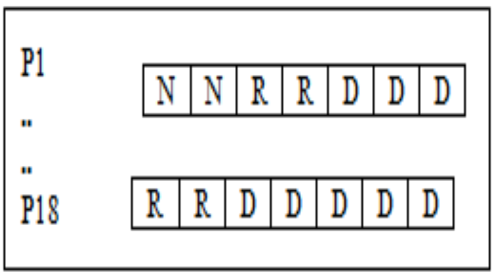


Figure 4: No violation of Soft constraints (called as zero cost patterns)

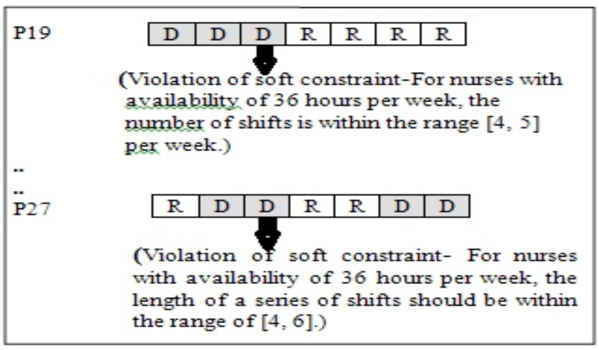


Figure 5: Violation of Soft constraints with Cost 10 (called as non-zero cost patterns)

Australian Journal of Basic and Applied Sciences, 7(13) Nov 2014, Pages: x-x

By combining the granulation of shift types and the granulation of shift sequences into patterns we can derive patterns of shifts in the **D**NR domain. Clearly such patterns represent sets of patterns in the original EDLNR domain. For example a pattern **DD**NNRRR can be considered as a representative of 9 patterns in the EDLNR domain as illustrated in the figure 6. We note that some of the sequences in the EDLNR domain have a non-zero cost due to the interrelationship that cannot be captured in the **D**NR domain since all EDL shifts are represented by the same shift **D**. Table 3 shows the interrelationship of the EDL shifts that gives rise to some cost.

Table 3: Interrelationship of the EDL and **D**NR with the cost associated.

Preceding Shifts	Succeeding Shifts			
	N	E	D	L
N	Ok	n/f	n/f	n/f
E	Ok	Ok	Ok	Ok
D	Ok	5	Ok	Ok
L	Ok	5	5	Ok

Preceding Shifts	Succeeding Shifts	
	D	N
D	Ok	Ok
N	n/f	Ok

n/f = not feasible

So, we adopt the lowest cost of the sequences in the EDLNR domain as an indication of the lower limit on the cost in the **D**NR domain as illustrated in Figure 5. Of course, this means that we are open to revising upward the cost of the schedule once we convert the solution from the **D**NR domain to the original EDLNR domain. The problem of scheduling shifts is therefore transformed into the problem of scheduling patterns. The computational gain that can be attained from this domain transformation depends on the number of patterns that need to be considered. It turns out that the number of zero-cost patterns and patterns with other pre-specified cost is relatively small.

Sequences of shifts							Cost
E	E	N	N	R	R	R	0
E	L	N	N	R	R	R	0
E	D	N	N	R	R	R	0
D	D	N	N	R	R	R	0
D	L	N	N	R	R	R	0
D	E	N	N	R	R	R	5
L	L	N	N	R	R	R	0
L	E	N	N	R	R	R	5
L	D	N	N	R	R	R	5

Pattern							Cost Limit
D	**D**	N	N	R	R	R	0

Figure 6 : EDLNR domain patterns and a representation of **D**NR pattern

Patterns construction

With the granulation of data described in the previous section, we can proceed with the construction of patterns that satisfy the various constraints. We start with all possible shift sequences and apply consecutively the hard constraints so as to eliminate these sequences that violate them (i.e. infeasible sequences). The remaining sequences are feasible as far as the individual hard constraints are concerned, but this set can be refined further by considering implicit hard constraints derived from the combination of hard constraints.

