University of Nottingham

A Domain Transformation Approach for Addressing Staff Scheduling Problems

by

Geetha Baskaran Dip.CS, BSc, MSE, PGCHE

Thesis submitted to The University of Nottingham for the degree of

Doctor of Philosophy

School of Computer Science

2015

Dedicated to my beloved

Mum and Dad (Mr & Mrs Baskaran), I have to praise the Almighty for giving me such loving parents. You were there all the time supporting and inspiring me to the completion of this study. I love both of you.

Prof. Andrzej Bargiela

Uncle Rajakumaran

Sister (Dr Vadsala B.)

Brothers (Dr Saravanan & Hari Karan)

Sister-In-Law (Anushia) & Brother-In-Law (Dr Cho Wai Sum) Little niece (Yasarnee) & little nephew (Sacheen)

My Loving Children, Ghashika and Sireeshtan. You are everything to me. I love both of you.

To His Divine Grace, Lord Krishna, My eternal guide and inspiration, by whose teachings I have learned to love the every-living example. This study is humbly offered at the LOTUS FEET of the ALMIGHTY Lord Krishna….

'Hare Kŗşņa, Hare Kŗşņa, Kŗşņa Kŗşņa Hare Hare
Hare Rāma, Hare Rāma, Rāma Rāma Hare Hare'

Abstract

Staff scheduling is a complex combinatorial optimisation problem concerning allocation of staff to duty rosters in a wide range of industries and settings. This thesis presents a novel approach to solving staff scheduling problems, and in particular nurse scheduling, by simplifying the problem space through information granulation. The complexity of the problem is due to a large solution space and the many constraints that need to be satisfied. Published research indicates that methods based on random searches of the solution space did not produce good-quality results consistently. In this study, we have avoided random searching and proposed a systematic hierarchical method of granulation of the problem domain through pre-processing of constraints. The approach is general and can be applied to a wide range of staff scheduling problems.

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. The schedule construction is done systematically, while assuring its feasibility and minimising the cost of the solution in the reduced problem space of weekly sequences. 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. This two-stage approach to solving the scheduling problem is referred to here as a domain-transformation approach.

The thesis reports computational results on both standard benchmark problems and a specific scheduling problem from Kajang Hospital in Malaysia. The results confirm that the proposed method delivers high-quality results consistently and is computationally efficient.

Publications/Disseminations During PhD Period

Delivered Workshop Presentation

Granular Approach to Personnel Scheduling: Inspiration from the Hirota Lab. In *2009 International Workshop on Advanced Computational Intelligence and Intelligent Informatics* (IWACIII2009), November 7th, Tokyo Institute of Technology, Tokyo, Japan. (EXCELLENT PRESENTATION AWARD: Appendix H)

Delivered Keynote Lecture:

Domain transformation in nurse scheduling. In *2014 International Conference on Artificial Intelligence, Energy and Manufacturing Engineering* (IIE ICAEME2015), Dubai, UAE, 7–8 January 2015 (Appendix J)

Participated in Competition:

Baskaran, G. (2015). Domain transformation in nurse scheduling problem using integer programming. In *2015 National Innovation and Invention Competition through Exhibition* (iCompEx'15), Politeknik Sultan Abdul Halim Mu'adzam Shah, Kedah, 24–26 March. (GOLD MEDAL: Appendix K).

Baskaran, G. (2013). Information granulation to the nurse scheduling problem. In *3MT: Three Minute Thesis*. University of Nottingham, Malaysia, 3 November.

Fully Refereed Papers Published:

Baskaran, G., Bargiela, A. & Qu, R. (2015). Domain transformation using greedy algorithm in nurse scheduling. In *2015 International Conference on Artificial Intelligence, Energy and Manufacturing Engineering (*IIE ICAEME2015). Dubai, UAE, 7–8 Jan, pp. 92–98, Paper ID: E0115039, ISBN: 978–93–84422–05–9. Doi: http://dx.doi.org/10.15242/IIE. E0115039

Baskaran, G., Bargiela, A. & Qu, R. (2014). Integer programming: Using branch-and-bound to solve the nurse scheduling problem. In *2014 International Conference on Artificial Intelligence and Manufacturing Engineering* (IIE ICAIME2014). Dubai, UAE, 25–26 Dec, pp. 211–217, Paper ID: U1214025, ISBN 978–93–84468–11–8.

Baskaran, G., Bargiela, A. & Qu, R. (2014). Seeing the bigger picture: Domain transformation approach to the nurse scheduling problem. *Australian Journal of Basic and Applied Sciences (AENSI Journal), 8*(24) Special: 308–320, ISSN:1991–8178. In: *International Conference on Science, Engineering and Technology*, Jakarta, Indonesia. (BEST PAPER AWARD: Appendix I).

Baskaran, G., Bargiela, A. & Qu, R. (2014). New solution for a benchmark nurse scheduling problem using integer programming. In *International Conference on Information, Management Science and Application* (ICIMSA 2014). Beijing, China, pp. 350–353, ISBN (IEEE Xplore): 978–1-4799–6541–0/14/$31.00

Baskaran, G., Bargiela, A. & Qu, R. (2014). Simulation of scheduling and cost effectiveness of nurses using domain transformation method. In *28th European Conference for Modelling and Simulation (*ECMS 2014). Brescia, Italy, pp. 226–234, doi:10.7148/2014–0226

Baskaran, G., Bargiela A. & Qu, R. (2013). A study of cost effective scheduling of nurses based on the domain transformation method. In *27th European Conference on Modelling and Simulation* (ECMS 2013). Alesund, Norway, doi:10.7148/2013–0309

Baskaran, G., Bargiela, A. & Qu, R. (2009). Hierarchical method for nurse rostering based on granular pre-processing of constraints. In *Proceedings of 23rd European Conference on Modelling and Simulation*. Madrid, Spain, doi:10.7148/2009–0855–0860

Abstracts Published

Baskaran, G., Bargiela, A. & Qu, R. (2013). Integer programming: Domain transformation in nurse scheduling problem: WASET, Melbourne, Australia. In *International Conference on Computer Science and Information Engineering* (ICCSIE 2013), p.1403.

Baskaran, G., Bargiela, A. & Qu, R. (2012). From simplified to detailed solutions to the nurse scheduling problem. In *25th European Conference on Operational Research (*EURO XXV). Vilnius, Latvia, pp. 134–135.

Posters Published

Baskaran, G., Bargiela, A. & Qu, R. (2015). Introduction to domain transformation in nurse scheduling. In *Research for Computer Science School*. University of Nottingham, Malaysia.

Baskaran, G., Bargiela, A. & Qu, R. (2009). Introduction to hierarchical method in nurse scheduling. In *Research for a Better Tomorrow: Impacts in the 21st Century*. University of Nottingham, Malaysia.

Participated in Research Talks

Faculty of Science's research talk, 10 July 2013: *A study of cost effective scheduling of nurses based on the domain transformation method.*

Faculty of Science's research talk, 16 November 2011: *Greedy algorithm in nurse scheduling.*

Postgraduate talk, 27October 2010: *Simplified to detailed solutions to nurse rostering problem.*

Postgraduate talk, 23 June 2010: *Pre-processing (granulation) of constraints.*

Faculty of Science's research talk, 5 August 2009: *Hierarchical method for nurse rostering.*

Acknowledgements

First and foremost, I owe my gratefulness to the Almighty Lord for having guided me and giving me the strength and willpower to complete this study. In my opinion, undertaking a PhD is the second biggest commitment after marriage in a young woman's life. In my case, this study would not have been possible without the help of many people. I would like to take this opportunity to express my deep appreciation to those who have helped me in one way or another in the completion and success of this study. I extend my apologies to anyone I may have failed to mention.

First, I am most grateful to my supervisor Professor Andrzej Bargiela (my principal supervisor) for his guidance throughout the labyrinth of the system, course of my study and most of all for his unstinting words of encouragement, which helped tremendously towards the completion of this dissertation. Without his generous help, this study could never have been materialised. I would also like to thank Dr Rong Qu (my co-supervisor). I have enjoyed working with both of you, and I would like to thank you both for training me to do scientific research. Your support, help and advice have been invaluable to me.

I would also like to extend my appreciation to Kajang Hospital, Malaysia for providing me with an opportunity to conduct my research. Special thanks to the hospital matron Pn. Mashita binti Khalid for her guidance in providing the constraints. I also thank the Medical Research and Ethics Committee who gave approval to conduct this study, and the Institute of Health Behavioural Research for allowing me to conduct this research in the selected hospital and include information on the hospital's constraints in one of my research papers. In particular, I would like to thank En. Nazrul and Cik Sharipazalira for helping my research paper for submission to the European Conference of Modelling and Simulation (ECMS) to obtain approval from the different committees. A special thanks to the Director General of Health of Malaysia for giving me this opportunity to conduct my research in Kajang Hospital and for his permission to publish a conference paper based on the experience. Special thanks also to the nurses who participated in my questionnaire survey for the comparison studies. I also thank everybody that one way or another participated in or assisted me directly or indirectly towards the completion of this study.

My gratitude also goes to Dr Tim Curtois for providing some information for the research paper; Dr K. Harikrishnan for his assistance in understanding the mathematical terms and analysis; and Antem for helping me to understand the simulation programming in C++. Thank you all for your most helpful suggestions and information for my study. I wish to express my heartfelt appreciation to my loving parents, uncle Rajakumaran, brothers and sisters for the encouragement, love and understanding they showed me during my period of study. I hope their patience has paid off. Deep appreciation is also extended to my sincere friends, Professor Mei Fong, Dr Siti Khatijah, and Dr Tissa, for their firm friendship, invaluable ready support, cheerfulness, encouragement and inspiration.

Last but not least, I express my deepest gratitude and love to my children Ghashika Nambiar and Sireeshtan Nambiar, who have given me their happiness, patience and encouragement and shared in my pleasure, sorrow and anxieties. Amma love you both.

Contents

List of Tables

List of Figures

List of Abbreviations

AI	Artificial Intelligence
AL	Annual Leave
CCU	Coronary Care Unit
EL	Emergency Leave
GT	Greedy Technique
HCI	Human Computer Interaction
IP-BB	Integer Programming Branch-and-Bound
NP	Non-deterministic Polynomial-time
NSP	Nurse Scheduling Problem
ORTEC	World’s leaders in optimization software and analytics solution
PC	Personal Computer
PH	Public Holiday
SD	Sleep Day
SSP	Staff Scheduling Problem

Glossary of Terms

Cover requirements represent the number of nurses required at work each day or at specific times (i.e. during shifts) each day. This may also be called **shift or coverage demand**.

Coverage demand. This indicates the required number of nurses with specific qualifications for each shift on a particular day during the planning period.

Cyclical scheduling. A cyclical work schedule establishes that shifts are performed in cyclical (rotating) patterns. A work schedule is specified for a certain planning horizon, and after this period the schedule is repeated. A cyclic schedule may be specified for either all or a subset of the employees of a department.

Days off requests specify that an employee requests not to work on a specific day, or on a specific part of a day. Days off requests are mostly modelled as soft constraints.

Days off scheduling constructs schedules that indicate working days and days off for each employee. The specific shifts performed by employees on working days are determined at a later stage. A days off schedule should satisfy labour legislation, specifying for example the maximum number of consecutive working days. In addition, days off scheduling should ensure that sufficient employees are available to be assigned to shifts.

Float nurses move between units and departments to cover gaps in staff cover due to absences e.g. sick, vacation leave etc.

Hard constraints are rules which must be satisfied for the roster to be feasible. They may also be called **binding constraints** or **imperative planning rules**.

Individual preferences. A nurse may request a rest day, annual leave or specific working shifts for some days of the planning period.

Planning period. This defines the time horizon over which nurses are scheduled. A typical planning period in nurse scheduling is four weeks (28 days).

Qualifications and skills. Personnel can be categorised based on a number of factors, such as qualifications, skills, experience and responsibility (Burke, De Causmaecker, Petrovic, and VandenBerghe, 2004a; Burke, De Causmaecker, and VandenBerghe, 2004b). In some cases, gender, nationality and personality are also considered.

Schedule. A schedule is an ordered list of working shifts and rest days, or an ordered list of shift sequences and rest day periods, or an ordered list of one or more shift patterns. The length of the schedule is the length of this ordered list, which must be the same as the planning period. A schedule can be either cyclic or non-cyclic. If a schedule is non-cyclic, staff members can indicate their preferences for working or being off on specific days.

Scheduling horizon is the time period over which the roster is provided. It may also be called the **planning period**.

Scheduling is the allocation, subject to constraints, of resources to objects placed in space-time, in such a way as to minimise the total cost of the resources used.

Self-scheduling. With self-scheduling, employees propose the work schedule they prefer to work during a given planning horizon. Since these proposed schedules possibly do not match the shift staffing demand as specified by the organisation, the planning problem is to reassign shifts in order to match the specified shift staffing demand. In Chapter 8, we propose a method that supports the planner to create feasible work schedules from the individual work schedules proposed by the employees.

Shift pattern. A fixed length set consisting of working shifts and non-working shifts (Aickelin and Dowsland, 2000). A rest day is considered a non-working shift.

Shift requests specify that an employee requests to work (or not to work) a specific shift on a specific day. Shift requests are mostly modelled as soft constraints. In addition to proposing schedules, some literature lets employees specify ‘importance’ of shift requests, where ‘strong’ shift requests or more important to satisfy.

Shift rostering is concerned with the assignment of employees to shifts. On a planning horizon of typically a couple of weeks or a month, for each day and employee it should be specified which shift the employee performs, such a schedule we refer to as a *work schedule*. Shift rostering is subject to labour legislation specifying constraints on assignment of a single shift, but also on combinations of shifts.

Shift rotation is the situation when an employee works a different shift to the one they worked previously. Depending on whether the start time is earlier or later than before, it is called backward or **forward rotation**.

Shift scheduling defines shifts that should be staffed for a period of, for example, a day, a week or a month. These shifts should respect a set of constraints and are supposed to cover given staffing levels, expressing the required number of employees in each time slot, as efficiently as possible. In addition to the required number of employees, staffing levels may also specify required skill levels. Thus, shift scheduling defines a set of shifts, which are not yet assigned to employees. Shift scheduling only defines the shifts that are required to be staffed.

Shift sequence. A set of shift types on consecutive days, one shift a day (Brucker *et al.*, 2010). Shift sequences often have different lengths.

Shift type. Hospital shifts with a typically well-defined start and end time. Many NSPs are concerned with the three traditional shifts: early (e.g., 7:00–15:00), late (15:00–22:00) and night (22:00–7:00).

Skill category. Each class of nurses has particular minimum qualifications, skills or responsibilities and experience. In hospitals, typical categories include matrons, head nurses, specialised nurses, regular nurses, junior nurses, caretakers and cleaners. These classes may also be referred to as ‘grades’.

Soft constraints are rules which should ideally be satisfied but in order to provide a feasible solution may be broken. They may also be called **non-binding constraints, floppy constraints, preference planning rules** or **aversion costs**. Soft constraints are often given priorities which are relative to each other. If the priorities are assigned using weights then a higher priority constraint may be violated if it means a number of lower priority constraints will be satisfied. 2 Literature Review 16

Split weekend is the situation where an employee works on only one day of the weekend (i.e. Saturday or Sunday). A **complete weekend** is the opposite (i.e. the employee works on neither or both days of the weekend). A **stand alone shift** or stand alone day is an *off-on-off* work pattern. It may also be called an **isolated work day**. A **work pattern** is an individual's schedule over a planning period. That is, the days they have on and off and possibly also the shifts they have on the days on. Predefined patterns may also be called **stints**.

Weekend shift rostering addresses the assignment of weekend shifts to employees. The weekday shifts are assigned to employees in a later stage. Also weekend shift rostering has to comply with labour legislation specifying for example constraints on the number of consecutive working weekends.

Work regulations. It is common for each nurse to have a personalised agreement and job description, specifying whether the nurse is full-time or part-time, permanent or temporary and whether they can do shift work.

Chapter 1: Introduction

This thesis focuses on a novel transformation approach to addressing staff scheduling problems (SSP), in particular nurse scheduling, involving different real-word problems. These problems are transformed into a more structured domain, in which a new representation of information through pre-processing (called 'patterns') is introduced. The study also implements several techniques, focusing on a general algorithm that enhances the solutions generated by the proposed approach. This chapter presents the introduction to this study, followed by the problem statement, research questions and scope of this study. Finally, the chapter presents the roadmap for the rest of the thesis.

1.1 Introduction

'Scheduling' is defined by Cambridge Dictionaries Online (see http://dictionary.cambridge.org) as 'the job or activity of planning the times at which particular tasks will be done or events will happen'. Scheduling problems are multi-faceted, meaning it is vital to understand the development of the different aspects involved in constructing a good schedule. The term 'scheduling' has several different meanings across the literature. For the purposes of this study, we use the definition of Wren (1996):

> **Scheduling** is the allocation, subject to constraints, of resources to objects placed in space-time, in such a way as to minimise the total cost of the resources used. (p. 53)

Scheduling deals with the allocation of resources to tasks over given periods to achieve certain objectives while meeting various constraints (Barker, 1974). The components involved in scheduling are characterised by complicated interrelationships. Due to this, the preparation of a schedule can become complex and expensive in terms of time and resources.

Problems with staff scheduling are common across a wide range of industries, including in manufacturing, service industries, resource allocation, transportation, project management and distribution settings. At its base, these problems are concerned with scheduling a workforce to meet demand for manpower that varies within a day and/or within a week. Dealing with SSPs means determining which staff should cover which shifts so that the demand for manpower is met at all times, taking into account organisational and legal rules. The major challenge is providing reasonable labour costs and customer satisfaction while meeting this varying demand.

One of the best-known areas where staff scheduling is a concern is the healthcare industry. In order to meet strict quality standards in patient care, the objective of personnel rostering in healthcare is to match the number of skilled people working at given time intervals to the demand for certain nurse services. Timetables are constrained by governmental and hospital rules, but also by personal preferences and work regulations. A key SSP in the healthcare industry is the scheduling of working hours for nurses, known as ‘nurse rostering’ or ‘nurse scheduling’ (Burke, 2004). This thesis focuses specifically on this nurse scheduling problem (NSP).

NSPs are well-known, having been studied by personnel managers, operations researchers and computer scientists for more than 45 years. They are unique compared to other SSPs mainly because of a presence of a range of different staff requirements on different days and shifts (Li and Aickelin, 2006). The wide fluctuation in demand that can occur throughout the day and from one day to the next—subject to some of the most difficult and specific constraints—is what makes NSPs so challenging and difficult. Maintaining an acceptable service level according to nurses’ preferences with minimal coverage requirements is also considered of extreme importance. In addition, hospital personnel rostering is a very complex scheduling domain because, unlike many other organisations, healthcare institutions operate 24 hours a day, every single day of the year (Li and Aickelin, 2006).

NSPs are not only one of the more commonly occurring problems in healthcare (the UK's NHS alone currently employs approximately 655,000 nurses, see Christie & Co., 2015) but also one of the most complex. This high complexity is due to a number of factors, some of which (but rarely all) may be found in other SSPs. These factors include:

- As stated, hospitals operate for 24 hours a day, seven days a week. This introduces a number of legal constraints and working preferences relating to night shifts, minimum rest times, working on weekends and national holidays, among others.
- The workforce consists of nurses with varying skills and grades, which need to be considered when constructing rosters.
- There are a variety of shifts. Even the more basic NSPs usually involve a minimum of three shift types (e.g., early, late and night). More frequently, there are a number of other shift types to assign, each with varying durations and associated constraints.
- There are a large number of employees.
- Cover requirements may not be uniform but vary from day to day.
- There are long planning horizons. They can range up to 12 weeks or even a year in some instances.
- There are many, often conflicting, constraints and objectives. For example, constraints or objectives relating to:
    - Cover requirements.
    - Day on/off and shift on/off requests.
    - Minimum and maximum length stretches of days on, off, or specific shifts.
    - Minimum and maximum hours and/or shifts worked during certain periods.
    - Shift rotations.
    - Desirable and undesirable work patterns.
    - Minimum and maximum numbers of specific shift types (possibly during certain periods).
    - Minimum and maximum ratios of shift types worked.
    - Tutorship requirements or the opposite, meaning ensuring certain employees do not work at the same time.

These features make NSPs not only hard to solve but also difficult to model. However, the effort required is worthwhile when high-quality rosters are produced.

At present, many nurses prepare their own schedules based on availability and contractual agreements, and make adjustments through consensus in the case of conflicts. If NSPs can be solved efficiently, this will have an immense impact on nurses' working environment, which in turn can strongly improve the quality of healthcare. Nurses preparing their own schedules can also improve nurses' satisfaction level and facilitate the recruiting of capable personnel.

This contrasts with the situation when schedules are created individually for each hospital unit by the head nurse (Berrada, Ferlandand and Michelon, 1996). The fundamental intention of scheduling in general is to ensure that the number of staff members is ample to cover the nursing requirements and individual nurse duties (Glass and Knight, 2010). Most hospital wards have head nurses or nurse managers who are responsible for manually constructing nurse schedules. Head nurses usually spend a substantial amount of time developing schedules, especially when faced with many requests for consideration of changes from their staffs. Additional time spent in handling ad-hoc changes to current duty schedules (Cheang, Li, Lim and Rodrigues, 2003). This makes manual procedures time consuming and inefficient. Consequently, more feasible approaches have been developed (Purnomoand Bard, 2007), by having significant benefits in terms of saving administrative staff time and generally improving the quality of the schedules produced.

NSPs are very complex real-world scheduling problems (Karp, 1972) and belong to a class of non-deterministic polynomial-time (NP)-hard problems. Generating good work schedules can greatly influence nurses' working conditions, which is strongly related to quality of healthcare. Nurse schedules are designed to ensure a reasonable (fair) and efficient schedule for nurses. The scheduling problem involves allocating suitably qualified staff to meet a time-dependent demand for different services. The most general form of a NSP 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, Qu, Burke and Post, 2005). Usually, there are a number of different constraints on the problem that must be satisfied, which can be split into hard and soft constraints, depending on whether they are essential or merely desirable, respectively. Problems with both hard and soft constraints essentially have two separate objectives. Firstly, all the hard constraints must be satisfied for the solution to be feasible and secondly, the soft constraints must be satisfied as far as possible. Where there are several soft constraints, this raises a further issue of which of these constraints are the most important and, from an algorithmic point of view, how to deal with the problem of setting appropriate weights. The complexities and challenges of NSPs arise from the fact that a large variety of constraints exist, some of which contradict each other. Often, different constraints will be in direct conflict with one another, and so a trade-off is necessary to find optimal solutions. Additional difficulties occur when the satisfaction of the hard constraints is non-trivial, which raises a further issue of how much bias can be given to the feasibility aspects of the problem without adversely affecting the final solution quality. If the bias is too much in favour of the soft constraints, resulting solutions will not be feasible; however, by focusing solely on the satisfaction of the feasibility constraints, optimal solutions may be missed. With very small problems, all possible solutions to the problem may be enumerated and finding an optimal solution is reduced to the task of merely choosing the solution with the best cost. With larger problems, this is not feasible. For NP-hard problems, the amount of time required to solve the model grows exponentially with problem size.

It is important to have careful balance between the different constraints in these types of large NP-hard problems, involving hierarchies of objectives and conflicting constraints, and the resulting trade- offs necessary to produce high-quality solutions. In this thesis, 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 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. In particular, attention is given to the solution of two real-world problems: ORTEC 01 and Kajang Hospital. For both problems, there are conflicting hard and soft constraints that must be successfully balanced to produce optimal solutions. For each available nurse, a single cost relating to the soft constraint is known for each complete set of potential shifts, or 'shift patterns'. These costs are based on the nurses' individual preferences; by keeping these costs low, staff satisfaction is increased.

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), and Cheang *et al.* (2003). Burke *et al.* (2004b) present a more extensive and excellent survey that mainly copes with nurse scheduling. These range from traditional mathematical programming methods and linear programming to heuristic methods that guarantee to find an optimal solution and 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-heuristic methods is that they can neither provably produce optimal solutions nor provably reduce the search space. They also tend to lack well-defined stopping criteria. Moreover, as most NSPs are highly constrained, 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 situations Burke *et al. (*2009).

This thesis is concerned with creating solutions within a constructive information granulation called domain transformation approach. The real-world benchmark and the newly introduced real-world NSP chosen are suitable for applying such a method. Further, the method is general enough to be used in other SSP applications. The techniques applied to execute the general algorithm (information granulation and pattern construction) in this thesis are constructive and deterministic, and are our main achievements.

1.2 Problem Statement and Scope

The problem statement of this thesis is: To what extent can information granulation be used to solve NSPs?

This study proposes solutions that can be applied to solve SSPs in healthcare, specifically NSPs. The research presented in this thesis focuses on developing a novel approach to information granulation for NSPs. It also seeks to develop and implement a general algorithm of generating schedules for nurses under the information granulation approach. The aim is to achieve a feasible solution with minimum cost, flexibility in staff scheduling and continuity in the scheduling process. The aim is also to ensure a balanced and equitable schedule between all employees, in terms of workload, and also to respect a predefined sequence of work shifts and days off, either following work rules or staff preferences.

The main objectives of this thesis relate to the exploitation of a high-quality constructive approach, transforming the original problem domain to a smaller domain. In the smaller domain, some shifts have the same set of constraints, so they may be considered as the same type; further, this domain involves fewer patterns to be scheduled. These observations mean that fewer shift types need to be considered, thus simplifying the problem. With conversion to a larger domain, the search space expands. This indicates that all schedules that can be derived directly in larger domain patterns are covered. This information granulation is called ‘domain transformation’. The domain transformation approach can produce high-quality solutions within a short implementation and development time. The domain transformation method is easily reproduced, in contrast to some meta-heuristic methods that tend to use extensive problem-specific information and random decision-making to arrive at solutions. Domain transformation can also be performed on any number of months or weeks of schedule. The schedules produced have continuity between the months. All the constraints are considered in order to generate the monthly schedule. In this thesis, the proposed solutions will be assessed using a real-world benchmark NSP (available for download from http://www.cs.nott.ac.uk/~tec/NRP/) and new data collected from Kajang Hospital.

1.3 Research Questions

The research questions of this thesis, following from the problem statement described above are:

1. What is a suitable design for information granulation to solve NSPs?
2. How can we achieve feasible nursing schedules?
3. To what extent can information granulation solve NSPs in the real world?
4. To what extent can the method used be generalised?
5. Can nurses' preferences be adopted into this new NSP approach?
6. Is continuity from one month to another maintained with feasibility using the proposed method?

1.4 Thesis Overview

This chapter has discussed the research problem and objectives and the contributions of the research. The remainder of this thesis is structured as follows:

- Chapter 2 contains the literature review. It outlines how previous literature has considered the complexity of NSPs. It also discusses the nurse scheduling classifications and techniques used in addressing NSPs, and reviews various surveys from the scheduling literature.
- Chapter 3 presents the real world benchmark datasets used to evaluate the novel algorithm. The chapter also presents background of important terminology related to NSPs.

- Chapter 4 presents the methodologies employed in the subsequent chapters. It introduces domain transformation as a novel state-of-the-art approach in addressing NSPs. The objective of domain transformation is to minimise cost and meet nurse demands by satisfying the constraints. The chapter also discusses the literature related to domain transformation.
- Chapter 5 presents a detailed result analysis, and analyses the effect of various instance parameters on the domain transformation results. To assess the performance of the domain transformation approach, experimental results from 18 real-world benchmark datasets and a new real-world dataset from Kajang Hospital, Malaysia, are presented. This chapter also describes the implementation of the method, with three different techniques used to solve NSPs. Zero-cost patterns using these techniques based on the main algorithm are explored and proposed as possibly enhancing final solution quality.
- Chapter 6 summarises the findings of this thesis and suggests possibilities for future work arising from the study.

Chapter 2: Literature Review

This chapter presents a literature review focusing on state-of-the-art developments in solving NSPs. The purpose is to provide a foundational understanding of nurse scheduling. To achieve this, the different approaches and methods used to solve NSPs are explored, and the various objectives and constraints applicable in NSPs are identified. Additionally, we also summarise the algorithm techniques proposed in this area by providing timeline surveying the past 15 years. Finally, we derive the gap between the way nurse scheduling is approached in theory and in practice.

2.1 Background of Scheduling and Staff Scheduling

Scheduling has been widely researched for decades. It covers a large variety of problems. Most of these problems are computationally hard to solve (in the sense of being NP-hard) and need complex algorithms (Pinedoe, 2008). Difficulty lies also in the modelling of the problems, and the mapping between high-level, declarative models and low-level, procedural search techniques (Michael, 2008). The goal of scheduling can be either satisfaction or optimisation of objectives (Draper, Jonsson, Clements and Joslin, 1999). Each objective may have a certain priority level, an earliest possible starting time and a due date. The objectives can also take many forms. One objective may be the minimisation of the completion time of the last task, while another may be the minimisation of the number of tasks completed after their respective due dates, or the minimisation of the cost associated with the schedule.

Beginning in the 1950s, attempts were made to solve scheduling problems using computers; however, at that time, computers had inadequate power for some formulations (Wren, 1996). More recently, it has been possible to solve much more complex scheduling problems using computers. Owing to the importance of scheduling problems in real life, a number of studies have sought to solve these problems in novel ways. In scheduling problems, time plays a central role. The schedule time horizon is the period for which the schedule is constructed. The schedule time horizon imposes constraints on the time range of an individual job in terms of its start and end time.

Staff scheduling is becoming a critical concern in service organisations such as emergency services and higher education, healthcare, hospitality and transportation systems. Scheduling in service organisations is different from that of manufacturing systems (Aggarwal, 1982). Some of the major differences are that the product of service systems cannot be placed into an inventory and that the customer receives the service directly from the server. While the primary objective of the manufacturing system is to minimise the total cost, service systems deal with conflicting objectives, such as minimising total cost and maximising staff satisfaction with their schedules.

Blöchliger (2004) introduced a tutorial to SSP using a hospital example. In this study, the focus was on how the scheduling problem could be analysed and modelled using various constraints, objectives and models. While Blöchliger's study did not give a solution to the problem, the modelling did suggest a number of algorithms for use in future studies, such as genetic algorithms, simulated annealing and tabu search. Ernst, Jiang, Krishnamoorthy, Owens, and Sier (2004) classified 16 different scheduling models based on rostering processes or the determination of staff requirements, such as task-based demand, shift-based demand, days- off scheduling, shift scheduling and task assignment. The authors also addressed 15 different application areas for scheduling problems, including airlines, buses, nurse scheduling, venue management, financial services and manufacturing. Staff scheduling did not get much attention in the area of artificial intelligence (AI) until the 1980s when Fox (1983) began their work on constraint-directed scheduling systems for the job–shop scheduling problem. A variety of service delivery settings have been studied in relation to staff scheduling, such as airline crews (Schindler and Semmel, 1993), hotel reservation personnel (Holloran and Byrn, 1986), telephone operators (Henderson and Berry, 1976), factory workers (Berman *et al.,* 1997), police officers (Taylor and Huxley, 1989) and security guards (Engku, 2001) and many more.

A large section of the studied SSP come from healthcare organisations such as hospitals and clinics and requires the scheduling of nurses. Sitompul (1992) notes that nurse scheduling shares much in common with other SSPs. All these problems require staff to be on duty 24 hours a day seven days a week, with fluctuating daily demand for services and fixed regulations as to acceptable work patterns. However, a NSP is further recognised by the following characteristics:

- *Staffing levels:* There can be four or more grades of nursing staff, each with a different skill level. Legal controls limit the tasks each grade of nurse can perform. Consequently, each shift can have a minimum staffing requirement for each grade of nurse.
- *Nurse preferences:* Due to the importance of maintaining nurse satisfaction and reducing turnover, schedules should reflect a nurse's preferences for shift patterns and days off.
- *Flexible scheduling:* To meet changing nurse requests for particular days off, a schedule should not be fixed or imposed. This means a new schedule needs to be calculated in each scheduling period, rather than rotating duties within an existing schedule.

The main feature that appears from these points is that nurse scheduling has multiple objectives (Ozkarahan and Bailey, 1988). Other sophisticated problems, such as the aircrew-scheduling problem, usually have the single objective of minimising costs after the basic constraints have been met (Graves, McBride, Gershkoff and Mahidhara, 1993; Hoffman and Padberg, 1993).

Chapter 1 discussed how NSPs are significant due to their importance, scientific challenges and complexity. As this thesis is primarily concerned with the nurse rostering problem, this is where we will focus most of our attention in this chapter.

2.2 Nurse Scheduling Problems

NSPs are well-known scheduling problems that arise in hospital wards globally. Although the details of NSPs vary in different countries, the essence of the problem is to allocate suitable nurses to shifts to meet service demand across different planning periods, while attempting to satisfy workplace regulations, nurses' preferences and other constraints, and minimising costs (Berrada, Ferland, and Michelon, 1996; Ernst *et al.*, 2004). In NSPs, the most used types of coverage are the night, evening and day shifts, although several other shift types can be defined.

Part of the scheduling problem is to determine the times at which shifts are allocated to each member of the nursing staff (Rowland and Rowland, 1997). Nurses' wellbeing and job satisfaction are affected by irregular shift work (Burke, De Causmaecker, Vanden Berghe, and Van Landeghem, 2004c). Properly scheduling the nursing staffs has a great impact on the quality of healthcare (Oldenkamp, 1996), the recruitment of nursing personnel, the development of a nursing budgets and various other functions of the nursing service. The process of constructing nurse schedules consists of determining the number of nurses, their skills and qualifications, nurses' preferences, workplace regulations or policies, service demand, the layout of the work timetable, the constraints and other criteria relevant to the specific hospital setting. Additionally, nurses have the right to request rest days and which shift pattern they would like to work. All nurses are different and should be treated as such when the schedules are created.

NSPs are known as NP-hard (Osogami and Imai, 2000). This complex problem cannot realistically be solved to optimality. This explains the scientific community's degree of interest in this research area over the last 45 years. A general overview can be found in (Hung, 1990) and (Sitompul and Radhawa, 1990). Producing good-quality nurse schedules greatly impacts on the quality of healthcare service (Cheang *et al.*, 2003; Landa-Silva and Le, 2008). Many hospitals use software to support the construction of nurse schedules; however, in many other cases, scheduling is still done manually. For problems of considerable size, the non-automated construction of nurse schedules is time consuming, difficult and prone to mistakes. As Burke *et al.* (2004c, p. 24) note, *'the automatic generation of high quality nurse schedules can lead to improvements in hospital resource efficiency, staff and patient safety, staff and patient satisfaction and administrative workload'.*

The following section provides the basic understanding in nurse scheduling and discusses work regulations and the other constraints in nurse scheduling commonly found mentioned in the literature.

2.2.1 Cyclic v. non-cyclic scheduling

There are a number of different lines of work models. Schedules can be cyclic or non-cyclic. In a cyclic model, all nurses of the same class perform exactly the same line of work but with different starting times for the first shift or duty. This schedule type is most applicable for situations with repeating demand patterns. In a cyclic schedule, a number of shifts are always grouped together and nurses rotate from one pattern to another. This involves generating a fixed roster that can satisfy staff requirements, without considering individual nurse requests. Nurses are then assigned schedules within the schedule. The basic problem with cyclic scheduling is its lack of flexibility (Smith and Wiggins, 1977): the schedule remains the same in each successive scheduling period. Nurses may also be unable to obtain their preferred holiday periods. Using this model, one schedule can be used for several months or even years. Due to this infrequent calculation, it may be more cost-effective to use human expertise to generate cyclic schedules, rather than to develop an automated solution (Megeath, 1978).

Meanwhile, in non-cyclic scheduling, shifts are considered independent, every shift is assigned individually and the schedule is reformulated before each scheduling period, with each schedule being matched to a particular nurse. This is done to accommodate individual nurses' preferences and allow for fluctuations in the number and type of staff assigned to a ward. This type of scheduling will usually result in longer work stretches and a more unbalanced distribution of shift types than would otherwise be necessary. More attention has been paid in the literature to the computerised generation of non-cyclic rosters than to the generation of cyclic rosters. This is due to the greater complexity of non-cyclic rosters.

2.2.2 Complexity of nurse scheduling

The nurse scheduling environment provides a complex problem because of the large number of conflicting constraints that must be balanced to create a schedule. These constraints cannot be prioritised because they are not independent or stable. This forces unique solutions to the scheduling problem (Jelinek and Kavois, 1992). According to Bard and Purnomo (2005) and Chiaramonte (2008), job dissatisfaction may affect turnover and absenteeism rates, further complicating the work of creating desirable schedules. The importance of mitigating scheduling problems for nurses has resulted in the emergence of various approaches and techniques for solving NSPs.

Within the hospital environment, staff members are organised into groups of nurses in a ward. Usually, each ward performs a set of fixed activities at a settled location (Burke and Newall, 2004), for the most part with a permanent team of nurses.

The nurse schedule configuration should fulfil both an agreed list of requirements as well as demand coverage. Normally, efforts are also made to minimise salary costs and satisfy nurse preferences as far as possible. This agreed list of requirements are the constraints that help to define acceptable schedules for individual nurses in terms of seniority, workload, holidays, weekends off, consecutive assignments and rotations (Jaumard, Semet and Vovor, 1998). During each day of a planning horizon, several shifts can be planned.

Due to the large number of possible schedules and the change of the cost with different combinations of shifts, the optimisation of the overall schedule by the modification of individual shifts and/or various groups of shifts is considered an NP-hard problem (Celia and Roger, 2010). However, this classification is predicated on the assumption of the deployment of scheduling algorithms that explore the solution space directly.

Miller *et al.* (1976) defined π_i as *'the set of feasible patterns for nurse i'*, and the solution space as the Cartesian product of all feasibility regions $\pi_1, \pi_2,$

$\pi_{3...}\ \pi_n$. For a single employee with four shifts worked over a period of 28 days, a single feasibility region contains:

$$\binom{n}{k} = \frac{n!}{x!\,(n-k)!} = \binom{28}{4} = \frac{28!}{4!\,(28-4)!} = 491\,400\ schedules$$

However, according to the author, in practice, the number of available solutions is smaller. The majority of solutions can be eliminated by applying constraints associated with the problem; for instance, the demand allocated for each nurse during each shift, or the legal limits to the number of consecutive shifts a nurse can work without a rest day. The total number of permutations is therefore also lower. If feasibility regions are defined, each region can be defined as an upper bound for the complexity of the problem. Even given the application of such constraints, realistically NSPs are still too complex to be solved by an exhaustive search methodology.

NSPs present a high degree of diversity in addition to complexity. de Causmaecker and Vanden Berghe (2010) have initiated the development of a general framework for categorising nurse rostering problems. The categorisation of the problems will help researchers to study the complexity and hardness of the problem instances and the efficiency of the corresponding algorithms. Categorisation is according to their properties such as the personnel environment, work characteristics and optimisation objective.

Vanhoucke and Maenhout (2009) have developed 10 complexity indicators in four groups for NSPs. The indicators are based on problem properties such as problem size, preferences of the nurses, coverage constraints and time-related constraints, which restrict the individual schedules of the nurses. The indicators can be used to predict the performance of exact and heuristic methods on a given problem instance. Moreover, they can assist to select the most promising algorithm from a set of algorithms to solve a given problem instance. Messelis *et al.* (2010) utilised a number of structural and formal features of NSPs to predict algorithm behaviour on a particular problem instance. The structural problem features were size-dependent features, coverage constraint structure, workforce structure, contract and request- related features. The resulting approach can be utilised in a system in which the solution quality of a problem instance needs to be calculated quickly without finding the actual solution, such as agent negotiation systems between hospital wards.

The constraints and multiple objectives of NSPs make them unique within the domain of SSP. The situation is further complicated by the existence of different policies and circumstances for different hospitals and wards. This has prevented existing solutions to the problem from being widely applied (Sitompul, 1992). In Sections 3.3 and 3.4, the existing approaches to nurse scheduling are considered in detail.

2.2.3 Constraints in nurse scheduling

Nurse scheduling is defined as the creation of a periodic (weekly, fortnightly or monthly) schedule for nursing staff of one or several wards, subject to constraints such as legal regulations, personnel policies, nurses' preferences and other hospital-specific requirements. Bechtold *et al.* (1991) list the constraints of NSPs as: (1) labour requirements (2) labour schedule duration (3) labour schedule start time (4) meal and rest breaks (5) consecutive/non-consecutive days off (6) labour productivity (7) number of employees (8) equipment capacity (9) labour availability (10) labour location site (11) hours per day of operation (12) schedule planning horizon, or (13) some combination of the above. While, Miller, William and Gustave (1976) groups constraints into feasibility set and non-binding constraints (also known as hard and soft constraints, respectively), which vary with legal regulations and individual preferences. In the scheduling literature, constraints can be classified into two categories; hard constraints and soft constraints (Qu *et al.,* 2009a).

Hard constraints are those that must be satisfied to obtain feasible solutions. They may include legal and hospital requirements enforced on the schedule. The legal requirements, either fixed or contract-based, usually limit the maximum time a nurse can work and describe combinations of shifts that can occur in the schedule. This generates a very complex set of constraints. Conversely, hospital requirements relate to the coverage needed to maintain an appropriate level of care quality.