Table 3: Night Sequences

	M	T	W	T	F	S	S
1	N	N	R	R	-	-	-
2	-	-	N	N	R	R	-
3	-	-	-	-	N	N	N

Such implicit hard constraints are illustrated in Table 3. The requirement for an uninterrupted sequence of 3 night shifts over the weekend and the maximum number of 2-3 night shifts during the week creates an implicit requirement that one may not have consecutive 3 night shifts on Monday-Tuesday-Wednesday or Tuesday-Wednesday-Thursday because the remaining day would have just a single night shift. After applying the implicit hard constraints (Table 3) the sequences satisfying these constraints are available for ranking with respect of their soft constraints violation cost. We start ranking these sequences from the highest cost of 1000 to zero-cost, (i.e. satisfying all the soft constraints). For the 36 hour full-time nurses

Australian Journal of Basic and Applied Sciences, 7(13) Nov 2014, Pages: x-x

and 32 hour part-time nurses there are 18 zero-cost patterns. Meanwhile, for the 20 hour part-time nurses there are 15 zero-cost patterns. Table 4 itemizes all these zero-cost patterns for the different contract of nurses.

Table 4: Numbering of the zero-cost patterns for the 36/32h nurse (18 patterns) and 20h nurses (15 patterns) in the DNR solution space.

	36hoursFT/32hours PT Nurses		**20 hours PT Nurses**
A1	NNRRRDD	B1	NNRRRRR
A2	NNRRDDD	B2	RRNNRRR
A3	DDNNRRR	B3	RDNNRRR
A4	DRRRNNN	B4	RRRRNNN
A5	DDRRNNN	B5	DDRRRRR
A6	RRRDNNN	B6	RDDRRRR
A7	DRRDNNN	B7	DDDRRRR
A8	RRDDNNN	B8	RRDDRRR
A9	DDDDRRR	B9	RDDDRRR
A10	RDDDDRR	B10	RRRDDRR
A11	DDDDDRR	B11	DRRDDRR
A12	DDRRRDD	B12	RRDDDRR
A13	DDDRRDD	B13	RRRRRDD
A14	DRRRDDD	B14	DRRRRDD
A15	DDRRDDD	B15	RRRRDDD
A16	RRRDDDD		
A17	DRRDDDD		
A18	RRDDDDD		
Full-time (FT) and Part- time (PT)			

If the zero-cost patterns are augmented by non-zero cost patterns e.g. patterns violating soft constraints with cost 10; then the set of available patterns for the 36 hour and 32 hour nurses are 30 and the set of patterns for the 20 hour nurses are 26. Clearly by including progressively higher cost patterns, the set of available patterns will increase but since the objective of the scheduling is to find the lowest cost schedule, typically there is no need to consider higher cost patterns.

Domain Transformation Scheduling Workflow

Scheduling involves selecting one out of the set of available patterns for each nurse. The selection is subject to the requirement that the cover specified in Table 2 is satisfied. Although the cardinality of the sets of patterns that can be assigned to individual nurses is relatively small (of the order of tens) the combinatorial space of schedules is still very large. Considering selection of 1 out of 18 patterns in each of the 5 weeks amounts to the consideration of 18^5=1889568 schedules. We proceed therefore with the simplification of the nurse scheduling problem from the full-scheduling-horizon to a recursive-weekly-scheduling. The proposed procedure deployed in the following stages:-

Step 1: Scheduling One Week Schedule (**D**NR)
Step 2: Expand schedule in N weeks (**D**NR)
Step 3: Convert **D**NR to EDLNR schedule
Each of these mentioned steps in our proposed domain transformation scheduling will be discussed in details within the following subsections.

Scheduling One Week Schedule (DNR)

In step 1, once we have identified the zero cost patterns as in Table 4, we construct the week 1 schedule in DNR domain. We associate patterns based on zero cost patterns with nurses based on full time or part time schedule. This is called *schedule set*. Schedule set is stored in a vector object (array). In order to construct the schedule, we must consider some specific measure. As a result, we first consider the shifts that are the most difficult to schedule. In this subject, the most important and difficult one to schedule is the night shift. Moreover, they are the most important shifts with a cost of 1000 if the length is not within the range of [2, 3]. Subsequently, we place the day only patterns in the array of day patterns. There are 18 zero cost patterns for the 32/36h nurse and 15 zero cost patterns for the 20h nurses. We noticed that the

Australian Journal of Basic and Applied Sciences, 7(13) Nov 2014, Pages: x-x

32/36h have the same patterns as they fall in the same category we refer to as set A. Similarly, we named set B for the 20h nurses. We then proceed scheduling 13 nurses using patterns from set A and 3 nurses using patterns from set B.

Firstly, we select one assignment of night patterns based on zero cost patterns. As shown in Table 6, the N (night shifts) are grouped together in pairs or triples at fixed days. Here, we can calculate the number of ways to combining them in the patterns, using mathematical combinations as described in Table 7. The objective is to satisfy the demand of 1111111 for the night shift.

Table 6: Night shifts based on zero cost patterns

	36/32 hours FT Nurses		**20 hours PT Nurses**
A1 A2	NNRRRDD NNRRDDD	B1	NNRRRRR
A3	DDNNRRR	B2 B3	RRNNRRR RDNNRRR
A4 A5 A6 A7 A8	DRRRNNN DDRRNNN RRRDNNN DRRDNNN RRDDNNN	B4	RRRRNNN

Table 7: Mathematical combinations of night shifts based on zero cost patterns

	M T	W T	F S S
FT FT + PT	${}_2C_1$ ${}_3C_1$	${}_1C_1$ ${}_3C_1$	${}_5C_1$ ${}_6C_1$
Demand Filled	11	11	111

Where M T = Monday, Tuesday
W T = Wednesday, Thursday
F S S = Friday, Saturday, Sunday

For the FT, we only have 2 night patterns which are the A1 and A2 for M and T. As for W and T we only have A3 pattern. While for F, S, and S we have 5 patterns which are from A4 to A8. So, if we are choosing for just FT, then we select 1 from 2 for M and T. Similarly for the W and T, we select 1 and we select 1 from 5 for F, S and S. This implies the same for FT+PT. For M and T, we select 1 pair from 3. Similarly, for W and T, we select 1 from 3. For F, S and S, we select 1 from 6. The total number of combinations of patterns that satisfy demand for night shift cover is as following: ${}_3C_1$ x ${}_3C_1$ x ${}_6C_1$ = 54 (where ${}_3C_1$ denotes number of different selections of one pattern out of the set of three patterns) Later the day shifts are assigned as illustrated in Table 8. Here we assign the day shifts on weekend using the zero cost patterns. To satisfy the demand for total D of 9999966, block A12 to A18, are chosen at first. This are basically patterns of days on weekends. So, the total number is ${}_7C_5$=21. Next, the rest shifts on weekend are assigned. To satisfy the demand for total R of 9999966, patterns of rest on weekends are chosen. Firstly, block A9 to A11 are chosen. Later on, block B5 to B12 are chosen next. If we did not find zero cost assignment, then increase number of pattern by involving the non-zero cost patterns. If the demand is over satisfied, we use the switching patterns as shown in Table 9. We can increase number of replacement by involving the switching patterns of non-zero cost patterns also. This can be done by move the shift or less the shift according to the days. A complete zero cost patterns switching is shown in Figure 7. Here it is shown in different shifts counts meaning the number of day shift in a pattern. For example: We can move shift from A17 to A16 or vice versa. This means we are moving from 5-day shift to 4-day shift on a Monday.

Table 11 shows an example of switching zero cost patterns. Initially by placing the nights and days pattern, the demand was not satisfied. It was "9979966". This was corrected by using the switching patterns. Hence, pattern A12 is shifted to pattern A13 for 2 nurses. This shifting of patterns satisfies the demand of "9999966".

Australian Journal of Basic and Applied Sciences, 7(13) Nov 2014, Pages: x-x

Table 8: Day shifts based on zero cost patterns

	36/32 hours FT Nurses	M T W T F S S		**20 hours PT Nurses**	M T W T F S S
A9	DDDDRRR	1 1 1 0 0 0 0	B5	DDRRRRR	1 1 0 0 0 0 0
A10	RDDDDRR	0 1 1 1 1 0 0	B6	RDDRRRR	0 1 1 0 0 0 0
A11	DDDDDRR	1 1 1 1 1 0 0	B7	DDDRRRR	1 1 1 0 0 0 0
A12	DDRRRDD	1 1 0 0 0 1 1	B8	RRDDRRR	0 0 1 1 0 0 0
A13	DDDRRDD	1 1 1 0 0 1 1	B9	RDDDRRR	0 1 1 1 0 0 0
A14	DRRRDDD	1 0 0 0 1 1 1	B10	RRRDDRR	0 0 0 1 1 0 0
A15	DDRRDDD	1 1 0 0 1 1 1	B11	DRRDDRR	1 0 0 1 1 0 0
A16	RRRDDDD	0 0 0 1 1 1 1	B12	RRDDDRR	0 0 1 1 1 0 0
A17	DRRDDDD	1 0 0 1 1 1 1	B13	RRRRRDD	0 0 0 0 0 1 1
A18	RRDDDDD	0 0 1 1 1 1 1	B14	DRRRRDD	1 0 0 0 0 1 1
			B15	RRRRDDD	0 0 0 0 1 1 1