When the hard constraints are satisfied, the generated schedule will be usable from the perspective of the law and the hospital. Much work remains to be done on workforce requirements, which are often incorporated into soft constraints.

Soft constraints are typically time-related. Their satisfaction is desirable but not compulsory, and thus they can be violated. Soft constraints are diverse and serve to encourage roster quality by satisfying the workforce, in turn helping to meet demands for high-quality care. As an incentive not to violate the more important soft constraints, they will bring high costs or penalties when violated. Soft constraints might include requests for rest days, shift type preferences or requests for longer free time blocks between worked shifts. However, one implicit soft constraint remains hidden if there is no nurse or a bad schedule, and the schedule can be improved with the exception of the head nurse.

The goal is always to schedule resources to meet the hard constraints while aiming at a high-quality result with respect to soft constraints. These two categories will not affect the three sets of constraints as defined by Cheang *et al.* (2003); that is, *coverage*, *work and contract regulations* and *nurse preferences*. Any constraint in any of these three sets can be considered a hard or soft constraint. The first constraint category, *coverage*, requires a set number of nurses of each skill category to be scheduled at the required period. This ensures an adequate level of staff to meet patient demands, which define the required number of nurses during the planning period. The *work and contract regulations* constraints ensure that shifts assigned to nurses respect the regulations outlined in their contracts and any other regulations that apply to all staff. The main types of work regulation constraints are:

- *Working hours*: the maximum/minimum hours that a nurse can work over a period (e.g., a week or a fortnight).
- *Consecutive working shifts/days*: the maximum/minimum number of shifts/days that a nurse can work in a row. Maximum consecutive working shifts/days allow regular breaks in a nurse’s schedule.

- *Shift patterns*: illegal and/or undesired patterns of shift types.
- *Shift assignments*: the maximum/minimum number of shifts that a nurse can work in the planning period.
- *Working weekends*: constraints related to weekend work. For example, the maximum/minimum number of weekends that nurses can work during the planning period, or whether nurses can work both days of a weekend.
- *Break periods*: the maximum/minimum length of breaks between consecutive working shift patterns.

The third category of constraints includes all *nurse preferences*. According to Ernst *et al.* (2004), the tendency in the modern workplace is to focus on individuals rather than on teams. Hence, personnel schedules should cater to individual preferences. This is mainly true in nurse scheduling because it is common that each nurse indicates their preferences and gets involved in the scheduling process. Complying with these preferences as much as possible may assist in increasing nurse satisfaction levels. Commonly occurring constraints listed below:

1. Nurses workload (minimum/maximum).
2. Consecutive same working shift (minimum/maximum/exact number).
3. Consecutive working shift/days (minimum/maximum/exact number).
4. Nurse skill levels and categories.
5. Nurses' preferences or requirements.
6. Nurses free days (minimum/maximum/consecutive free days).
7. Free time between working shifts (minimum).
8. Shift type(s) assignments (maximum shift type, requirements for each shift types).
9. Holidays and vacations (predictable), e.g., bank holiday, annual leave.
10. Working weekend, e.g., complete weekend.
11. Constraints among groups/types of nurses, e.g., nurses not allowed to work together or nurses who must work together.
12. Shift patterns, Historical record, e.g., previous assignments.
13. Other requirements in a shorter or longer time period other than the planning time period, e.g., every day in a shift must be assigned.
14. Constraints among shifts, e.g., shifts cannot be assigned to a person at the same time.
15. Requirements of (different types of) nurses or staff demand for any shift (minimum/maximum/exact number).

2.2.4 Objective functions

Objective functions are calculated to measure the quality of schedules. Depending on the model used to represent the schedule, different approaches can be engaged to evaluate objective functions. It is common that objective functions are related to the constraints in the model and hence can measure the violations of the constraints or the cost of constraint violation. The objective function criteria that have been used or suggested in past solutions include: (1) total labour hours scheduled, (2) total number of employees, (3) labour costs, (4) unscheduled labour costs, (5) customer service, (6) over-staffing, (7) understaffing, (8) number of schedules with consecutive days off, (9) number of different work schedules utilised, or (10) some combination of the above (Bechtold, Brusco and Showalter, 1991). These criteria are not limited to the nurse scheduling environment alone, and some are not appropriate for certain NSPs where part-time personnel are not allowed.

Bechtold *et al.* (1991) mentioned that total labour hours scheduled is the performance criteria most frequently used by scheduling researchers. In Dowsland's (1998) nurse scheduling solution, individual preferences and requests for days off were taken into consideration when formulating the objective function. The lower the cost obtained, the better is the quality of the schedule. Of course, due to the often conflicting and large number of constraints, there is rarely a perfect roster with penalty zero Burke *et al.* (2013).

2.3 Summary of Nurse Scheduling Approaches and Techniques

Modelling nurse scheduling is not a new idea. Until the 1960s, scheduling tools consisted only of graphical devices such as the Gantt Chart. Howell (1966) outlined the procedure necessary to develop a cyclical schedule accommodating the work patterns and individual preferences of nurses. In the early 1970s, scheduling systems began to be based on heuristic models (Isken and Hancock, 1991; Smith, Wiggins and Bird, 1979). These models represented an improvement because they could theoretically take into account all scheduling constraints in solving the problem. Maier-Rothe and Wolfe (1973) developed a cyclical scheduling procedure that assigned different types of nurses to each unit based on average patient care requirements, hospital personnel policies and nursing staff preferences. Howell (1966) and Frances (1966) laid down some basic principles for manual cyclic rostering. Rosenbloom and Goertzen (1987) developed a computer algorithm for the generation of cyclic rosters. Warner (1976) described an early approach combining manual planning and integer programming. The author defined five criteria for the scheduling part of the problem: coverage, quality, stability, flexibility and cost. Arthur and Ravindran (1981) formulated NSPs as a goal programming problem—an approach that was taken up by hospital schedulers for building real-life schedules. Their research has been described as innovative because the scheduler makes different changes in the final solution and integrates AI techniques into the interface of a decision support system to facilitate manual changes.

Turning to survey articles, in 1976, Fries (1976) compiled an early bibliography of applications of operations research methods in healthcare systems. Hung (1995) collected 128 articles on nurse scheduling, from the 1960s to 1994, and presented an overview from a variety of research domains, where most papers study the experience of new work- week arrangements. Ernst *et al.* (2004) present a very comprehensive overview of the literature on staff scheduling and rostering. The authors divided their paper into three main parts: definitions, classification of personnel-scheduling problems and a classification of the literature into application areas and solution methods, with comments on applicability. They also pointed out some areas for improvement, including greater consideration of individual preferences and the generalising of the scheduling algorithms, models and methods. De Vries (1987) developed a 'management control framework' to balance supply and demand, replacing strict balance for nursing care. The author believed the flexibility of setting parameters separately per ward, and according to the expert knowledge on the floor that is used for forecasting the workload, have mainly seen the satisfactory performance of the framework. Silvestro and

Silvestro (2000) discuss the results of a survey of nurse scheduling practices in the UK National Health Service. The authors define three different scheduling policies: departmental rostering, team rostering and self-rostering. They conclude that the benefits and limitations of these policies depend on the operational context, such as ward size, predictability of demand, demand variability and complexity of nurses' skill mix.

Many different techniques for solving NSPs have been proposed in the literature. One of the first techniques used for solving NSP (dated back to the 1970s) is mathematical programming (Abernathy *et al.*, 1973; Trivedi and Warner, 1976; Miller, 1976; Warner, 1972, 1976). Traditional techniques from linear programming and integer programming have been employed to solve NSPs (see, for example, Beasley, 1996; Jaumard *et al.*, 1998; Miller, 1976; Warner, 1972, 1976). Integer-programming techniques designed to find optimal solutions to linear programming problems that have integer variable restrictions. However, integer-programming algorithms are computationally expensive, and models with large numbers of variables soon become time consuming to solve (Chow and Hui, 1993).

Warner (1976) uses a multiple-choice programming algorithm based on the work of Healy (1964) to solve a nurse rostering problem in the University of Michigan Hospital. Following on from Warner's work, Kostreva,

Lescyski and Passini (1978) developed a mixed-integer programming formulation of NSPs. Then, using a suitable computational technique, the value of the objective function is maximised or minimised (Papadimitriou and Steiglitz, 1982). Bailey (1985) developed a cyclical scheduling model with integer programmingThe branch-and-bound algorithm is a classic method to solve the integer program (Wolsey and Nemhauser, 1999; Thorton and Sattar, 1997). Maenhout and Vanhoucke (2010) present an exact branch-and-price algorithm for NSPs that incorporates different branching strategies. Balakrishnan and Wong (1990) used network model to solve the workforce scheduling problem. The decomposition technique involves intelligently breaking larger problems into smaller ones that are easier to manage. By aggregating these subgroups, all the hard constraints must be satisfied. Dealing with each sub- problem in turn has been shown to work well in nurse rostering (Aickelin and Dowsland, 2000) and other scheduling problems (Burke *et al.,* 2004). A constructive based on successive resolution was proposed (Ademir, Dario, Everton, and Wesley, 2011) in which the algorithm first constructs an initial solution by solving successive bottleneck assignment problems. Subsequently, in the second phase, two improvement procedures based on reassignment steps are applied. The basic principles of the method used by Warner 1976 illustrated in Figure 2.1. Firstly, a set of feasible schedules generated for each nurse. These schedule sets combined until the best staffing levels for the complete roster found. In the second phase, the algorithm calculated the best combination of schedules according to the nurses' preferences. In both phases, the multiple-choice algorithm used a linear programming method to arrive at an initial solution and then searched for the best integer solution.

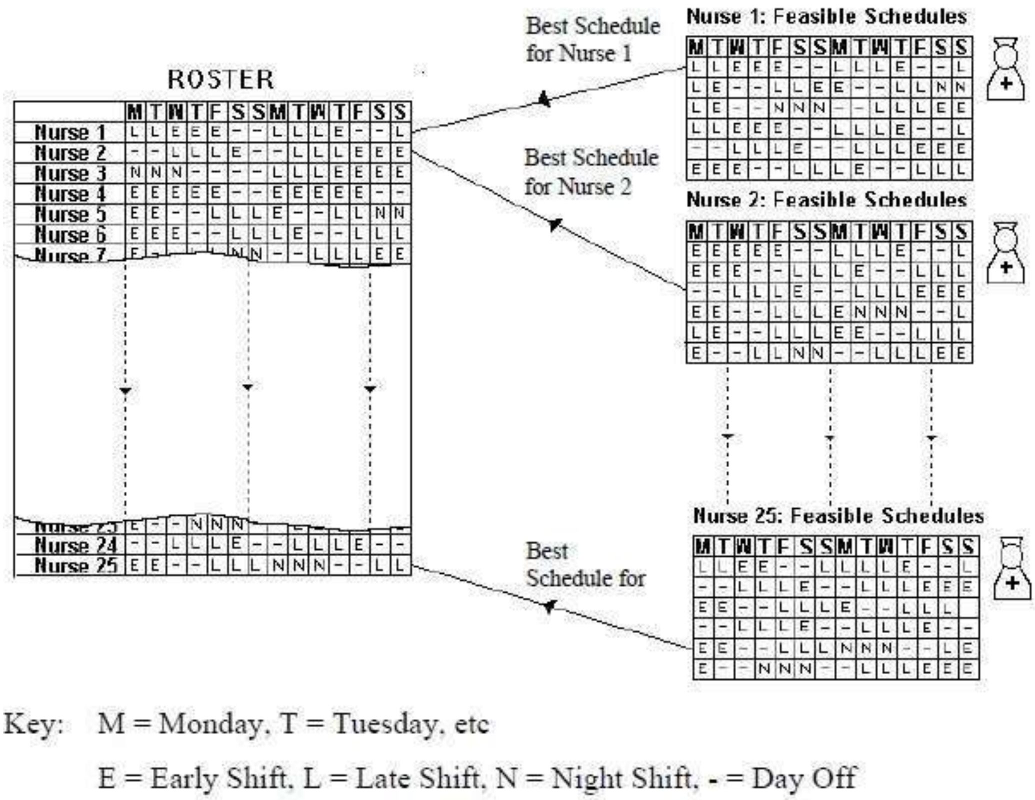


Figure 2.1. Warner's feasible schedule approach to nurse scheduling.

Following from Warner's work, Kostreva, Lescyski, and Passini (1978) developed a mixed-integer programming formulation of NSPs. The first phase involved heuristically generating a complete schedule that fulfilled all the constraints of the problem. The aim was for the schedule to meet the minimum standard and all nurse requirements; that is, at least one of the schedules generated should afford the nurses the days off that they requested. The second phase of the approach used a mixed-integer programming technique to assign schedules to individual nurses. The objective of phase two was to minimise the total 'hate points' score; where nurses were provided with questionnaire and a matrix of 'hate points' was calculated for each nurse in relation to each schedule (Kostreva *et al.*, 1978, p. 287). The algorithm iterates between phases one and two to generate a new schedule with each iteration, as illustrated in Figure 2.2. However, the heuristic is not specified in detail and its performance is not comparatively tested; therefore, the approach cannot be fully assessed.

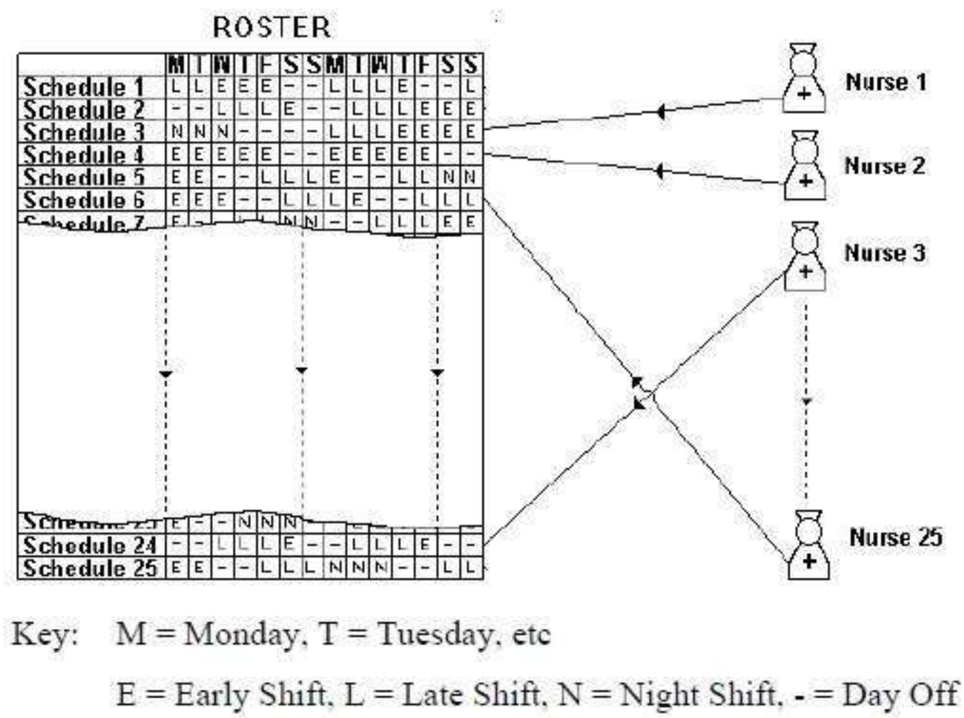


Figure 2.2. Kostreva *et al.*'s assignment approach to nurse rostering.

Working with NP-hard scheduling problems, Huarng (1997) proposed the approach of sub-grouping, by splitting nurses and workloads into several subgroups, and obtained a very satisfactory computational result.

However, this approach is model dependent. Li, Lim and Rodrigues (2003) presented a hybrid AI approach using a class of over-constrained NSPs. Their approach was two-phased: first, a solution was obtained for a relaxed version of the problem that included only the hard constraints and part of the nurses' requests for shifts. In the second phase, adjustments were made by descending local search and tabu search to improve the solution. Glover and McMillan (1986), Valouxis and Housos (2000) aimed to combine the strength of mathematical programming and AI approaches. The problem was formulated as an approximate integer programming model, where the integer programming problem is first solved and its solution further improved using tabu search.

Many heuristic approaches were straightforward automation of manual practices, which have been widely studied and documented in nursing administration literature (see, for example, Hung, 1995; Jelinek and Kavois, 1992). Heuristic searches apply heuristic models to find feasible schedules. A heuristic model is a set of rules constructed based on some level of knowledge; it does not guarantee an optimal solution. This type of method is ideal for solving problems with soft constraints, although it may have problems dealing with hard constraints. When the constraint conditions are numerous, it is generally difficult for the heuristic scheduling approach to attain a reasonable solution. It is thus not easy to process NSPs using this approach (Millar and Kiragu, 1998). However, the heuristic search approach is useful to adopt to address some of the weaknesses of other approaches (Smith and Wiggins, 1977); for example, local searches, the tabu search or simulated annealing methods are likely to be weak on their own and usually need to be combined with other techniques. Further, it seems almost impossible to define a simple hierarchy or set of priorities to enable a completely mechanical relaxation of the constraints.

Okada and Okada (1988) aimed to solve NSPs by applying a state- space search procedure similar to the manual method of the human scheduler. These search algorithms can produce high-quality solutions, but often at a considerable computational cost. Another example of the heuristic search approach was developed by Randhawa and Sitompul (1993), whose model consists of a best-first search algorithm to generate work patterns. Meta-heuristics represent a higher level of abstraction. They are usually implemented as a heuristic scheduler on top of low-level heuristics (Burke *et al.*, 2004a), which are treated as black boxes. In certain cases, the heuristic search approach increases the efficiency of the state-space search. Chiaramonte and Chiaramonte (2008) proposed a heuristic method using a competitive agent-based negotiation that focused on nurses' preferences. However, this heuristic search approach does not do very well with regard to job satisfaction, because personal requests will only be granted whenever these requests do not conflict with other priorities. Further, a heuristic approach is implicitly based on a certain view of nursing schedule quality, which makes it less useful whenever another view is applied.

Rosenbloom and Goertzen (1987) presented an algorithm with three stages: generate a set of possible schedules which are seven-tuples of 0–1 depending on whether the day is off or on, formulate the problem as an IP, and produce a solution. For example, Arthur and Ravindran (1981) used 0–1 goal programming to solve two-stage cyclical scheduling problems. Musa and Saxena (1984) used a 0–1 goal- programming formulation for nurse scheduling in one unit of a hospital. In their study, goals with different priority levels represented hospital policies and nurses' preferences. Berrada *et al.* (1996), in their 0–1 goal programming model for nurse scheduling, set the hard constraints based on administrative and union contract specifications, while work patterns and nurses' preferences determined the soft constraints. Moores, Garrod and Briggs (1978) also applied 0–1 goal programming to formulate the student nurse allocation problem. The main drawback of the exhaustive search approach is its rigidity concerning the priority structure of the optimisation algorithm. Although both goal programming and constraint programming offer more flexibility in choosing priorities, they still require a fully specified hierarchy of priorities. Therefore, the problem for Moores *et al.* (1978) was to produce a three-year schedule for student nurses that complied with the minimum practical and theoretical standards, while also being suitable for use as part of the hospital work force. Ozkarahan and Bailey (1988) utilised goal programming to search for a schedule with the traditional 'set covering' model. Throughout this paper, the importance of flexibility in the nurse scheduling environment was emphasised where each solution can be disaggregated into specific assignments for specific units and nurses.

The main idea of local search is to take a possible solution to the problem as a start (even if it is bad), and slowly modify it according to predetermined rules with the hope of creating better solutions. In its default form, the local search process is a hill-climbing algorithm. Important variations on hill-climbing algorithms are tabu search and simulated annealing. Bellanti, Carello, Della Croce and Tadei (2004) developed an approach in which they use both tabu search and simulated annealing in a largely similar problem with an initial solution created using a heuristic method. Thereafter, a set of neighbourhood operators is defined and tabu search or simulated annealing is applied to improve the solution.

Aickelin and Dowsland (2000) used genetic algorithms to solve NSPs. Dowsland and Thompson (2000) combined tabu search and network programming to establish a non-cyclical scheduling system, while Knjazew (2002) used genetic algorithms to solve cyclical scheduling problems. Li and Aickelin (2006) used a Bayesian optimisation algorithm to solve NSPs. Several nurse scheduling models were based on linear programming (Ozkaharan, 1989), penalty-point algorithms and mixed-integer programming (Harmeier, 1991). Another example of exhaustive search uses constraint programming for solving NSPs (Weil, Heus, Francois and Poujade, 1995) is the constraint programming combines logic programming and an AI technique with operations research techniques. It enables the problem modelling to be dissociated from the algorithms used for the solution, which provides flexibility in adjusting the formal model of NSPs. Jaumard *et al.* (1998) solve a NRP with the objective of reducing salary cost and improving nurse preference satisfaction. They also use column generation techniques, where the columns correspond to individual schedules for each nurse. Darmoni *et al.* (1995) use constraint programming to solve the scheduling problems in a French hospital. A fair scheduling among nurses is applied using a search strategy over a planning horizon of up to six weeks. Abernathy, William, Nicholas, John and Sten (1973) present two solution procedures to determine the staffing level: the first approach iteratively uses a penalty function for understaffing and overstaffing, whereas the second approach determines a required staffing level based on the chance-constraints. Other optimisation techniques have been used in nurse scheduling particularly for the non-cyclical type. These include the assignment problem (Gaetan, Pierri and Brigitte, 1999), non-linear programming (Warner, 1976) and goal programming (Ozkaharan, 1989). Blau and Sear (1983) applied a cyclic descent approach to another NSP and reported the successful implementation of the algorithm on a microcomputer; however, they did not evaluate the quality of the rosters generated.

2.4 Survey Review of the Nurse Scheduling Problem

This research field has grown rapidly over the past decades. We focus on the period 2000–2015, selecting 88 articles that focus on algorithmic techniques that have been successfully applied to NSPs and specifically target approaches using real-world benchmark problems from various places. In addition to explaining and summarising the characteristics and algorithms of techniques (such as in Section 2.3). Table 2.1 gives an overview of the selected articles on algorithmic techniques for solving NSPs. We categorise the most broadly used and well-cited literature (up to 2015) on algorithmic techniques based on four classification criteria: integer programming, construction techniques, heuristic and others (methods were hybridised or integrated with other techniques). Therefore, recent methods that are not as well established are not represented in this table. We also did not include in the categorisation any articles that present general staffing or SSPs.

Table 2.1. Techniques for the Nurse Scheduling/Rescheduling Problem

Year	Authors	Integer Programming	Construction Technique	Heuristic	Others
2000	Aickelin & Dowsland				GA+H
	Dowsland & Thompson				TS+IP
	Cai & Li			GA	
2001	Burke *et al.*			MA	
	Brusco & Jacobs	ILP			
2002	Knjazew			GA	
2003	Soubeiga			HH	
	Li *et al.*				H+LS+TS
	Dias *et al.*			TS, GA	
	Ikegami *et al.*	MIP			
2004	Aickelin & White				GA+IP
	Aickelin & Dowsland			IGA	
	Isken	IP			
	Winstanley				CLP+AB
	Bard (2004b)	MIP			
	Bard (2004a)	IP			
	Burke (2004a)			VNS	
2005	Bard & Purnomo (2005a)	IP (B&P)			
	Bard & Purnomo (2005b)	IP			
	Azaiez & Al Sharif	0-1LinearGP			
	Bard & Purnomo (2005c)				CGB+IP
	Bard & Purnomo (2005d)				CGB+IP+H
	Matthews	LP			
	Horio		CH		
	Fung *et al.*				GCS/SS
	Brucker *et al.*		CH		
2006	Beddoe & Petrovic				CBRG+GA
	Suman & Kumar		SA		
	Belien	MIP(B&P),DA			
	Li et al				BOA
	Dowsland *et al.*		GA		
2007	Moz & VazPato				GA+CH
	Bard & Purnomo	IP(LR)			
	Purnomo & Bard	IP(B&P)			
	Burke *et al.*			EA(SS)	
	Burke *et al.*				H+VNS
	Thompson			LS,SA	
	Bai *et al.*				GA+SA +HH
	Beddoe & Petrovic				CBR+TS
	Bester *et al.*			TS	
	Majumdar & Bhunia			GA	
	Aickelin& Li				ED+LD
	Aickelin& Li				ED
2008	Chiaramonte			AB	
	Landa-Silva & Le	SEAMO			
	Vanhoucke & Maenhout	IP			
	Burke *et al.*				H+VNS
2009	Brucker *et al.*				CH+LS

Year	Authors	Integer Programming	Construction Technique	Heuristic	Others
	Goodman *et al.*				GRASP
	Burke *et al.*				IP+VNS
	Beddoe *et al.*				HM+CBR
	Tsai & Li			GA	
2010	Glass & Knight	MIP			
	Maenhout				MO
	Brucker		CH		
	Bai				EHA
	Bouarab *et al.*	MP			
2011	Constantino *et al.*				GA+SS
	Vlah *et al.*			VNS	
	Ramli & Ahmad			TS (Enhanced)	
2012	Burak *et al.*			VNS	
	Gino *et al.*				MO
	Fang He & Qu				CP–CG
	Naudin *et al.*	MM			
	Birgin *et al.*			LS	
	Valouxis	MIP			
2013	Burke *et al.*			VDS	
	Awadallah *et al.*				HHS
	Messalis *et al.*				AP
	Solos IP			VNS	
	Maenhout *et al.*				IP–CG
2014	Liang & Turkcan				MOO
	Kim *et al.*			GA	
	Leksakul & Phetsawat			GA	
	Kumar *et al.*	LP			
	Legrain *et al.*				MOO
	Wong *et al.*				H+LS+GP
2015	Gonsalves & KoheiKuwata			MA	
	Chun *et al.*			GA	
	Jafari & Salmasi			SA	
	Jafari *et al.*	MP			
	Agyei *et al.*				GA+PGA
	Bagheri *et al.*	0-1GP			
	Swapnaja *et al.*	SP			

Note: GA=Genetic algorithm, H=Heuristic, TS=Tabu search, IP=Integer programming, GP=Goal Programming, LP=Linear programming, MA=Memetic algorithm, ILP=Integer linear programming, HH=Hyper-heuristic, IGA=Indirect GA, CLP=Constraints logic programming, AB=agents-based, MM=Mathematical Model, MIP=Mixed-integer programming, B&P=Branch & Price, CGB=Column generation based, GCS/SS=Guided complete search/Simplex solver, NN=Neural network, CH=Constructive heuristics, CBRG=Case-based repair generation, CBR=Case-based reasoning, SA=Simulated annealing, DA=Decomposition approach, SS=Scatter search, VNS=Variable neighbourhood search, LS=Local search, GRASP=Greedy random adaptive search procedure, SEAMO=Simple Evolutionary Algorithm for Multi-objective Optimisation, CP–CG=Constraint programming based column generation., MOO=Multi-objectives Optimisation. PSO=Particle Swarm Optimisation, MP=Mathematical Programming, EHA=Evolutionary Hybrid Algorithm, HM=Hybrid Meta-heuristic, BOA=Bayesian Optimisation Algorithm, ED=Estimation Distribution, VDS=Variable neighbourhood search, HHS=Hybrid Harmony Search, AP=Algorithm prediction, SP=Stochastic Programming, PGA=Parallel Genetic Algorithm Summary

2.5 Summary

This chapter presented basic concepts and models for scheduling, SSPs and NSPs. The chapter also presented a brief description of the constraints that are found most often in the literature, as well as in real-world nurse scheduling scenarios, although solutions to these constraints are not presented. The extensive review of the extant literature in this chapter leads us to draw several conclusions that may be useful for guiding further research. First, it is clear that this research field is growing rapidly. Researchers are increasingly creative in applying multiple methodologies and techniques to optimise NSPs, and thus meet a myriad of objectives and performance constraints. In this chapter, these techniques have been placed in four different categories. Arguably, these categories could have been further divided and, in future, novel methods for solving this problem are likely to appear.

Despite the many models and approaches proposed to counter NSPs in the literature, there is still a significant gap between nurse scheduling in theory and the challenging requirements in a real hospital environment. This is because models in the research are often an over-simplification of real-world NSPs. The current trend is to address the requirements of the real world (Burke *et al.*, 2004) and try to bridge the gap between research models and real-world models. This aim could be pursued by including many constraints in the research models but still allowing flexibility of models.

Benchmark problems from real-world environments would be particularly useful as a means for improving and validating the algorithms. Creating useful real-world benchmark examples is not easy, however, as they are nearly always very complicated problems. Most of the approaches in the literature have been shown to produce high-quality rosters and have real-life implementation. However, despite the many methods proposed to date, there is no single heuristic that is able to solve all scheduling problems effectively (Burke *et al.*, 1994a). That is, there is no way of knowing which is the ‘best’ method. Implementing and comparing the different algorithms across all the literature would be an impractical task.

In the author's experience, although there are advantages and competencies in the many approaches reported in the literature, several results are not easily reproducible because most of the algorithms depend on some random number generation. This means that a simple change in the generation of random numbers may affect very significantly the direction of the optimisation process. As a result, randomness generates different results and makes the results only statistically comparable. Since the results are hard to reproduce, it is difficult to determine whether they are optimal or not and it is not possible guarantee the quality of every individual solution.

The survey of the literature also showed that the previous approaches used are highly reliant on the technology available at the time. Early systems were severely constrained by computational limitations in terms of the problem complexity that was examinable. For example, in some of the early approaches, punch cards were used to input data and paper forms were needed for data collection. As computing power has increased, scheduling approaches have become more flexible and take into account more working preferences. Some of the current state-of-the-art approaches to automate nurse scheduling require similar run times of algorithm on personal computers with 3000MHz processors and numerous other improvements. This highlights either a serious lack of progress over the past 25 years, or more likely, limitations on the size and complexity of the problems that could be solved in the past, and the increase in complexity of the problems that are solved now. This increase in computing power is expected to continue in the future, so we should anticipate even better solutions to be produced more quickly for even harder real-world problems. In contrast to the gaps in the existing research, this thesis is significant in its flexibility of approach, applicability in practice and generic problem formulation.

Chapter 3: Datasets and Background of the Nurse Scheduling Problem

From the published research it is clear that benchmark datasets were used quite extensively. The usage of the same standard benchmark datasets in different research conducted by all researchers in this area is very important in order to have a fair judgment about the efficiency and efficacy of a particular approach. This chapter explains in detail the datasets used in the thesis to test the performance of the analysed approach. In addition, it gives a background of NSPs, to situate all subsequent chapters.

3.1 Datasets

The real-world datasets used in this study were available for download for scientific research from http://www.cs.nott.ac.uk/~tec/NRP/. The primary use of these datasets was to obtain information related to the nurses to be scheduled. In this study, we interpret a NSP as a problem of constructing appropriate information granules and using these granules to design an optimised schedule. The schedule 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. A large set of constraints are accessible for this NSP. Some of these datasets have logic constraints and are very complex to handle. Others are over-constrained, making it difficult to find a feasible solution to satisfy all constraints. Therefore, soft constraints are used to represent the conflicting preferences of nurses. We search for feasible solutions that minimise this violation of soft constraints.

3.1.1 ORTEC

Without the loss of generality, we discuss our contribution in the context of a specific NSP as encountered by ORTEC, the Netherlands, in intensive care units in Dutch hospitals. ORTEC supports hospitals and other organisations internationally with automated workforce management solutions. ORTEC provided data that showed a challenging real-world problem very typical of their clients' needs. Over the years, this problem has been tested by a range of meta-heuristic algorithms (Burke *et al.*, 2004a, 2008; Brucker *et al.*, 2005; Li *et al.*, 2012), and has become a benchmark dataset in the literature. The characteristics of this problem have been discussed in Baskaran, Bargiela and Qu (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 shift to 16 nurses in a ward within a scheduling period of five weeks. Twelve of the nurses are full-time and have a contract of 36 hours/week. One other full-time nurse works 32 hours/week and the remaining three part-time nurses work 20 hours/week. Each instance also has a number of specific personal requests, such as particular shifts and/or days requested off or on.

3.1.2 Shifts and shift demand

There are four different shift types in the problem: day, early, late and night shifts. All the shifts except night shifts cover nine hours including one hour of rest time. So the actual number of working hours for each shift type is eight. Night shifts last eight hours but include no rest time and so are counted as eight working hours. The total demand requirement for each shift for each day varies between instances. Generally, larger wards require more nurses on duty during each shift but similar sized wards can also have different demand requirements. The required number of nurses on individual shifts for different days of the week is summarised in Table 3.1. The hard and soft constraints that need to be satisfied are described in turn below.

Table 3.1. Shift Types and Daily Demand of 16 Nurses 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 shift								

3.1.2.1 Constraints

The NSP involves allocating the required workload to nurses subject to a number of hard and soft constraints, as detailed below.

3.1.2.1.1 Hard constraints

The hard constraints (denoted by HC) listed below must be met in all circumstances; otherwise, the schedule is considered infeasible and unacceptable.

HC1. Demands need to be fulfilled.

HC2. For each day, one nurse may start only one shift.

HC3. 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 four hours.

HC4. The maximum number of night shifts is three per period of five consecutive weeks.

HC5. A nurse must receive at least two weekends off duty per five- week period. A weekend off duty lasts 60 hours including Saturday 00:00 to Monday 04:00.

HC6. Following a series of at least two consecutive night shifts, a 42- hour rest period is required.

HC7. 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 of rest. An exception is that once in a period of 21 days for 24 consecutive hours, the resting time may be reduced to eight hours.

HC8. The number of consecutive night shifts is at most three.

HC9. The number of consecutive shifts (workdays) is at most six.

HC10. One of the full-time nurses requires not receiving any late shifts.

HC11. The maximum labour time averages 36 hours/week over a period of 13 consecutive weeks if this period does not include work during night shifts.

3.1.2.2 Soft constraints

The soft constraints (denoted by SC) in the problem we are dealing with are listed in Table 3.2. These constraints should be satisfied as much as possible; however, in real-world circumstances, it is usually unavoidable that some will be violated. Depending on how strongly these soft constraints are desired, a weight (a simple number) is assigned to each to reflect its importance (especially in comparison to other soft constraints). The highest weight is 1000, denoting a strong desire that this constraint be satisfied. The lowest weight is 1, indicating the relative unimportance of satisfying this constraint. The penalty of a feasible schedule is the sum of the weights of all the violations of soft constraints in the schedule. The weights are fixed either by the head nurses or through feedback from the nurses about what qualities they desire in their schedules. As a rough guide, the weights are described as follows:

Weight 1000: The constraint should not be violated unless absolutely necessary.

Weight 100: The constraint is strongly desired.

Weight 10: The constraint is desired but not critical.

Weight 1: Try to obey this constraint if possible, but it is not essential.

In practice, exponentially scaled weights like these are the most common type used. However, users do have the option of setting and changing the weight for each constraint to any positive integer value.

Table 3.2. Soft Constraints and their Weights

	Soft Constraints	Weights
SC1	For the period of Friday 23:00 to Monday 0:00, a nurse should have either no shifts or at least two shifts (complete weekend).	1000
SC2	Avoid sequences of shifts with length of one for all nurses.	1000
SC3a	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
SC3b	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
SC4	The rest after a series of *day*, *early* or *late* shifts is at Least two days.	100
SC5a	For nurses with availability of 30–36 hours per week, the number of shifts is within the range [4, 5] per week.	10
SC5b	For nurses with availability of 0–30 hours per week, the number of shifts is within the range [2, 3] per week.	10
SC6a	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
SC6b	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
SC7	For all nurses, the length of a series of *early* shifts should be within the range [2, 3]. It could be within another series of shifts.	10
SC8	For all nurses, 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
SC9a	An *early* shift after a *day* shift should be avoided.	5
SC9b	An *early* shift after a *late* shift should be avoided.	5
SC9c	A *day* shift after a *late* shift should be avoided.	5
SC10	A *night* shift after an *early* shift should be avoided.	1

To have the same evaluation functions as those of other approaches in the literature, the above soft constraints are measured by the quadratic function. That is, the measure of violations is squared and multiplied by the corresponding weight.

3.1.3 Kajang Hospital

3.1.3.1 Problems faced by head nurse (matron)

The general process of manual roster runs is as follows. Early in each week, the head nurse of Medical Ward 2 will draft the roster for each nurse. The process of producing the roster begins with the collection of information from each nurse, including their preference of days off and shifts. The head nurse faces a few problems during the production of the roster:

1. They need to reproduce drafts until nurses with adequate skills and experiences are equally mixed in each shift.

2. When new nurses need to attend training/courses, the workload of these leaving nurses has to be equally distributed. Therefore, the roster needs to be reshuffled.

3. When certain nurses have to be transferred to other wards for a few weeks because their expertise is needed, the roster needs to be reshuffled.

It is inefficient for the head nurse to spend his/her time and effort to arrange the schedule. Moreover, the task is made difficult by the problems stated above. Therefore, a solution is needed to make scheduling quicker and easier. The new scheduling problem presented in this thesis has been studied for three wards in a large Malaysian hospital; that is, the coronary care unit (CCU), medical ward and male ward in Kajang Hospital. We outline the following characteristics.

1. We have to adhere to Malaysian national laws and the collective labour agreements enforced in Malaysian hospitals.

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 Kajang Hospital.

3. It is not necessary to consider qualifications, as all personnel are highly qualified. However, specialised nurses are required to oversee all tasks in each shift.

All 28 nurses in Kajang Hospital are full-time and have a contract of 40 hours per week. There are 10 specialised U29 grade nurses and 18 normal U29 grade nurses working across different types of shift, as illustrated in Table 3.3. This satisfies the daily coverage requirements for these shift types.

3.1.4 Shifts and shift demand

There are four different shift types in the problem: day, early, late and night shifts. The hospital uses the terminology of 'morning', 'office hours', and 'evening' shifts; however, for the purpose of this study, and with the consent of the matron, we have renamed these shifts using the terms common in the nurse- scheduling literature: early, day, late and night. These shift types vary in their duration, but all include one hour of rest time. The early and late shifts have a seven-hour duration, the dayshift is nine hours and the night shift is 10 hours. The hospital's scheduling period is two weeks long, and the hospital practices a number of types of rest day, including Sleep Day (SD), Day Off (DO), Public Holiday (PH), Annual Leave (AL) and Emergency Leave (EL).

The required number of nurses on individual shifts for different days of the week is summarised in Table 4.7. The hard and soft constraints that need to be satisfied are described in turn below.

Table 3.3. Shift Types and Daily Demand of 28 Nurses During a Week

Shift type	Start time	End time	Demands						
			M	T	W	T	F	S	S
Early	07:00	14:00	6	6	6	6	6	6	6
Day	08:00	17:00	1	1	1	1	1	1	1
Late	14:00	21:00	6	6	6	6	6	6	6
Night	21:00	07:00	3	3	3	3	3	3	3

3.1.4.1 Constraints

3.1.4.1.1 Hard constraints

The hard constraints listed below must be met in all conditions; otherwise, the schedule is considered infeasible and unacceptable.

HC1: Demands need to be fulfilled.

HC2: For each day, one nurse may start only one shift.

HC3: One of the nurses requires performing only the Office Hour shift per day.

HC4: At least one skilled nurse must be scheduled to each shift.

HC5: The number of consecutive shifts (night) is at most three.

HC6: The number of consecutive shifts (workdays) is at most six.

HC7: Following a series of three consecutive night shifts, a 48-hour rest is required.

HC8: Following a series of six consecutive day shifts, a 24-hour rest is required.

HC9: The maximum number of night shifts is three per period of two consecutive weeks.

3.1.4.1.2 Soft constraints

The soft constraints listed below represent the preferences of the nurses and hospital requirements at Kajang Hospital. These soft constraints should be satisfied as far as possible; however, in real-world circumstances, it is usually unavoidable that some of these soft constraints will be violated. A numerical penalty weight is given for each soft constraint based on the importance of that constraint. A weighting is simply a number. Depending on how strongly these soft constraints are desired (especially in comparison to other soft constraints), a weight is assigned to each (see Table 3.4). The higher the weight, the more strongly the satisfaction of this constraint is desired. The penalty of a feasible schedule is the sum of the weights of all the violations of soft constraints in the schedule. One key issue regarding setting the weights of constraints in NSPs is that there are no standard weights for soft constraints, as they vary widely from one hospital to another. To serve as a guide, the weights shown in Table 3.4 can be understood as follows:

Weight 1000: The constraint should not be violated unless absolutely necessary.

Weight 100: The constraint is strongly desired.

Weight 10: The constraint is desired but not critical.

Weight 5: The constraint is favoured but not crucial.

Weight 1: Try to obey this constraint if possible, but it is not essential.

In practice, exponentially scaled weights like these are most commonly used. However, the users do have the option of setting and changing the weight for each constraint to any positive integer value.

Table 3.4. Soft Constraints and their Weights

	Soft Constraints	Weights
SC1	Avoid sequences of shifts with length of one for all nurses.	1000
SC2	The rest after a series of *morning* or *evening* shifts is at least two days.	100
SC3	The number of shifts is within the range [4, 6] per week.	10
SC4	The length of a series of shifts should be within the range of [4, 6].	10
SC5	Days on/off requests: Requests by nurses to work or not to work on specific days of the week should be respected; otherwise, solution quality is compromised.	10
SC6	Shift on/off requests: Similar to SC5 but relating to specific shifts on certain days.	10
SC7	For all nurses, the length of a series of *morning* shifts should be within the range [1, 4]. It could be within another series of shifts.	10
SC8	For all nurses, the length of a series of *evening* shifts should be within the range of [1, 4]. It could be within another series of shifts.	10
SC9a	A *morning* shift after the *office hour* shift should be avoided.	5
SC9b	An *evening* shift after the *office hour* shift should be avoided.	5
SC10	An *evening* shift after a day off followed by a night shift should be avoided.	1

3.1.4.2 Proposed solution of simplified plan for simulation in Kajang Hospital

In this study, we proposed a simulation model for Kajang Hospital that describes the functioning of the main processes in the NSP (Baskaran *et al.*, 2013a). This study was conducted by request from Kajang Hospital (see Appendix C for the related article). Figure 3.1 shows the design process to achieve an efficient scheduling simulation and presents a cost-effective schedule by executing the demand simulation. One example that we showed to the hospital used integer programming to find the results. Interactive scheduling is facilitated in our novel approach. Interactive scheduling allows human abilities to be extended and a scheduling approach applied to solve real problems. It provides a means for modifying the solution to cater for factors that had been assumed away during problem simplification. Generally, our solution is focused on solving complex problems based on well-justified simplifications of the original problem. We systematically subdivided the problem into smaller sub-problems capable of reproducing the result. This identification of interactive scheduling is dynamic. It is thus fully independent, user-oriented and compatible with the new human-centred computing paradigm. It is important to have easily understandable results in both domains. Another benefit in this simulation model is that the domain transformation can reduce computational complexity and thus computational time.

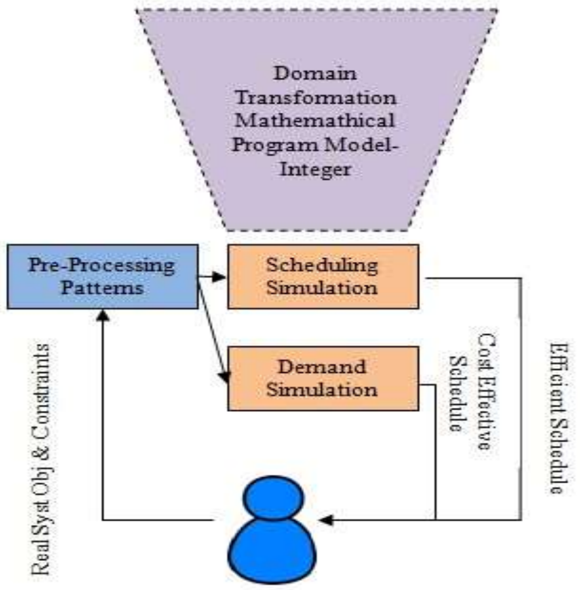


Figure 3.1. Design of the simulation model.

The simulation model also reduces the cross-referencing over the detailed swapping of shifts for individual nurses. The goal of this scheduling simulation is to test how the different schedules perform when, for instance, the workload or capacity has to cope with uncertainty. To retrieve meaningful results, the simulation was tested intensively with a range of different parameters. The results were then discussed with the real system matron to identify a number of service criteria in coordination with the hospital. If the first results indicate that the schedule does not meet the goals set in the simulation model, it can be adjusted or added to using some of the constraints in the mathematical programming model. Schedulers are provided with all of the information they need concerning the different steps so that matrons can choose which schedule to implement. During this simulation model, the schedules obtained will not make any difference in terms of the different order of processing. The schedule is the same when we change the order of individual patterns or nurses.

3.1.5 Other real-world benchmark problems.

To validate our algorithms and encourage more competition and collaboration between researchers addressing scheduling, we have built a collection of diverse and challenging benchmark datasets. The collection has grown over several years, has been sourced from 13 different countries and the majority are based on real-world scheduling scenarios. Table 3.5 lists these benchmark instances. All are available for download from http://www.cs.nott.ac.uk/~tec/NRP/.

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 organisations prefer to minimise overstaffing, whereas others prefer to maximise staff satisfaction and so set a higher importance weighting for those objectives.

Table 3.5. 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	

3.1.5.1 Initial study on selected real-world benchmark datasets: Hard and soft constraints

Table 3.6. Hard Constraints

HC	Category	Details
GPost	One shift per day	One shift per day (D, N, R)*
	Coverage (no over/ under cover)	Weekday: 3D 1N; Weekend: 3D, 1N
	Working type	Full-time: 18 shifts; Part-time: 10 shifts
	Shift patterns	Maximum consecutive working days: 6
		Maximum consecutive N shifts: 3

HC	Category	Details
		Maximum consecutive working weekends: 3
		After a series of work, at least 2 days rest
		Complete weekends, i.e. rest or work on both days
		After N shifts, at least 2 days of rest
Valouxis	One shift per day	One shift one day (D, E, N, O)*
	Coverage (no over/ under cover) Weekday:	4D 4E 2N; Weekend: 3D 3E 2N
	Working type	18 shifts
	Shift patterns	Maximum consecutive working days: 5
		Maximum consecutive N shifts: 3
		Maximum consecutive working weekends: 3
		After a series of work, at least 2 days off
		Complete weekends, i.e. free or work on both days
		After N shifts, at least 2 days off
WHPP	One shift per day	One shift one day (D, E, N)*
	Coverage (no over/ under cover) Weekday:	10D on Mon and Tues; 5D on Wed to Sun;10E on Mon, Tues, Wed &Fri; 5E on Thurs & Weekend; 5N
		30 standard nurses
	Working type	
	Shift patterns	Maximum consecutive working days: 7
		At least 2 days off after 7 working days
		Maximum consecutive N shifts: 4
		At least 2 days off after 4N
		No N-D, N-E, E-D
Ozkarahan	One shift per day	One shift one day (12,8)
	Coverage (no over/under cover) Weekday:	Skill AID: Shift 12(1) and Shift 8(0); Skill RN: Skill 8(2) and Skill 12(4) Weekday and 12(3) Weekend
	Working type	AID: 3 nurses and RN: 11 nurses

Note: D=day shift; E=Early, N=night shift; O=off day, Mon=Monday, Tues=Tuesday, Wed=Wednesday, Thurs=Thursday, Fri=Friday, Sat=Saturday, Sun=Sunday.

Table 3.7. Soft Constraints

SC	Category	Details	Weights
GPost	Balanced workload	Full-time: [4,5] shift/week	1*
		Part-time: [2,3] shift/week	1*
		Full-time: shifts series length [4,6] Part-time: shifts series length [2,3]	100
	Pattern preference	No standalone shift, i.e. single day on	
		No one shift over a weekend	100
		No one day off between shift series	10
Valouxis	Balanced workload	No. of D shifts: [5, 8] in the schedule	100
	Pattern preference	No. of E shifts: [5, 8] in the schedule	100
		No. of N shifts: [2, 5] in the schedule	100
		No. of O shifts: [10, 13] in the schedule	100
		No standalone shift, i.e. single day on	1000
		No one shift over a weekend	1000
		A D after E should be avoided	1000
		A E after N should be avoided	1000
		A D after N should be avoided	1000
		At least 2 days off between shift series	100
WHPP	Pattern preference	Series of D/E/N shift length: 3	40
		Series of D/E/N shift length:3	
Ozkarahan	Pattern preference	No On-Off-On should be avoided	20
		1* No Off-On-Off should be avoided	
		1* Work both Sat and Sunday	
		1* Max 1 working weekend	
		1* Weekend On-Off, Off-On or On-On	400

Note: To have the same evaluation functions as those of other approaches in the literature, the constraints denoted by * are measured by the quadratic function. That is, the measure of violations is squared and multiplied by the corresponding weight.

3.2 Further Definition of Shifts, Sequence of Shifts, Schedules and Scheduling

To understand better the definitions based on Baskaran, Bargiela and Qu (2014c), an example of shifts is given in Table 3.8. These shifts are taken from the ORTEC dataset explained in Section 3.1.1.

Table 3.8. Shift Types and Daily 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 shift								

There are five *shift types* in the problem presented in Table 3.8: day, early, late, night and rest shifts. The total demand requirement for each shift for each day varies from three nurses Monday to Friday to two nurses on weekends. There is no difference in night demand. Every day, only one nurse covers the night shift. Feasible sequences of shifts must satisfy all hard constraints. For the above problem, 16,768 feasible sequences of shifts were identified for a one-week period. An example of a feasible sequence of shift for a one-week period (Monday to Sunday) is EDLLRRR, while an infeasible sequence of shifts might be EDLNNNN. The latter is infeasible because the sequence of night shifts violates hard constraint where the number of consecutive night shift is at most three for this real-world dataset, which requires no more than three night shifts in a row. The identified feasible sequences can be classified into zero-cost or non-zero-cost sequences of shifts,

zero-cost means the sequence does not violate any soft constraints and non-zero-cost means the sequence does violate one or more soft constraints.

Cost can range from a smaller value to larger value. The value can go higher when sequences are connected, due to the greater number of possible violations of soft constraints. Sequences of shifts depend on satisfying constraints. Further explanations based on soft constraints and weight are illustrated in Section 4.6. The weight for each soft constraint is calculated either linearly or quadratically using the violation measurement factors (Li *et al.*, 2012). A soft constraint with a linear penalty function is calculated as: violation measurement factor multiplied by weight. Alternatively, a quadratic penalty function is calculated as: violation measurement factor squared and multiplied by weight. An example of a feasible sequence of shifts that does not violate any soft constraints (i.e., a zero-cost sequence) is ELLLRRR. Similarly, an example of a feasible sequence of shifts that violates soft constraint two by having a sequence of shifts with a length of one, giving a cost of 1000, is ELLLRRE. Likewise, when there are few combinations of violations happen, an example of feasible sequence having three violation of soft constraint one, two and six by not having a complete weekend, having a sequence of shifts with a length of one and violating the length of series of shift, giving a cost of 2010, is DRRRRDR where violation 1 gives cost 1000, violation 2 gives cost 1000 and violation 3 gives cost 10. Specifically on ORTEC study, among all feasible sequences, there are 193 zero-cost sequences for the 36/32hours/week nurses and 66 zero-cost sequences for the 20 hours/week nurses. The remaining 16,510 feasible shift sequences have a non-zero-cost. It is now possible to define the objective of the problem: To find a feasible schedule with the lowest possible weight caused by soft constraint violations. From the viewpoint of the head nurse, the actual weight hides a lot of information about the solution but is not totally meaningless. By investigating the weight of each schedule, it is possible to gain some idea of the schedule quality. For example, if the weight is less than 1000, then we know that all the constraints with weight 1000 have been satisfied. However, the key to producing satisfactory schedules is setting the correct weights and ensuring that all the required constraints are defined. Therefore, it is important that the end user either has a good understanding of how to fix the weights and define constraints or has clearly described the requirements that they need.

A schedule is a set of sequences allocated to each nurse such that they add up to the coverage requirement, as described in Table 3.8. In our ORTEC case study, schedules are constructed to minimise the cost of sequences of shifts over a period of five weeks. Table 3.9 provides an example of a five-week schedule with a specific number of nurses that meets the coverage requirement of each different shift. This schedule is feasible since it satisfies all the hard constraints, especially the demand or cover. As we can see, each week, the total demand requirement for each shift for each day that varies from three nurses Monday to Friday to two nurses on weekends is satisfied. There is no difference in night demand. Every day, only one nurse covers the night shift.

Table 3.9. Example of Five-week Schedule with 16 Nurses

	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
.	....	....			
12	LRRRDDD	DLLRRLL	LDNNRRR	RDDLLRR	DDDDLRR
.	....	....			
16	....	....			
Cover E	3333322	3333322	3333322	3333322	3333322
Cover D	3333322	3333322	3333322	3333322	3333322
Cover L	3333322	3333322	3333322	3333322	3333322
Cover N	1111111	1111111	1111111	1111111	1111111

Note: E=Early, D=Day, L=Late, N=Night

Scheduling is a process of allocating shifts over a predefined period subject to various constraints. Scheduling that satisfies the hard constraints on sequences of shifts and the cover requirement will generate a feasible schedule (see Table 3.9). This can then be refined to lower the cost of the schedule by ensuring satisfaction of as many soft constraints as possible. The scheduling problem in the above scenario presents a combinatorial optimisation problem in a space of **16*535=4.6*1025** possible schedules, which is clearly a computationally prohibitive task (Baskaran, Bargiela and Qu, 2013a). Most of the methods highlighted in Chapter 2 perform optimisation on feasible schedules by adjusting individual shifts. This can involve the replacement of one shift type with another and subsequent balancing of the required cover. Alternatively, optimisation may involve swapping shifts allocated to two nurses on the same day, which does not alter staff cover. As Table 3.10 shows, swapping a shift can produce a lower-cost schedule.

Table 3.10. Shift Swapping to Achieve a Schedule with a Lower Cost

	Sequences of Shifts				
	Week 1	**Week 2**	**Week 3**	**Week 4**	**Week 5**
Nurses	MTWTFSS	M……. S	M……. S	M……. S	M……. S
1	ELLRRLL	**L**DNNRRR	RLLLDRR	EEELLRR	LLLRRLL
.	…………	**Swapping**			
12	LRRRDDD	**D**LLRRLL	LDNNRRR	RDDLLRR	DDDDLRR
.	…………	….			
16	…………	….			
Cover E	3333322	3333322	3333322	3333322	3333322
Cover D	3333322	3333322	3333322	3333322	3333322
Cover L	3333322	3333322	3333322	3333322	3333322
Cover N	1111111	1111111	1111111	1111111	1111111

Note: E=Early, D=Day, L=Late, N=Night

Both a simple change of a single shift and the swapping of two shifts imply non-monotonic changes in the cost of a schedule (non-monotonic=*defeasible inference*, i.e., inference in which reasoners draw conclusions tentatively, reserving the right to retract them in the light of further information). In other words, a decrease in the number of violated soft constraints does not necessarily imply a decrease in the cost function. Therefore, the process of optimisation of the non-monotonic cost may converge to local optima rather than global optima. For example, if there are 15 constraints with different cost values, the challenge is to choose which constraints to violate and which not to violate. If the schedule violates one of the more expensive constraints (i.e., with an associated cost closer to 1000), it can be replaced with a schedule that violates one or more less expensive constraints at a lower total cost. Unfortunately, local optima evaluated in this way do not provide any guidance concerning the required adjustment of the independent variables to facilitate convergence to global optima. This means that existing methods have to perform combinatorial searches in a large problem space. Figure 3.2 illustrates the challenge of the scheduling problem by providing an example of two possible sequences of shifts allocated to Nurse 1 in week 1. A simple change of one shift implies a non-monotonic change in the cost associated with the violation of constraints. Due to the large number of possible schedules and the non-monotonic change of the cost with different combinations of shifts, the optimisation of the overall schedule by the modification of individual shifts and/or various groups of shifts is considered an NP-hard problem (Celia and Roger, 2010). However, this classification is predicated on the assumption of the deployment of scheduling algorithms that explore the solution space directly.

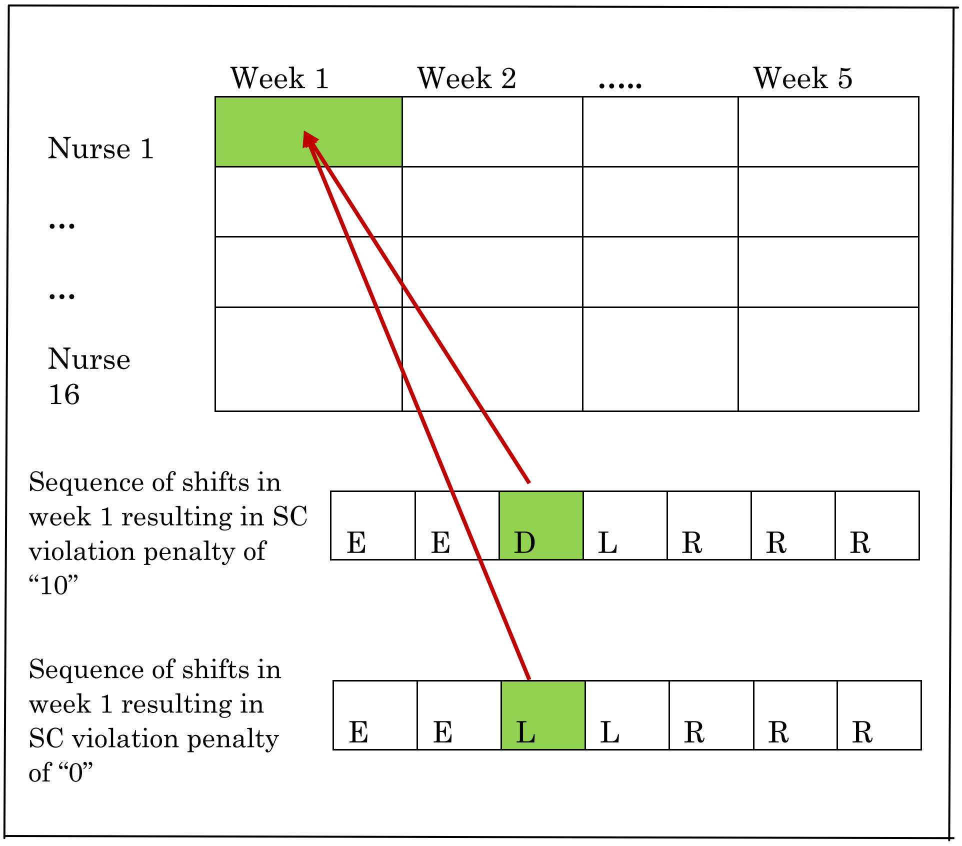


Figure 3.2. Example of a single shift change in a schedule.

Chapter 4: Domain Transformation Approach

This chapter presents the proposed approach for the NSP that will be employed in the subsequent chapters of this thesis. First, a 'bigger picture' overview summary of previous studies and the proposed study is illustrated in a diagram. Background to the domain transformation approach is given to explain the decision of using the selected approach. Finally, to understand better how the structure of the problem influences the behaviour of the NSP, a novel approach is discussed that explains how costs are minimised in a schedule by transforming the original problem domain into a smaller domain that is easier to manage. We then provide the general novel algorithm designed based on this approach to solve any NSP or SSP.

4.1 The Bigger Picture

During the last decades, many scientific studies have been conducted in order to support the task of nurse scheduling using computer programs, as discussed in Chapter 2. The concepts of the previous studies are summarised in Figure 4.1 below.

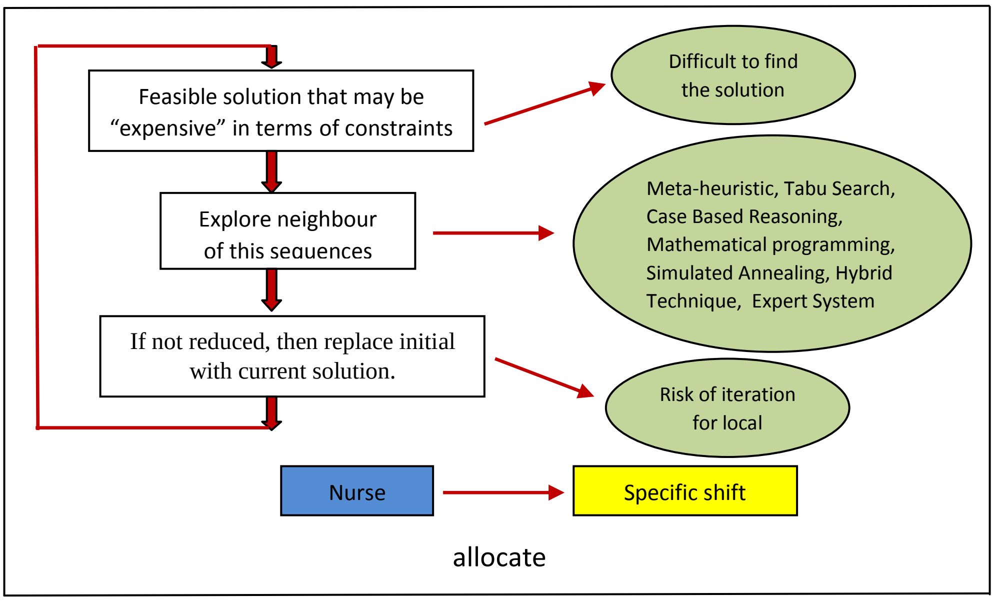


Figure 4.1: Summary of previous studies on NSPs

As mentioned previously, a feasible schedule is a schedule that satisfies all the hard constraints. In most cases, feasible solutions that are found may be ‘expensive’ in terms of the constraint weight or cost. This leads to difficulty in finding a solution. Hence, in order to find a feasible solution, the neighbourhood of these sequences are explored. From the literature, there are many approaches, methods and techniques used. A few techniques are highlighted in Figure 4.1. In general, if the cost of the schedule is not reduced, then the initial solution will be replaced with the current solution found. In this process, nurses are allocated to a specific shift most of the time. As an alternative, we transform the original problem domain into a smaller domain that is easier to manage. Nurses are allocated with patterns instead of shifts. In the offline preparation, as labelled in Figure 4.2, initially we identified all shift sequences with zero-cost. These sequences are called ‘patterns’, and these patterns are allocated to a specific shifts sequence. Later, we use this sequence of patterns to design and obtain an optimal schedule. This is done in an analytical preparation. This approach contrasts with the standard, detailed level of problem representation, which requires arrangement of various heuristic methods to manage computational complexity.

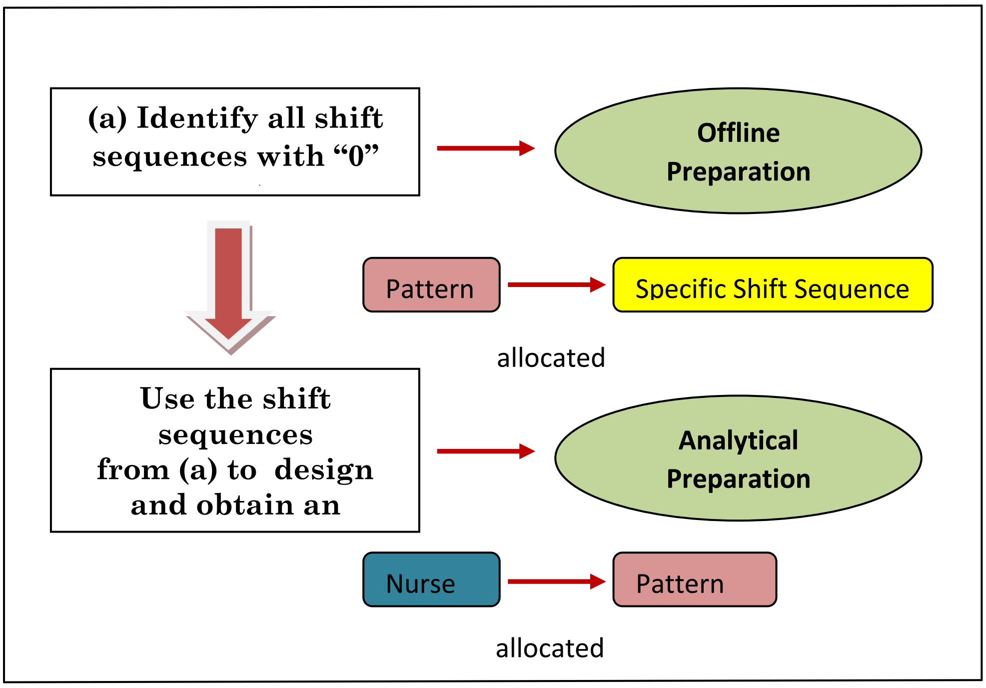


Figure 4.2: Outline of proposed approach for the NSP

The fundamental hypothesis of this thesis is that the information granulation of the pre-processing of initial problem information can lead to a transformation of the NSP into a new solution domain in which the problem is solved more easily. This aggregated information from the modified information domain, grouped properly, will be much easier to handle and perform efficiently. This is because it will be generating reproducible results, as opposed to dealing with the original information as in many previous studies.

4.2 Overview of Information Granulation

As a simple way of understanding information granulation, consider the following example: when one is travelling, it is most useful to know first about the weather conditions in a place rather than the exact temperature. The less precise but more general notion of weather is more appropriate at the planning stage then the precise information about the temperature at a specific instant. In another example, to establish a course view of the world map, the focus is on high-level information such as the placement of the continents, countries and oceans. Only when one needs more detail is it necessary to move down to finer-scale information such as the location of regions, provinces and states.

To simplify a concept while maintaining its accuracy is one of the objectives of the emerging computing paradigm of granular computing (Zadeh, 1979). Granular computing views the world as divided into entities called information granules that are grouped together due to their similarity, functional adjacency, in distinguish ability or coherence (Bargiela and Pedrycz, 2003). A highly detailed granular world can be abstracted into lower granulation using formal frameworks that approximate the original representation. This can be formally written as:

$$\mathbf{G} = < \mathbf{X}, G, \mathbf{A} \ldots >$$

Where **G** is the granulation process, **X** is the element to be granulised, *G* is a family of reference and **A** refers to abstractions (Bargiela and Pedrycz, 2003). Briefly, granular computing is geared towards representing and processing basic chunks of information; that is, information granulation (Kasabov, 1996; Zadeh, 1997). In 1979, Zadeh first introduced and discussed the notion of information granulation, pioneering the explicit study of granular computing (Zadeh, 1979). In 1982, Pawlak proposed the theory of rough sets (Pawlak, 1982, 1991), which provides a concrete example of granular computing. To some extent, rough set theory brought increased attention to the importance of granulation.

According to Zadeh, granules are constructed and defined based on the concept of generalised constraints. A granule may be interpreted as a subset of a universal set; while in programming, a granule can be a program module (Yao, 2004b). To quote Zadeh's (1997, p. 111) definition, '*granulation* involves a decomposition of whole into parts. Conversely, organization involves an integration of parts into whole; causation involves association of causes with effects'. An important property of granules and granular level is their granularity. The granularity of a level refers to the collective properties of granules in a level with respect to their size. Granularity is reflected by the size of all granules involved in the level, and it enables the construction of a hierarchy. Thus, information granules are divided into layers or hierarchies to build an information pyramid in which the granules at the bottom are concerned with numeric processing and the granules at the top are solely devoted to symbol-based processing. Information granules are most commonly encountered at the intermediate level. This is illustrated in Figure 4.3.

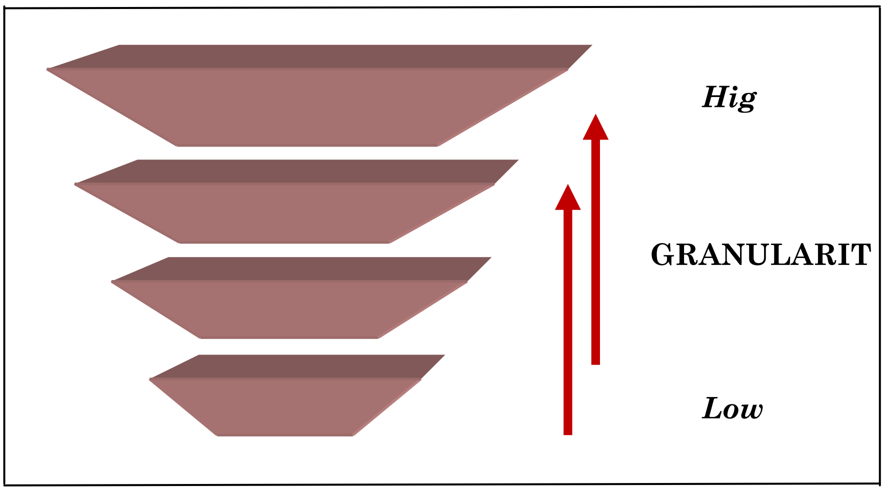


Figure 4.3. An information-processing pyramid (Bargiela and Pedrycz, 2003).

The issues of relevance and defining the 'size' of an information granule are of fundamental importance in the field of granular computing and depend on the problem in which the granules are used. In general, high information granularity levels are associated with a decrease in the usefulness of the concept. Granular computing, therefore, focuses on every day and commonly used concepts.

4.3 Granular Computing Work

Stepaniuk and Skowron (2005) studied granulated information systems and granular approximate space and discussed the granular framework of approximation and dependency relationships between concepts. Initiatives include granular computing as a way of problem solving (Yao, 2004a, 2004b, 2007; Zhang and Zhang, 2007) and granular computing as a paradigm of information processing (Bargiela and Pedrycz, 2002, 2008). Based on granularity and abstraction, many authors have studied certain fundamental topics of AI, such as knowledge representation (Giunchglia and Walsh, 1992; Zhang and Zhang, 1992), theorem proving (Giunchglia and Walsh, 1992), planning (Knoblock, 1993), natural language understanding (Mani, 1998), intelligent tutoring systems (McCalla, Greer, Barrie and Pospisil, 1992), machine learning (Saitta and Zucker, 1998) and data mining (Han, Cai and Cercone, 1993).

Chen, Chen, Hsu and Zeng (2008) present a novel model called the 'information granulation based data mining approach' to tackle the imbalanced data of many real-world datasets. This method imitates the human ability to process information, acquires knowledge from information granules rather than from numerical data, and introduces a latent semantic indexing-based feature-extraction tool by using singular value decomposition to reduce the data dimensions dramatically. In another study, Li, Qiu, Liu and Bai (2010) propose an algorithm for generating a domain concept granule lattice. Li *et al.* illustrated that ontology building can be attained from a given incomplete multi-valued information system, and can automatically construct a basic domain ontology based on the domain concept granule lattice. Further, granulation is one of the most common techniques used in sound design, where the samples within the grain are identical to those found in the original (Truax, 1990a, 1994b). Information granulation is a powerful approach for emphasising the relevant information rooted completely in the raw data. In their granular model, Rahim and Bargiela (2009) showed that by capturing the persistent feature of potential (as opposed to actual) conflicts in the conflict chain information granule, one could construct a much simpler model of an exam-scheduling task.

There are many reasons for studying domain transformation using methods derived from information granulation. Human problem solving is based crucially on levels of granularity and change between granularities (Hobbs, 1985; Zadeh, 1997); therefore, the implementation of information granulation principles extracts the common elements from human problem solving, leading to more effective information-processing systems. Moreover, a multiple-level representation reveals orderliness, control and the complex system or problem. Thus, by omitting unnecessary, irrelevant details and focusing on the correct level of abstraction, it is possible to simplify a complex system or problem. Further, by considering the same problem at different levels of granularity, some details may be ignored. While this may result in approximate and inaccurate solutions (Zadeh, 1997), it also brings the benefit that such solutions can normally be obtained at a fraction of the cost. Granular computing provides true and natural representations of the real- world NSP. Through multiple-level representations, one can obtain a full understanding of a system.

4.4 Proposed Approach of Domain Transformation

The need for effective and efficient scheduling is becoming increasingly important. In private hospitals, this importance lies in the need to control hospital costs efficiently through optimising nursing salaries. The main challenge in nurse scheduling is to allocate specific shifts to nurses while ensuring minimum costs or penalties. In this study, we adopt a novel information granulation approach to nurse scheduling. Granular computing can be understood as 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 of building on information granules provides a simplified representation of the actual scheduling problem, but one with enhanced generality because of the degree of abstraction from non-critical information 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 a multilevel and multi-view representation of the problem (Yao, 2007). Information granules are collections of entities that usually originate at the numeric level and that are arranged together due to their similarity, functional or physical adjacency, indistinguishability or coherency. The information granules produced in this study are aggregated shift types and patterns representing shift sequences with soft constraints taken into consideration. This data processing creates a significant methodological development of nurse scheduling practice. The aggregation was inspired by insights from previous studies conducted by the authors (Bargiela, 1985; Peytchev *et al.*, 1996) and was formalised as a granular computing methodology (Bargiela and Pedrycz, 2003, 2004, 2008).

Our novel approach to the solution of the scheduling problem is referred to here as the *domain transformation approach* (in the context of information granulation). The domain transformation approach introduced in Baskaran, Bargiela and Qu (2012) departs from the orthodoxy of direct exploration of the space of schedules. It is an effective methodological approach to dealing with a complex NSP. Examples of the domain transformation approach in other applications include the subdivision of a problem domain into multiple sub-problems (e.g., the Danzig-Wolfe decomposition for solving linear programming problems), and the transformation from continuous to discrete functional description (e.g., the Z-transform converting time domain signals into discrete domain of trains of pulses).

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, Taylor, Villella, Foss, Ryner, Baker, and Hall, 2000). The combined computational effort of the domain transformation in addition to the solution of the transformed problem and the conversion from the transformed to the original domain is significantly smaller than what would be required for the solution in the original problem domain. The same problem-solving philosophy is proposed here in the context of nurse scheduling. Our approach can be summarised into a three-stage process:

1. Convert the problem from the original edlNR domain into a problem in the smaller DNR domain ('edlNR' and 'DNR' are explained below).

2. Solve the problem in the DNR domain.

3. Convert the DNR solution into a solution in the original edlNR domain.

As far as the NSP is concerned, a domain transformation approach could be applied successfully to produce feasible and good-quality nurse schedules. Information granulation (Bargiela and Pedrycz, 2002) serves as an important medium to simplify a problem that needs to be split into smaller sub-tasks. It provides an abstraction mechanism that reduces the overall conceptual burden in the original domain. A systematic approach that involves information granulation will create new data representation (patterns), which will provide valuable and meaningful information that could definitely ease the scheduling task. By having different sizes or representations of the information granules, a certain amount of details can be hidden during the problem solving. This offers an advantage in terms of reducing the complexities of NSPs.

Our main example in Chapter 5 is the ORTEC dataset used for the algorithm evaluation. The constraints on the ORTEC real-world dataset, listed in Section 3.1.4.1, are taken as the sample for the discussion in this chapter (Baskaran, Bargiela and Qu, 2014d, 2015).

4.4.1 Granulation of constraints

The ORTEC constraints are defined at a very detailed time resolution. In this form, they can be overwhelming given the number of shifts in the planning horizon and the number of nurses to be scheduled. Following from the observation of only three feasible night-shift patterns that satisfy ORTEC's hard constraints (as shown in Table 3.6), we propose to identify feasible 'merged-Day' (**D** shift) (as discussed in Section 4.4.2) in a similar way in this chapter. From the description of constraints in Section 3.1.4.1 (on ORTEC), we noticed that soft constraints 1 to 6 are used to identify the feasible 'merged- Day'. Later the soft constraints 7 to 10 are used for the conversion from the 'merged-Day' to the day shifts which consist of Early (E), Day (D) and Late (L). Figure 4.6 explains this, which involves abstraction from the detailed specification of the hard constraints and the development of new semantic entities of feasible day- and night-shift patterns to express NSPs. In the context of granular computing, we interpret this as the granulation of constraints.

4.4.2 Granulation on shifts types

The edlNR domain is the problem domain with five types of shift, as defined in Table 3.1. 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 applicable work regulations. This justifies considering these three shifts as one shift of type 'merged-Day' (**D** shift), which simplifies the scheduling task. By contrast, the Night (N) shift has a clearly distinct set of work regulations; therefore, N shift is retained. Similarly, the rest shift (R) is retained. It is thus proposed that the NSP be expressed at a more abstract level using just three types of shift: merged-Day (**D**), night (N) and rest (R).

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. 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 (five weeks) and involving five shift types, we have **16*5^35=4.6*10^25** different schedules in the overall edlNR solution space. By contrast, in the DNR solution space for schedules for the same number of nurses over the same duration but with only three shift types, the number of possible schedules is considerably smaller at **16*3^35=8*10^17**. Despite remaining computationally prohibitive, this represents a reduction by a factor of 108. Moreover, we notice the potential for additional domain transformation, with the associated computational gain. 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 adopting a week-at-a-time approach, whereby the problem space reduces to W*(N*S^7), which is 5*(**16*5^7**) in the edlNR domain and 5*(**16*3^7**) in the DNR domain. Thus, the reduction is from 80*5^7 to 80*3^7 (i.e., from 78,125 in the edlNR domain to 2187 in the DNR domain). Of these 625000 and 174960 sequences in the edlNR and DNR domains, 16,768 and 160, respectively, are feasible sequences. This has been generated by the pattern generator. Table 4.1 summarises the staff cover requirements for the corresponding edlNR shifts in the DNR domain during one week.

Table 4.1. Demand Summarisation

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 shift								

	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 shift						

4.4.2 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's duration that do not have any penalties associated with them and sequences that have some arbitrary penalties. We will call these sequences 'patterns' and use them as the basic building blocks for the schedules. The distinct value added by patterns is that, because of their prior assessment with regard to the satisfaction of soft constraints, they can be used in the scheduling process without the need for additional checking of the constraints. This is an advantage compared to scheduling with sequences of shifts, wherein a change of a single shift requires the evaluation of all constraints, both hard and soft. Figures4.2 and 4.3 provide examples of zero-cost and non-zero-cost patterns, respectively.

P1	N	N	R	R	D	D	D
…							
…							
P18	R	R	D	D	D	D	D

Figure 4.4. Zero-cost patterns—No violation of soft constraints.

P19	D	D	D	R	R	R	R

(Violation of SC- For nurses with availability of 36 hours per week, the number of shifts is within the range [4,5] per week)

…

…

P18	R	D	D	R	R	D	D

(Violation of SC- For nurses with availability of 36 hours per week, the length of series of shifts should be within the range of [4,6] per week)

Figure 4.5. Non-zero-cost pattern—Violation of soft constraints with cost 10.

The problem of scheduling shifts is therefore transformed into a 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 is found that the number of zero-cost patterns and patterns with other pre-specified costs is relatively small. In the scenario considered above, there are only 18 zero-cost patterns. This means that there are only **16*185=3*107** five-week schedules that can be constructed from 18 patterns for 16 nurses. This number of schedules can be completely enumerated within seconds on an average personal computer (PC). The combined two domain transformations have achieved an enormous reduction of the space of possible schedules by a factor 1018. In other words, one second of computations in the domain of patterns is equivalent to 100,000,000,000 years of computations in the edlNR domain. Of course, the solution of the scheduling problem in the domain of patterns needs to be converted back into the original edlNR domain. This involves a small computational effort, primarily concerned with the specific requirements with regard to the precedence of E-, D- and L-shifts, as summarised in Table 4.2.

Table 4.2. Interrelationship of the edl Shifts and DNR Domain, with Associated Costs.

	Succeeding Shifts				
Preceding 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

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

By combining the granulation of shift types and sequences into patterns, we can derive patterns of shifts in the DNR domain. Such patterns represent sets of patterns in the original edlNR domain. For example, the pattern **DD**NNRRR can be considered representative of nine patterns in the edlNR domain, as illustrated in Figure 4.6. We note that some sequences in the edlNR domain have a non-zero-cost due to interrelationships that cannot be captured in the DNR domain, as all edl shifts are represented by the same shift **D**. Table 4.2 shows the interrelationships of the edl shifts that give rise to some cost.

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 4.6. edlNR domain patterns and a representation of a DNR pattern.

We adopt the lowest cost sequences in the edlNR domain as an indication of the lower limit on the cost in the DNR domain, as illustrated in Figure 4.6. This means that we are open to revising the cost of the schedule upwards once we convert the solution from the DNR domain to the original edlNR domain.

4.4.3 Pattern 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 the hard constraints consecutively to eliminate all sequences that violate them (i.e., the 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 the implicit hard constraints derived from the combination of hard constraints. One such implicit hard constraint is illustrated in Table 4.3. The requirement for an uninterrupted sequence of three night shifts over the weekend and a maximum number of two to three night shifts during the week creates an implicit requirement that one may not work three consecutive night shifts on Monday–Wednesday or Tuesday–Thursday, because the remaining day would have just a single night shift. After applying the implicit hard constraints (see Table 4.3) the sequences satisfying these constraints are available for ranking with respect to their soft constraints violation cost.

Table 4.3. Night Sequences

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

We start ranking these sequences from the highest cost of 1000 to zero-cost. For the 36/32hours/week full-time nurses, there are 18 zero-cost patterns. Meanwhile, for the 20 hours/week part-time nurses, there are 15 zero-cost patterns. Table 4.4 itemises these zero-cost patterns for the different nurse contract types.

Table 4.4. Numbering of the Zero-Cost Patterns in the DNR Solution Space by Nurse Contract Type.

	36/32 hours FT nurses		20hours 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		

Note: FT=Full-time, PT=Part-time

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 36/32hours/week nurses is increased to 30, and the set of patterns for the 20hours/week nurses is increased to 26. By including progressively higher cost patterns, the set of available patterns will increase; however, since the objective of scheduling is to find the lowest cost schedule, there would typically be no need to consider higher cost patterns.

4.5 Important Novel Design of Domain Transformation Algorithm

Having defined the granular entities of shifts and patterns, we proceed to formulate the NSP as a recursive process with duration of one week. We postulate that a granulation of the scheduling horizon from individual days into weeks correlates closely with the granulation of constraints into patterns, and consequently provides a natural simplification of the scheduling problem from the full-scheduling horizon to a recursive-weekly-scheduling. The search space pertinent to finding a weekly schedule is significantly smaller than the search space for the corresponding full-scheduling-horizon task.