Table 9: Switching patterns based on zero cost patterns

Replacement of patterns		Day of the week
Move shift	Less shift	
A17	A16	Monday
A9	C9	Monday
..	..	..
A15	A14	Tuesday
..	..	..

Shift/Day	5	4	3	2
Monday		A4 →	B4	
		A3 →	B3	
	A7 →	A6		
		A9 →	B9	
	A11 →	A10		
		A14 →	B16	
	A17 →	A16		
			B7 →	B6
			B11 →	B10
			B14 →	B13
Tuesday	A5 →	A4		
		A10 →	B12	
		A12 →	B14	
	A15 →	A14		
			B2 →	B3
			B8 →	B9
Wednesday	A8 →	A6		
	A13 →	A12		
	A18 →	A16		
			B7 →	B5

Shift/Day	5	4	3	2
Wednesday			B12 →	B10
Thursday	A7 →	A4		
		A6 →	B4	
		A9 →	B7	
	A17 →	A14		
		A16 →	B15	
			B9 →	B6
	A2 →	A1		
	A11 →	A9		
		A10 →	B9	
	A15 →	A12		
		A14 →	B14	
			B12 →	B8
			B15 →	B13
Friday	A2 →	A1		
	A11 →	A9		
	A15 →	A12		
		A14 →	B14	
			B12 →	B8
			B15 →	B13

Figure 7 : Switching patterns based on shifts of zero cost patterns according to days

Table 11 : Example of week 1 schedule using zero cost patterns

Nurse number	Pattern number / Switch	Patterns	Initial Cover M T W T F S S	Pattern after switch	Cover after switch M T W T F S S
1	A4	DRRRNNN	1 0 0 0 0 0 0		1 0 0 0 0 0 0
2	A3	DDNNRRR	1 1 0 0 0 0 0		1 1 0 0 0 0 0
3	A1	NNRRRDD	0 0 0 0 0 1 1		0 0 0 0 0 1 1
	Partial cover1 of D		**2 1 0 0 0 1 1**		**2 1 0 0 0 1 1**
4	A12->A13	DDRRRDD	1 1 0 0 0 1 1	DDDRRDD	1 1 1 0 0 1 1
5	A17	DRRDDDD	1 0 0 1 1 1 1		1 0 0 1 1 1 1
6	A18	RRDDDDD	0 0 1 1 1 1 1		0 0 1 1 1 1 1
7	A12->A13	DDRRRDD	1 1 0 0 0 1 1	DDDRRDD	1 1 1 0 0 1 1
8	A14	DRRRDDD	1 0 0 0 1 1 1		1 0 0 0 1 1 1
	Partial cover2 of D		**6 3 1 2 36 6**		**6 3 3 2 3 6 6**
9	A10	RDDDDRR	0 1 1 1 1 0 0		0 1 1 1 1 0 0
10	A10	RDDDDRR	0 1 1 1 1 0 0		0 1 1 1 1 0 0
11	A11	DDDDDRR	1 1 1 1 1 0 0		1 1 1 1 1 0 0
12	A11	DDDDDRR	1 1 1 1 1 0 0		1 1 1 1 1 0 0
13	A9	DDDDRRR	1 1 1 1 0 0 0		1 1 1 1 0 0 0
14	B6	RDDRRRR	0 1 1 0 0 0 0		0 1 1 0 0 0 0
15	B10	RRRDDRR	0 0 0 1 1 0 0		0 0 0 1 1 0 0
16	B10	RRRDDRR	0 0 0 1 1 0 0		0 0 0 1 1 0 0
	TOTAL OF D		**9 9 7 9 9 6 6**		**9 9 9 9 9 6 6**