The feasible schedules for week 1 for the granular NSP can be enumerated relatively easily and be seen as a new granular search space for the solutions for subsequent weeks. Since these schedules capture the essence of hard constraints and the requirements for personnel cover (which are pre- defined for the scheduling horizon), this set is exhaustive and can only be reduced by the introduction of additional constraints. Owing to the above granulations, the scheduling task becomes spanned by the relatively small vector of feasible patterns and schedules for week 1. The overall idea outlined above is to reduce the problem complexity by granulating the search space. We now introduce one of the benchmark NSPs that we will tackle later in the chapter. As explained in Chapter 3, the ORTEC dataset is derived from real- world problems in intensive care units at a Dutch hospital. The main algorithm for implementing the schedule in the smaller space is as follows (Baskaran *et al.*, 2009):

1. Convert the problem space from {e, d, l, N, R} to the smaller space of {**D**, N, R}.

[The result of the granulation of the shift types is the reduction of the number of feasible shift patterns by several orders of magnitude. This point is illustrated in Figure 4.6, where a large number of patterns are represented by the single pattern DDNNRRR. The size reduction brought by this granulation of shift types becomes even more pronounced when considering several weeks.]

2. Identify all the shift patterns with '0' cost for week 1 in the {N, d, R} space.

[This is carried out by the offline processing of all hard and soft constraints. Although some hard constraints, such as constraints 4 and 5 as listed in Section 3.1.2.1.1 are specified in the context of the five-week scheduling horizon, they can be easily interpreted in the context of a one-week scheduling horizon with an appropriate additional constraint imposed onto subsequent weeks. The resulting set of 18 zero-cost patterns for full-time, 36/32hours/week nurses and 15 zero-cost patterns for part-time, 20hour/week nurses are listed in Table 4.4.]

3. Within the space of feasible shift patterns for week 1, we identify sets of patterns that satisfy the personnel cover requirements, which have cardinality equal to the number of nurses on the ward.

[By specifying the differing cardinality of the sets, we can generate feasible schedules for different numbers of staff. We can also easily take into account the different requirements for personnel cover.]

4. Extending the scheduling horizon from week 'w' to 'w+1', we identify which of the feasible shift patterns need to be excluded for each specific pattern deployed in week 'w'.

[This step ensures that the hard constraints on the number of consecutive shifts on the interface of two weeks and the number of night shifts for a single nurse are satisfied. Note that this step reduces the search space of feasible schedules for week 'w+1'.]

5. Given the set of feasible schedules identified in step 3 above, and the set of feasible patterns for the week 'w+1' identified at step 4, we perform a search in this schedule space to find feasible schedules for week 'w+1'.

[The elimination of some of the patterns as a possible continuation after a specific pattern in week 'w' (as highlighted in step 4) reduces the number of possible schedules that can be generated from this smaller set of patterns in week 'w+1'. Consequently, some of the schedules that were feasible in week 'w' will not be feasible in week 'w+1'. An important conclusion is that a set of feasible schedules for week 'w+1' is contained in the set of feasible schedules for week 'w'. This provides an upper limit on the extent of the search that is needed to identify all feasible schedules in week 'w+1'.]

6. The process of scheduling personnel for subsequent weeks is implemented by repeating steps 4 and 5 above until the set of feasible patterns is empty if the planning horizon has been reached.

4.6 Pattern Generator

Using a pattern generator (see Figure 4.7) was very important in this domain transformation approach to NSPs. This generator class contains the logic to generate all possible patterns to construct the schedule for a single week. In Section 4.4.3, we described the pattern construction. As revealed in Section 3.1, different hospitals will have their own specific operating policies concerning the hard and soft constraints. Based on these constraints, we generate the patterns without any cost (zero-cost patterns) and those with cost (non-zero-cost patterns). The cost comes from the soft constraints, which are each assigned a weight. Using a pattern generator, each pattern is passed through the selected hard and soft constraint checkers. The pattern generator class also has the logic to expand the patterns from DNR to edlNR. Along with the pattern, the generator stores which soft constraints are violated, and it can return the total cost for a pattern.

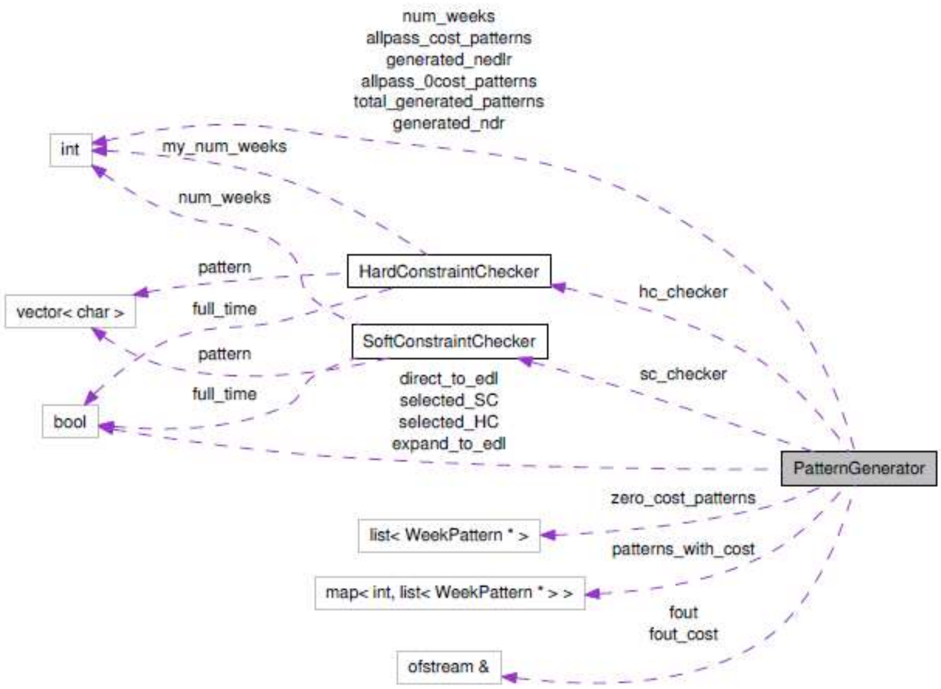


Figure 4.7. Pattern generator class.

4.7 Summary

Nurses' performance in a hospital can be managed and coordinated with the aid of nurse scheduling. We use the domain transformation method introduced in as a practical illustration of the information granulation methodology to generate multiple feasible low-cost rosters, which are evaluated with simulation. Domain transformation is an approach to solving complex problems that relies on a well-justified simplification of the original problem. We deal with several corresponding problem descriptions at different levels of generality or accuracy. The more general descriptions serve to facilitate an approximate problem solution in a smaller search domain and more detailed representations preserve the possibility of refinement of the solutions. We subdivided the problem into smaller sub-problems in a systematic way and capable to reproduce the result. This approach is able to conquer solution easily by avoiding random search. Conversely, in other methods, some failed to reproduce results, and produce inconsistent performance, some works best on some datasets but failed to repeat the good characteristics on other datasets. It represents a departure from the conventional one-shift-at-a-time scheduling approach.

It offers the advantage of efficient and easily understandable solutions, as well as offering deterministic reproducibility of the results. The models and algorithms involved in generating the schedule should have a strong yet flexible structure to adapt to the various unexpected situations that can occur in the hospital setting (discussed further in Chapter 5 and Appendix C). The previous state-of-the-art never used information granulation (domain transformation approach), thus dealing with a lot of cross-referencing and checking of data. We note, however, that this cannot guarantee the global optimum will be achieved.

Chapter 5: Algorithm Evaluation

This chapter explains the three different techniques implemented in the main algorithm to solve NSPs using the domain transformation approach with real-world datasets. This chapter also discusses the exhaustive analysis conducted using the three techniques by presenting the results obtained from the experiments. These techniques were able to find the adequate coverage breakpoint in a wide variety of real-world NSPs. The aim of this analysis is to: a) propose standard configurations for handling small and large datasets, b) compare the complexity of the obtained solutions and the time required to achieve them, c) compare these domain transformation approaches using real-world datasets and d) determine possible areas of improvement in domain transformation. Moreover, in the final validation stage, our approach was computationally up to nine times faster than the best-known result for challenging real-world nurse scheduling datasets. Each algorithm is also tested using data collected from schedules actually worked at Kajang Hospital, a public hospital in Malaysia. This allows the results obtained using the domain transformation approach to be compared with the manually generated solutions developed by hospital staff.

5.1 Introduction

Based on the main algorithm in Section 4.5, we have tried to plug many techniques to evaluate the performance of the algorithm. In this chapter we present three main technique that produced good computational results based on the real-world benchmark NSP.

5.2 Technique 1 (T1)

Scheduling involves selecting one out of a set of available patterns for each nurse. The selection is subject to the requirement that the cover specified in Table 4.1 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 remains very large: the selection of one out of 18 patterns in each of the five weeks of the planning horizon amounts to 18^5=1889568 schedules. We therefore proceed with the further simplification of the NSP from the full-scheduling-horizon to a recursive-weekly-scheduling. The proposed procedure is deployed in the following stages (Baskaran *et al.*, 2014c):

Step 1: Scheduling week1schedule (DNR)

Step 2: Expand schedule in N weeks (DNR)

Step 3: Convert **D**NR to edlNR schedule.

These steps are now discussed in turn.

5.2.1 Scheduling week1schedule (DNR)

In step 1, once we have identified the zero-cost patterns as in Table 4.4, we construct the week 1 schedule in the DNR domain. We associate patterns based on zero-cost patterns with nurses based on full-time or part- time schedules. This is called schedule set. Schedule set is stored in a vector object (array). To construct the schedule, we must consider some specific measures. First, we consider the shifts that are the most difficult to schedule; in our case, this is the night shift. This is also the most important shift, with a cost of 1000 if the length of the shifts is not within the range of [2, 3]. Subsequently, we place the day-only patterns in the array of day patterns. The result is 18 zero-cost patterns for the 32/36 hours/week full-time nurses and 15 zero-cost patterns for the 20 hours/week part-time nurses. As the full- time nurses working both 32 and 36 hours have the same patterns, they fall in the same category: set A. The 20 hours/week part-time nurses are set B. Scheduling can then proceed for the 13 nurses using patterns from set A, and three nurses using patterns from set B. We select one assignment of night patterns based on zero-cost patterns. As shown in Table 5.1, the night shifts (N) are grouped together in pairs or triples at fixed days. We can calculate the number of ways to combine them into patterns using mathematical combinations as described in Table 5.2. The objective is to satisfy the demand of 1111111—one nurse every day (as referred to in Table 4.1) for the night shift.

Table 5.1. Zero-Cost Patterns with Night Shifts

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

Table 5.2. Mathematical Combinations of Night Shifts Based on Zero-Cost Patterns

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

Note: M T=Monday, Tuesday; W T=Wednesday, Thursday; F S S=Friday, Saturday, Sunday

The notation for the combinations is given as:

$$_nC_r \; :$$

which means the number of combinations of *n* items taking *r* items at a time.

For the full-time nurses, for M and T, we have only two night patterns: A1 and A2. For W and T, we have only the A3 pattern. For F, S and S, we have five patterns: A4 to A8. If we are choosing for just the full-time nurses, then we select 1 from 2 for M and T; 1 for W and T; and 1 from 5 for F, S and S. Similarly, when selecting for both full- and part-time nurses, for M and T, we select 1 pair from 3; for W and T, we select 1 from 3; and for F, S and S, we select 1 from 6. The total number of combinations of patterns is thus 3C1 x 3C1 x 6C1=54.

Later, the day shifts are assigned as illustrated in Table 5.3. Here, we assign the day shifts on weekends using the zero-cost patterns. To satisfy the demand for total D of 9999966 (as referred to Table 4.1), with nine nurses on weekdays and six nurses on weekends, blocks A12 to A18 are chosen first. These are the patterns of days on weekends. Thus, the total number is 7C5=21. Next, the remaining shifts on weekends are assigned. To satisfy the demand for total R of 9999966, patterns of rest on weekends are chosen. Firstly, blocks A9 to A11 are selected, followed by blocks B5 to B12. If no zero-cost assignments are found, the number of patterns is increased by including non-zero-cost patterns.

Table 5.3 Zero-Cost Patterns with Day Shifts only

	36/32hours FT nurses	M T W T F S S		20hours 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 5.4. Switching Patterns Based on Zero-Cost Patterns

Replacement of patterns		**Day of the week**
More shifts	Fewer shifts	
A17	A16	Monday
A9	C9	Monday
A15	A14	Tuesday

If demand is over-satisfied, we use the switching patterns, as shown in Table 5.4. We can increase the number of replacements by also including the switching patterns of non-zero-cost patterns. This can be done by move the shift (more shift) or less the shift (fewer shift) according to the days. A complete zero-cost pattern switching is shown in Figure 5.1. The different shift counts indicate the number of day shifts in the pattern. For example, we can move a shift from A17 to A16 or vice versa. This means we are moving from a five-day shift to a four-day shift on a Monday.

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 5.1. Switching patterns based on shifts of zero-cost patterns according to days.

Table 5.5 shows an example of switching zero-cost patterns for the week 1 scheduling. Initially, placing the night and day patterns failed to satisfy the demand of 9999966—the result was 9979966. This was corrected by using the switching patterns: shifting A12 to pattern A13 for two nurses.

Table 5.5. Example of Week 1 Schedule using Zero-Cost Patterns

Number of nurses	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 Cover 1 of D		**2 1 0 0 0 1 1**		**2 1 0 0 0 11**
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 Cover 2 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**

5.2.2 Expand schedule for N+1 Week (DNR)

Based on the week N schedule detailed in the previous section, we next construct the N+1 week schedule in the 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 at week 1 is the night shifts constraint. According to hard constraints 4 and 9, listed in Section 3.1.2.1.1, the maximum number of night-shift a nurses can work is 2 to 3. Two sets can be used in this placement: 1) we can choose one of three night-shift patterns from pattern A, or 2) we can choose two night-shift patterns from pattern A and one night- shift pattern from pattern B. Figure 5.2 illustrates the placement of night shifts in the schedule. Indirectly, this night-shift placement will satisfy hard constraint 11 in Section 3.1.2.1.1.

As an example, we now calculate the night shifts per week over the five-week scheduling period. Referring to Figure 5.2, night shifts placed using pattern A are 3+3+2+2+2=12 per period of five consecutive weeks, while night shifts placed using pattern B are 1+1+1=3.

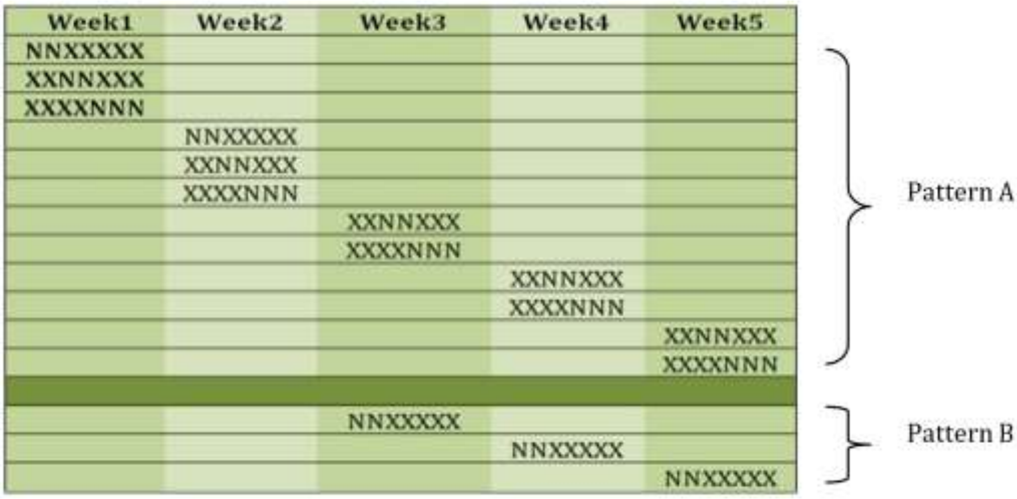


Figure 5.2. Night placement over five weeks.

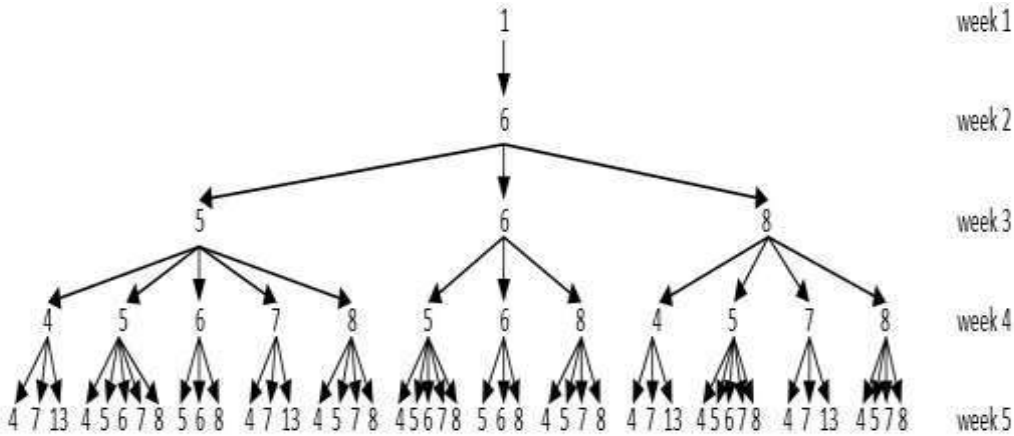
Next, we need to satisfy the hard constraints 5, 6 and 9, as given in Section 4.5.1.1.1. Table 5.6 shows a good sample of patterns that need to be considered when selecting schedules from the generated zero-cost week 1 schedules for N weeks. For example, patterns of A4 to A8 need to be followed by a minimum of two days' rest. Accordingly, we see that pattern A16 satisfies hard constraint 6. Further, since pattern A16 has four **D**s (workdays), this pattern can be followed by pattern A12. In this way, it satisfies hard constraint 9 because the number of consecutive shifts is at most six. The example shown in Table 5.6 also satisfies hard constraint 5, with two weekends off duty.

Table 5.6. Possible Five-Week Patterns

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

To find all possible five-week schedules for each nurse, the week 1 zero-cost schedules are converted into tree structures, as shown in Figure 5.3. In this figure, we see that all week 1 zero-cost schedules are assigned a number for marking purposes. These numbers are used to check the feasible patterns of shifts that can follow a pattern from a previous week.

Feasibility is mainly checked according to hard constraints 5, 6, 9 and the night-shift constraints. In Figure 5.4, the example of the possible shift patterns over five weeks for Nurse 1begins with a schedule generated for week 1 (see Table 5.5): in this case, pattern DRRRNNN, marked as 1. This pattern can only be followed with RRDDDDD (marked 6) for week 2. Moving to week 3 for nurse 1, patterns 5, 6 and 8 can follow. In the case of multiple pattern options for a week, the pattern assigned the lowest number is listed first. These steps are repeated for weeks 4 and 5. This simplification is important for recursive- weekly-scheduling, both to limit the use of large loops and to ease computational time.



The numbers and the patterns relationship

1	6	5	4	4
DRRRNNN	RRDDDDD	DRRDDDD	DDRRRDD	DDRRRDD

Figure 5.3. Possible patterns of shifts for a five-week period, arranged as a subset list.

In this study, building schedules using only zero-cost patterns could be achieved only until week 3 following all subset lists of which Figure 5.3 gives one example. From week 4 onwards, other low-cost patterns are incorporated to satisfy the demand. Checking is done on switching patterns to determine whether these patterns could be used to improve the initial solutions. Our aim at this stage is to decrease the penalty cost from the use of non-zero-cost patterns.

5.2.3 Convert DNR to edlNR schedule

In step 3, once the DNR domain schedule is constructed, we convert this result to the edlNR domain. First, we obtain the schedule array from determining whether it is a four- or five-week schedule. Next, we find the '**D**' index in the array. For example, if we have (RRDDDDD RRRDDDD) as the array, the position of '**D**' is (3 4 5 6 7 11 12 13 14). We convert this '**D**' to D, L, E, making the permutation for shift L C^9_3. After selecting 3 L shift, the 6 D shift remains. Hence, the permutation for shift E is C^6_3 . All possible permutations of edl for some day equal $C^9_3\,C^6_3$. This loop is continued until the demand is satisfied. Figure 5.4 shows the best schedule using this method; it was generated in only 45 seconds, at a cost of 100.

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'	RRREELL			
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 5.4. Best result schedule for the NSP.

5.3 Technique 2 (T2)

Another method applied in this thesis is integer programming, which is specific case of linear programming that constrains variables to integer values (Ballnski, 1965). In particular, we use a branch-and-bound (IP- BB) algorithm (Baskaran, Bargiela and Qu 2013b, 2014a, 2014b, 2014d).

Integer programming seeks to solve problems that require integer solutions.

To specify the problem, the objective is to minimise the value of individual variables.

We formulate the problem of a two-week scheduling period with an integer-programming model that can be altered to adapt to any other problem with different constraints. The above patterns for the two weeks scheduling 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 5.5. This will allow for the representation of three states.

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

Figure 5.5. Binary pattern matrix.

This binary pattern matrix will be called **B**. This matrix is replicated for each nurse, and the combined pattern matrix, **C**, is shown in Figure 5.6.

C=[

Day	-	-	-	-	-	-	Night		-	-	-	-	-
1	1	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	1	1	1	1	1	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	1	1	1
1	1	0	0	0	0	0	0	0	0	0	1	1	1
0	0	0	1	0	0	0	0	0	0	0	1	1	1
1	0	0	1	0	0	0	0	0	0	0	1	1	1
0	0	1	1	0	0	0	0	0	0	0	1	1	1
1	1	1	1	0	0	0	0	0	0	0	0	0	0
…….													
…….													
1	1	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	1	1	1	1	1	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	1	1	1
1	1	0	0	0	0	0	0	0	0	0	1	1	1
0	0	0	1	0	0	0	0	0	0	0	1	1	1
1	0	0	1	0	0	0	0	0	0	0	1	1	1
0	0	1	1	0	0	0	0	0	0	0	1	1	1
1	1	1	1	0	0	0	0	0	0	0	0	0	0

];

Figure 5.6. 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’ (5.1)

Where **x** is the unknown binary vector, representing a solution to the scheduling problem, and **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:

A’ * x <=b’ (5.2)

Where **A** is a matrix where the number of columns corresponds to the number of nurses and the number of rows is equal to the product of the number of nurses **n** and the number of patterns **p**, represented as:

m=n * p (5.3)

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

Figure 5.7. Example of rows=number of nurses (say 15) * number of patterns (say 18).

Figure 5.7 represents matrix A for n=15 and p=18. The vector **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 assigned in week 1. The objective of the optimisation of the scheduling defined as trying to satisfy the cover requirement with the minimum number of nurses. This expressed simply as:

Min NP * x (5.4)

where NP is a vector of 1s of size **m**. The cost function defined as a sum of penalties representing a nurse working a given shift on a day. Therefore, our aim is either to minimise the penalty subject to a nurse should have no shifts or at least two shifts (complete weekend) and avoid sequences of shifts with length of one for all nurses.

5.3.1 Branch-and-bound

Branch-and-bound (algorithms, see Lawler and Wood, 1966 for examples), methods implicitly enumerate all possible solutions to an integer-programming model. The basic concept underlying the branch-and-bound 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 increasingly 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 (Fi) and the one created most recently was selected.

2. *Fathoming:* If the sub-problems were not feasible, they were discarded.

3. *Bounding:* The new sub-problems were solved and a lower bound b(Fi) for the sub-problem was computed.

4. *Fathoming:* For each new sub-problem, if b(Fi) ≥ U, then the current best upper solution was bound and the fathomed sub-problem was discarded.

5. *Optimality test/partitioning:* If no unfathomed sub-problems remain, either they were obtained at an optimal solution to the sub-problem (stop), or the corresponding problem was broken into further sub-problems to perform another iteration.

5.3.2 User interface

The Create Patterns button runs the generating pattern code for creating patterns. The Create Schedule button reads input.txt for configuration and a common patterns document and stores that information in a database. It then creates the schedules. When the Create Schedule button is pressed, a layout such as in Figure 5.8 appears.

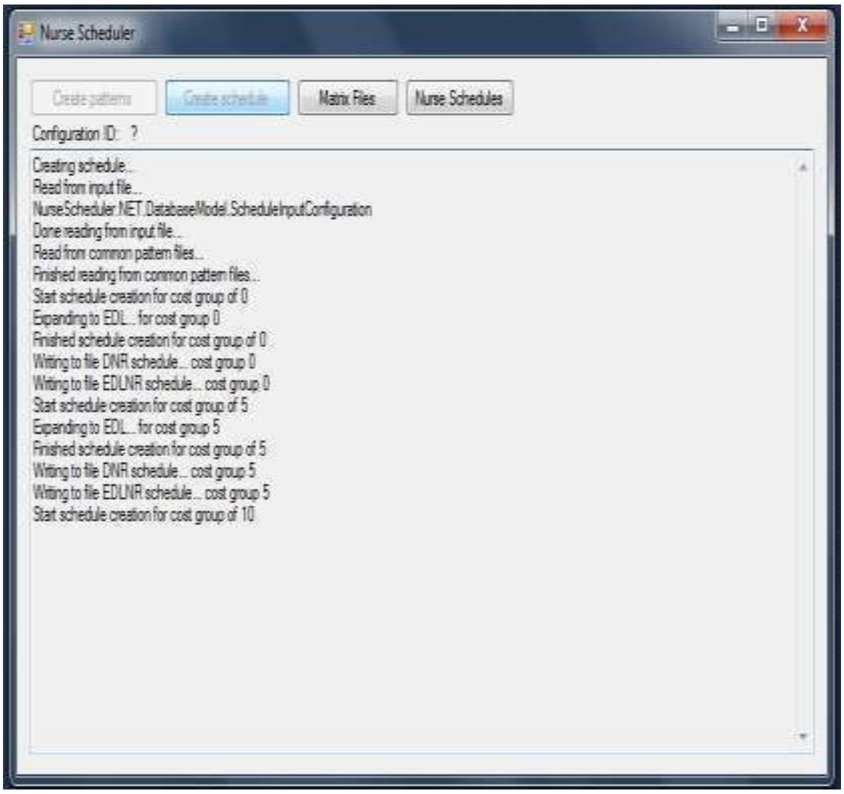
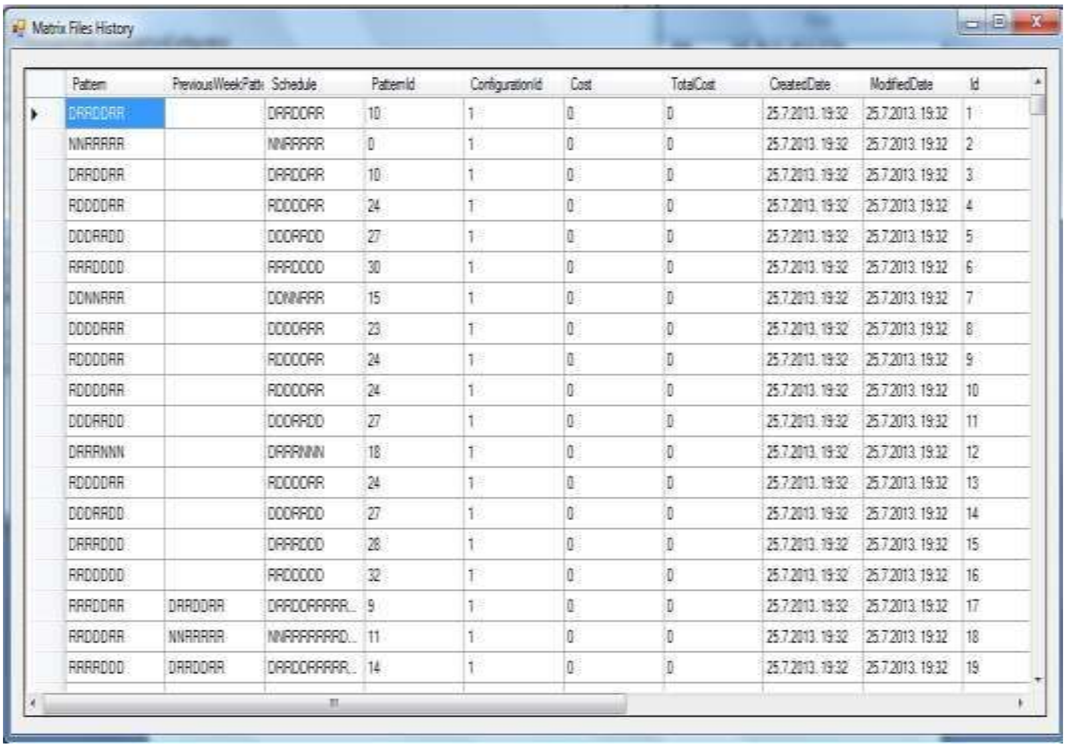


Figure 5.8. Layout of window opened by the Create Schedule button.

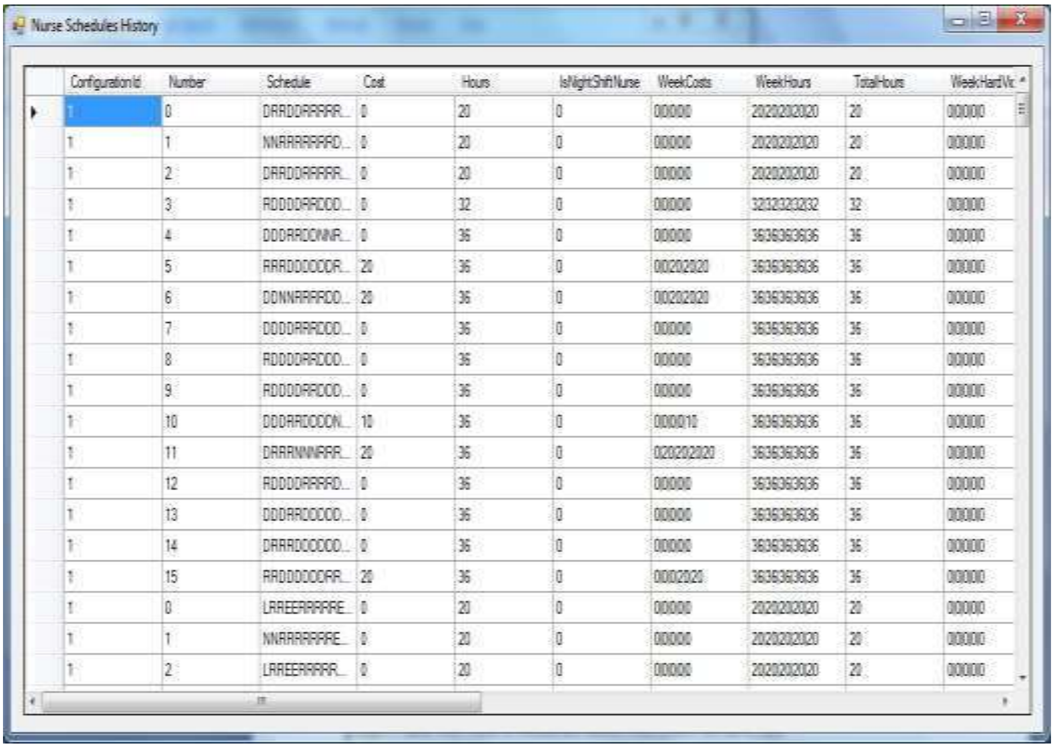
The Matrix Files history shown in Figure 5.9 illustrates the schedules generated by weeks and stored in a Matrix Files table created with the Create Schedule button. All patterns used in this process are saved in the history for all weeks and all costs used. This history does not use matrices or integer programming.



Pattern	PreviousWeekPatt	Schedule	PatternId	ConfigurationId	Cost	TotalCost	CreatedDate	ModifiedDate	Id
DRRDDRR		DRRDDRR	10	1	0	0	25.7.2013. 19:32	25.7.2013. 19:32	1
NNRRRRR		NNRRRRR	0	1	0	0	25.7.2013. 19:32	25.7.2013. 19:32	2
DRRDDRR		DRRDDRR	10	1	0	0	25.7.2013. 19:32	25.7.2013. 19:32	3
RDDDDRR		RDDDDRR	24	1	0	0	25.7.2013. 19:32	25.7.2013. 19:32	4
DDDRRDD		DDDRRDD	27	1	0	0	25.7.2013. 19:32	25.7.2013. 19:32	5
RRRDDDD		RRRDDDD	30	1	0	0	25.7.2013. 19:32	25.7.2013. 19:32	6
DDNNRRR		DDNNRRR	15	1	0	0	25.7.2013. 19:32	25.7.2013. 19:32	7
DDDDRRR		DDDDRRR	23	1	0	0	25.7.2013. 19:32	25.7.2013. 19:32	8
RDDDDRR		RDDDDRR	24	1	0	0	25.7.2013. 19:32	25.7.2013. 19:32	9
RDDDDRR		RDDDDRR	24	1	0	0	25.7.2013. 19:32	25.7.2013. 19:32	10
DDDRRDD		DDDRRDD	27	1	0	0	25.7.2013. 19:32	25.7.2013. 19:32	11
DRRRNNN		DRRRNNN	18	1	0	0	25.7.2013. 19:32	25.7.2013. 19:32	12
RDDDDRR		RDDDDRR	24	1	0	0	25.7.2013. 19:32	25.7.2013. 19:32	13
DDDRRDD		DDDRRDD	27	1	0	0	25.7.2013. 19:32	25.7.2013. 19:32	14
DRRRDDD		DRRRDDD	28	1	0	0	25.7.2013. 19:32	25.7.2013. 19:32	15
RRDDDDD		RRDDDDD	32	1	0	0	25.7.2013. 19:32	25.7.2013. 19:32	16
RRRDDRR	DRRDDRR	DRRDDRRRRR...	9	1	0	0	25.7.2013. 19:32	25.7.2013. 19:32	17
RRDDDRR	NNRRRRR	NNRRRRRRRD...	11	1	0	0	25.7.2013. 19:32	25.7.2013. 19:32	18
RRRRDDD	DRRDDRR	DRRDDRRRRR...	14	1	0	0	25.7.2013. 19:32	25.7.2013. 19:32	19

Figure 5.9. Layout of schedule generated by weeks.

The Nurse Schedules button creates the final nurse schedules in both the DNR and edlNR domains for all three cost groups (see Figure 5.10).



ConfigurationId	Number	Schedule	Cost	Hours	IsNightShiftNurse	WeekCosts	WeekHours	TotalHours	WeekHardVi
1	0	DRRDDRRRRR...	0	20	0	00000	2020202020	20	00000
1	1	NNRRRRRRRD...	0	20	0	00000	2020202020	20	00000
1	2	DRRDDRRRRR...	0	20	0	00000	2020202020	20	00000
1	3	RDDDDRRDDD...	0	32	0	00000	3232323232	32	00000
1	4	DDDRRDDNNR...	0	36	0	00000	3636363636	36	00000
1	5	RRRDDDDDDR...	20	36	0	00202020	3636363636	36	00000
1	6	DDNNRRRRDD...	20	36	0	00202020	3636363636	36	00000
1	7	DDDDRRRDDD...	0	36	0	00000	3636363636	36	00000
1	8	RDDDDRRDDD...	0	36	0	00000	3636363636	36	00000
1	9	RDDDDRRDDD...	0	36	0	00000	3636363636	36	00000
1	10	DDDRRDDDDN...	10	36	0	000010	3636363636	36	00000
1	11	DRRRNNNRRR...	20	36	0	020202020	3636363636	36	00000
1	12	RDDDDRRRRD...	0	36	0	00000	3636363636	36	00000
1	13	DDDRRDDDDD...	0	36	0	00000	3636363636	36	00000
1	14	DRRRDDDDDD...	0	36	0	00000	3636363636	36	00000
1	15	RRDDDDDDRR...	20	36	0	0002020	3636363636	36	00000
1	0	LRREERRRRE...	0	20	0	00000	2020202020	20	00000
1	1	NNRRRRRRRE...	0	20	0	00000	2020202020	20	00000
1	2	LRREERRRR...	0	20	0	00000	2020202020	20	00000

Figure 5.10. Layout of schedule generated in both domains.

A new configuration ID is assigned each time Create Schedule is started. This ID can be used as a reference number in tables with nurse schedules (see Figure 5.11).

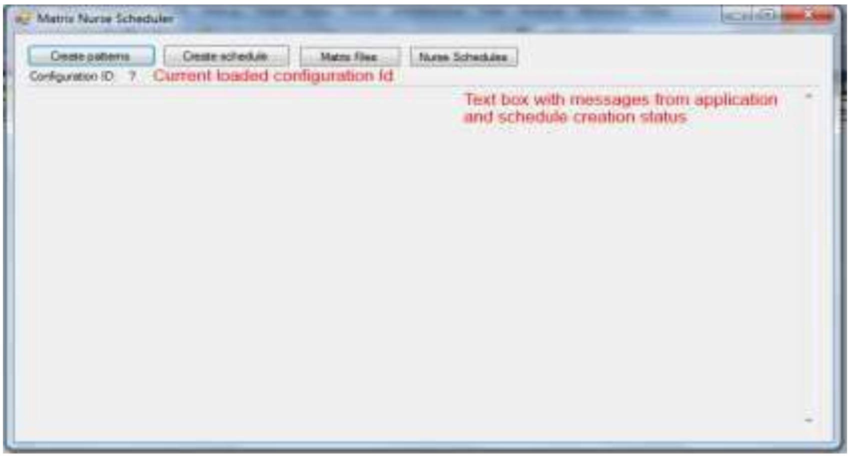


Figure 5.11. Layout of interface and the position of the configuration ID.

5.3.3 IP-BB computational result

In the DNR domain for the week 1 schedules, IP-BB managed to find zero-cost patterns only, satisfying the demand of '9999966'; that is, nine nurses with the D shift each weekday and six nurses with the D shift each weekend day. Demand of nights '1111111' is also satisfied, as shown in Figure 5.12.

```
#Number of nurses = 16

###### OUTPUT SCHEDULE (DNR format) ######

---------------------------------
Computed Schedule for 1 week(s)
Start date: 1-1-2003
End date:   31-1-2003
---------------------------------

Days->       | MTWTFSS  ctot
-----------------------------
Nurse  1(20): RDDRRRR    0
Nurse  2(20): RRRDDRR    0
Nurse  3(20): RRRDDRR    0
Nurse  4(32): RDDDDRR    0
Nurse  5(36): RDDDDRR    0
Nurse  6(36): NNRRRDD    0
Nurse  7(36): DDDDDRR    0
Nurse  8(36): DDDRRDD    0
Nurse  9(36): DRRDDDD    0
Nurse 10(36): DDDDDRR    0
Nurse 11(36): RRDDDDD    0
Nurse 12(36): DDDRRDD    0
Nurse 13(36): DRRRDDD    0
Nurse 14(36): DRRRNNN    0
Nurse 15(36): DDNNRRR    0
Nurse 16(36): DDDDRRR    0

Verifying total nurses available each day:
------------------------------------------

Total D:      9999966

Total N:      1111111
```

Figure 5.12. Week 1 IP-BB schedule in the DNR domain.