Expand schedule for N+1 Week (DNR)

Based on the week N schedule, we could construct the N+1 week schedule in DNR domain. For N=1 we have 45 zero cost schedules. Week 1 selection is very important because it underpins finding a good schedule in the following weeks. The most important hard constraint to be checked is the night shifts constraints. Refer to hard constraint 4 and 9, the night shift nurses needed are 2<=night shift nurses<=3. There are 2 sets used in this placement. First set is when we choose one set of 3 night shift patterns from pattern A. While the second set is when we place 2 night shift patterns from pattern A and 1 night shift pattern from pattern B. Figure 9 illustrates the placement of night shift in the schedule. Indirectly, this night placement will actually satisfy the hard constraint 11. For example when we calculate night shift per week over the 5 week scheduling period:

Night shift placed using patterns A are 3+3+2+2+2 = 12 per period of 5 consecutive weeks. Meanwhile night shifts for patterns B are 1+1+1 = 3.

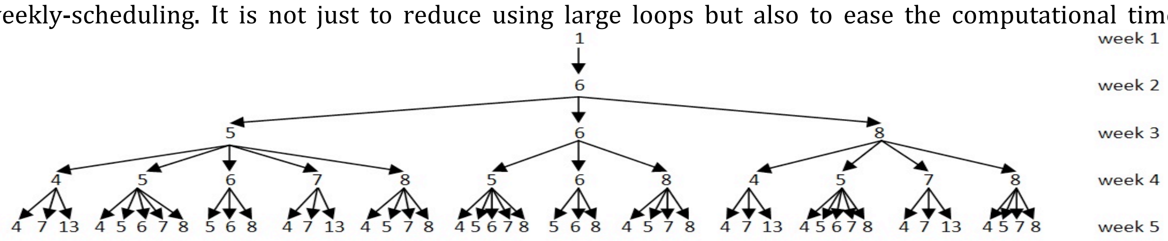
Figure 9: Night placement over 5 week

Subsequently, we need to satisfy the hard constraint 5, 6 and 9. Table 12 shows a good sample of patterns that need to be considered when we are selecting schedules from the generated zero cost weeks 1 schedule for the N weeks. For example; if we have patterns of A4 to A8, this need to be followed by minimum two days' rest. So, this means pattern A16 satisfies the hard constraints 6. Since pattern A16 has four **D**'s (workdays), and then this pattern can be followed by pattern A12. This way it is satisfying the hard constraint 9 because the number of consecutive shifts is at most 6 days. Besides, the example showed in table 12 is also satisfying the hard constraint 5 with 2 weekends off duty.

Table 12: Possible week's pattern

A4-A8	A16	A12	A9	A9
XXXXNNN	RRRDDDD	DDRRRDD	DDDDRRR	DDDDRRR

Therefore, week 1 zero cost schedules are converted into tree structure as shown in figure 8. All the week 1 zero cost schedules are given a number just for a marking purpose. It is shown in first column of table 11. This numbers are used to check the feasible patterns of shifts that can follow a pattern from a previous week. They are mainly checked according to the hard constraints 5, 6, 9 and the night constraints. This is best illustrated in figure 8 based on the example of schedule generated for week 1 from the table 11. For example nurse 1 have pattern DRRRNNN and marked as 1. This pattern can be followed by RRDDDDD (marked 6) for the week 2. Moving to week 3 for nurse 1, we have pattern marked 5,6 and 8 that can be followed. So, we always choose the first marking which is 5 (DRRDDDD) in this case. These steps are also used in week 4 and week 5. This simplification is very important in the recursive-weekly-scheduling. It is not just to reduce using large loops but also to ease the computational time.



The numbers and the patterns relationship

1	6	5	4	4
DRRRNNN	RRDDDDD	DRRDDDD	DDRRRDD	DDRRRDD

Figure 8: Possible patterns that can be followed as subset list

In this study, building schedules using only zero cost patterns could be done only until week 3 following the subset list that is created in the tree structure. From week 4 onwards the low cost patterns are incorporated to satisfy the demand. Checking is done on switching pattern to see if we can apply these patterns to improve the initial solutions. Our aim, at this stage, is to decrease the penalty cost that occurred due to the use of non-zero cost patterns.