Upon conversion to the original domain of edlNR for a month, non-zero-cost patterns were incorporated, increasing the cost of the schedule to 90, as shown in Figure 5.13.

```
---------------------------------
Computed Schedule for 4 week(s)
Start date: 1-1-2003
End date:   31-1-2003
---------------------------------

Days->         | MTWTFSS ctot | MTWTFSS ctot | MTWTFSS ctot  | MTWTFSS ctot  | MTWTFSS ctot

Nurse  1(20): --ERRRR    0 | RRRRNNN    0 | RRREERR    0 | RRRRREE    0 | RRRRE--    0
Nurse  2(20): --REERR    0 | EERRRRR    0 | NNRRRRR    0 | EERRRRR    0 | RRRRR--    0
Nurse  3(20): --REERR    0 | RRREERR    0 | RRRRNNN    0 | RREEERR    0 | REERR--    0
Nurse  4(32): --ELLRR    0 | EELLRRR    0 | RREEELL    0 | LRRRNNN    0 | RRREE--   10
Nurse  5(36): --ELLRR    0 | RREEELL    0 | LRRRLLL    0 | NNRRREE    0 | ELLLR--    0
Nurse  6(36): --RRREE    0 | ELLLRRR    0 | REELLRR    0 | REELLRR    0 | EELLL--    0
Nurse  7(36): --LLLRR    0 | REELLRR    0 | RREEEDD    0 | DRREELL    0 | NNRRL--    0
Nurse  8(36): --LRREE    0 | LLNNRRR    0 | RDLLLRR    0 | RREEELL    0 | LRREE--    0
Nurse  9(36): --REELL    0 | NNRREEL    0 | LLRRREE    0 | ELLLRRR    0 | RREED--    0
Nurse 10(36): --DDDRR    0 | DDDDDRR    0 | EEDDDRR    0 | EELLLRR    0 | LLNNR--    0
Nurse 11(36): --DDDLL    0 | LRRRDDD    0 | LLLRRRR   10 | RDDDDRR   10 | EERRN--   20
Nurse 12(36): --LRRDD    0 | LLLRRDD    0 | DDNNRRR    0 | RDDDDRR    0 | RRDDD--    0
Nurse 13(36): --RRDDD    0 | DDDRREE    0 | ELLLRRR    0 | LLNNRRR    0 | DDLLL--    0
Nurse 14(36): --RRNNN    0 | RREEDRR   20 | EEDRREE   20 | LLLRRDD   20 | DDDDR--   20
Nurse 15(36): --NNRRR    0 | DDDDLRR   10 | DDDDDRR   10 | DDDDDRR   10 | DDDDD--   10
Nurse 16(36): --DDRRR    0 | RRRDLLE    5 | DRRDDDD   15 | DRRRLDD   30 | LLERR--   30

Verifying total nurses available each day:
------------------------------------------

Total E:       0033322      | 3333322      | 3333322      | 3333322      | 3333300

Total D:       0033322      | 3333322      | 3333322      | 3333322      | 3333300

Total L:       0033322      | 3333322      | 3333322      | 3333322      | 3333300

Total N:       0011111      | 1111111      | 1111111      | 1111111      | 1111100
```

Figure 5.13. One month schedule on the edlNR domain.

```
---------------------------------
Computed Schedule for 5 week(s)
Start date: 1-10-2012
End date:   4-11-2012
---------------------------------

Days->         | MTWTFSS ctot | MTWTFSS ctot | MTWTFSS ctot | MTWTFSS ctot | MTWTFSS ctot

Nurse  1(20): REERRRR    0 | RRRRNNN    0 | RRREERR    0 | RRRRREE    0 | RRRREEE    0
Nurse  2(20): RRREERR    0 | EERRRRR    0 | NNRRRRR    0 | EERRRRR    0 | RRRRREE    0
Nurse  3(20): RRREERR    0 | RRREERR    0 | RRRRNNN    0 | RREEERR    0 | REERRRR    0
Nurse  4(32): REELLRR    0 | EELLRRR    0 | RREEELL    0 | LRRRNNN    0 | RRREELL   10
Nurse  5(36): REELLRR    0 | RREEELL    0 | LRRRLLL    0 | NNRRREE    0 | ELLLRRR    0
Nurse  6(36): NNRRREE    0 | ELLLRRR    0 | REELLRR    0 | REELLRR    0 | EELLLRR    0
Nurse  7(36): DDLLLRR    0 | REELLRR    0 | RREEEDD    0 | DRREELL    0 | NNRRLLL    0
Nurse  8(36): LLLRREE    0 | LLNNRRR    0 | RDLLLRR    0 | RREEELL    0 | LRREEDD    0
Nurse  9(36): LRREELL    0 | NNRREEL    0 | LLRRREE    0 | ELLLRRR    0 | RREEDRR   20
Nurse 10(36): EDDDDRR    0 | DDDDDRR    0 | EEDDDRR    0 | EELLLRR    0 | LLNNRRR    0
Nurse 11(36): RRDDDLL    0 | LRRRDDD    0 | LLLRRRR   10 | RDDDDRR   10 | EERRNNN   20
Nurse 12(36): LLLRRDD    0 | LLLRRDD    0 | DDNNRRR    0 | RDDDDRR    0 | RRDDDDD    0
Nurse 13(36): ERRRDDD    0 | DDDRREE    0 | ELLLRRR    0 | LLNNRRR    0 | DDLLLRR    0
Nurse 14(36): ERRRNNN    0 | RREEDRR   20 | EEDRREE   20 | LLLRRDD   20 | DDDDRRR   20
Nurse 15(36): DDNNRRR    0 | DDDDLRR   10 | DDDDDRR   10 | DDDDDRR   10 | DDDDDRR   10
Nurse 16(36): DLDDRRR   15 | RRRDLLE   20 | DRRDDDD   30 | DRRRLDD   30 | LLERRRR   40

Verifying total nurses available each day:
------------------------------------------

Total E:       3333322      | 3333322      | 3333322      | 3333322      | 3333322

Total D:       3333322      | 3333322      | 3333322      | 3333322      | 3333322

Total L:       3333322      | 3333322      | 3333322      | 3333322      | 3333322

Total N:       1111111      | 1111111      | 1111111      | 1111111      | 1111111
```

Figure 5.14. Five-week schedule in the edlNR domain using IP-BB.

Figure 5.13 shows that from week 2 onwards, patterns of cost 5, 10 and 20 were incorporated. As the weeks progressed, at least one non-zero-cost pattern was incorporated to fulfil the demand of the nurses in the schedule. However, when the schedule was extended to five weeks, as in Figure 5.14, non-zero-cost patterns were also incorporated in week 1. This is because it was necessary to fulfil the demand for nurses for the five-week duration using patterns able to satisfy the continuity from one week to the next.

5.4 Technique 3 (T3)

Addition to the above two methods, we have also used the greedy technique in domain transformation to solve NSPs problem (Baskaran, Bargiela and Qu, 2015). The initial solution is computed by means of a greedy algorithm. A complete solution for this problem is defined for each day of the month and for each nurse on the shift associated.

5.4.1 Step 1: Obtain the week 1cost groups

First, weekly sequences of patterns consisting of high-quality shift sequences are generated (Baskaran *et al.*, 2009). They can then be used to schedule the week 1 patterns in the DNR domain. We group the patterns into three categories: cost 0, cost 5 and cost 10, according to the full-time and part- time nurses' patterns. We list all the night patterns (which may or may not include day patterns) as well as all of the day-only patterns. We also list the schedule set that contains the week 1 schedule and nurse information. The schedule set is tied to the nurses, so the number of elements in a schedule set is the same as the number of nurses. Further, the schedule set contains nurse information, such as costs, constraint violations and hours.

5.4.2 Step 2: Generate the week 1schedule

The schedule set comprises associated patterns based on zero-cost patterns with nurses, based on full- or part-time schedules. The schedule set is stored in an array. The night shift is the most important shift in the NSP, as it involves a number of hard constraints (HC5, HC7 and HC9) as well as highly weighted soft constraints (SC2, SC3). Therefore, as it is difficult to satisfy the night shifts, we begin by assigning the night patterns in an array. A greedy algorithm examines all the days with the aim to guarantee the requested coverage for each shift. This is done by selecting, for each given shift, the best nurse to be assigned to that shift. Thus, day-only patterns are placed in the array. Demand is calculated to check the remaining number of day and night shifts to be filled per day.

The remaining demand is calculated by looping all nurses for each day, counting the number of shifts and subtracting them from the total allowed demand for each type of shift. We use two nested loops (nurses and days) for counting shifts used in total for a day. Then, when calculating the difference, *if*s (for shift types) and *for*s (for days) are used, separating shift types when calculating the difference, per day. For example, if for some days D is allowed (DNR mode), demand is 9, four nurses only have D shift and remaining demand is 5. A similar approach is used for other shift types. Nurses are looped to assign a pattern to the nurse and meaning to the schedule set. For the first week, nurses are considered carriers of the schedule set.

1.0 The nurse assignment method is invoked where:

> 1.1 we check if night shift is allowed
>
> 1.2 all day-only patterns are looped (*for* loop)
>
> > 1.2.1 patterns are checked and validated regarding demand (length of pattern is checked depending on number of hours the nurse works, which cannot be more than six days in one week)
> >
> > 1.2.2 pattern value is calculated (number of working shifts in pattern)
>
> 1.3 go back to 1.0 until the end of the number of day-only patterns
>
> 1.4 all night-shift patterns are looped, if nurse can work night shifts (*for* loop)

1.4.1 patterns are checked and validated regarding demand (length of pattern is checked depending on number of hours the nurse works, which cannot be more than six days in one week)

1.5 pattern value is calculated (number of working shifts in pattern). It is checked which pattern is the best fit

1.5.1 if night shift is not allowed, the best day-only-valued pattern that fits the demand is assigned to the nurse

1.5.2 if night shift is allowed, the best night-shift-valued pattern is assigned if existing or not used; otherwise, the day pattern is assigned

1.5.3 a pattern is assigned: if possible, night pattern; otherwise, day pattern

1.6 if too many night shifts are present in the schedule and exceed the demand, excessive night shifts are replaced with R (free days)

1.7 end of nurse assignment method.

Later, a check is made of whether the demand has been met. If not, the nurse repair function is called to repair the missing demand by replacing patterns with other day or night patterns. A check is also made of whether the patterns are correct. The function is run repeatedly. Later, the number of demand-remaining patterns is calculated.

2.0 for each day in a week:

2.1 if more than the needed day shifts are assigned, the fix pattern method is called

2.1.1 the fix pattern replaces someday shifts with a free day

2.1.2 pattern validity is checked

2.2 if more than the needed night shifts are assigned, the fix pattern method is called

2.2.1 the fix pattern replaces some night shifts with a free day

2.2.2 pattern validity is checked

3.0 the number of remaining demand patterns is calculated

4.0 for each day in a week

4.1 if less than the needed day shifts are assigned, the fix pattern method is called

4.1.1 the fix pattern replaces someday shifts with a free day

4.1.2 pattern validity is checked

4.2 if less than the needed night shifts are assigned, the fix pattern method is called

4.2.1 the fix pattern replaces some night shifts with a free day

4.2.2 pattern validity is checked.

Thus, first we use a greedy algorithm to create a schedule. Then, we try to fit improved patterns to the schedule using the nurse repair function recursively. The function is limited by time, so it will not try to find a better long-term pattern than what can be found in an optimal number of seconds while checking the results in the experiment. Next, the fix pattern function is used to address demand, removing shifts from the schedule when the demand is overbooked. For the first week, we use zero-cost patterns. Figure 5.15 provides a graphical illustration of the generation of the week 1 schedule.

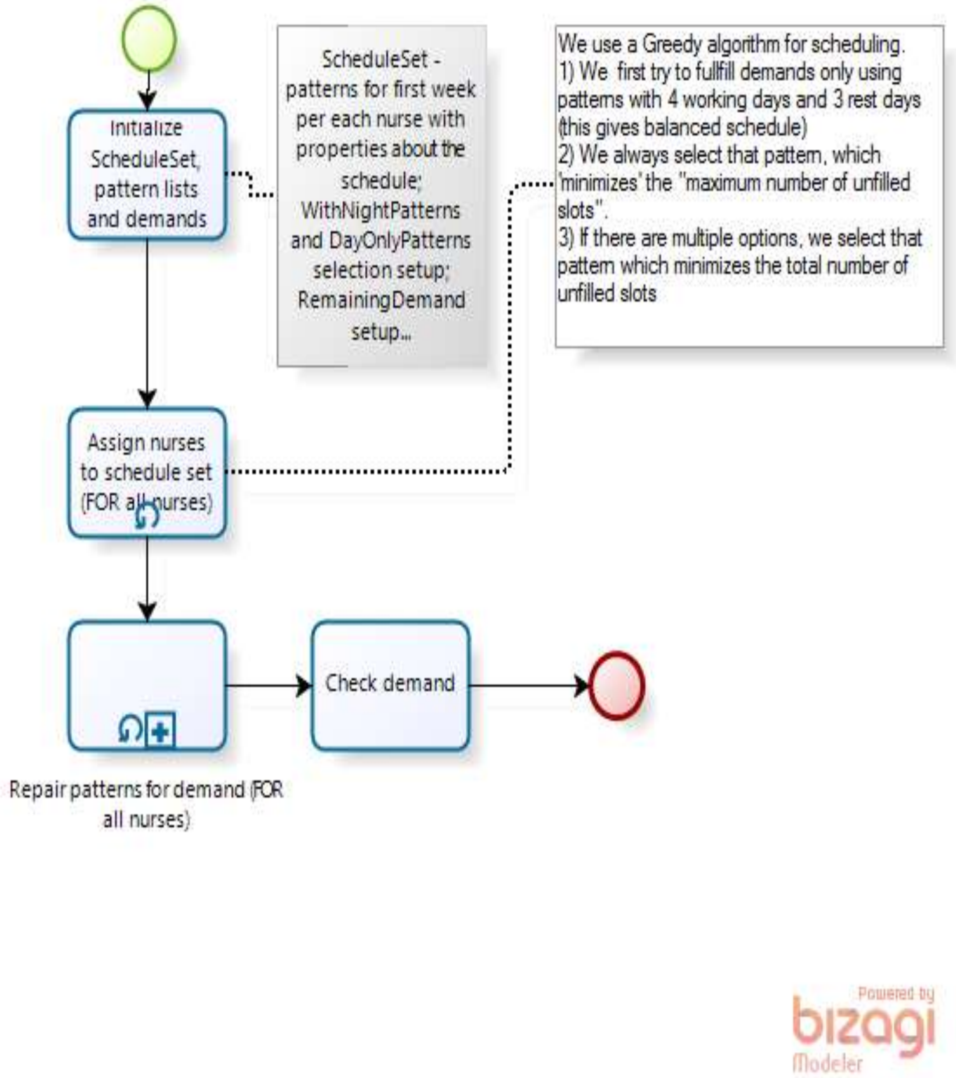


Figure 5.15. The process of generating a week one schedule.

5.4.3 Step 3: Generate the N week's schedule

To expand the schedule to N weeks, matrix base subsets are used to set the combinations of nurses and patterns that can or cannot be used. First, the subset is based on pattern validity inside the first *for* loop. In the following *for* loop, it is set based on which nurses will and will not take night patterns. This function uses similar methods as for the first week, but with added checks of the current week's schedule against those of previous weeks. The function also calculates the costs and violations, and finally corrects the schedule in relation to the demand. The changes incorporated to modify the week 1 function are at 1.2.1 and 1.4.1. The patterns are checked and validated regarding cost and violations, where zero-cost and zero violations are allowed. There are also some loops added at 1.5, which for week N we name 5.5.

5.5 it is checked which pattern is the best fit

5.5.1 if no pattern is the best fit, we take a similar approach as for 1.2 but with increased cost

5.5.1.1 all day-only patterns are looped, and patterns are checked and validated regarding cost and violations, allowing up to 20 cost and zero violations. The same is done with night patterns. Pattern value is retrieved

5.5.1.2 if the best fit is still not found, we exit the assign nurse function

5.5.2 if night shift is not allowed, the best day-only-valued pattern is assigned to nurses that fit the demand. We also check the cost of the pattern inside the schedule and the violations, where zero-cost is allowed

5.5.3 if night shift is allowed, the best night-shift-valued pattern is assigned if existing or not used; otherwise, the day pattern is assigned. We also check the cost of the pattern inside the schedule and the violations, where zero-cost is allowed

5.5.4 a pattern is assigned: if possible, night pattern; otherwise, day pattern.

Again, a check is made of whether the demand has been met, and all other steps as in the generation of the week 1 schedules are followed. The difference here is that the check made for cost allows for both zero-cost and zero violation. Once more, the number of remaining demand patterns is calculated as shown in 2.0 to 4.2.2. Here, at each *if* selection, before the pattern validity is checked, costs and violations are checked. Between 0 and

20 cost patterns are allowed. This is because patterns formed together can incur costs. Thus, the pattern is checked to ensure it forms a cost within the schedule and that it has zero hard violations. First, whether the schedule can be made with zero-cost is checked, and if not, cost up to 20 is allowed per nurse.

5.4.4 Step 4: Convert DNR to edlNR

At this stage, we convert the DNR domain to the original edlNR domain. Here, the 'edl' patterns are involved. We look for the next series with a day pattern and check its position of next day pattern, night pattern and series length (this attempt is based on solving series). First, we check if the pattern ends with a night series. If yes, we place 2–3 Ls or Es before the N shift series, provided this is possible with the demand. We keep the Ds where it is not possible to replace them with L or E, depending on demand. If the series is the first in the schedule, we allow one day's length of miniseries (of Ls or Es); otherwise, if the pattern does not end with a night series, we place Es at the beginning of all short series with length 2–3, depending on demand. We also place Ls at the end of all series with length of 2–3 and retain the Ds where it is not possible to replace them with L or E, depending on demand. We then look for the next series. At this point, we always check the demand.

Following this, we backtrack all nurses. We loop days and check for missing demand for the day and for E or L. Then, we place E or L for found D shifts. Again, we check the demand and 'edl' costs and violations. Finally, we check if the schedule is required for incomplete weeks. If the schedule begins later than Monday and ends earlier than Sunday, these extra shifts are removed. Cost is recalculated with the extra days removed from the schedule, so it is possible that the schedule cost is lower than for full weeks. The complete process is illustrated in Figure 5.16.

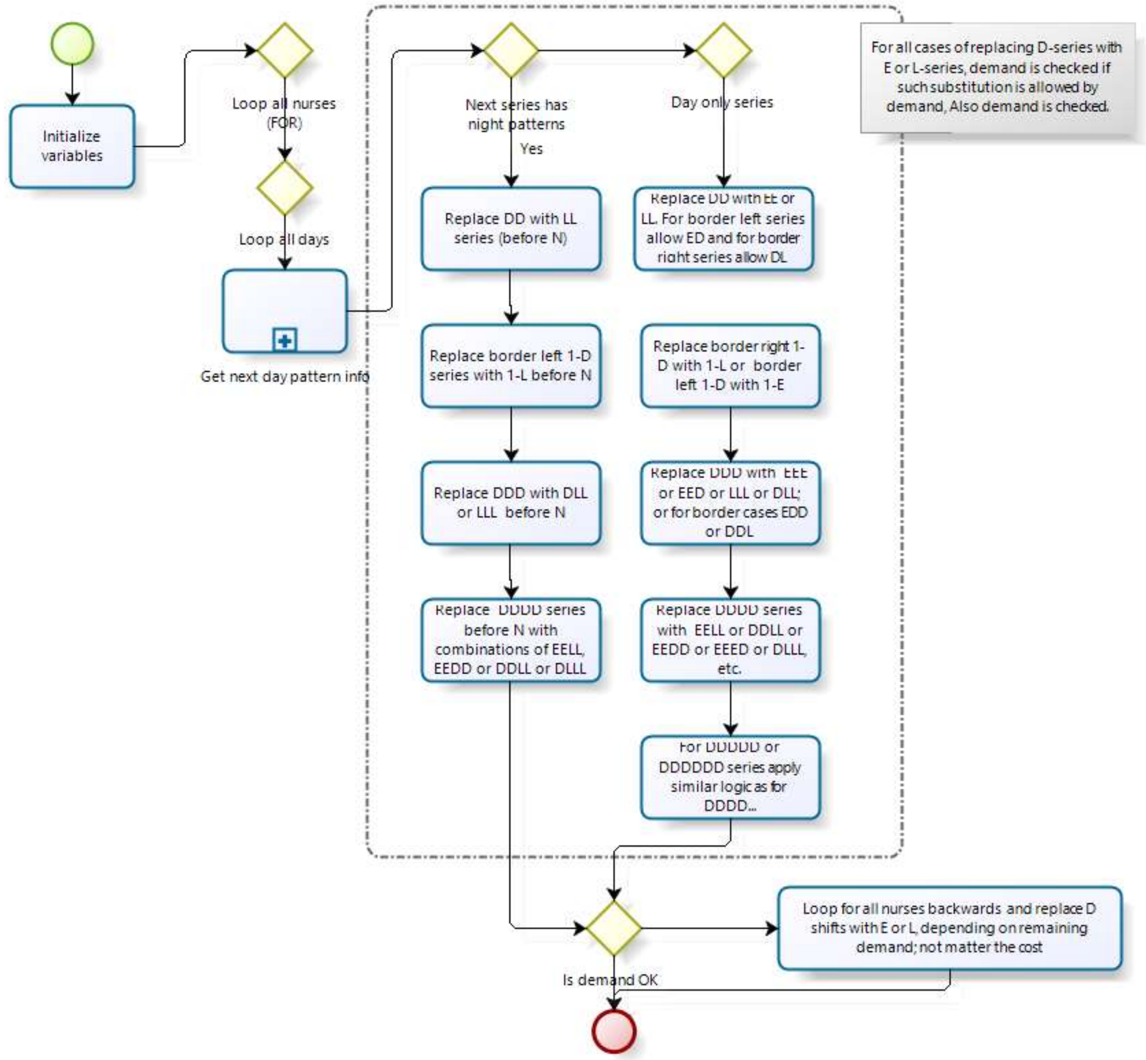


Figure 5.16. Process of converting from DNR to edlNR.

5.4.5 Greedy technique computational result

As in the greedy technique, the patterns with cost 15 and 20 are used in generating the schedules for a month. The total cost for a month schedule is 130. However, the cost increased to 150 when extended to a five-week schedule (see Figure 5.17).

```
----------------------------------
Computed Schedule for 5 week(s)
----------------------------------
Days->       | MTWTFSS cost | MTWTFSS cost | MTWTFSS cost | MTWTFSS cost | MTWTFSS cost

Nurse 01(20):LRREERR    0 | RRREERR    0 | NNRRRRR    0| RRRREEE    0 | RRRREEE    0|
Nurse 02(20):NNRRRRR    0 | RREEERR    0 | REEERRR    0| EEERRRR    0 | EERRRRR    0|
Nurse 03(20):LRREERR    0 | RRRREEE    0 | RRRREEE    0| RRRRNNN    0 | RRREERR    0|
Nurse 04(32):REELLRR    0 | EELLRRR    0 | REELLRR    0| RRREELL    0 | NNRREEE    0|
Nurse 05(36):EEERRLL    0 | NNRRDDD    0 | LLLRREE    0| ELLLRRR    0 | EELLLRR    0|
Nurse 06(36):RRREEEL    0 | LLRRNNN    0 | RRREERR   20| EELLLRR   20 | EEERRLL   20|
Nurse 07(36):LLNNRRR    0 | REELLRR    0 | EEERRRR   20| REEELLL   20 | RRRDLLL   20|
Nurse 08(36):EELLRRR    0 | EELLLRR    0 | DDLLLRR    0| DDLLLRR    0 | RREEDDD    0|
Nurse 09(36):RDLLLRR    0 | DDDDDRR    0 | RDLLLRR    0| DDDDDRR    0 | RRLLNNN    0|
Nurse 10(36):RDDDDRR    0 | DDDDRRR    0 | LLNNRRR    0| RREEEDD    0 | DRRDDDD    0|
Nurse 11(36):DLLRREE    0 | LLNNRRR    0 | RRDDDLL    0| LRRRDDD    0 | LLLRRRR   10|
Nurse 12(36):ERRRNNN    0 | RRRRREE   20 | ERREEDL   20| LLRRRRR   20 | DDDLLRR   20|
Nurse 13(36):RDDDDRR    0 | RREEDLL    0 | LRRRDDD    0| NNRRREE    0 | DDDDRRR    0|
Nurse 14(36):DLDRRDD    0 | DDDDRRR    0 | DDDDDRR    0| DDDDRRR    0 | LLNNRRR   15|
Nurse 15(36):DRRRDDD    0 | LLLRRDD    0 | DLDRRLD    0| LLNNRRR    0 | RDDEDRR   30|
Nurse 16(36):RREDLLE    0 | ERRRLLL    0 | ERRDNNN    0| RRDDDRR   20 | LLERRRR   35|

Verifying total nurses available each day:
------------------------------------------

Total E:   3333322      | 3333322      | 3333322      | 3333322      | 3333322

Total D:   3333322      | 3333322      | 3333322      | 3333322      | 3333322

Total L:   3333322      | 3333322      | 3333322      | 3333322      | 3333322

Total N:   1111111      | 1111111      | 1111111      | 1111111      | 1111111
```

Figure 5.17. Five-week schedule in the edlNR domain using the greedy technique.

5.5 Comparison Result in DNR Domain and EdlNR Domain

It is interesting to compare the total costs of the patterns in the different domains. In the DNR domain, patterns of cost 0, 10 and 20 are used. In the edlNR domain, the same pattern category types are used, and patterns of cost 5 are included, mainly because of the soft constraints related to edl shifts. During conversion, patterns of cost 1 or 5 can be incorporated. Figure 5.18 shows the results of the three different techniques used to evaluate the DNR domain for one month. It is shown that T1 used the lowest total cost pattern, at a cost of 10, while T2 and T3 used patterns of the same cost, 20, in week 2. Looking at week 3, of the three techniques, T2 used the lowest cost pattern. However, at the end of the week, the patterns used by T1 and T3 had decreased to zero-cost. When we compare the three techniques in the DNR domain for five weeks, T1 started with patterns of a total cost of 15, while T2 and T3 started with patterns of a total cost of 20 in week 2. In week 4, T2 used zero-cost patterns; however, in week 5, the patterns had a total cost of 30. For T1 and T3, both had an increased cost of patterns in week 3, decreasing to zero-cost for T1 and a total cost of 10 for T3 in week 5.

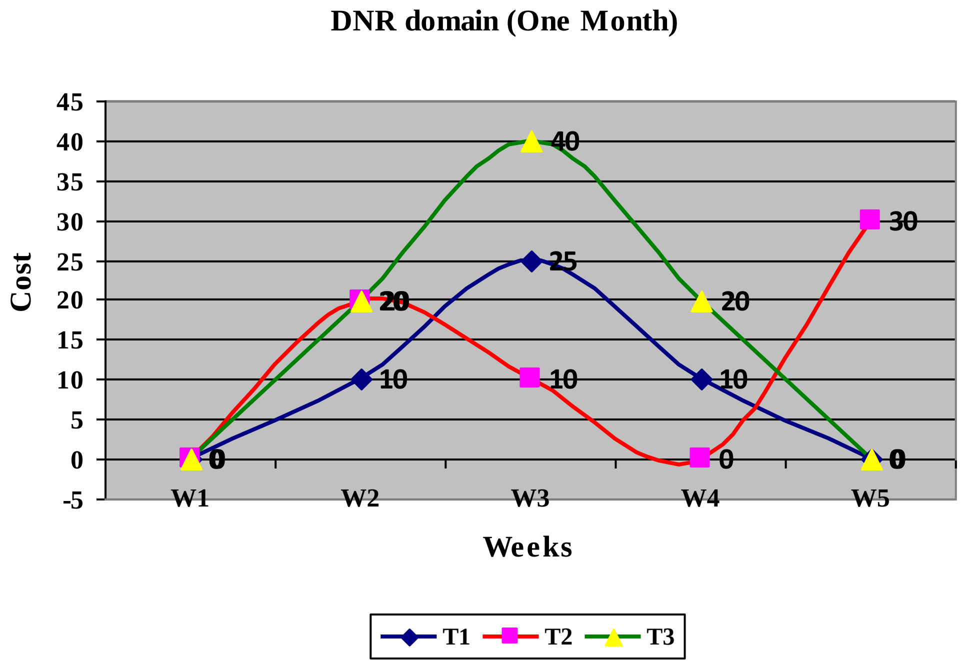


Figure 5.18. Comparison of T1, T2 and T3 in the DNR domain for one month.

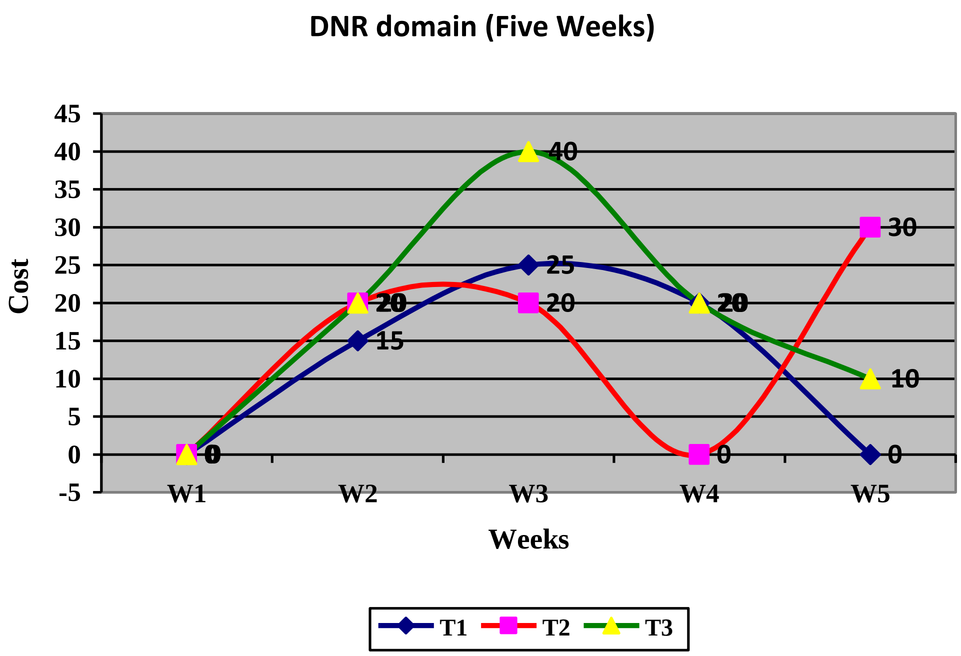


Figure 5.19. Comparison of T1, T2 and T3 in the DNR domain for five weeks.

Among the three techniques used to generate schedules in the edlNR domain for one month (see Figure 5.20), T2 had the highest total cost of patterns in week 2, but the lowest total cost in week 4. Conversely, T3 started with the lowest cost of patterns in week 2, but used the highest cost patterns in the final week. T1 started with zero-cost patterns, and ended with zero-cost patterns in the final week.

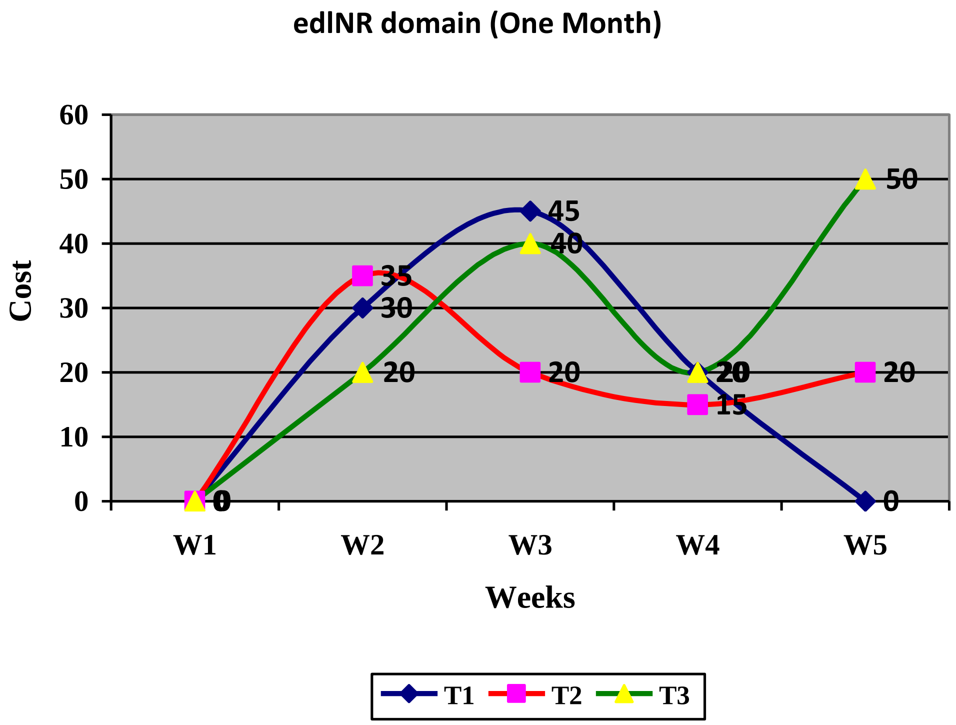


Figure 5.20. Comparison of T1, T2 and T3 in the edlNR domain for one month.

Figure 5.21 shows the results of the three techniques used in the edlNR domain for five weeks. T2 started with cost patterns in week 1, but used zero-cost patterns in week 4. For T3, the cost of patterns increased to week 3, before dropping slightly in week 4 and increasing again in week 5. The pattern costs for T1 were similar to in the one-month generation in Figure 5.20, with the exception of the incorporation of a pattern of cost 5 in week 5.

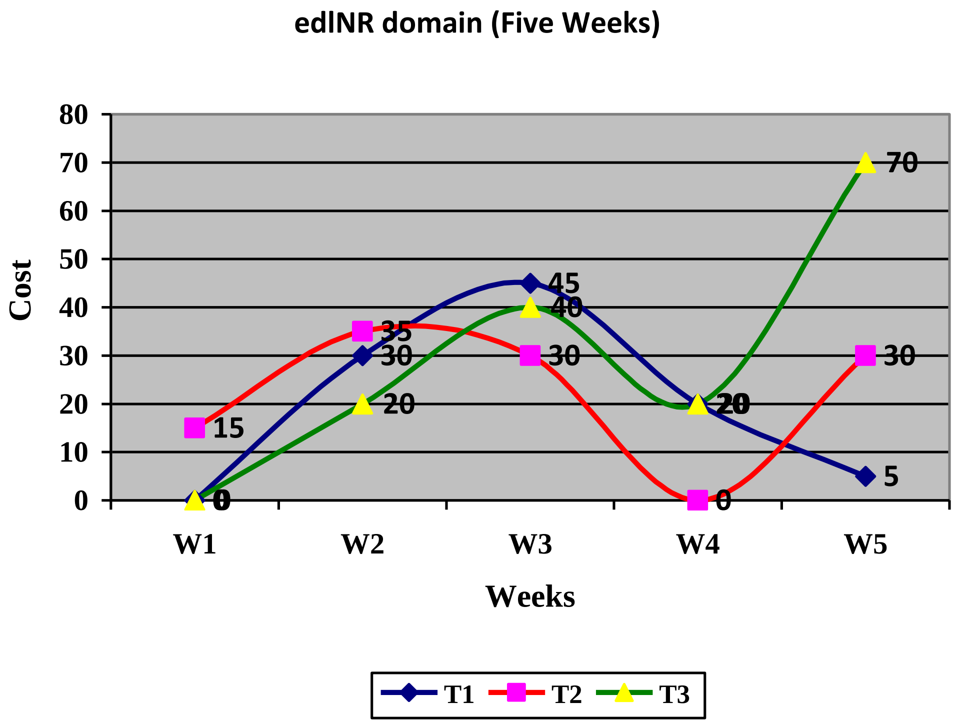


Figure 5.21. Comparison of T1, T2 and T3 in the edlNR domain for five weeks.

Table 5.7. Comparison of total cost in DNR and edlNR domain

Domain	One Month/Total Cost			Five Weeks/Total Cost		
	T1	T2	T3	T1	T2	T3
DNR	45	60	80	60	70	90
edlNR	95	90	130	100	120	150

As shown in Table 5.7, the total cost of patterns in the DNR domain is smaller than in the edlNR domain. This is because the conversion incorporates all soft constraints. Therefore, higher cost patterns must be used to satisfy the demand. T1 was found to perform better in the DNR domain for both one month and five weeks, and in the edlNR domain for five weeks. However, T2 outperformed T1 in the edlNR domain for one month. It has been demonstrated that domain transformation is a simple and economical approach for generating reliable high-quality NRP schedules. This approach has facilitated the transformation of this complex problem in benchmark studies into a more cost-effective schedule. Finding the solution in a smaller domain (DNR) and generating a database of feasible patterns offline is an important feature of our approach, allowing the saving of computational effort and production of low-cost schedules.

5.6 Computational Results for One Month and Five Weeks

During the offline preparation process using the ORTEC dataset in the DNR domain, 18 feasible zero-cost shift patterns were generated for the 36/32 hours/week FT nurses, while 15 feasible zero-cost shift patterns were generated for the 20hours/week PT nurses. These zero-cost patterns were used for the allocation of schedules for different types of nurses. Using another dataset, we also generated some one-week sets of zero-cost patterns for FT and PT nurses, which we subsequently expanded to a five-week solution. In this case, all patterns were generated with a zero-cost solution for the DNR domain until week 3. There was no evidence that having zero-cost solutions in early weeks forced the later adoption of expensive (non-zero-cost) patterns, as in our problem solution we had all 20 cost patterns in the DNR domain solution. These results were subsequently converted to the edlNR space. The patterns generated had to satisfy the problem constraints. During the conversion, a zero-cost schedule was again produced for the five-week period. However, this may not hold true when trying to satisfy a greater number of constraints or different sets of problems. In the case of infeasibility, the constraints can be relaxed incrementally in order of cost until a feasible solution is found. Once the zero-cost patterns do not fit the schedule, the lowest cost patterns should be chosen. Both the one-week sequences and the five-week solution then have a non-zero-cost schedule to satisfy the nurse demand for each day. By comparing against previous solutions reported in the literature (see Table 5.8 for five weeks and Table 5.9 for one month), our approach is shown to produce the optimal solution for the problem. Our particular implementation is an adaptation of Baskaran *et al.* (2009).

Table 5.8. Previous Solutions to this Same Problem Statement (5 weeks/35 days)

Penalty	Approach	Execution time	Author
170	Decomposition + VNS Iterative	<1 minute	Brucker *et al.*, 2005
100*	Domain Transformation T1	45 seconds	Baskaran *et al.*, 2014c
120*	Domain Transformation T2	150 seconds	Baskaran,G *et al.*, 2014d
150*	Domain Transformation T3	90 seconds	Baskaran,G *et al.*, 2015

Table 5.9. Previous Solutions to this Same Problem Statement (4 weeks/one month)

Penalty	Approach	Execution time	Author
775	GA	1 hour	Burke *et al.*, 2008
681	GA	24 hour	Burke *et al.*, 2008
706	HO/VNS	1 hour	Burke *et al.*, 2008
541	HO/VNS	12 hour	Burke *et al.*, 2008
360	VDS	25 minutes	Burke *et al.*, 2007
465	VDS	600 seconds	Burke *et al.*, 2014
270	MIP	2 minutes	Glass and Knight, 2009
270	Branch-and-Price	69.3 seconds	Burke *et al.*, 2014
***95**	Domain Transformation T1	30 seconds	Baskaran *et al.*, 2014c
***90**	Domain Transformation T2	135 seconds	Baskaran,G *et al.*, 2014d
***130**	Domain Transformation T3	85 seconds	Baskaran,G *et al.*, 2015

According to the literature that tests the ORTEC dataset (see Table 5.9), the best result was a 270 cost solution after an execution time of two minutes (Glass and Knight, 2009) while Burke *et al.* produced the same results with a shorter computational time of 69.3 second. We have thus achieved an improvement by obtaining a 90 cost solution using T2. The execution times were achieved using comparable computers. Burke *et al.* (2007, 2008) used a P4, 2.4 GHz processor PC. While Glass and Knight

(2010) used a desktop PC with a P4 2.67 GHz processor whereas ours had a clock speed of 2.64 GHz. Our shorter execution time is partly due to the accessibility of a feasible solution with no penalties; that is, a zero-cost solution. Should a zero-cost solution have proved infeasible, we would have relaxed one or more of the higher penalty constraints, with run time increasing as a result. We recognise that using this domain transformation approach with problems of higher complexity may be challenging computationally, but not impossible.

5.6.1 Continuity

An important issue in nurse scheduling is the continuity from one scheduling period to the next. This has been a gap in the literature, as highlighted in Celia *et al.* (2010). The NSP benchmark instances tested in this thesis are designed to produce schedules for an isolated period. The penalties are applied in accordance with the standard that all possible violations are counted at the beginning of the period, and not ignored at the end. We recognise that the benchmark instances are intended as a basis for comparison between alternative scheduling methodologies, and that the consideration of isolated schedule periods serves this purpose. However, in a practical environment, information relating to one scheduling period is carried forward to the next, creating additional issues of 'continuity'. For example, while the scheduling period may only be one month in length, the constraints do not primarily relate to that one-month period. In those constraints relating to periods, some relate to one week, others to a rolling five-week period, or even a rolling 13-week period. To illustrate this point, Appendices 1 and 2 show, respectively, the schedule of T2 for 13 weeks, with a total cost of 250, and the schedule of T2 for 52 weeks, with a total cost of 580. Table 5.10 shows the results of the three techniques used for the domain transformation to generate the 52-weekschedules. T2 performed better on cost compared to T3 or T1. However, while T3 generated the 52- week schedule at a higher cost, it did so with the lowest computational time, at only 21 minutes.

Table 5.10. Result of three techniques generating schedule for 52 weeks

ORTEC 52 weeks	T2	T3	T1
Cost	580	640	600
Time (minutes)	25	21	24

Domain transformation is an effective approach, designed to handle the constraints relating to various periods. Table 5.11 presents the results for our approach compared to two other approaches on the 12 large real-world NSP datasets of ORTEC (January to December). The first approach for comparison is a hybrid genetic algorithm developed by ORTEC in the commercialised software Harmony TM (Fijn van Draat, Post, Veltman and Winkelhuijzen, 2006). The second approach is a hybrid Variable Neighbourhood Search with a heuristic ordering as the construction method (Burke, Curtois, Post, Qu and Veltman, 2008). The Hybrid IP method (Burke, Li and Qu, 2009) applies an IP model to construct the initial solution and a Variable Neighbourhood Search to make improvements to the results. Our approach obtained better results on all 12 datasets for all three techniques, generating the one-month schedules with <2.25 minutes computational time. Minor cost variations occur between months in our approach because of the different start days, which affects some of the hard constraints. This may require the incorporation of non-zero-cost patterns. Overall, however, the results demonstrate that domain transformation can find good-quality solutions in less computational time for highly constrained NRPs, compared to the current best approaches in the literature.

Table 5.11. Comparison of Results for ORTEC January to December

ORTEC Jan–Dec	Hybrid GA	Hybrid VNS	Hybrid IP	Domain Transformation		
	[125]	[115]	[123]	**T2**	**T3**	**T1**
Jan	775	735	460	**90**	**130**	**95**
Feb	1791	1866	1526	**70**	**100**	**75**
Mac	2030	2010	1713	**75**	**115**	**80**
Apr	612	457	391	**85**	**120**	**85**
May	2296	2161	2090	**95**	**135**	**100**
Jun	9466	9291	8826	**90**	**130**	**95**
July	781	481	425	**85**	**125**	**90**
Aug	4850	4880	3488	**95**	**130**	**100**
Sept	615	647	330	**80**	**120**	**90**
Oct	736	665	445	**90**	**130**	**95**
Nov	2126	2030	1613	**95**	**135**	**100**
Dec	625	520	405	**85**	**125**	**90**

5.7 Comparison of Performance Reported in the Literature of Techniques Using Domain Transformation in NSP

We assess our domain transformation approach upon a set of benchmark real-world NSP datasets, publicly available at http://www.cs.nott.ac.uk/~tec/NRP. The chosen benchmark datasets are the most tested problems in the literature because of their complex constraints.

The rules, regulations and objectives have been taken directly from the real- world cases and preserved with their essential characteristics. The difficulty of the problems not only depend on the number of shift types, number of nurses and length of the scheduling period, but also on the complex constraints involved.

Within our approach, the IP process was solved by using the latest GNU Octave's GLPK (4.45). We have also used the database engine SQL Server Compact 3.5. The results obtained through solving the T1, T2, and T3 are presented in Table 5.12. The problem is one of minimisation, and the results in bold indicate the optimal solutions. All the techniques using the domain transformation approach were able to solve most of the instances to optimality; however, the computation time varied from <0.1 second to 2.25 minutes in the case of the hardest instance. In comparison with the best-known results, we achieved a new result for GPost-B with a cost of2 in 15 seconds with T2 and WHPP, zero-cost with T2 in 17 seconds and T1 in 5 seconds. One result for large instances outperformed the other examples: ORTEC. This appears significantly better than the best results achieved in the existing literature. Our approach achieved cost 90 within 135 seconds for ORTEC01 and 155 seconds for ORTEC02. Overall, our results represent the best-known cost.

Table 5.12. Results on Benchmark Datasets

Data Sets	Best known Cost	BUR 14		MET 09		BUR 09b (SS2)		BUR 09b (MEH)		Our Approach T2		Our Approach T3		Our Approach T1	
		Cost	Time	Cost	Time	Cost	Time	Cost	Time	Cost	Time	Cost	Time	Cost	Time
Musa	**175**	175	<0.1	175	39					**175**	1	**175**	1	**175**	1
GPost	**5**	5	2	8	234	9	4305	915	605	**5**	3	**5**	20	**5**	5
GPost-B	**2**	3	29.3			5	3955	789	475	**3**	**8**	**2**	**15**	**3**	**10**
Ozkarahan	**0**	0	<0.1	0	1					**0**	**<0.1**	**0**	1	**0**	**<0.1**
Millar-2Shift-Data1	**0**	0	<0.1	0	1	0	910			**0**	**<0.1**	**0**	43	**0**	1
Millar-2Shift-Data1.1	**0**	0	<0.1			0	20			**0**	1	**0**	30	**0**	2
Azaiez	**0**	0	0.3	0	233					**0**	1	**0**	30	**0**	2
WHPP	**0**	5	17.6							**0**	**5**	**0**	**17**	**5**	**10**
Valouxis-1	**20**	80	909.6	160	3780	100	4000			**20**	**40**	**20**	**42**	**20**	**40**
Ikegami-2Shift-Data1	**0**	0	41.7							**0**	**40**	**0**	**40**	**0**	**40**
Ikegami-3Shift-Data1	**2**	2	597.8	63	671					**2**	**50**	**2**	**68**	**2**	**55**
Ikegami-3Shift-Data1.1	**3**	4	995.2							**3**	**80**	**3**	**88**	**3**	**85**
Ikegami-3Shift-Data1.2	**3**	5	5411.9							**3**	**90**	**3**	**95**	**3**	**95**
ORTEC01	**90**	270	69.3			365	3400	535	7580	**95**	**30**	**90**	135	**130**	**85**
ORTEC02	**90**	270	105.1							**95**	**60**	**90**	155	**130**	**99**
QMC-1	**13**	13	57.6			20	4435	39	3160	**13**	60	**13**	62	**13**	60
QMC-2	**29**	29	1.9							**29**	**1**	**29**	3	**29**	2
SINTEF	**0**	0	10.5			4	4105			**0**	**10**	**0**	48	**0**	22

We have presented a novel information-granulation-based formulation of the NSP and have solved it using T1, T2 and T3. The results show that all three techniques can solve some instances very effectively. For other instances, the time and resource requirements may be restrictive. However, with more development of new ideas, it may be possible to improve the performance of our method further. The domain transformation approach uses a number of novel ideas that we believe are general enough to be adapted to other problem domains. All instances tested were modelled using a generic model. Tables 5.13and 5.14 below show, respectively, the comparison of the average cost percentage from the optimal cost, and the average time percentage from the optimal cost, with other methods solving the benchmark problems reported in the literature. We evaluate the mean percentage differences between the results distributed by a given method and the best-known results reported in the literature, showing the standard deviation of the difference. Our domain transformation approach is shown to be competitive, with the lowest mean percentage of 0 (T2), 7.7 (T3) and 3.4 (T1) for cost average and 49.6 (T2), 34.1 (T3) and 26.1 (T1) for time average when comparing our approach to the best-known costs. Among our three techniques, T2 is nine times faster on average compared to Burke *et al.*'s results (2014). Moreover, our approach is the most consistently reliable, as indicated by standard deviations of 0 (T2), 17.8 (T3) and 11.8(T1) for the differences where the cost is clustered closely around the mean. This is supported by the coefficient of variation, which is less than 5% for our domain transformation approach which generally give us a feeling of good method performance. The standard deviations for time were also lower for our method (44.3 [T2], 36.7 [T3] and 21.8 [T1]) than for other methods reported in the literature. This indicates that our domain transformation method is reliable and stable and is more capable of reducing the computational effort required than other methods reported in the literature.

Table 5.13. Average Cost Percentage from Optimal Cost

Data Sets	Best known Cost	BUR 14		MET 09		BUR 09b (SS2)		BUR 09b (MEH)		Our approach T2		Our approach T3		Our approach T1	
		C	%	C	%	C	%	C	%	C	%	C	%	C	%
Musa	**175**	175	0.0	175	0.0					**175**	0.0	**175**	0.0	**175**	0.0
GPost	**5**	5	0.0	8	60	9	80	915	18200	**5**	0.0	**5**	0.0	**5**	0.0
GPost-B	**2**	3	50.0			5	66.7	789	26200	**2**	**0.0**	**3**	50.0	**3**	50.0
Ozkarahan	**0**	0	0.0	0	0.0					**0**	0.0				
Millar-2Shift-Data1	**0**	0	0.0	0	0.0	0	0.0			**0**	0.0	**0**	0.0	**0**	0.0
Millar-2Shift-Data1.1	**0**	0	0.0			0	0.0			**0**	0.0	**0**	0.0	**0**	0.0
Azaiez	**0**	0	0.0	0	0.0					**0**	0.0	**0**	0.0	**0**	0.0
WHPP	**0**	5	∞							**0**	**0.0**	**5**	∞	**0**	**0.0**
Valouxis-1	**20**	80	300	160	700	100	400			**20**	0.0	**20**	0.0	**20**	0.0
Ikegami-2Shift-Data1	**0**	0	0.0							**0**	0.0	**0**	0.0	**0**	0.0
Ikegami-3Shift-Data1	**2**	2	0.0	63	3050					**2**	0.0	**2**	0.0	**2**	0.0
Ikegami-3Shift-Data1.1	**3**	4	33.3							**3**	0.0	**3**	0.0	**3**	0.0
Ikegami-3Shift-Data1.2	**3**	5	66.6							**3**	0.0	**3**	0.0	**3**	0.0
ORTEC01	**90**	270	200.0			365	305.6	535	494.4	**90**	**0.0**	**130**	**44.4**	**95**	**5.6**
ORTEC02	**90**	270	200.0							**90**	**0.0**	**130**	**44.4**	**95**	**5.6**
QMC-1	**13**	13	0.0			20	53.9	39	200	**13**	0.0	**13**	0.0	**13**	0.0
QMC-2	**29**	29	0.0							**29**	0.0	**29**	0.0	**29**	0.0
SINTEF	**0**	0	0.0			4	∞			**0**	0.0	**0**	0.0	**0**	0.0
Average % Difference to Best known Cost		**mean% = 47.2** **std% = 90.1** **cv% =1.9**		**mean%=544.3** **std% =1134.5** **cv% =2.1**		**mean%=113.3** **std%= 153.3** **cv% =1.4**		**mean%=11175.4** **std% =13143.4** **cv% =1.2**		**mean%=0** **std% =0** **cv% =0**		**mean% =7.7** **std% = 17.8** **cv% =2.3**		**mean%=3.4** **std%=11.8** **cv% = 3.5**	

Note: std=standard deviation, CV=coefficient of variation

Table 5.14. Average Time Percentage from Optimal Cost

Data Sets	BUR 14	MET 09	BUR 09b (SS2)	BUR 09b (MEH)	Our approach T2	T3	T1
	Time (s)	Time (s)	Time (s)	Time (s)	Time (s)	Time (s)	Time (s)
Musa	**<0.1**	39			1	1	1
GPost	**2**	234	4305	605	20	5	3
GPost-B	29.3		3955	475	**15**	**10**	**8**
Ozkarahan	**<0.1**	1			1	**<0.1**	**<0.1**
Millar-2Shift-Data1	**<0.1**	1	910		43	1	**<0.1**
Millar-2Shift-Data1.1	**<0.1**		20		30	2	1
Azaiez	**0.3**	233			30	2	1
WHPP	17.6				**17**	**10**	**5**
Valouxis-1	909.6	3780	4000		**42**	**40**	**40**
Ikegami-2Shift-Data1	41.7				**40**	**40**	**40**
Ikegami-3Shift-Data1	597.8	671			**68**	**55**	**50**
Ikegami-3Shift-Data1.1	995.2				**88**	**85**	**80**
Ikegami-3Shift-Data1.2	5411.9				**95**	**95**	**90**
ORTEC01	69.3		3400	7580	135	85	**30**
ORTEC02	105.1				155	**99**	**60**
QMC-1	**57.6**		4435	3160	62	60	60
QMC-2	1.9				3	2	**1**
SINTEF	10.5		4105		48	22	**10**
Average Percentage Difference to Best Known Cost	**mean =458.3 std =1276.6**	**mean =708.4 std =1374.5**	**mean =3141.3 std =1696.4**	**mean =2955 std = 3322**	**mean =49.6 std =44.3**	**mean =34.1 std =36.7**	**mean =26.1 std =21.8**

5.8.1 Hospital Kajang

Currently, the matron in charge of the nursing department in Hospital Kajang, Malaysia, creates all nurses' schedules manually using a trial and error approach. Producing satisfactory schedules using this approach is costly and inefficient. These manual schedules do not satisfy a number of important criteria for efficient scheduling. In particular, they fail to satisfy the hard constraints of balanced schedules, fairness considerations and nurses' preferences, ergonomic considerations, and staffing requirements related to quality and size. Therefore, we cannot compare the cost of creating these manual schedules to those generated by our approach, as even when the manual schedules satisfy demand, they will include some infeasible patterns (see Appendix F). Our domain transformation approach, as well as being a practical computerised tool, provides important improvements in relation to the feasibility of schedules produced.

Our approach considers such restraints as nurses' preferences, current hospital policies, recommended policies drawn from the literature, and ergonometric issues. Satisfying simultaneously all these factors may not be feasible; however, some are considered hard constraints that must be satisfied. The hard constraints were designated as such based on feedback from the matron. The remaining constraints, considered soft constraints, were also assigned their priority levels and weights based on the judgment of the matron. The hands-on experience of the matron and nurses of Hospital Kajang offered valuable insight into our subject. The hospital matron prepares the nurses' schedules manually every two weeks. However, we have shown that four- and five-week schedules can also be generated in a short computational time at not too high a cost. Figures 5.22–5.23 below show the schedules generated for two weeks with T2, four weeks with T1 and five weeks with T3.

```
-----------------------------------
Computed Schedule for 2 week(s)
-----------------------------------
Days->   | MTWTFSS viol cost | MTWTFSS viol cost

Nurse 01(40, ):LLNNRRR  0    0|REEEERR  0   0|
Nurse 02(40, ):NNRRELE  0    0|EEERREE  0   0|
Nurse 03(40, ):RRRLNNN  0    0|RREEEER  0   1|
Nurse 04(40,O):DDDDRRR  0    0|RRRDDDD  0   0|
Nurse 05(40, ):RREEEER  0    0|LENNRRR  0   0|
Nurse 06(40, ):EEERRRL  0    0|EEEERRR  0   0|
Nurse 07(40, ):EERRRLE  0    0|NNRRREE  0   0|
Nurse 08(40, ):ERRRLLE  0    0|NNRRREE  0   0|
Nurse 09(40, ):RRRELEE  0    0|EERRREE  0   0|
Nurse 10(40, ):LLNNRRR  0    0|RDEEERR  0   5|
Nurse 11(40, ):NNRREEE  0    0|ERRRELE  0   0|
Nurse 12(40, ):REEEERR  0    0|RRDEELR  0   5|
Nurse 13(40, ):RREEEER  0    0|LLNNRRR  0   0|
Nurse 14(40, ):EEERRRD  0    0|ELLRRRL  0   5|
Nurse 15(40, ):RELEDRL  0    0|LLNNRRR  0   0|
Nurse 16(40, ):LLNNRRR  0    0|RLLLLRR  0   0|
Nurse 17(40, ):NNRRLEL  0    0|DRRRLLL  0   0|
Nurse 18(40, ):ERRRNNN  0    0|RRLLLRL  0  10|
Nurse 19(40,S):EEEERRR  0    0|RRRLNNN  0   1|
Nurse 20(40,S):RRLLEER  0    0|ELEEEER  0   0|
Nurse 21(40,S):LLLRRRE  0    0|LELRRRE  0   0|
Nurse 22(40,S):LLRRRLL  0    0|NNRRLLL  0   0|
Nurse 23(40,S):LLLRRRL  0    0|LLLRRRL  0   0|
Nurse 24(40,S):RRLLLLR  0    0|RRRLNNN  0   1|
Nurse 25(40,S):RRRRRRR  0   10|RRRRRRR  0  10|
Nurse 26(40,S):RRRLNNN  0    0|RRRLLLL  0   1|
Nurse 27(40,S):RRRLLDL  0    0|LRRRNNN  0   5|
Nurse 28(40,S):RRLLLLR  0    0|RRLLLLR  0   0|

verifying total nurses available each day:
-------------------------------------------

Total E:  6666666 | 6666666

Total D:  1111111 | 1111111

Total L:  6666666 | 6666666

Total N:  3333333 | 3333333
```

Figure 5.22. Schedule for 2 weeks (T2).

```
Computed Schedule for 4 week(s)
-----------------------------------
Days->   | MTWTFSS viol cost | MTWTFSS viol cost | MTWTFSS viol cost | MTWTFSS viol

Nurse 01(40, ):LLNNRRR  0    0|REEEERR  0    0|RRRLNNN  0    0|RREEEER  0   1|
Nurse 02(40, ):NNRRELE  0    0|EEERRLL  0    0|NNRRREL  0    0|EEEERRR  0   0|
Nurse 03(40, ):RRRLNNN  0    0|RREEEER  0    0|RREEEER  0    0|EEEERRR  0   1|
Nurse 04(40,O):DDDDRRR  0    0|RRRDDDD  0    0|RRDDDDR  0    0|RDDDDRR  0   0|
Nurse 05(40, ):RREEEER  0    0|LENNRRR  0    0|EEEERRR  0    0|RRRLNNN  0   1|
Nurse 06(40, ):EEERRRL  0    0|EEEERRR  0    0|LENNRRR  0    0|REEEERR  0   0|
Nurse 07(40, ):EERRRLE  0    0|NNRRRLE  0    0|EEEREEE  0    0|ERRRNNN  0   0|
Nurse 08(40, ):ERRRLLE  0    0|NNRRREL  0    0|EEEERRR  0    0|RRREEEE  0   0|
Nurse 09(40, ):RRRELEE  0    0|EERRRLL  0    0|NNRRREE  0    0|EERRREE  0   0|
Nurse 10(40, ):LLNNRRR  0    0|RDEEERR  0    0|RRLLNNN  0    0|RRRLEEE  0   5|
Nurse 11(40, ):NNRREEE  0    0|ERRREEE  0    0|ERRREEE  0    0|ERRREEE  0   0|
Nurse 12(40, ):REEEERR  0    0|RRDEEER  0    0|LENNRRR  0    0|RRELLDR  0   5|
Nurse 13(40, ):RREEEER  0    0|LLNNRRR  0    0|RRRLLLE  0    0|NNRRRLE  0   0|
Nurse 14(40, ):EEERRRD  0    0|ELLRRRE  0    0|EDERRRE  0    0|DELRRRD  0   5|
Nurse 15(40, ):RELEDRL  0    0|LLNNRRR  0    0|RLLEERR  0    0|LLNNRRL  0   0|
Nurse 16(40, ):LLNNRRR  0    0|RLLLLRR  0    0|RLLEERR  0    0|LLNNRRR  0   0|
Nurse 17(40, ):NNRRLEL  0    0|DRRRLEE  0    0|DRRRNNN  0    0|RRLLLRL  0  10|
Nurse 18(40, ):ERRRNNN  0    0|RRLLLRL  0   10|NNRRLLR  0   10|RLLRRLL  0  11|
Nurse 19(40,S):EEEERRR  0    0|RRRLNNN  0    0|RRREEEE  0    0|LRRREEL  0   1|
Nurse 20(40,S):RRLLEER  0    0|ELEEEER  0    0|LLNNRRR  0    0|RRLELLR  0   0|
Nurse 21(40,S):LLLRRRE  0    0|LELRRRE  0    0|EEERRLL  0    0|EEERRLL  0   0|
Nurse 22(40,S):LLRRRLL  0    0|NNRRLLE  0    0|LRRRLLL  0    0|NNRRLLL  0   0|
Nurse 23(40,S):LLLRRRL  0    0|LLLRRRL  0    0|LLLRRRL  0    0|LLNNRRR  0   0|
Nurse 24(40,S):RRLLLLR  0    0|RRRLNNN  0    0|RRRLLLL  0    0|RLLLLRR  0   1|
Nurse 25(40,S):RRRRRRR  0   10|RRRRRRR  0   10|RRRRRRR  0   10|RRRRRRR  0  10|
Nurse 26(40,S):RRRLNNN  0    0|RRRLLLL  0    0|LLRRRLL  0    0|NNRRLLE  0  11|
Nurse 27(40,S):RRRLLDL  0    0|LRRRNNN  0    0|RRLLLRD  0   10|LRRRNNN  0  15|
Nurse 28(40,S):RRLLLLR  0    0|RRLLLLR  0    0|RLLLLRR  0    0|LLLLRRR  0   0|

Verifying total nurses available each day:
-------------------------------------------

Total E:  6666666 | 6666666 | 6666666 | 6666666

Total D:  1111111 | 1111111 | 1111111 | 1111111

Total L:  6666666 | 6666666 | 6666666 | 6666666

Total N:  3333333 | 3333333 | 3333333 | 3333333
```

Figure 5.23. Schedule for 4 weeks (T1).

```
---------------------------------
Computed schedule for 5 week(s)
---------------------------------
Days->   | MTWTFSS viol cost | MTWTFSS viol cost | MTWTFSS viol cost | MTWTFSS viol cost | MTWTFSS vi
Nurse 01(40, ):LENNRRR  0   0|RREEEER  0   0|LENNRRR  0   0|REEEERR  0   0|EEEEERE  0  10|
Nurse 02(40, ):NNRRREE  0   0|EERRRLE  0   0|NNRRRLE  0   0|EEEERRR  0   0|REEEERR  0  10|
Nurse 03(40, ):RRRENNN  0   0|RREEEER  0   0|EEEEERR  0   0|RRREEEE  0   0|REEEERR  0  10|
Nurse 04(40,O):RDDDDRR  0   0|RRRDDDD  0   0|RDDDDRR  0   0|RRDDDDR  0   0|DDDDRRR  0   0|
Nurse 05(40, ):RREEEER  0   0|REEREER  0  10|RRNNRRR  0  20|RRRRRRR  0  20|RRRRRRR  0  20|
Nurse 06(40, ):EEERRRE  0   0|LENNRRR  0   0|RRRLELE  0   0|NNRRREE  0   0|EERRREE  0   0|
Nurse 07(40, ):ERRREEE  0   0|ERRRNNN  0   0|RRRRRRR  0  20|EEERNNN  0  20|RRRREER  0  20|
Nurse 08(40, ):RRREEEE  0   0|ERRREEE  0   0|EERRREE  0   0|EEERREE  0   0|EEERREE  0  10|
Nurse 09(40, ):LENNRRR  0   0|EEEERRE  0   0|EEERELE  0   0|NNRRREE  0   0|ELEERRR  0   0|
Nurse 10(40, ):NNRRREE  0   0|EEERRRE  0   0|EEERRRE  0   0|EEERRLE  0   0|NNRREEE  0  10|
Nurse 11(40, ):RRRENNN  0   0|RRDEEER  0   0|RRRENNN  0   0|RRRLELD  0   0|NNRRDEE  0   5|
Nurse 12(40, ):EEEERRR  0   0|RRRLLLE  0   0|NNRREED  0   0|EDLRRRL  0   0|ELLRLDD  0  10|
Nurse 13(40, ):REELERR  0   0|DDLERRR  0   0|LLNNRRR  0   0|RRLEEER  0   0|RRRELLL  0   5|
Nurse 14(40, ):RRELEDR  0   0|LLNNRRR  0   0|RLEEERR  0   0|RRLEELR  0   0|RRRLLLL  0   0|
Nurse 15(40, ):ELLRRRD  0   0|LLLLLRR  0   0|RRRLLLL  0   0|NNRRLLL  0   0|LRLLLLR  0  15|
Nurse 16(40, ):ERRRLLL  0   0|NNRRLLL  0   0|ERRRLEL  0   0|DRRRLLL  0   0|LRRRNNN  0   0|
Nurse 17(40, ):NNRRRLL  0   0|ELLRRRL  0   0|DLERRRL  0   0|LLNNRRR  0   0|LLLLRRR  0   5|
Nurse 18(40, ):DRRRNNN  0   0|RRRRRRR  0  20|NNRRRRR  0  20|LLRLLRR  0  20|RRNNRRL  0  20|
Nurse 19(40,S):REEEERR  0   0|LENNRRR  0   0|RREEEER  0   0|LENNRRE  0   0|EEERRRE  0   0|
Nurse 20(40,S):RRLLLER  0   0|RRRENNN  0   0|RRLLLER  0   0|LLEERRR  0   0|LLNNRRR  0   0|
Nurse 21(40,S):LLLRRRE  0   0|LLERRRE  0   0|EELRREL  0   0|ELNNRRL  0   0|LLLERRR  0   0|
Nurse 22(40,S):RRRLLLL  0   0|RLLLERR  0   0|RRRELDE  0   0|RLLLERR  0   0|RRRLNNN  0   6|
Nurse 23(40,S):RLLLLRR  0   0|RRRLNNN  0   0|RRLLLLL  0   0|LRRRNNN  0   0|RRRREER  0  31|
Nurse 24(40,S):RRLLLLR  0   0|RRLLLER  0   0|RRLENNN  0   0|RRLLLLL  0   0|LRRRLLL  0  10|
Nurse 25(40,S):LLLRRRL  0   0|LLLRRRL  0   0|LLLLRRR  0   0|RLLLLRR  0   0|RRLLNNN  0  10|
Nurse 26(40,S):ELRRRLL  0   0|NNRRLLL  0   0|LRRRLLL  0   0|LRRRLEL  0   0|NNRRRLL  0   0|
Nurse 27(40,S):LLNNRRR  0   0|RRRLLLL  0   0|LLRRNNN  0   0|RRRRRRR  0  20|RLNNRRR  0  31|
Nurse 28(40,S):LRRRLLL  0   0|NNRRRLL  0   0|LLLLRRR  0   0|RRRLNNN  0   0|RRLLLLL  0  11|

Verifying total nurses available each day:
------------------------------------------
Total E:  6666666 | 6666666 | 6666666 | 6666666 | 6666666
Total D:  1111111 | 1111111 | 1111111 | 1111111 | 1111111
Total L:  6666666 | 6666666 | 6666666 | 6666666 | 6666666
Total N:  3333333 | 3333333 | 3333333 | 3333333 | 3333333
```

Figure 5.24. Schedule for 5 weeks (T3).

Table 5.15. Summary of the Results of the Techniques for Kajang Hospital

	T2		T1		T3	
	Cost	Time (s)	Cost	Time (s)	Cost	Time (s)
2 weeks	44	21	50	19	65	9
4 weeks	77	47	77	35	107	30
5 weeks	100	52	109	40	249	38
Mean	73.7	40	78.7	31.3	140.3	25.7
Standard deviation	28.2	16.6	29.5	11	96.4	15

Table 5.15 shows that T2 outperformed T1 and T3 on cost for all time horizons, but that it was the slowest in computational time. T3 had the best computational time results: producing schedules in just nine seconds for two weeks, 30 seconds for four weeks and 38 seconds for five weeks. However, the standard deviation for T1 was lower than for either T2 or T3. Regardless of technique, our approach produced feasible schedules, satisfying the constraints. This represents an advantage to the hospital, whose existing method for generating schedules fails to satisfy even the

demand for nurses. Using our method, we could produce two-, four- and five-week schedules for their perusal.

The results from the real-world datasets can provide insight into domain transformation using information granulation. In almost all cases, the computerised method produced better-quality schedules (figure 5.23) than the manually generated solutions used in Kajang Hospital (Appendix G). Scheduling quality was measured using the five factors developed by Oldenkamp and Simons (1995):

- *Optimality*: where the granulation of shifts helps in representing the degree to which nursing expertise is distributed over the different shifts.
- *Completeness*: where the model is formulated as hard constraints and is always satisfied in the domain transformation approach to represent the degree to which the quantitative demands for nurses per shift are met.
- *Proportionality*: where the granulation of patterns and the checking of shifts to satisfy the demand hours actually represents the degree to which each nurse has been given about the same amount of working days (morning, evening and night shifts).
- *Healthiness*: where the incorporation of preferred shifts or rest days as hard and soft constraints represents the degree to which the welfare and health of the nurses has been considered.
- *Continuity factor*: analysed systematically using the hard constraints provided to represent the degree to which there is continuity in the nursing group during the different shifts.

5.8 Conclusion

Nurses' performance in the hospital setting can be managed and coordinated with the aid of nurse scheduling. Nurse scheduling helps departments in the hospital to organise the number of nurses working in a day, on either day or night shifts. Using proper scheduling methods, a high- quality schedule can be produced. From the research presented above, it is clear that preparing a nurse schedule requires the assessment of a wide range of criteria, including organisational rules, personal data and legal regulations. No single software can provide the solution, as each hospital has its own unique requirements and constraints; however, the same method can achieve a good solution. For proper results, the models and algorithms involved in generating the schedule should have a strong but flexible structure to adapt to the various unexpected situations that occur in a hospital. Further, the complexity of the NSP dataset makes it difficult to find feasible schedules with the precise number of nurses demanded per day and fulfilling a majority of soft constraints.

Our approach is not ‘hard coded’ to certain instances; rather, it has been designed keeping in mind the goal of learning about new problem- solving situations. In particular, our approach is applicable to most NSPs found in the literature. The proposed domain transformation approach to nurse scheduling represents a significant departure from the heuristic or meta-heuristic approaches that rely on the randomisation of the search procedure in the vast search space. Our approach offers deterministic reproducibility of solutions, as the domain of patterns allows for full enumeration of solutions. However, although our proposed method provides competitive and transparent results, it does not guarantee the global optimum. This is because the selection of non-zero-cost patterns for use in a specific scheduling process is guided only by the (rational) notion of making use of cheaper patterns first.

Chapter 6: Conclusions and Future Work

This chapter presents a summary of the work conducted in this thesis to improve the efficiency of nurse scheduling when handling large-scale domains. This chapter summarises the methodologies used and the findings of each chapter. This chapter also highlights the contributions made by this thesis to the area of nurse scheduling, and suggests some future research directions regarding domain transformation and the research field in general.

6.1 Outcomes of the Research

In this thesis, we discussed our novel proposed approach to solving NSPs. NSPs are usually highly constrained, have a large number of possible solutions and are complex (Okada and Okada, 1988). In our preliminary study, we suggested that when enough information is delivered, using a systematic method, a real-world NSP can be solved efficiently and effectively. Although real-world NSPs are complex and very challenging to solve, they can be solved by simplifying the problem and ensuring that solutions are always reproducible. It is important to look at the 'big picture' in order to grasp the core of the problem before focusing on the small details. In this process, information is pre-processed to group related data so we can avoid too much detail. This simplification can cause common characteristics to emerge, which can provide a balance between the accuracy and generality of problem illustration. The easiest and least determined way of problem simplification is when the problem is divided and then solved in stages. This can be achieved through information granulation, explained in Chapter 4. The concept of granular computing (Bargiela and Pedrycz, 2002; Bargiela *et al.*, 2004; Bargiela and Pedrycz, 2008) provided important information for efficient scheduling.

This thesis proposed a domain transformation approach to solving NSPs. The domain transformation approach transforms the original scheduling problem domain into smaller sub-domains. This reduces the problems' complexities effectively and can be converted back into the original problem domain. By subdividing the real-world NSPs into smaller sub-problems, each problem was solved more effectively. By performing this step, we successfully mapped the original problem expressed in the multi-dimensional domain onto a reduced dimensionality domain. Another important highlight of our proposed approach is the pattern generation. We generated the zero-cost and non-zero-cost patterns by granulating data and constraints. This provides more meaningful information that can be utilised at a later stage in the scheduling process. When performing this granulation, certain data and constraints are grouped according to certain conditions (e.g., granulation of constraints, granulation of shift and granulation of patterns into sequence). This reduces cross-checking and cross-referencing in big data during scheduling. Further, the schedule is feasible (in terms of demand of nurse cover, and all hard constraints are satisfied) and is generated in a short amount of time.

An important issue when solving NSPs is how to handle complex constraints. The complexity of the dataset of NSPs requires finding a feasible schedule with a precise number of nurses per day, which fulfils many soft constraints as well. The large number of complicated constraints cannot easily be applied to limit the space of acceptable solutions (Smith and Wiggins, 1977; Weil *et al.*, 1995). These constraints concern, for example, continuity in service, personnel policies, staff preferences, operating budgets and labour constraints (see Rosenbloom and Goertzen, 1987). Additionally, some of these considerations may be in conflict with others, such as employment requests and the need to balance the workload (Randhawa and Sitompul, 1993). Also Ozkarahan and Bailey (1988, p. 306) stress the conflicting objectives and constraints of NSPs.

This is proved in our domain transformation algorithm, which runs quickly and produces good solutions. In real nurse scheduling settings, we noticed that the problems are nearly always over-constrained. The feasible solutions produced in previous studies have tended to be expensive in terms of violating constraints. Moreover, it has often been difficult to find a solution (Focacci *et al.*, 1999). A common characteristic of many methods in the surveyed literature is that they have a tendency to converge towards feasible solutions when small modifications made after the schedule is generated will produce a better result (Forest and Michell, 1993; Rawlins, 1991).

The Nottingham benchmarks (Burke *et al.,* 2010) are a collection of SSP instances gathered from various resources and published online (see http://www.cs.nott.ac.uk/_tec/NRP/). The diversity of its resources makes the Nottingham benchmarks a valuable sample of international SSPs. These benchmark instances allow researchers to compare their algorithms to other approaches that have been independently implemented. This increases the credibility of results and conclusions and helps reviewers better gauge the strength of new methods.

The proposed method for NSPs utilising a general novel algorithm and being evaluated using three techniques are described in Chapter 5. All techniques have been shown to deliver consistently competitive results for all real-world benchmark datasets. Comparing the average mean deviations from the best-known solution for each benchmark dataset, our method shows that the performances are consistent for all types of problem and on average outperforms all other results. The variance of these deviations is smallest for our method when compared to others reported in the literature—other constructive methods found in the literature demonstrated an irregular performance, where they performed well on some benchmark problems and less well on others (Junker *et al.,* 1999; Yunes *et al.,* 2000). Further, by automating NSPs, the scheduling effort and calculation time are reduced considerably from the manual approach that was previously used in our new real-world dataset (Kajang Hospital). The quality of the automatically produced schedules is much higher than the quality of the manual schedules.

Existing research shows the inefficiency of manual schedules for large and complex NSPs (Howe, 2001; Burke *et al.,* 2010), especially in the continuity of scheduling from month to month. This is also observed through our experiments in Chapter 5. Our approach has successfully maintained continuity for highly constrained and large-scale NSPs. However, some of these methods, especially the design of neighbourhood structures in the literature, are tailored to a specific problem instance

(Ahuja, 2002). With each alteration in a local search, solutions need to be checked to preserve feasibility (Kilby *et al.*, 2000). Conversely, our approach is tailored in a more widely applicable and general way. According to Dowsland (1997, page 394), *a general algorithm is like a size 48 cloth. It will cover everybody but it does not fit anyone very well.* Moreover, general algorithms are designed from the management viewpoint and do not consider special constraints, like ergonomic criteria. Other sources used in building scheduling policies are the current applied policies in the hospital, as well as recommended policies displayed in the literature that account for ergonomic factors. Therefore, the developed model performs quite well based on the quality criteria. The model has been found not only to satisfy hospital objectives but also, and to a larger extent, nurses' preferences (proportionality, days off, isolated days on and off, etc.).

Due to the need to ensure the feasibility of schedules, our general algorithm checks if patterns allocated to nurses satisfy the demand for that particular week. For conventional approaches, without the information granulation stage, clashing information is implicit in data; thus, a lot of permutations requiring a lot of time need to be conducted in order to create a feasible timetable. This problem can be avoided using the approach proposed in this study. The information granulation stage is one of the biggest contributions of this thesis towards solving and minimising the search domain.

Many techniques have been applied to explore the neighbourhood in the schedule or select elements randomly; however, the majority suffer from the risk that if the cost is not reduced, the initial cost will be replaced by the current solution (Nareyek, 2001; Junker *et al.*, 1999; Yunes *et al.,* 2000; Fahle *et al.,* 2002). Also problematic is the fact that sometimes the methods used are not reproducible (Gendron, *et al.* 2005). Conversely, the approach taken in this thesis is systematic. It transforms the NSP into a simpler problem with fewer shift types and constant nurse demand and availability.

To summarise, it is clear that the domain transformation approach proposed in this study is very simple and competitive in terms of generating reliably high-quality schedules. By transforming the original real-world scheduling problem into smaller sub-problems and applying appropriate granulations, we managed to reduce the complexities of the problem, thus saving a substantial computational effort compared to other methods. The implication of this is that different simplifications can be obtained at different levels of construction. Further, we can reduce the search space at a more abstract level, while retaining the opportunity of improvement by using a more detailed representation. Such simplification can save computation effort, as well as allow for greater ease of handling the data. The problems are not characterised just at the most detailed level, but include a diverse construction at a higher level.

The approach proposed in this study to solve NSPs is efficient and reliable in producing high-quality schedules. It has consistently produced encouraging results for all real-world benchmark datasets, which is not the case for some other constructive methods in the literature. They perform well on some benchmark problems and not as well on others, and in a few cases some methods fail to produce a solution. This is a rather unwelcome characteristic from the user's perspective, as there is no way of predicting the quality of the solution that will be obtained using a particular method on a new dataset. Since the proposed approach produces consistent results when tested on different real-world benchmark datasets and a new dataset, the method is shown to be very flexible and has quality, consistency and potential for universal application.

This proposed approach is not limited to NSPs; it is a general approach of looking at problems at different levels of construction. By doing so, a spectrum of possibilities becomes available: we can reduce either the search domain or the complexity of the problem by looking at a more abstract level, or we can gain a more detailed description of the problem by examining a more detailed level. We would also like to highlight that an important feature of the proposed approach is that its deterministic pattern in the results generated for all datasets is always preserved, which makes it a novel contribution to the NSP research field.

The content of the present dissertation has been the subject of journal articles, conference papers and conference abstracts (see 'Publications/Disseminations during PhD Period').

6.2 Future Research

In this section, we identify a number of opportunities for future research in the field of nurse scheduling. In our future work, a few extensions of this work could prove interesting. First, the domain transformation approach developed here could be extended to solve a wider range of NSPs and other SSP. Hospitality management is a promising area that should be more explored, namely hotels (housekeeping staff) and restaurants.

Combining the approach with state estimation could provide better judgment with a greater number of more diverse schedules. The estimation of the state of a system that is monitored through measurements that have limited accuracy has long been recognised as a challenging practical problem. This is primarily because it becomes necessary to identify a much larger set of possible system states (Bargiela, Pedrycz and Tanaka, 2003; Hashemian and Armaou, 2014). This can be evaluated using the sensitivity matrix method to find a set of efficient solutions. The basis of this method is the observation that when the measurement set is minimal (i.e., it is observable and contains no observable subset), the linearized uncertainty bounds can be calculated without recourse to a linear programming procedure (Bargiela, 2001). Bargiela *et al.* (2003), Yang Fan and Xiao Deyun (2008) and Lou (2015) present a sensitivity matrix method that offers a good compromise between the accuracy and efficiency of estimation of the state uncertainty set. Thirdly, as our domain transformation approach has demonstrated computational efficiency comparable to previous approaches reported in the literature, implementing this approach to find diverse solutions in NSPs could be interesting. An extension to the Kajang Hospital NSP could include simulation of patient number per nurse. This would involve modifying the solution procedure. The domain transformation model could also be expanded for all units, departments and wards in Kajang Hospital.

Discussions have already begun regarding the possibility of designing the graphical user interface for our model for use in Kajang Hospital. Future work may also focus on building a user-friendly computer package. Importantly, there is also a need to improve the system to allow the head nurse to regenerate the system in the case of unexpected occurrences (i.e., no feasible solution). In this thesis, we have constructed a deterministic nurse- scheduling model capable of defining the demand for each nurse. However, hospital administration systems operate in a dynamic and uncertain environment in which unexpected events lead to schedule disruptions and infeasibilities. Therefore, most hospitals will at some time confront the problem of rescheduling, where it is necessary to update the activity schedule. Rescheduling is thus an important and interesting topic, and one that would be valuable to investigate using the Hospital Kajang dataset, to assess how our modelled system reacts to unexpected events.

6.3 Conclusions

The approach this thesis proposes to solving NSPs has proven to be very effective in generating feasible schedules. A general algorithm was developed and applied with three techniques to 27 real-world benchmark datasets in NSPs. ORTEC benchmark datasets, which are most challenging and most studied in the literature, were used in order to test the insight of our proposed approach. Our results were compared to those of other approaches tested in the literature for all real-world benchmark datasets.

Three main goals of the proposed model were to develop and evaluate general algorithms using a novel approach of information granulation to achieve a feasible solution with minimum cost, flexibility in staff scheduling solutions and continuity in the scheduling process. These goals were achieved by implementing the domain transformation approach based on the insight of information granulation allowing the transformation of the NSP into a smaller problem domain that could be solved in stages and a reasonable amount of time. This resulted in reducing the complexity of the problem domain.

This original formulation also makes it possible to control the periodicity of rest days as well as the length of the planning horizon. The definition of the planning period is not a much-explored issue in the literature, since it is closely related to demand forecast periods and is often an input parameter. However, the initially set planning period may not be the best fit to a problem's features, and so it was pertinent to determine the 'ideal' planning period for a specific instance. This experiment was carried out in Kajang Hospital, Malaysia. The proposed approach developed in this study was shown to be generalizable and flexible, with several degrees of freedom and with the capacity of being easily applied to different real-life SSPs.

This is a new finding and represents a novel contribution to the academic literature. From a healthcare organisation (Kajang Hospital) point of view, the use of the flexible scheduling approach proposed in this study represents a powerful tool for increasing both the efficiency and the effectiveness of the staff scheduling process, leading to higher profitability and productivity. However, the implementation of such a solution into practice is not always easy; it depends on the involvement of the organisation's management in the whole development process. This was positively achieved in our new real-world dataset from Kajang Hospital.