Convert DNR to EDLNR schedule

In step 3, once the **D**NR domain schedule is constructed, we convert this result to the EDLNR domain. Initially, obtain the schedule array from determining whether it is 4 weeks or 5 week schedule. Later, find the 'D' index in the array. For example; we have (RRDDDDD RRRDDDD) as the array. Here, find the 'D' index of array. From above example the position of 'D' is (3 4 5 6 7 11 12 13 14). We convert this '**D**' to D, L, E. So, the permutation for shift L is C_9^3. After select 3 L shift then remains 6 D shift. Hence, permutation for shift E is C_6^3. All possible permutation of EDL in some day is $C_9^3 C_6^3$. This loop is continued until it satisfies the demand. The best schedule found in this method is presented in Figure 10 which is just cost 100 generated in 45 seconds.

Australian Journal of Basic and Applied Sciences, 7(13) Nov 2014, Pages: x-x

Computed Schedule for 5 week(s):								
Week->	**Week 1**	**Week 2**	**Week 3**	**Week 4**	**Week 5**			
Nurse 1	'NNRRRRR'	'EERRREE'	'RREEEDD'	'LRRRELL'	EEERRRR			
Nurse 2	'LLNNRRR'	'EEERREE'	'EEELRRR'	'LLRRLLL'	EERREEE	7	10	EEE
Nurse 3	'RREENNN'	'RREEEDD'	'LRRRELL'	'DLLRRRR'	REEELRR	8	10	LLL
Nurse 4	'EERRREE'	'LLNNRRR'	'LLRRLLL'	'LLLRREE'	ELLLRRR	9a	5	DE
Nurse 5	'EEERREE'	'DLLLRRR'	EENNRRR'	'RRREEDD'	LRRREEE	9b	5	LE
Nurse 6	'LRRRELL'	'NNRRRRR'	'DDLLLRR'	'DDLLLRR'	LLLRRLL	9c	5	LD
Nurse 7	'LLRRLLL'	'LRRRELL'	'LLLRREE'	'EEELRRR'	LLNNRRR			
Nurse 8	'RRREEDD'	'LLRRLLL'	'DLLRRRR'	'RDDLLRR'	RREENNN			
Nurse 9	'RREEEDD'	'RRREEDD'	'EERRREE'	'EENNRRR'	DDLLLRR			
Nurse 10	'EELLLRR'	'RREENNN'	'RRREEDD'	'EERRREE'	DDDLRRR			
Nurse 11	'DLLLRRR'	'EELLLRR'	'RREENNN'	'RREEEDD'	RRRREELL			
Nurse 12	'RDLLLRR'	'RDLLLRR'	'RDDLLRR'	'RREENNN'	RRRDDDD			
Nurse 13	'DDDDDRR'	'DDDDDRR'	'DDDDDRR'	'DDDDDRR'	DDRRDDD			
Nurse 14	DDDRRRR	DDDRRRR	RRREDRR	RRREDRR	NNRRRRR			
Nurse 15	RRRDDRR	RRRDDRR	RRDDDRR	NNRRRRR	RRDDLRR			
Nurse 16	RRDDDRR	RRDDDRR	NNRRRRR	RRDDDRR	RRDDDRR			
Verifying total nurses available each day:								
Total D:	9999966	9999966	9999966	9999966	9999966			
Total N	1111111	1111111	1111111	1111111	1111111			

Figure 10: Result schedule for the nurse scheduling problem

COMPUTATIONAL RESULTS

During the offline preparation process, there are 18 feasible zero cost shift patterns being generated for the 36hours FT nurses and 32hours PT nurses. There are also 15 feasible zero cost shift patterns being generated for the 20hours PT nurses. These are the zero cost patterns which were identified for the sample problem of a ward in a Dutch hospital in the **D**NR space solution. These patterns were used according to the allocation of different type of nurses. We also manage to generate a few set of zero-cost patterns for one-week FT and PT nurse sequences. Later on, the above mentioned week one solution was expanded to a 5-week solution. All the patterns were generated with zero-cost solution for the **D**NR domain until week 3. There is no evidence that having zero-cost solutions in early weeks forces one to adopt "expensive" (non-zero-cost patterns later on) as in our problem solution we have all 20 cost patterns in the **D**NR domain solution. These results were soon converted to the EDLNR space.

The patterns generated here have to satisfy the problem constraints. During the conversion period, again a zero-cost schedule was produced for the period of 5 weeks. However, the situation may change in when try to satisfy more constraints or different set of problem. Should this lead to infeasibility, we relax the constraints incrementally in order of their cost, until a feasible solution is found. The minimal cost patterns were chosen once the zero cost patterns do not fit in the schedule. Both the one week sequences and the 5 week solution will be then having a non-zero-cost schedule to satisfy the demand of the nurses in each day. So, our approach reports the optimal solution for the above problem as 100 cost solution after an execution time of 45 second for a 5 week schedule as shown in table 14. Besides, 95 cost solution for 4 weeks schedule as shown in table 13. The algorithm was implemented using Microsoft Visual C++ 2008. The experiments were performed using a PC with a Dual Core 2.64GHz.

According to the literature which tests this problem, the best known result is 270 by (Glass and Knight, 2009) after an execution time of only 2 minutes. The summary of these solutions is shown in Table 13. We have thus achieved an improvement, obtaining a 100 cost solution. The execution times were achieved using comparable/not comparable computers. Burke et al. (2007, 2008) used a P4, 2.4 GHz processor PC, whereas ours has a clock speed of 2.64 GHz. We note that our very short execution time is partly due to the accessibility of a feasible solution with no penalties which is the zero cost solution. Should this have proved infeasible, we would have relaxed one or more of the higher penalty constraints and would expect the run time to increase as a result. Our particular implementation is an adaptation of (Geetha et al., 2009). We recognize that problems of higher levels of complexity may be computationally challengeable but not impossible to try with this approach.

Table 13: Previous solutions to this same problem statement (4 weeks/one month)

Penalty	**Approach**	**Execution time**	**Author**
775	GA	1 hour	
681	GA	24 hours	
587	GA	(iterative) "Long"	
706	HO/VNS	1 hour	
541	HO/VNS	12 hours	
460	IP/VNS	50 minutes	
360	VDS	25 minutes	Burke et al. (2007

280	VDS	16+ hours	
270	IP	2 minutes	Glass and Knight, 2009
95[x]	Domain Transformation Granular Information	30 seconds	Baskaran, G et al. 2014

Table 14 : Previous solutions to this same problem statement (5 weeks/35 days)

Penalty	**Approach**	**Execution time**	**Author**
170	decomposition + VNS iterative		Brucker et al. 2005
100[x]	Domain Transformation Granular Information	45 seconds	Baskaran, G et al. 2014

CONCLUSIONS

Information granulation was found to be very useful in providing good quality solutions by significantly reducing the possible explorations. The extraction of good features in granulation is used more explicitly reduce the search space and hence to solve the problem. This method has been shown to be a relatively straightforward but highly effective for the nurse scheduling problem concerned in this work, and is a feasible and more effective alternative to the existing algorithms for the commercial workforce management and planning. The conversion of two different spaces is actually a very efficient and effective method of exploring the search space. Besides, we are not overlooking any valid patterns because all the possible patterns are generated using the constraints. The generation of patterns actually provides us high quality schedules. The time taken to do this would depend upon the machine executing it and the dimensions of the problem instance. With the specification of PC used in implementing the algorithm and the experiment, the optimal result has been produced using this method with a shorter time, where it just takes less than 8 seconds to even generate the schedule. Moreover, our approach is not "hard coded" to any constraints. It has been designed with the aim of being able to learn about new problem solving situations in mind. So, this actually gives the advantage of adaptability as it can be applied to other hospital environments by simply altering the formulations of constraints and requirements. In addition, it is flexible as it provides dual criteria of solution space acceptance during the search, thus enabling users more degree of freedom for a better decision making. The study also shows the efficiency of identifying feasible patterns for subsequent weeks. Conceptually this approach can be generalized to non-zero cost pattern sets. Automating the nurse scheduling can actually reduce the scheduling effort and time. It can also take care of all the constraints and give a quick evaluation of schedules.

FUTURE WORK

In our future work, the approach can be extended by employing mathematical model and make a comparison with the current approach. Although our approach is tested on a specific problem, the idea of this approach can be extended to solve a wider range of nurse scheduling 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 scheduling techniques. Our future work will also investigate the identification and extraction of good features in the solutions. Such features can be used more explicitly to solve the problem and hence reduce the search space. Furthermore, they will help the human scheduler to understand the problem better and actually learn from the search process for future rescheduling.