Information granulation and pattern generation leads to cost reduction in producing a feasible schedule. The proposed method is systematic, robust and proven flexible, since it has been tested in real-world benchmark applications. Further, through the avoidance of meticulous searches, which normally use the random selection process, the proposed approach is capable of generating a solution that is reproducible and consistent. The generality of the algorithm used in various different nurse scheduling benchmark problems demonstrates that the proposed method can also be applicable to a wide range of SSPs. The deterministic process of both information granulation and pattern generations is an unique achievement, since there appears to be no prior research of deterministic patterns used in scheduling.

In conclusion, this thesis has proposed a generic, novel and valuable approach to SSPs, which was tested in NSP real-world benchmark studies. We developed general methodologies, showed their flexibility and solved 27 real-world benchmark datasets of NSPs. We developed an innovative formulation of sequences and consecutiveness of shifts. We contend that this study can add value to any healthcare application by leading to cost reduction and an increase in productivity.

References

Abdennadher, S. & Schlenker, H (1999). Nurse scheduling using constraint logic programming. In *Proceedings of the 16th National Conference on Artificial Intelligence and the 11th Innovative Applications of Artificial Intelligence Conference (IAAI99)*, Orlando, Florida, pp. 838–843.

Abernathy, W. J., Baloff, N., Hershey, J. & Wandel, S. (1973). A three-stage manpower planning and scheduling model—a service- sector example. *Operations Research, 21*(3), 693–711.

Abernathy, W., Balo, N., Hershey, J. & Wandel, S. (1973). A three-stage manpower planning and scheduling model: A service sector example. *Operations Research, 22*, 693–711.

Ademir A. C., Dario L. S., Everton L. d. M. & Wesley R (2011). A heuristic algorithm for nurse scheduling with balanced preference satisfaction. In *Proceedings of the 2011 IEEE Symposium on Computational Intelligence in Scheduling (IEEE-CISched 2011)*, pp. 39–45, IEEE Press, Paris France, April 2011.

Aggarwal S. (1982). A focused review of scheduling in services. European *Journal of Operational Research, 9*, 114–121.

Agyei, W., Obeng-Denteh, W. & Andaam, E. (2015). Modeling nurse scheduling problem using 0–1 goal programming: A case study of Tafo Government Hospital, Kumasi-Ghana. *International Journal of Scientific & Technology Research, 4*.

Ahuja, R.K. *et al.* (2012). A survey of very large-scale neighbourhood search techniques. *Discrete Appl. Math., 123*(1–3), 75–102.

Aickelin, U. & Dowsland, K. (2000). Exploiting problem structure in a genetic algorithm approach to a nurse rostering problem. *Journal of Scheduling, 3*(3), 139–153.

Aickelin, U. & Dowsland, K. A. (2004). An indirect genetic algorithm for a nurse scheduling problem. *Computers and Operations Research, 31*(5), 761–778.

Aickelin, U. & Li, J. (2007). An estimation of distribution algorithm for nurse scheduling. *Annals of Operations Research, 155*(1), 289–309.

Aickelin, U. & White, P. (2004). Building better nurse scheduling algorithms. *Annals of Operations Research, 128*, 159–177.

Aickelin, U., Burke, E. K. & Li, J. (2007). An estimation of distribution algorithm with intelligent local search for rule-based nurse rostering. *Journal of the Operational Research Society, 58*(12), 1574–1585.

Aiken, L. H., Clarke, S. P., Sloane, D. M., Sochalski, J. & Silber, J. H. (2002). Hospital nurse staffing and patient mortality, nurse burnout, and job dissatisfaction. *The Journal of American Medical Association, 288*(16), 1987–1993.

Arthur, J. L. & Ravindran, A. (1981). A multiple objective nurse scheduling model. *AIIE Transaction, 13*(1), 56–60.

Awadallah, M.A., Khader, A.T., Al-Betar, M.A. & Bolaji, A.L. (2013). Hybrid harmony search for nurse rostering problems. In *Proceedings of the IEEE symposium on computational intelligence in scheduling (SCIS)* (pp. 60–7).

Azaiez, M. N. & Al Sharif, S. S. (2005). A 0–1 goal programming model for nurse scheduling. *Computers and Operations Research, 32*, 491–507.

Bagheri, M., Gholinejad Devin, A. & Izanloo, A. (2014). A two-stage stochastic programming for nurse scheduling in Razavi Hospital. *Razavi Int J Med, 3*(1), e23622.

Bai, R. Burke, E. K., Kendal, G., Li, J. P. & McCollum, B. (2007). A hybrid evolutionary approach to the nurse rostering problem. *IEEE Transaction on Evolutionary Computation*, pp. 1–10.

Bai, R., Burke, E. K., Kendall, G., Li, J. & McCollum, B. (2010). An efficient evolutionary hybrid algorithm for the nurse rostering problem. *IEEE Transactions on Evolutionary Computation, 14*(4), 580–590.

Bailey, J. (1985). Integrated days off and shift personnel scheduling. *Computer and Operations Research, 9*(4), 395–402.

Balakrishnan, N. & Wong, R. T. (1990). A network model for the rotating workforce scheduling problem. *Networks, 20*, 25–41.

Balekar, S. & Mhetre, N. (2015). Parallelizing genetic operators for solving nurse scheduling problem for GPGPU. *International Journal of Computer Science and Information Technology Research, 3*, 78–83.

Ballnski, M. L. (1965). Integer programming: Methods uses computation, *Management Science, 12*, 253–313.

Bard, J. F. & Purnomo, H. W. (2005a). Hospital-wide reactive scheduling of nurses with preference considerations. *IIE Transactions, 37*, 587–608.

Bard, J. F. & Purnomo, H. W. (2005b). Short-term nurse scheduling in response to daily fluctuations in supply and demand. *Healthcare Management Science, 8*, 315–324.

Bard, J. F. & Purnomo, H. W. (2005c). A column generation-based approach to solve the preference scheduling problem for nurses with downgrading. *Socio-Economic Planning Sciences, 39*, 193–213.

Bard, J. F. & Purnomo, H. W. (2005d). Preference scheduling for nurses using column generation. *European Journal of Operational Research, 164*, 510–534.

Bard, J. F. & Purnomo, H. W. (2007). Cyclic preference scheduling of nurses using a lagrangian based heuristic. *Journal of Scheduling, 10*, 5–23.

Bard, J. F. (2004a). Selecting the appropriate input dataset when configuring a permanent workforce. *Computers and Industrial Engineering, 47*(4), 371–389.

Bard, J. F. (2004b). Staff scheduling in high volume service facilities with downgrading. *IIE Transactions, 36*(10), 985–997.

Bargiela A. (1985). An algorithm for observe ability determination in water- system state estimation. *IEEE Proceedings Part D, 132*, 245–249.

Bargiela A., Pedrycz W. & Hirota K. (2004). Granular prototyping in fuzzy clustering. *IEEE Transactions on Fuzzy Systems, 12*(5), 697–709

Bargiela, A. & Pedrycz, W. (Eds.). (2002). *Granular computing: An introduction*. Boston: Kluwer Academic Publishers.

Bargiela, A. & Pedrycz, W. (Eds.) (2003). *Granular computing: An introduction*. Boston: Kluwer Academic Publishers.

Bargiela, A. (2001). Interval and ellipsoidal uncertainty in water system state estimation. In W. Pedrycz (Eds.), *Granular computing* (pp. 23–57). New York: SpringerVerlag.

Bargiela, A. and 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.2007.905 912.

Bargiela, A., Pedrycz, W. & Tanaka, M. (2003). A study of uncertain state estimation, *IEEE Transactions on Systems, Man and Cybernetics: Part A: Systems and Humans, 33*(3), 288–301. doi:10.1109/TSMCA.2002.806500

Barker, K. R. (1974). *Elements of sequencing and scheduling*. New York: John Wiley and Sons.

Baskaran, G., Bargiela, A. & Qu, R. (2009). Hierarchical method for nurse rostering based on granular pre-processing of constraints. In *Proceedings of the 23rd European Conference on Modelling and Simulation (ECMS)*, Madrid, Spain, pp. 855–861. doi: 10.7148/2009–0855–0861

Baskaran, G., Bargiela, A. & Qu, R. (2012). From simplified to detailed solutions to the nurse scheduling problem. In *25th European Conference on Operational Research (EURO)*, Vilnius, Latvia, pp. 134–135.

Baskaran, G., Bargiela, A. & Qu, R. (2013b). Integer programming: Domain transformation in nurse scheduling problem: WASET, Melbourne, Australia. In *International Conference on Computer Science and Information Engineering (ICCSIE 2013)*, p. 1403.

Baskaran, G., Bargiela, A. & Qu, R. (2014a). Simulation of scheduling and cost effectiveness of nurses using domain transformation method. In *28th European Conference for Modelling and Simulation (ECMS 2014)*, Brescia, Italy, p. 28.

Baskaran, G., Bargiela, A. & Qu, R. (2014b). New solution for a benchmark nurse scheduling problem using integer programming. In *International Conference on Information, Management Science and Application (ICIMSA 2014)*, Beijing, China.

Baskaran, G., Bargiela, A. & Qu, R. (2014c). Seeing the bigger picture: Domain transformation approach to the nurse scheduling problem. *Australian Journal of Basic and Applied Sciences (AENSI Journal), 8*(24) Special, 308–320.

Baskaran, G., Bargiela, A. & Qu, R. (2014d). Integer programming: Using branch-and-bound to solve the nurse scheduling problem. In *2014 International Conference on Artificial Intelligence and Manufacturing Engineering (IIE ICAIME2014)*, Dubai, UAE, 25–26 December, Paper ID: U1214025, ISBN 978–93–84468–11–8.

Baskaran, G., Bargiela, A. & Qu, R. (2015). Domain transformation using greedy algorithm in nurse scheduling. In *2015 International Conference on Artificial Intelligence, Energy and Manufacturing Engineering (IIE ICAEME 2015)*, Dubai, UAE, 7–8 January, Paper ID: E0115039, ISBN: 978–93–84422–05–9

Baskaran, G., Bargiela, A. and Qu, R. (2013a). A study of cost effective scheduling of nurses based on the domain transformation method. In *27th European Conference on Modelling and Simulation (ECMS 2013)*, Alesund, Norway, p. 27.

Beasley, J. E. (1996). *Advances in linear and integer programming*. Oxford University Press

Bechtold, E., Brusco, M. J. & Showalter, M. J. (1991). A comparative evaluation of labour tour scheduling methods. *Decision Sciences, 22*, 683–699.

Beddoe, G. R. & Petrovic, S. (2006). Selecting and weighting features using a genetic algorithm in a case-based reasoning approach to personnel rostering. *European Journal of Operational Research, 175*, 649–671.

Beddoe, G. R. & Petrovic, S. (2007). Enhancing case-based reasoning for personnel rostering with selected tabu search concepts. *Journal of Computer Science and Information Technology*, University of Nottingham.

Beddoe, G. R., Petrovic. S. & Li J. (2009). A hybrid meta-heuristic case based reasoning system for nurse rostering. *Journal of Scheduling 12*(2), 99–119.

Belian, J. (2006). Exact and heuristic methodologies for scheduling in hospitals- problems, formulations and algorithms. *4OR: A Quarterly Journal of Operations Research, 5*(2), 157–160.

Bellanti, F., Carello, G., Della Croce, F. & Tadei, R. (2004). A greedy-based neighborhood search approach to a nurse rostering problem. *European Journal of Operational Research, 153*, 28–40.

Berman, O., R. C. Larson & E. Pinker (1997). Scheduling workforce and workflow in a high volume factory. *Management Science, 43*(2), 158–172.

Berrada, I., Ferland, J. A. & Michelon, P. (1996). A multi-objective approach to nurse scheduling with both hard and soft constraints. *Socio- Economic Planning Sciences, 30*(3), 183–193.

Bester, M., Nieuwoudt, I. & Vuuren, J. H. V. (2007). Finding good nurse duty schedules: A case study. *Journal of Scheduling, 10*(6), 387–405.

Bilgin, B., Causmaecker, P. D & Rossie, B. (2012) Local search neighbourhoods for dealing with a novel nurse rostering model. *Journal: Annals of Operations Research – Annals*, 1–25.

Blau, R. A. & Sear, A. M. (1983). Nurse scheduling with a microcomputer. Journal of Ambulatory Care Management, 6, 1–13.

Blöchliger, I. (2004). Modelling staff scheduling problems: A tutorial. *European Journal of Operational Research, 158*(3), 533–542.

Blythe, J., Baumann, A., Zeytinoglu, I., Denton, M. & Higgins, A. (2005). Full-time or part-time work in nursing: Preferences, trade-offs and choices. *Healthcare Quarterly, 8*(3), 69–77.

Bouarab, H., Champalle, S., Dagenais, M., Lahrichi, N., Legrain, A. & Taobane, M. (2010). Nurse scheduling: From theoretical modeling to practical resolution. *Society for Health Systems-Online material in Staffing and Labour Management*, 1–10.

Brigitte, J., Semet, F. & Vovor, T. (1998). A generalized linear programming model for nurse scheduling. *European Journal of Operational Research, 107*, 1–18.

Brucker, P., Burke, E., Curtois, T., Qu, R. & Berghe, G. V. (2009). A shift sequence based approach for nurse scheduling and a new benchmark dataset. *Journal of Heuristics*, 1–13.

Brucker, P., Burke, E., Curtois, T., Qu, R. & Berghe, G. V. (2010). A shift sequence based approach for nurse scheduling and a new benchmark dataset. *Journal of Heuristic, 16*(4), 559–573.

Brucker, P., Qu, R., Burke, E. & Post, G. (2005). A decomposition, construction and post-processing approach for nurse rostering. In *Multidisciplinary International Scheduling Conference* (pp. 397–406).

Brusco, M. & Jacobs, L. (2001). Starting-time decisions in labor tour scheduling: an experiment analysis and case study. *European Journal of Operational Research, 131*, 459–475.

Burke, E. K. & Newall, J. P. (2004). Solving examination timetabling problems through adaptation of heuristic orderings. *Annals of Operations Research, 129*, 107–134.

Burke, E. K., Cowling, P., Caumaecker, P. D. & Berghe, G. V. (2001). A memetic approach to the nurse rostering problem. *Applied Intelligence special issue on Simulated Evolution and Learning, 15*(3), 199–214.

Burke, E. K., Curtois, T., Post, G., Qu, R. & Veltman, B. (2008). A hybrid heuristic ordering and variable neighourhood search for the nurse rostering problem. *European Journal of Operational Research, 188*(2), 330–341.

Burke, E. K., Curtois, T., Qu, R. & Berghe, G. V. (2007). *A scatter search for the nurse rostering problem*. Computer Science Technical Report. University of Nottingham.

Burke, E. K., Curtois, T., Qu, R. & Berghe, G. V. (2013). A time predefined Variable Depth Search for nurse rostering. INFORMS *Journal on Computing, 25*(3), 411–419.

Burke, E. K., Curtois, T., Qu, R. & Vanden Berghe, G. (2010). A scatter search methodology for the nurse rostering problem. *Journal of the Operational Research Society*, *61*, 1667–1679.

Burke, E. K., De Causmaecker, P. & VandenBerghe, G. (2004b). Novel metaheuristic approaches to nurse rostering problems in Belgian hospitals. In J. Leung (Eds.), *Handbook of scheduling: Algorithms, models, and performance analysis* (pp. 44.1–44.18). CRC Press.

Burke, E. K., De Causmaecker, P., Petrovic, S. & VandenBerghe, G. (2004a). Variable neighbourhood search for nurse rostering problems. In M. G. C. Resende & J. P. De Sousa (Eds.), *Metaheuristics: Computer decision-making* (pp. 153–172). Kluwer.

Burke, E. K., De Causmaecker, P., VandenBerghe, G. & Van Landeghem, H. (2004c). The state of the art of nurse rostering. *Journal of Scheduling, 7*(6), 441–499.

Burke, E. K., Elliman, D. G. & Weare, R. F. (1994a). A genetic algorithm for university timetabling. In *Proceedings of the AISB workshop on Evolutionary Computing*, University of Leeds, UK.

Burke, E. K., Li, J. & Qu, R. (2009). A hybrid model of integer programming and variable neighbourhood search for highly-constrained rostering problems. *European Journal of Operational Research, 203*(2), 484–493.

Cai, X. & Li, K. N. (2000). A genetic algorithm for scheduling staff of mixed skills under multicriteria. *European Journal of Operational Research, 125*, 359–369.

Carlos, C. C. (2001). A short tutorial on evolutionary multiobjective optimization. In *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization* (pp. 21–40). Springer.

Celia, A. G. & Roger, A. K. (2010). The nurse rostering problem: A critical appraisal of the problem structure. *European Journal of Operational Research, 202*, 379–389. doi:10.1016/j.ejor.2009.05.046

Cheang, B., Li, H., Lim, A. & Rodrigues, B. (2003). Nurse rostering problems—A bibliographic survey. *European Journal of Operational Research, 151*, 447–460.

Chen, M.-C., Chen, L.-S., Hsu, C.-C. & Zeng, W.-R. (2008). An information granulation based data mining approach for classifying imbalanced data. *Information Sciences, 178*, 3214–3227

Chiaramonte, M. V. & Chiaramonte, L. M. (2008). An agent-based nurse rostering system under minimal staffing conditions. *International Journal of Production Economics, 114*(2), 697–713.

Chiaramonte, M. V. (2008). *Competitive nurse rostering and rerostering*. Arizona State University.

Chow, K. P. & Hui, C. K. (1993). Knowledge-based system for rostering. *Expert Systems with Applications, 6*, 361–375.

Christie + Co. (2015). *The UK nursing workforce crisis or opportunity?, Nurse Staff in the NHS 2015.* Retrieved from http://www.nursingtimes.net/download?ac=1303857.

Constantino, A. A., Landa-Silva, D., de Melo, E. L. & Romao, W. (2011). A heuristic algorithm for nurse scheduling with balanced preference satisfaction. In *IEEE Symposium on Computational Intelligence in Scheduling* (pp. 39–45).

Costa, G. (1996). The impact of shift and night work on health. *Applied Ergonomics, 27*, 9–16.

Darmoni, S. J., Fajner, A., Mahe, N., Leforestier, A., Vondracek, M., Stelian, O. & Baldenweck, M. (1995). Horoplan: Computer-assisted nurse scheduling using constraint-based programming. *Journal of Society for Health Systems, 5*, 41–54.

De Causmaecker, P. & VandenBerghe, G. (2010). Towards a reference model for timetabling and rostering. *Annals of Operations Research, 63*, 105–128.

De Vries, G. (1987). Nursing workload measurement as management information. *European Journal of Operational Research, 29*, 199–208.

Dias, T. M., Ferber, D. F., Souza, C. C. & Moura, A. V. (2003). Constructing nurse schedules at large hospitals. *International Transactions in Operational Research, 10*, 245–265.

Dowsland, K. (1998). Nurse scheduling with Tabu search and strategic oscillation. *European Journal of Operational Research, 106*, 393–407.

Dowsland, K. (1998). Nurse scheduling with Tabu search and strategic oscillation. *European Journal of Operational Research*, *106*, 393–407.

Dowsland, K. A. & Thompson, J. M. (2000). Solving a nurse scheduling problem with knapsacks, networks and Tabu search. *Journal of the Operational Research Society, 51*, 825–833.

Dowsland, K. A., Herbert, E. A, Kendall, G. & Burke, E. (2006). Using tree search bounds to enhance a genetic algorithm approach to two rectangle packing problems. *European Journal of Operational Research, 168*, 390–402.

Draper, D., Jonsson, A., Clements, D. & Joslin, D. (1999). Cyclic scheduling. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence* (pp. 1016–1021).

Engku Abu Bakar, E. M. N. (2001). A heuristic to scheduling security personnel at local universities in Malaysia (Doctoral dissertation). Universiti Utara Malaysia, Malaysia.

Ernst, A. T., Jiang H., Krishnamoorthy M. & Sier D. (2004). Staff scheduling and rostering: A review of applications, methods and models. *European Journal of Operational Research, 153*(1), 3–27.

Ernst, A. T., Jiang, H., Krishnamoorthy, M., Owens, B. & Sier, D. (2004). Annotated bibliography of personnel scheduling and rostering. *Annals of Operations Research*, 127.

Fahle, T. *et al.* (2002). Constraint programming based column generation for crew assignment. *Journal of Heuristics, 8*(1), 59–81.

Fijn van Draat, L., Post, G. F., Veltman, B. & Winkelhuijzen, W. (2006). Harmonious personnel scheduling. *Medium Econometrische Toepassingen, 14*, 4–7.

Focacci, F., Lodi, A. & Milano, M. (1999). Cost-based domain filtering. In *Principles and practice of constraint programming – CP'99* (pp. 189–203).

Forrest, S. & Mitchell, M. (1993). What makes a problem hard for a genetic algorithm? Some anomalous results and their explanations. *Machine Learning, 13*, 285–319.

Fox, M. S. (1983). *Constraint–directed search: A case study of job-shop scheduling.* Technical Report, CMU-RI-TR-83–22, Robotics Institute, Carnegie Mellon University.

Frances, M. A. (1966). Implementing a program of cyclical scheduling of nursing personnel. *Hospitals, J. A. H. A., 40*, 108–125.

Fries, B. E. (1976). Bibliography of operations research in health-care systems. *Operations Research, 24*(5), 801–814.

Fung, S. K. L., Leung, H. F. & Lee, J. H. M. (2005). Guided complete search for nurse rostering problems. In *Proc. of the 17th IEEE*

International Conference on Tools with Artificial Intelligence, pp. 1082–3409.

Gaetan, C., Pierri, H. & Brigitte, J. (1999). The assignment problem with seniority and job priority constraints. *Operations Research, 47*, 449–453.

Gendron, B., Lebbah, H. & Pesant, G. (2005). Improving the cooperation between the master problem and the subproblem in constraint programming based column generation. In *Integration of AI and OR techniques in constraint programming for combinatorial optimization problems* (pp. 217–227).

Gino, J. L., Mobasher, A. & Murray, J. C. (2012). Multi-objective nurse scheduling models with patient workload and nurse preferences. *Management, 2*(5), 149–160. doi:10.5923/j.mm.20120205.03

Giunchglia, F. & Walsh, T. (1992). A theory of abstraction. *Artificial Intelligence, 56*, 323–390.

Glass, C. & Knight, R. (2010). The nurse rostering problem: A critical appraisal of the problem structure. *European Journal of Operational Research, 202*(2), 379–389.

Glover, F. & Mcmillan. C. (1986). The general employee scheduling problem: An integration of management science and artificial intelligence. *Computers and Operations Research, 13*, 563–573

Gonsalves, T. & Kuwata, K. (2015). Memetic algorithm for the nurse scheduling problem. *International Journal of Artificial Intelligence & Applications, 6*(4), 43–52. doi:10.5121/ijaia.2015.6404 43–52

Goodman, M. D., Dowsland, K. A. & Thompson, J. M. (2009). A grasp-knapsack hybrid for a nurse-scheduling problem. *Journal of Heuristics, 15*(4), 351–379.

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.

Graves, G. W., McBride, R. D., Gershkoff, D. A. & Mahidhara, D. (1993). Flight crew scheduling. *Management Science, 39*(6), 736–745.

Han, J., Cai, Y. And Cercone, N. (1993). Data-driven discovery of quantitative rules in data bases. *IEEE Transactions on Knowledge and Data Engineering, 5*, 29–40.

Harmeier, P. E. (1991). Linear programming for optimization of nurse scheduling. *Computers in Nursing, 9*(4), 149–151.

Hashemian, N. & Armaou, A. (2014). Moving horizon estimation using Carleman linearization and sensitivity analysis. *Annual Meeting AIChE*, Atlanta, GA.

He, F. & Qu, R. (2012). A constraint programming based column generation approach to nurse rostering problems. *Computers & OR, 39*(12), 3331–3343.

Healy, W. C. (1964). Multiple choice programming. *Operations Research, 12*, 122–138.

Henderson, V. A. (2006). The concept of nursing. *Journal of Advance Nursing, 53*(1), 21–34.

Henderson, W. B. & Berry, W. L. (1976). Heuristic methods for telephone operator shift scheduling: An experimental analysis. *Management Science, 22*, 1372–1380.

Hobbs, J. R. (1985). Granularity. In *Proceedings of the 9th International Joint Conference on Artificial Intelligence* (pp. 432–435).

Hofe, H.M.A.M. (2001). Nurse rostering as constraint satisfaction with fuzzy constraints and inferred control strategies. In *DIMACS workshop on constraint programming and large scale discrete optimization* (pp. 67–101). Piscataway, NJ: American Mathematical Society.

Hoffman, K. L. & Padberg, M. (1993). Solving airline crew scheduling problems by branch-and-cut. *Management Science, 39*(6), 657–682.

Holloran, T. J. & Byrn, J. E. (1986). United airlines station manpower planning system. *Interfaces, 16*, 39–50.

Horio, M. (2005). *A method for solving the 3-shift nurse scheduling problem through a general project scheduler based on the framework of RCPSP/τ.*

Howell, J. P. (1966). Cyclical scheduling of nursing personnel. *Hospitals, 40*, 77–85.

Huarng F. (1997). Integer goal programming model for nurse scheduling: A case study. *Lecture Notes in Economic and Mathematical System, 448*, 634–643.

Hung, R. (1995). Hospital nurse scheduling. *Journal of Nursing Administration, 25*(7/8), 21–23.

Ikegami, A. & Niwa, A. (2003). A sub problem-centric model and approach to the nurse scheduling problem. *Mathematical Programming, 97*(3), 517–541.

Isken, M. (2004). An implicit tour scheduling model with applications in healthcare. *Annals of Operations Research, 128*(1–4), 91–109.

Isken, M. W. & Hancock W. M. (1991). A heuristic approach to nurse scheduling in hospital units with non-stationary, urgent demand, and fixed staff size. *Journal of the Society for Health Systems, 2*(2), 24–40.

Jafari, H. & Salmasi, N. (2015). Maximizing the nurses' preferences in nurse scheduling problem: mathematical modeling and a meta-heuristic algorithm. *J Ind Eng Int, 11*, 439–458. doi:10.1007/s40092–015–0111–0

Jafari, H., Bateni, S., Daneshvar, P., Bateni, S. & Mahdioun, H. (2015). Fuzzy mathematical modeling approach for the nurse scheduling problem: a case study. *Int. J. Fuzzy Syst*. doi:10.1007/s40815–015–0051–2

Jaumard, B., Semet, F. & Vovor, T. (1998). A generalized linear programming model for nurse scheduling. *European Journal of Operational Research, 107*, 1–18.

Jelinek, R. C. & Kavois, J. A. (1992). Nurse staffing and scheduling: Past solutions and future directions. *Journal of Social Health System, 3*(4): 75–82.

Junker, U. *et al.* (1999). A framework for constraint programming based column generation. In *Principles and practice of constraint programming* (pp. 261–275).

Karp, R. M. (1972). Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher (Eds.), *Complexity of computer computations* (pp. 85–103). New York: Plenum Press.

Kasabov, N. (1996). *Foundations of neural networks, fuzzy systems, and knowledge engineering*. Cambridge, MA: MIT Press.

Kilby, P., Prosser, P. & Shaw, P. (2000). A comparison of traditional and constraint based heuristic methods on vehicle routing problems with side constraints. *Constraints, 5*(4), 389–414.

Kim, S., Ko, Y., Uhmn, S. & Kim, J. (2014). A strategy to improve performance of genetic algorithm for nurse scheduling problem. *International Journal of Software Engineering and Its Applications, 8*(1), 53–62.

Knauth, P. (1996). Design better shift system. *Applied Ergonomics, 27*, 39–44.

Knjazew, D. O. (2002). *A competent genetic algorithm for solving permutation and scheduling problems*. Norwell, MA: Kluwer Academic Publishers.

Knoblock, C. A. (1993). *Generating abstraction hierarchies: An automated approach to reducing search in planning*. Boston: Kluwer Academic Publishers.

Kostreva, M. M., Lescyski, M. D. and Passini, F. T. (1978). The nurse scheduling decision via mixed-integer programming. In *Proceedings of American Hospital Association Forum on Nurse Scheduling* (pp. 291–305).

Landa-Silva, D. & Le, K. (2008). A simple evolutionary algorithm with self-adaptation for multi-objective nurse scheduling. In K. Srensen, C. Cotta and M. Sevaux (Eds.), *Adaptive and multilevel metaheuristics* (pp. 133–155). Springer.

Lawler, E. W. & Wood, D. E. (1966). Branch-and-bound methods: A survey. *Operations Research, 14*, 699–719

Legrain, A., Bouarab, H. & Lahrichi, N. (2015). The nurse scheduling problem in real-life. *J Med Syst, 39*(1),160. doi: 10.1007/s10916–014–0160–8.

Leksakul, K. & Phetsawat, S. (2014). Nurse scheduling using genetic algorithm. *Mathematical Problems in Engineering*. http://dx.doi.org/10.1155/2014/246543

Li, H., Lim, A. & Rodrigues, B. (2003). A hybrid AI approach for nurse rostering problem. In *ACM Symposium on Applied Computing* (pp. 730–735).

Li, J. & Aickelin, U. (2006). BOA for nurse scheduling. In: Pelican, M., Sastry, K. & Cantu-Paz, E. (eds), *Scalable optimization via probabilistic modelling: From algorithms to applications* (pp. 315–332). Springer.

Li, J., Burke, E. K. & Qu, R. (2012). A pattern recognition based intelligent search method and two assignment problem case studies. *Applied Intelligence, 36*, 442–453.

Li, X., Qiu, T., Liu, Q. & Bai, X. (2010). Ontology building from incomplete information system based on granular computing. In *IEEE International Conference on Granular Computing*, San Jose, California.

Liang, B. & Turkcan, A. (2015). Acuity-based nurse assignment and patient scheduling in oncology clinics. *Healthcare Manag Sci*. doi:10.1007/s10729–014–9313-z

Lin, C., Kang, J., Chiang, D. & Chen, C. (2015). Nurse scheduling with joint normalized shift and day-off preference satisfaction using a genetic algorithm with immigrant scheme. *International Journal of Distributed Sensor Networks.*

Maenhout, B. & Vanhoucke, M. (2010). Branching strategies in a branch-and-price approach for a multiple objective nurse scheduling problem. *Journal of Scheduling, 13*(1), 77–93.

Maenhout, B. & VanHoucke, M. (2013). An integrated nurse staffing and scheduling analysis for long-term nursing staff allocation problem. *Omega, 41*, 485–99.

Maier-Rothe, C. & Wolfe, H. B. (1973). Cyclical scheduling and allocation of nursing staff. *Socio-Economic Planning Science, 7*, 471–487.

Majumdar, J. & Bhunia, A. K. (2007). Elitist genetic algorithm for assignment problem with imprecise goal. *European Journal of Operation Research, 177*, 684–692.

Mani, I. (1998). A theory of granularity and its application to problems of polysemy and under specification of meaning. In *Principles of*

Knowledge Representation and Reasoning, Proceedings of the Sixth International Conference (pp. 245–255).

Matthews, C. H. (2005). Using linear programming to minimize the cost of nurse personnel. *Journal of Healthcare Finance, 32*(1), 37–49.

McCalla, G., Greer, J., Barrie, J. & Pospisil, P. (1992). Granularity hierarchies. Computers and Mathematics with Applications, 23, 363–375.

Megeath, J. D. (1978). Successful hospital personnel scheduling. *Interfaces, 8*(2), 55–59.

Messelis T., De Causmaecker, P. & Vanden Berghe, G. (2013). Algorithm performance prediction for nurse rostering. In *Proceedings of the 6th multidisciplinary international scheduling conference: theory and applications (MISTA)*.

Messelis, T., Haspeslagh, S., Bilgin, B. VandenBerghe, G. & De Causmaecker, P. (2010). *Algorithm performance prediction for a real world optimisation problem*. Technical Report, K. U. Leuven Campus Kortrijk, Department of Computer Science, KAHO St.-Lieven, Information Technology.

Millar, H. H. & Kiragu, M. (1998). Cyclic and non-cyclic scheduling of 12h shift nurses by network programming. *European Journal of Research, 104*, 582–592.

Miller, H. E., William, P. & Gustave, J. R. (1976). Nurse scheduling using mathematical programming. *Operations Research, 24*(5), 857–870.

Moores, B., Garrod, N. & Briggs, G. (1978). The student nurse allocation problem: A formulation. *Omega, 6*(1), 93–96.

Moz, M. & Pato, M. V. (2007). A genetic algorithm approach to a nurse re-rostering problem. *Computers and Operations Research, 34*, 667–691.

Musa, A. A. & Saxena, U. (1984). Scheduling nurses using goal programming techniques. *IIE Transactions, 16*(3), 217–221.

Nareyek, A. (2001). Using global constraints for local search. In *DIMACS workshop on constraint programming and large scale discrete optimization* (pp. 9–29).

Naudin, E., Chan, P. Y. C., Hiroux, M., Zemmouri, T. & Weil, G. (2012). Analysis of three mathematical models of the staff rostering problem. *Journal of Scheduling, 15*, 23–38.

Okada, M. & Okada, M. (1988). Prolog-based system for nursing staff scheduling implemented on a personal computer. *Computers and Biomedical Research, 21*, 53–63.

Oldenkamp, J. H. & Simons, J. L. (1995). Quality factors of nursing schedules. *Planning and Evaluation-Proceeding AMICE, 95*, 69–74.

Oldenkamp, J. H. (1996). *Quality in fives: On the analysis, operationalization and application of nursing schedule quality*. University of Groningen

Osogami, T. & Imai, H. (2000). Classification of various neighbourhood operations for the nurse scheduling problem. *Lecture Notes in Computer Science, 1969*, 72–83.

Ozkaharan, I. (1989). A flexible nurse scheduling support system. *Computer Methods and Programs in Biomedicine, 30*(2/3), 145–153.

Ozkarahan, I. & Bailey, J. E. (1988). Goal programming model subsystem of a flexible nurse scheduling support system. *IIE Transactions, 20*(3), 306–316.

Papadimitriou, C. H. & Steiglitz, K. (1982). *Combinatorial optimisation: Algorithms and complexity*. Englewood Cliffs, New Jersey: Prentice-Hall.

Pawlak, Z. (1982). Rough sets. *International Journal of Computer and Information Sciences, 11*, 341–356.

Pawlak, Z. (1991). *Rough sets, theoretical aspects of reasoning about data*. Dordrecht: Kluwer Academic Publishers.

Peytchev, E., Bargiela, A. & Gessing, R. (1996). A predictive macroscopic city traffic flows simulation model. In *Proceedings of European Simulation Symposium ESS'96*, Genoa, 38–42.

Pinedoe, M. (2008). *Scheduling: Theory, algorithms and systems* (3rd ed.). Springer.

Purnomo, H. W. & Bard, J. F. (2007). Cyclic preference scheduling for nurses using branch and price. *Naval Research Logistic, 54*(2), 200–220.

Qu, R., Burke, E. K., McCollum, B., Merlot, L. T. & Lee, S. Y. (2009a). A survey of search methodologies and automated system development for examination timetabling. *Journal of Scheduling, 12*(1), 55–89.

Rahim, S. and Andrzej, B. (2009). Granular modelling of exam to slot allocation. In *Proceedings of the 23rd European Conference on Modelling and Simulation (ECMS 2009),* Madrid, Spain, pp. 861–866.

Ramli, R & Ahmad, S. N. I. (2011). Innovative neighbor generations in Tabu search technique for a nurse rostering problem. In *International Proceedings of Computer Science and Information Technology: Information and Electronics Engineering (ICIEE)*, Bangkok, Thailand, 28–29 May.

Randhawa, S. U. & Sitompul, D. (1993). A heuristic-based computerized nurse scheduling system. *Computers and Operations Research, 20*(8), 837–844.

Rawlins, G. J. E. (1991). *Foundations of genetic algorithms*. Morgan Kaufmann Publishers.

Rosenbloom, E. S. & Goertzen, N. F. (1987). Cyclic nurse scheduling. *European Journal of Operational Research, 31*, 19–23.

Rowland, H. & Rowland, B. (1997). *Nursing administration handbook* (4^{th} ed.). Aspen.

Saitta, L. & Zucker, J.-D. (1998). Semantic abstraction for concept representation and learning. In *Proceedings of the Symposium on Abstraction, Reformulation and Approximation* (pp. 103–120).

Satheesh Kumar, B., Nagalakshmi, G. & Kumaraguru, S. (2014). A shift sequence for nurse scheduling using linear programming problem. *Journal of Nursing and Health Science, 3*(6), 24–28.

Schindler, S. & Semmel, T. (1993). Station staffing at Pan American world airways. *Interfaces, 23*, 91–98.

Silvestro, R. & Silvestro, C. (2000). An evaluation of nurse rostering practices in the national health service. *Journal of Advanced Nursing, 32*(3), 525–535.

Sitompul, D. & Randhawa, S. (1990). Nurse rostering models: A state-of-the- art review. *Journal of the Society of Health Systems, 2*, 62–72.

Sitompul, D. (1992). Design and implementation of a heuristic-based decision support system for nurse scheduling (Doctoral dissertation). Oregon State University.

Smith, D. L, Wiggins, A. & Bird, D. (1979). Post-implementation experience with computer-assisted nurse scheduling in a large hospital. *INFOR, 17*(4), 309–321.

Smith, L. D. & Wiggins, A. (1977). A computer-based nurse scheduling system. *Computers and Operations Research, 4*, 195–212.

Solos, I.P., Tassopoulos, I.X. & Beligiannis, G.N. (2013). A generic two-phase stochastic variable neighborhood approach for effectively solving the nurse rostering problem. *Algorithms, 6*(2), 278–308.

Soubeiga, E. (2003). *Development and application of hyper heuristics to personnel scheduling*. The University of Nottingham.

Stepaniuk, J. & Skowron, A. (2005). Ontological framework for approximation. *Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing Lecture Notes in Computer Science, 3641*, 718–727.

Suman, B. & Kumar, P. (2006). A survey of simulated annealing as a tool for single and multi-objective optimization. *Operational Research Society, 57*, 1143–1160.

Taishan, L. (2015). Consider uncertain parameters based on sensitivity matrix. Retrieved from http://arxiv.org/abs/1503.08379.

Taylor, P. E. & Huxley, S. J. (1989). A break from tradition for the San Francisco police: Patrol officer scheduling using an optimization-based decision support system. *Interfaces, 19*, 4–24.

Thompson, G. M. (2007). Solving the multi-objective nurse scheduling problem with a weighted cost function. *Ann Oper Res., 155*, 279–288.

Thornton, J. & Sattar, A. (1997). Nurse rostering and integer programming revisited. In B. Verma & X. Yao (Eds.), *International conference on computational intelligence and multimedia applications* (pp. 49–58).

Trivedi, V. M. & Warner, M. (1976). A branch-and-bound algorithm for optimum allocation of oat nurses. *Management Science, 22*, 972–981.

Trivedi, V. M. (1981). A mixed integer goal programming model for nurse service budgeting. *Operations Research, 29*(5), 1019–1034.

Truax, B. (1990a). Time shifting of sampled sound with a real-time granulation technique. In *Proceedings of the 1990 International Computer Music Conference*, San Francisco, CA: Computer Music Association.

Truax, B. (1994b). Discovering inner complexity: Time-shifting and transposition with a real-time granulation technique. *Computer Music Journal, 18*(2), 38–48.

Tsai, C. C. & Li, S. H. A. (2009). A two-stage modelling with genetic algorithms for the nurse scheduling problem. *Expert Systems with Applications an International Journal, 36*, 9506–9512.

Ulrich, C. M., Wallen, G., Grady, C., *et al.* (2002). The nursing shortage and the quality of care. *The New England Journal of Medicine, 347*(14), 1118–1119.

Valouxis, C. & Housos, E. (2000). Hybrid optimization techniques for the work shift and rest assignment of nursing personnel. *Artificial Intelligence in Medicine, 2*0, 155–175.

Valouxis, C., Gogos, C., Goulas, G., Alefragis, P. & Housos, E. (2012). A systematic two phase approach for the nurse rostering problem. *European Journal Operation Research, 219*, 425–33.

Vanhoucke, M. & Maenhout, B. (2009). On the characterization and generation of nurse scheduling problem instances. *European Journal of Operational Research, 196*(2), 457–467.

Vlah, S., Lukač, Z & J Pacheco (2011). Use of VNS heuristics for scheduling of patients in hospital. *Journal of the Operational Research Society, 62*, 1227–1238.

Warner, D. M. (1976). Scheduling nursing personnel according to nursing preference: A mathematical programming approach. *Operations Research, 24*(5), 842–856.

Warner, M. & Prawda, J. (1972). A mathematical programming model for scheduling nursing personnel in a hospital. *Management Science, 19*, 411–422.

Weil, G., Heus, K., Francois, P. & Poujade, M. (1995). Constraint programming for nurse scheduling. *IEEE Engineering in Medicine and Biology Magazine, 14*(4), 417–422.

Winstanley, G. (2004). Distributed and devolved work allocating planning. *Applied Artificial Intelligence, 18*, 97–115.

Wolsey, L. A. & Nemhauser, G. L. (1999). *Integer and combinatorial optimization*. Wiley.