REFERENCE

Baskaran, G., Bargiela,A. & Qu, R. (2009). Hierarchical Method For Nurse Rostering Based On Granular Pre-Processing of Constraints. 23rd European Conference on Modeling Simulation, ECMS 2009, Madrid, Spain, pp.855-861

Bargiela, A.; and Sterling, M. (1985). Adaptive forecasting of daily water demand-Comparative models for electrical load forecasting. D.W Bunn and E.D. Farmer, eda; John Wiley and Son, New York, N.Y.

Bargiela,A., Pedrycz, W. (2008), Toward a theory of Granular Computing for human-centred information processing, IEEE Transactions on Fuzzy Systems, 16(2), 320-330, doi: 10.1109/TFUZZ.2007.905912

Bargiela,A. &Pedrycz, W. (2003). Granular Computing: An Introduction. Kluwer Academic Publisher, Boston, Dordrecht, London.
Bargiela, A.; Pedrycz, W and Hirota, K. (2004). Granular Prototyping in Fuzzy Clustering, IEEE Transactions on Fuzzy Systems, 12(5):697-709, doi: 10.1109/TFUZZ.2004.834808
Brucker, P., Qu, R., Burke, E.K. &Post, G. (2005). A Decomposition, Construction and Post-processing Approach for a Specific Nurse Rostering Problem. Proceedings of the 2nd Multidisciplinary International Scheduling: Theory and Applications 2005, pp. 397-406. New York, USA.
Burke, E., De Causmaecker, P., VandenBerghe, G. & Van Landeghem, H. (2004b). The State of the Art of Nurse Rostering.*Journal of Schedulin,.*7(6).441-499.
Burke, E. K Qu, R; and He, F. (MISTA 2009). Hybridizing Integer Programming Models with an Adaptive Decomposition Approach for Exam Timetabling Problems. Proceedings of the 4th Multidisciplinary International Scheduling Conference: Theory and Applications, 10-12 Aug 2009, Dublin, Ireland, pages 435-446, 2009.
Burke, E.K. ; Curtois, T.E. ; Post, G. Qu,R. and B. Veltman. (2008). A Hybrid Heuristic Ordering and Variable Neighbourhood Search for the Nurse Rostering Problem. European Journal of Operational Research, 188(2), 330-341
Burke, E.K. ; Li, J. and Qu. R. (2011). A Pareto-Based Search Methodology for Multi-objective NurseScheduling. Annals of OR.
Cheang, B., Li, H., Lim, A. & Rodrigues, B. (2003). Nurse Rostering Problems – a Bibliographic Survey. *EJOR,* 151.447-460.
Goodwin, S.F., Taylor, B.J., Villella, A., Foss, M., Ryner, L.C., Baker, B.S., Hall, J.C. (2000). Aberrant splicing and altered spatial expression patterns in fruitless mutants of Drosophila melanogaster. Genetics 154(2): 725--745.
Hadi, W., Purnomo& Jonathan, F. (2007).Cyclic Preference Scheduling for Nurses Using Branch and Price.Bard *Naval Research Logistic,* Vol. 54 No. 2. 200-220, Published online in Wiley InterScience .
Isken and Hancock, 1990
Li, J. ; Burke, E.K.; Curtois, T. ; Petrovic, S. and R. Qu. (2012). The Falling Tide Algorithm: a New Multi-objective Approach for Complex Workforce Scheduling. Omega – International Journal of Management Science, 40(3): 283-293
Karp, R.M. (1972). Reducibility among combinatorial problems. In R.E. Miller and J.W.Thatcher (eds), *Complexity of Computer Computations*, Plenum Press, New York, pp. 85–103.
Peytchev, E; Bargiela, A and Gessing, R (1996). A Predictive Macroscopic City Traffic Flows Simulation Model . In Proceedings of European Simulation Symposium ESS'96
Sitompul, D. &Randhawa, S.(1990). Nurse Rostering Models: A State-of-the-Art Review. Journal of the Society of Health Systems, 2: 62-72.
Yao, Y.Y. (2006). Three perspectives of granular computing.Journal of Nanchang Institute of Technology, 25, 16-21, 2006.