Wong, T. C. & Xu, M. & Chin, K. S. (2014). A two-stage heuristic approach for nurse scheduling problem: a case study in an emergency department. *Computers & Operations Research, 51*, 99–110.

Wren, A. (1996). Scheduling, timetabling and rostering: A special relationship? In E. K. Burke and P. Ross (Eds.), *Practice and Theory of Automated Timetabling: First International Conference Edinburgh* (pp. 45–75). Springer.

Wright, P. & Mahar, S. (2013). Centralized nurse scheduling to simultaneously improve schedule cost and nurse satisfaction. *Omega, 41*(6), 1042–1052.

Yang, F. & Xiao, D. (2008). Interval state estimation in signed directed graph based on sensitivity matrix method. In *2008 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings*, Victoria, Canada, pp. 672–675.

Yao, Y. Y. (2004a). A partition model of granular computing. *LNCS Transactions on Rough Set, 1*, 232–253.

Yao, Y. Y. (2004b). Granular computing. Com*puter Science (JiSuanJiKeXue), 31*, 1–5.

Yao, Y. Y. (2007). The art of granular computing. In M. Kryszkiewicz, J. F. Peters, H. Rybinski and A. Skowron (Eds.), *Proceeding of the International Conference on Rough Sets and Emerging Intelligent Systems Paradigms, LNAI 4585* (pp. 101–112). Berlin: Springer.

Yunes, T.H., Moura, A.V. & de Souza, C.C. (2000). Solving very large crew scheduling problems to optimality. In *ACM symposium on Applied computing* (pp. 446–451).

Zadeh, L. A. (1979). Fuzzy sets and information granulation. In M. Gupta, R. Ragade and R. Yager (Eds.), *Advances in fuzzy set theory and applications* (pp. 3–18). Amsterdam: North Holland Publishing.

Zadeh, L. A. (1997). Towards a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. *Fuzzy Sets and Systems, 19*, 111–127.

Zhang, B. & Zhang, L. (1992). *Theory and applications of problem solving*. North-Holland, Amsterdam.

Zhang, L. & Zhang, B. (Eds.) (2007). *Theory and application of problem solving theory and application of granular computing in quotient spaces* [in Chinese] (2^{nd} ed.). Beijing: Tsinghua University Press.

Appendix A: Schedule for 13 weeks

```
---------------------------------
Computed Schedule for 13 week(s)
---------------------------------
```

Days->	MTWTFSS	cost	MTWTFSS	cost	MTWTFSS	cost	MTWTFSS	cost	MTWTFSS	cost	MTWTFSS	cost	MTWTFSS	cost
Nurse 01(20):	--REERR	0	RRREERR	0	NNRRRRR	0	RRRREEE	0	RRRREEE	0	RRNNRRR	0	NNRRRRR	0
Nurse 02(20):	--RRRRR	0	RREEERR	0	REEERRR	0	EEERRRR	0	EERRRRR	0	NNRRRRR	0	EERRRRR	0
Nurse 03(20):	--REERR	0	RRRREEE	0	RRRREEE	0	RRRRNNN	0	RRREERR	0	RREERRR	0	REEERRR	0
Nurse 04(32):	--ELLRR	0	EELLRRR	0	REELLRR	0	RRREELL	0	NNRREEE	0	LLLRREE	0	RRREERR	20
Nurse 05(36):	--ERRLL	0	NNRRDDD	0	LLLRREE	0	ELLLRRR	0	EELLLRR	0	EELLLRR	0	REELLRR	0
Nurse 06(36):	--REEEL	0	LLRRNNN	0	RRREERR	20	EELLLRR	20	EEERRDL	20	LLRRRRR	20	DDLLLRR	20
Nurse 07(36):	--NNRRR	0	REELLRR	0	EEERRRR	20	REEELLL	20	RRRDDLL	20	LRRRNNN	20	RREEELL	20
Nurse 08(36):	--LLRRR	0	EELLLRR	0	DDLLLRR	0	DDLLLRR	0	RREEDDD	0	DRRREEE	0	LLLRREE	0
Nurse 09(36):	--LLLRR	0	DDDDDRR	0	RDLLLRR	0	DDDDDRR	0	RRLLNNN	0	RRREELL	0	LRRDNNN	0
Nurse 10(36):	--DDDRR	0	DDDDRRR	0	LLNNRRR	0	RREEEDD	0	DRRDDLD	0	DDRRDDL	0	LLRREEE	0
Nurse 11(36):	--LRREE	0	LLNNRRR	0	RRDDDLL	0	LRRRDDD	0	LLLRRRR	10	EELLRRR	10	RRDDLLL	10
Nurse 12(36):	--RRNNN	0	RRRRREE	20	ERREEDL	20	LLRRRRR	20	DDDLLRR	20	EEELLRR	20	DDDDDRR	20
Nurse 13(36):	--DDDRR	0	RREEDLL	0	LRRRDDD	0	NNRRREE	0	DDDDRRR	0	RDDDDDD	10	RRDLDRR	20
Nurse 14(36):	--DRRDD	0	DDDDRRR	0	DDDDDRR	0	DDDDRRR	0	LLNNRRR	0	RDDDDRR	0	DDNNRRR	0
Nurse 15(36):	--RRDDD	0	LLLRRDD	0	DLDRRLD	0	LLNNRRR	0	RDDELRR	0	RREEELD	0	ERRRDDD	0
Nurse 16(36):	--EDLLE	0	ERRRLLL	0	ERRDNNN	0	RRDDDRR	20	LLERRRR	20	DLDDLRR	20	ELLRRDD	20

Days->	MTWTFSS	cost	MTWTFSS	cost	MTWTFSS	cost	MTWTFSS	cost	MTWTFSS	cost	MTWTFSS	cost
Nurse 01(20):	EELLLRR	20	EELLRRR	20	NNRREEL	20	LLRRREE	20	LLLRRRR	20	REELLRR	20
Nurse 02(20):	RREEERR	0	NNRRRRR	0	EEERRRR	0	RDNNRRR	0	RRNNRRR	0	REERRRR	0
Nurse 03(20):	REEERRR	0	REERRRR	0	EERRRRR	0	EELLLRR	20	EERRREE	20	LLRRRRR	20
Nurse 04(32):	NNRREEE	20	RRREERR	20	REEELLL	20	RREEELL	20	LRRRNNN	20	RRNNRRR	20
Nurse 05(36):	REELLRR	0	LLNNRRR	0	DDLLLRR	0	EELLLRR	0	EELLRRR	0	EEERREE	10
Nurse 06(36):	LLLRREE	20	LLRRREE	20	LLNNRRR	20	EELLLRR	20	EERRREE	20	ERRREEL	20
Nurse 07(36):	LRREEDL	20	LLRREEE	20	LLLRREE	20	DRREELL	20	LRREERR	20	NNRRRRR	20
Nurse 08(36):	LLRRDDD	0	DRRRDLL	0	LRRDNNN	0	RRDDNNN	0	RREEELL	0	LRREELL	0
Nurse 09(36):	RRRDDLD	0	DRRRDDD	0	DRRRDDD	0	DDDRRDD	0	NNRRDLL	0	LRREEDD	0
Nurse 10(36):	DRRDNNN	0	RREERRR	20	RDLLLRR	20	RRRRRRR	20	RDDDDDD	20	RRLLNNN	20
Nurse 11(36):	RRNNRRR	20	EELLLRR	20	RREEEDD	20	DRRREEE	20	RREEEDD	20	DRRDDDD	20
Nurse 12(36):	DLLRRRR	20	DDLLLRR	20	DDDDDRR	20	LDDDRRR	20	DDLLLRR	20	RDLLLRR	20
Nurse 13(36):	DDDRRRR	20	EDDDDRR	20	RLDDRRR	20	RRRRRRR	20	RDDLLRR	20	DDDDDRR	20
Nurse 14(36):	RDDLLRR	0	RREEEDD	0	ERRRDLE	0	NNRRDDD	0	DRRRRRR	10	DDDDDRR	10
Nurse 15(36):	ERRDDLL	0	RRDDNNN	0	RRREERR	20	LLEDDRR	20	DLDDDRR	20	ELDELRR	20
Nurse 16(36):	EDDRRRR	20	RDDDLLL	20	RRDLRRR	20	RLEEDRR	20	RLEDLRR	20	RLLRRLE	20

Verifying total nurses available each day:

Total E:	0033322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322
Total D:	0033322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322
Total L:	0033322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322
Total N:	0011111	1111111	1111111	1111111	1111111	1111111	1111111	1111111	1111111	1111111	1111111	1111111	1111111

Appendix B: Schedule for 52 weeks

Week 1 to 13 of 52 weeks schedule

```
---------------------------------
Computed Schedule for 52 week(s)
---------------------------------
```

Days->	MTWTFSS	cost	MTWTFSS	cost	MTWTFSS	cost	MTWTFSS	cost	MTWTFSS	cost	MTWTFSS	cost	MTWTFSS	cost	MTWTFSS	cost	MTWTFSS	cost	MTWTFSS	cost	MTWTFSS	cost	MTWTFSS	cost	MTWTFSS	cost
Nurse 01(20):	--REERR	0	RRREERR	0	NNRRRRR	0	RRRREEE	0	RRRREEE	0	RRNNRRR	0	NNRRRRR	0	EELLLRR	20	EELLRRR	20	NNRREEL	20	LLRRREE	20	LLLRRRR	20	REELLRR	20
Nurse 02(20):	--RRRR	0	RREEERR	0	REEERRR	0	EEERRRR	0	EERRRRR	0	NNRRRRR	0	EERRRRR	0	RREEERR	0	NNRRRRR	0	EEERRRR	0	RDNNRRR	0	RRNNRRR	0	REERRRR	0
Nurse 03(20):	--REERR	0	RRRREEE	0	RRRREEE	0	RRRRNNN	0	RRREERR	0	RREERRR	0	REEERRR	0	REEERRR	0	REERRRR	0	EERRRRR	0	EELLLRR	20	EERRREE	20	LLRRRRR	20
Nurse 04(32):	--ELLRR	0	EELLRRR	0	REELLRR	0	RRREELL	0	NNRREEE	0	LLLRREE	0	RRREERR	20	NNRREEE	20	RRREERR	20	REEELLL	20	RREEELL	20	LRRRNNN	20	RRNNRRR	20
Nurse 05(36):	--ERRLL	0	NNRRDDD	0	LLLRREE	0	ELLLRRR	0	EELLLRR	0	EELLLRR	0	REELLRR	0	REELLRR	0	LLNNRRR	0	DDLLLRR	0	EELLLRR	0	EELLRRR	0	EEERREE	10
Nurse 06(36):	--REEEL	0	LLRRNNN	0	RRREERR	20	EELLLRR	20	EEERRDL	20	LLRRRRR	20	DDLLLRR	20	LLLRREE	20	LLRRREE	20	LLNNRRR	20	EELLLRR	20	EERRREE	20	ERRREEL	20
Nurse 07(36):	--NNRRR	0	REELLRR	0	EEERRRR	20	REEELLL	20	RRRDDLL	20	LRRRNNN	20	RREEELL	20	LRREEDL	20	LLRREEE	20	LLLRREE	20	DRREELL	20	LRREERR	20	NNRRRRR	20
Nurse 08(36):	--LLRRR	0	EELLLRR	0	DDLLLRR	0	DDLLLRR	0	RREEDDD	0	DRRREEE	0	LLLRREE	0	LLRRDDD	0	DRRRDLL	0	LRRDNNN	0	RRDDNNN	0	RREEELL	0	LRREELL	0
Nurse 09(36):	--LLLRR	0	DDDDDRR	0	RDLLLRR	0	DDDDDRR	0	RRLLNNN	0	RRREELL	0	LRRDNNN	0	RRRDDLD	0	DRRRDDD	0	DRRRDDD	0	DDDRRDD	0	NNRRDLL	0	LRREEDD	0
Nurse 10(36):	--DDDRR	0	DDDDRRR	0	LLNNRRR	0	RREEEDD	0	DRRDDLD	0	DDRRDDL	0	LLRREEE	0	DRRDNNN	0	RREERRR	20	RDLLLRR	20	RRRRRRR	20	RDDDDDD	20	RRLLNNN	20
Nurse 11(36):	--LRREE	0	LLNNRRR	0	RRDDDLL	0	LRRRDDD	0	LLLRRRR	10	EELLRRR	10	RRDDLLL	10	RRNNRRR	20	EELLLRR	20	RREEEDD	20	DRRREEE	20	RREEEDD	20	DRRDDDD	20
Nurse 12(36):	--RRNNN	0	RRRRREE	20	ERREEDL	20	LLRRRRR	20	DDDLLRR	20	EEELLRR	20	DDDDDRR	20	DLLRRRR	20	DDLLLRR	20	DDDDDRR	20	LDDDRRR	20	DDLLLRR	20	RDLLLRR	20
Nurse 13(36):	--DDDRR	0	RREEDLL	0	LRRRDDD	0	NNRRREE	0	DDDDRRR	0	RDDDDDD	10	RRDLDRR	20	DDDRRRR	20	EDDDDRR	20	RLDDRRR	20	RRRRRRR	20	RDDLLRR	20	DDDDDRR	20
Nurse 14(36):	--DRRDD	0	DDDDRRR	0	DDDDDRR	0	DDDDRRR	0	LLNNRRR	0	RDDDDRR	0	DDNNRRR	0	RDDLLRR	0	RREEEDD	0	ERRRDLE	0	NNRRDDD	0	DRRRRRR	10	DDDDDRR	10
Nurse 15(36):	--RRDDD	0	LLLRRDD	0	DLDRRLD	0	LLNNRRR	0	RDDELRR	0	RREEELD	0	ERRRDDD	0	ERRDDLL	0	RRDDNNN	0	RRREERR	20	LLEDDRR	20	DLDDDRR	20	ELDELRR	20
Nurse 16(36):	--EDLLE	0	ERRRLLL	0	ERRDNNN	0	RRDDDRR	20	LLERRRR	20	DLDDLRR	20	ELLRRDD	20	EDDRRRR	20	RDDDLLL	20	RRDLRRR	20	RLEEDRR	20	RLEDLRR	20	RLLRRLE	20

verifying total nurses available each day:

Total E:	0033322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322
Total D:	0033322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322
Total L:	0033322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322
Total N:	0011111	1111111	1111111	1111111	1111111	1111111	1111111	1111111	1111111	1111111	1111111	1111111	1111111

Continue Week 14 to 26 of 52 weeks schedule

MTWTFSS	cost	MTWTFSS	cost	MTWTFSS	cost	MTWTFSS	cost	MTWTFSS	cost	MTWTFSS	cost	MTWTFSS	cost	MTWTFSS	cost	MTWTFSS	cost	MTWTFSS	cost	MTWTFSS	cost	MTWTFSS	cost	MTWTFSS	cost
RREEERR	20	EELLLRR	20	EELLLRR	20	EELLLRR	20	EERRREE	20	RRREERR	20	EELLLRR	20	RRRRREE	20	ERRRNNN	20	RRRRRRR	20	RRNNRRR	20	EELLLRR	20	EEERRRR	20
EELLLRR	20	RDNNRRR	20	REEELLL	20	RREELLL	20	RRNNRRR	20	REEERRR	20	RRRRREE	20	RRNNRRR	20	RREELLL	20	RREELLL	20	RREERRR	20	REEELLL	20	RRNNRRR	20
EELLLRR	20	EEERRRR	20	EELLLRR	20	EEERRRR	20	EERRNNN	20	RRNNRRR	20	NNRRRRR	20	EEERREE	20	ELLLRRR	20	NNRRLLL	20	NNRREEE	20	LLLRRRR	20	EELLRRR	20
EERRREE	20	LLRRNNN	20	RRRRRRR	20	EELLLRR	20	EELLLRR	20	EERRREE	20	ELLLRRR	20	EELLLRR	20	EELLLRR	20	EERRNNN	20	RRRRLLL	20	NNRREEL	20	LRREERR	20
DLLLRRR	10	RRRRLLL	20	NNRRREE	20	DLLLRRR	20	DDLLLRR	20	RRLLNNN	20	RRREELL	20	LRRRDLL	20	LRREELL	20	LRRREEE	20	DRREERR	20	EELLLRR	20	REERREE	20
LRRREEE	20	RREEELL	20	LRRREEL	20	LRRRDLL	20	NNRRLLL	20	NNRREEL	20	LRRREEE	20	RRREERR	20	REELLRR	20	REEEDDD	20	RREERRR	20	DDDDDRR	20	DDLLLRR	20
RRREELL	20	LRRREEE	20	DRREERR	20	RDDRRRR	20	RREERRR	20	REERRDD	20	DLLLRRR	20	EELLLRR	20	LLLRREE	20	ELLLRRR	20	REEDLLL	20	RRREEED	20	DRRREEE	20
RRDDNNN	0	RRREEED	0	DRREEDD	0	RREEEDD	0	DRRREEL	0	LRRRDDL	0	LLRRDDL	0	LLRRNNN	0	RRNNRRR	20	EELLLRR	20	EELLLRR	20	RRRDNNN	20	RRREELL	20
LLRRLLL	0	NNRRRRR	10	RDDDDRR	10	NNRREEE	20	LLRRDDD	20	LLLRRLD	20	RREEERR	20	NNRRRDL	20	LLRREEE	20	RREEEDD	20	DRRDDRR	20	DDDDDRR	20	RDLLLRR	20
RREERRR	20	RREEDDD	20	DRRRDDD	20	DRRDNNN	20	RREEEDD	20	DRREELE	20	DRRRNNN	20	RREEEDD	20	RRREEDD	20	DRRREEE	20	LLLRREE	20	EDDLRRR	20	DDDDDRR	20
NNRRDDD	20	DRRLLRR	20	LLLLRRR	20	DDDDRRR	20	RDDDDRR	20	DDLLLRR	20	REEEDRR	20	DDLLLRR	20	NNRRDDD	20	DRRRRRR	20	EEDRRRR	20	DLERRDD	20	RRRDNNN	20
LLNNRRR	20	EELLRRR	20	LDDRRRR	20	RRREERR	20	DDDDDRR	20	DDDLLRR	20	REERRDD	20	DDDDRRR	20	DDDDDRR	20	DLLLRRR	20	LLRRRDD	20	LRRRRRR	20	NNRRDDL	20
DDDDDRR	20	DDDDRRR	20	EDDRRLE	20	LLRRREE	20	LLLLRRR	20	LDDDDRR	20	DDDDDRR	20	RDDDDLD	20	RRRRRRR	20	RDDDDRR	20	DDDDDRR	20	RREEEDE	20	LRRDDRR	20
RRRDDRR	20	DDDDDRR	20	RLEDNNN	20	RRNNRRR	20	RREEELE	20	ERRRRRR	20	LDDDLRR	20	RLDDDRR	20	DDDRRRR	20	RDDDDRR	20	EDDLDRR	20	LLRRDLE	20	LLDRRDD	20
RDDRRDD	20	LLLRRRR	20	RLEDDRR	20	RDDDDRR	20	RRDDRRR	20	RLDDDRR	20	EDNNRRR	20	DLEEERR	20	REEDDRR	20	LLNNRRR	20	LLRRNNN	20	RRRRRRR	20	RLDELLD	20
DDERRRR	20	RLDDDDE	20	RRNNRRR	20	LLRRDDD	20	LLRRRRR	20	ELEDLRR	20	RRDDLLD	20	LRRRRRR	20	DDDDRRR	20	LDDDRRR	20	RDLLEDD	20	RRNNRRR	20	ELERRRR	20

3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322
3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322
3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322
1111111	1111111	1111111	1111111	1111111	1111111	1111111	1111111	1111111	1111111	1111111	1111111	1111111

Continue Week 27 to 39 of 52 weeks schedule

MTWTFSS	cost	MTWTFSS	cost	MTWTFSS	cost	MTWTFSS	cost	MTWTFSS	cost	MTWTFSS	cost	MTWTFSS	cost	MTWTFSS	cost	MTWTFSS	cost	MTWTFSS	cost	MTWTFSS	cost	MTWTFSS	cost	MTWTFSS	cost
EERRREE	20	RREERRR	20	RREEELL	20	LRREERR	20	RRREERR	20	EERRNNN	20	RREERRR	20	RREERRR	20	EELLLRR	20	EEERRRR	20	EELLLRR	20	EELLLRR	20	EELLLRR	20
RRRREEL	20	LRREERR	20	EELLLRR	20	RDNNRRR	20	RREERRR	20	EELLRRR	20	RRRRNNN	20	RRREERR	20	EEERRRR	20	EELLLRR	20	EELLRRR	20	REEELLL	20	RRRRRRR	20
EELLLRR	20	RRNNRRR	20	EELLRRR	20	REEELLL	20	RRRDNNN	20	RREEERR	20	REEELLL	20	RREELLL	20	RRNNRRR	20	RRREERR	20	EERRLLL	20	NNRREEL	20	LRRDNNN	20
EELLRRR	20	REEELLL	20	RRRRRRR	20	EERRREE	20	LLLRREE	20	RRREERR	20	EELLLRR	20	RRRRRRR	20	REELLRR	20	REEELLL	20	RREEERR	20	REELLRR	20	EERRRRR	20
DRRRDDD	20	LLLRRRR	20	EELLRRR	20	EELLLRR	20	EELLLRR	20	EELLRRR	20	RRRRRRR	20	EELLRRR	20	LLRREEL	20	LLRRNNN	20	RREERRR	20	RRRRRRR	20	EELLLRR	20
RDDDLLL	20	RRLLLRR	20	NNRREEL	20	LRRRNNN	20	RRRRRRR	20	NNRREEL	20	LRREELL	20	LRRDNNN	20	RRREELL	20	LRRREEE	20	LLLRREE	20	ELLLRRR	20	NNRRRRR	20
RRREERR	20	EELLLRR	20	DDDDDRR	20	RLLRRRR	20	EELLLRR	20	RDLLLRR	20	REELLRR	20	EELLLRR	20	DLLLRRR	20	RREERRR	20	DDDDRRR	20	RRRDNNN	20	RREELLL	20
RREEERR	20	RRRDDRR	20	DDDDDRR	20	DDLLLRR	20	RRNNRRR	20	DDDDDRR	20	DDDDDRR	20	REELLRR	20	DDDDDRR	20	RDDDLLL	20	RRRRRRR	20	DDDDDRR	20	RREEELL	20
DLLLRRR	20	EERREEL	20	LRREERR	20	DDDDDRR	20	EEELLRR	20	DDDDRRR	20	LLLRREE	20	LLLRREE	20	LLLRREE	20	LLNNRRR	20	DDDLLRR	20	LLRREEE	20	LLLRREE	20
DDDDDRR	20	LLRRREE	20	LLNNRRR	20	RREEERR	20	DDDDDRR	20	RREEDLL	20	LRRDDRR	20	DDDDDRR	20	DDDDDRR	20	DLLLRRR	20	RRRDNNN	20	RREEERR	20	DDDDDRR	20
RREERRR	20	DDRRNNN	20	RRRRDDD	20	DLDRRLL	20	NNRREEE	20	RRDDDDD	20	RRDDDRR	20	DDDDDRR	20	RRDDNNN	20	RRDDDDD	20	DRRREEL	20	LRRRRRR	20	RDDDDRR	20
LLRRNNN	20	RRDDDDD	20	RRRDLDD	20	NNRREEE	20	DDDRRLL	20	LRRRLDD	20	DDRRREE	20	LLRREEE	20	LDRRRRR	20	RDDDDRR	20	RDDDDRR	20	LLNNRRR	20	RDNNRRR	20
LDDRRDD	20	NNRRRDD	20	DDDRREE	20	ELDDRRR	20	RDDDDRR	20	LLNNRRR	20	DDNNRRR	20	RDNNRRR	20	RRRRDDD	20	DRRREEE	20	LLRRDDD	20	DDDRRDD	20	DRREEED	20
NNRRDLE	20	DDDRRLE	20	LRRRLLE	20	LRRDDDD	20	LLRRDDL	20	LLRRRLE	20	NNRREDD	20	NNRRDLL	20	NNRRRDD	20	NNRRRRR	20	LLNNRRR	20	DDDDRRR	20	RLDLDDD	20
RRNNRRR	20	DDDDDRR	20	RLERRRR	20	RRELRRR	20	DLERRDD	20	DLERRRR	20	ELDLERR	20	DLDRRDD	20	ERREELE	20	DRRLDDD	20	RREEEDD	20	RRLEDDD	20	DRRREDE	20
LLEDLRR	20	ELELERR	20	RLEENNN	20	RRRRDDD	20	LRREELD	20	RRRRLEE	20	ELLRRDD	20	ERRREDD	20	RREELRR	20	EDLRRRR	20	NNRRDLE	20	ERRRDLE	20	LLERRRR	20

3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322
3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322
3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322
1111111	1111111	1111111	1111111	1111111	1111111	1111111	1111111	1111111	1111111	1111111	1111111	1111111

Continue Week 40 to 52 of 52 weeks schedule

MTWTFSS	cost	MTWTFSS	cost	MTWTFSS	cost	MTWTFSS	cost	MTWTFSS	cost	MTWTFSS	cost	MTWTFSS	cost	MTWTFSS	cost	MTWTFSS	cost	MTWTFSS	cost	MTWTFSS	cost	MTWTFSS	cost	MTWTFSS	cost	MTWTFSS	cost
RRREERR	20	EELLLRR	20	RRREERR	20	EELLRRR	20	EERRREE	20	LLRRRRR	20	RRNNRRR	20	EEERREE	20	RRLLNNN	20	RRRREEE	20	RRRREEE	20	RRLLNNN	20	RRRRRRR	20	REE----	20
REERREE	20	LLRREEE	20	RRNNRRR	20	EELLRRR	20	RRRREEE	20	RRREERR	20	NNRRRRR	20	EELLRRR	20	EERREEE	20	LLLRREE	20	RRRREEE	20	LLLRRRR	20	RREELLL	20	RRR----	20
RREEERR	20	RRREELL	20	RREERRR	20	LLNNRRR	20	REERRRR	20	EELLLRR	20	EERREEL	20	LRREELL	20	LRRREEL	20	LLRRRRR	20	EELLLRR	20	EERREEE	20	RREERRR	20	NNR----	20
EERREEE	20	RREEERR	20	EERREEE	20	RREEERR	20	REELLRR	20	NNRREEL	20	LLRRNNN	20	RREEERR	20	RREEERR	20	EELLLRR	20	NNRRRRR	20	RREEERR	20	EELLRRR	20	EEL----	20
EELLRRR	20	EELLLRR	20	EELLLRR	20	EEERREE	20	LLNNRRR	20	EEERREE	20	ERRREEL	20	LRRREEE	20	RRRLLRR	20	EEERRDD	20	DLLLRRR	20	EELLLRR	20	RRREERR	20	EER----	20
LLLRRDD	20	LLLRREE	20	LLRREEL	20	LLRREEL	20	LRRRDDD	20	LLLRRRR	20	EELLLRR	20	RRRRRRR	20	EELLLRR	20	EELLLRR	20	REERRLL	20	RREEERR	20	LLNNRRR	20	DDL----	20
RREEDLL	20	LRRRLLL	20	NNRRDDL	20	LLRRNNN	20	RREERRR	20	EEERRDD	20	LLLRRDD	20	DLLLRRR	20	EEEDRRR	20	RREEELL	20	LRRDNNN	20	RRRDDLL	20	LRRREEL	20	LLR----	20
LRRRDDD	20	DDRRNNN	20	RREEDDD	20	DRRRDLL	20	LRREERR	20	RRRRDLL	20	LRRDDDD	20	NNRRDDL	20	LLRRRDL	20	LLNNRRR	20	EELLLRR	20	EEEDRRR	20	REEDDDD	20	RRE----	20
LLNNRRR	20	RRNNRRR	20	EELLLRR	20	RREERRR	20	DLLLRRR	20	RDDDDDD	20	RREEELE	20	DRRRNNN	20	RRRRDDD	20	DRREEDL	20	LLRRDDL	20	LLRRLLL	20	NNRREEE	20	RRE----	50
DLLLRRR	20	EEERRRR	20	DDLLLRR	20	RDDDDDD	20	RRLLNNN	20	RREERRR	20	DDRRRRR	20	EELLRRR	20	DDDDDRR	20	RRRRNNN	20	RREEERR	20	RRNNRRR	20	EEDRRRR	20	DDD----	50
RDDLLRR	20	RDDDDRR	20	DDDDDRR	20	RRREEDD	20	DDRRLLL	20	RRLLLRR	20	REELLRR	20	RDDDDRR	20	DDDDRRR	20	DDDLLRR	20	RRRDDDD	20	DRRDDDD	20	LLRRDDD	20	DRR----	50
DDDDDRR	20	DDDDDRR	20	LLRRNNN	20	RRLLLRR	20	DDDDRRR	20	RDDDDLE	20	RREERRR	20	RDDDDDD	20	RREERRR	20	RDDDRRR	20	DDDDDRR	20	NNRRDDD	20	DDRRDLE	20	LRR----	50
DRRDLRR	20	DLDDRRR	20	DDDDRRR	20	RDDDDRR	20	RRDDDLL	20	LRRDNNN	20	RRDDDLE	20	DRRRLLD	20	NNRRDLD	20	RRRDDRR	20	DDDELRR	20	LLRRRRR	20	DDLLLRR	20	EDD----	55
RRDDLRR	20	RREERRR	20	LLDRRLD	20	NNRRLLE	20	NNRRDDD	20	DRREERR	20	DLLLRRR	20	RDNNRRR	20	LLLRRRR	20	DDDDDRR	20	EDDRRLD	20	DDDRREE	20	RRLLNNN	20	RRL----	55
EDRRNNN	20	RRRLDDD	20	RRRRRLE	20	DRRRRRR	20	EDDDLRR	20	DDDLLRR	20	DDDDDRR	20	RLDDLRR	20	DDNNRRR	20	NNRRRRR	20	LLNNRRR	20	DDDLLRR	20	DDDDRRR	20	RLD----	55
NNRRRLL	20	NNRRRDD	20	RREDRRR	20	DDDDLRR	20	ELLEERR	20	DLNNRRR	20	RDDELRR	20	LLEELRR	20	RLDELLE	20	RREEDLD	20	RREERRR	20	RDDERRR	20	ELDDLRR	20	LLN----	55

3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3330000
3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3330000
3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3333322	3330000
1111111	1111111	1111111	1111111	1111111	1111111	1111111	1111111	1111111	1111111	1111111	1111111	1111111	1110000

Appendix C: Cost-Effectiveness in Nurse Scheduling

This chapter analyses and discusses the trade-off in the context of the NSP between the flexibility afforded by greater numbers of staff and the implied cost of employing extra staff. If the number of staff is constant, the method used in this study allows quantification of the degree of pressure put on the staff resulting from schedules that do not satisfy their preferences for shift allocation. Moreover, results from real-world data problem sets show that domain transformation facilitates the computation of feasible schedules in a relatively short time, with non-critical constraints being satisfied to a large degree. The resulting solutions facilitated conducting cost–benefit analysis of different staffing levels.

6.1 Introduction

The NSP involves assigning appropriate and efficient work regimes for nurses in either private and government hospitals. According to Henderson (2006, page 26): *the unique function of the nurse is to assist the individual, sick or well, in the performance of those activities contributing to the heath or its recovery that he would perform unaided if he had the necessary strength, will or knowledge.*

For nurses to perform their job well, they need to be organised through effective nurse scheduling. Nurse scheduling is often done manually; however, this takes a great deal of time and seldom generates the best quality results (Bouarab *et al.*, 2010). Several requirements must be taken into account in nurse scheduling, including the minimal allocation of the ward, legal regulations and nurses' personal needs (Abdennadher and Schlenker, 1999). A common problem in healthcare systems worldwide is the shortage of nursing staff (Ulrich, Wallen, Grady, *et al.* 2002). Nursing managers continue to struggle with high turnover levels (Wright, 2013). This can be partly attributed to the demanding nature of the nursing profession, which requires nurses to be available for shift work. In this context, producing work schedules that satisfy not only the clinical cover demands but also the specific requests and preferences of nurses becomes a key to staff satisfaction and retention. Failing to deal with this issue would inevitably lead to a lack of skilled nurses in clinical settings, resulting in a significant negative impact on patient outcomes, including mortality (Aiken *et al.*, 2002).

From the perspective of hospital management, there is an inseparable link between the scheduling activity and the decision about the total number of staff employed. In their publications, Costa (1996) and Knauth (1996) provided some guidelines on this issue. Hospitals prefer to avoid the expense of employing more nurses than is needed to meet required clinical care standards. Hospitals can be aided in striking this delicate balance by the use of computationally efficient software tools capable of constructing work schedules in both a long and medium-term planning mode, as well as in response to immediate staff requests. Independent studies have supported the view that investment in the advanced scheduling of nursing staff translates into the significant enhancement of job satisfaction, as well as savings in labour costs due to reduced nurse turnover (Bester, Nieuwoudt and Vuuren,

2007; Blythe *et al.*, 2005). However, producing schedules that meet hospital requirements and satisfy individual preferences and immediate requests is an extremely complex task (Gino, Mobasherand Murray, 2012).

Hospitals are constantly looking to optimise the cost of their nurses. To this end, it is crucial to ensure the available nurses match the expected demand for the workload. Automation of nurse scheduling is one aspect of optimising the nurse cost. In this chapter, we present an alternative way of tackling a large, real-world NSP. We have approached the problem of cost- effective nurse scheduling using the domain transformation method introduced in Baskaran *et al.* (2009, 2012) as a practical illustration of the information granulation methodology (Bargiela and Pedrycz, 2002, 2008) to generate multiple feasible low-cost schedules, which are subsequently evaluated. In this approach, the hospital is supplied with detailed information about the schedule, which they can use to make their selection objectively. Based on this solution, this chapter also investigates the optimum balance between the staffing levels of a ward and the ability to achieve good-quality schedules. Without the loss of generality, we consider a nurse-scheduling scenario based on the operation of an intensive care unit in a Dutch hospital (Baskaran *et al.,* 2009, 2012). To appreciate the computational complexity of the scheduling problem, we consider a specific case of a ward with 16 nurses employed on 36hours/week contracts, with a scheduling period of five week (35 days).

6.2 Methodology

The domain transformation approach introduced in Baskaran *et al.* (2012) departs from the orthodoxy of direct exploration of the space of schedules, as described in Chapter 4, section 4.3. Domain transformation is an approach to solving complex problems that relies on well-justified simplification of the original problem. We subdivided the problem into smaller sub-problems in a systematic way that remained capable of reproducing the result. Another benefit of this model is that domain transformation can reduce computational complexity and therefore computational time. It also reduces the need for cross-referencing over the detailed swapping of shifts for individual nurses. Through this model, the schedule obtained will not make any difference in terms of the order of processing. The schedule is the same when we change the order of individual patterns or nurses. This approach is able to offer solutions easily by avoiding random searching. By contrast, some other methods have failed to reproduce results, have performed inconsistently or have demonstrated good characteristic with some datasets but not others. Previous state-of-the-art methods do not use information granulation and have thus involved much cross-referencing and checking of data.

6.3 Balancing the Cost of Soft Constraints and the Cost of Staff

Nurse scheduling is inextricably linked with determining how many nurses should be employed. Most healthcare systems are under pressure to control costs while trying to provide high levels of service. This is a difficult balance to strike. Having a small number of nurses may affect quality of care, while employing a large number of nurses and not utilising their contractual hours is wasteful. In the approach presented in this chapter, the aim is to balance these concerns by combining the cost for the under-utilisation of nurses with the costs of violation of the soft constraints into a single performance index. The expectation is that with the decreasing number of nurses, we will find a progressively higher cost of violation of constraints up to the point that hard constraints would have to be broken, rendering solution infeasible. Conversely, with the increasing number of nurses, we expect that, although it will become easier to find schedules that do not violate soft constraints (i.e., one may find low- or zero-cost schedules), the pro-rata cost of the unused contractual hours of extra nurses will dominate the balance.

For the sake of clarity, we demonstrate our balancing approach only in the context of full-time nurses employed on 36 hours/week contracts. The under-utilisation of nurses (U) is measured as:

$$\mathbf{U=TC-TW} \qquad \mathbf{(6.1)}$$

Where TC is the total number of contractual hours per week and Tw is the total number of hours worked by all nurses in one week (as defined by the shift-cover requirement).

6.3.1 Process of checking the demands

It is important to appreciate that the number of hours defined by the shift-cover requirement (TW) does not determine, on its own, the required staff numbers. A simple division of TW by the number of hours per week stipulated by the nurses' employment contract provides only a lower bound on the number of required staff; it does not provide a good estimate of the actual staffing requirement. This is because the varying hard and soft constraints may imply the need for extra staff, despite identical shift-covers and nurse contracts.

For balancing the nurse schedules and staff numbers, we consider only positive values of U in equation (6.1). This is because negative numbers represent the requirement that the nurse works longer hours than stipulated in his/her employment contract, which is already penalised through the hard and soft constraints. The comparison of the cost associated with violation of constraints and the monetary cost of employing extra staff requires the adoption of some convention that would make these costs comparable. We assume that the following represents well the notional cost of under- employment of staff:

CU=U * 10 (6.2)

Where CU is cost of under-utilisation.

6.4 Numerical Results

The numerical experiments described in this section provide a representative sample of the experiments conducted to balance the degree of satisfaction of soft constraints and the decisions on employing additional nursing staff. We have varied the required cover on individual shifts to simulate the decision support functionality that may be required by the hospital management. To understand the behaviour of our model, multiple demand versus number of nurses scenarios were generated. For each scenario, the solution time is calculated in seconds.

6.4.1 Test data on original demands

Based on the original problem, we performed some sample runs for different numbers of nurses (Cases 1–3). Table 6.1 presents the results for the best set of nurses satisfying the demand of the original problem with a reasonable cost for the month. Tables 6.2 and 6.3 show the alternative demand scenarios, and Graphs 6.1–6.3 illustrate the costs for each case.

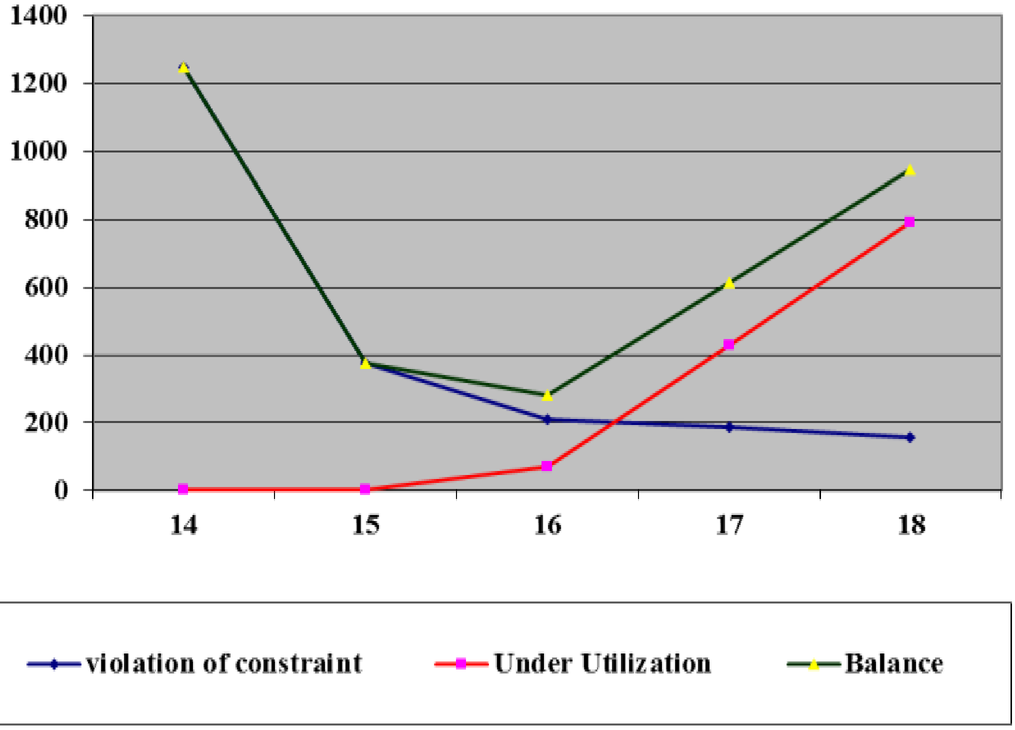
6.4.1.1 Case 1

Case 1 used '9999966' **D**-shift and the '1111111' N shift cover (TW=57*9+7*8=569 hours).

Table 6.1. Balance of Violation of Soft Constraints and the Under-Utilisation of Nurses for Case 1

TN	TC	U(h/w)	CSC	CU	T(s)
14	504	0	1250	0	706
15	540	0	375	0	503
16	576	7	210	70	139
17	612	43	185	430	31
18	648	79	180	790	22

Note: TN=Total number of nurses, TC=Total number of contractual hours, TW=Total number of hours works (in hours), U(h/w)=Under-utilisation of nurses (hours/week), CSC=Cost of violating soft constraint, CU=Cost of under-utilisation, T(s)=Time (in seconds) to execute the software.



Graph 6.1. Balance of constraint costs and cost of under-utilisation for Case 1.

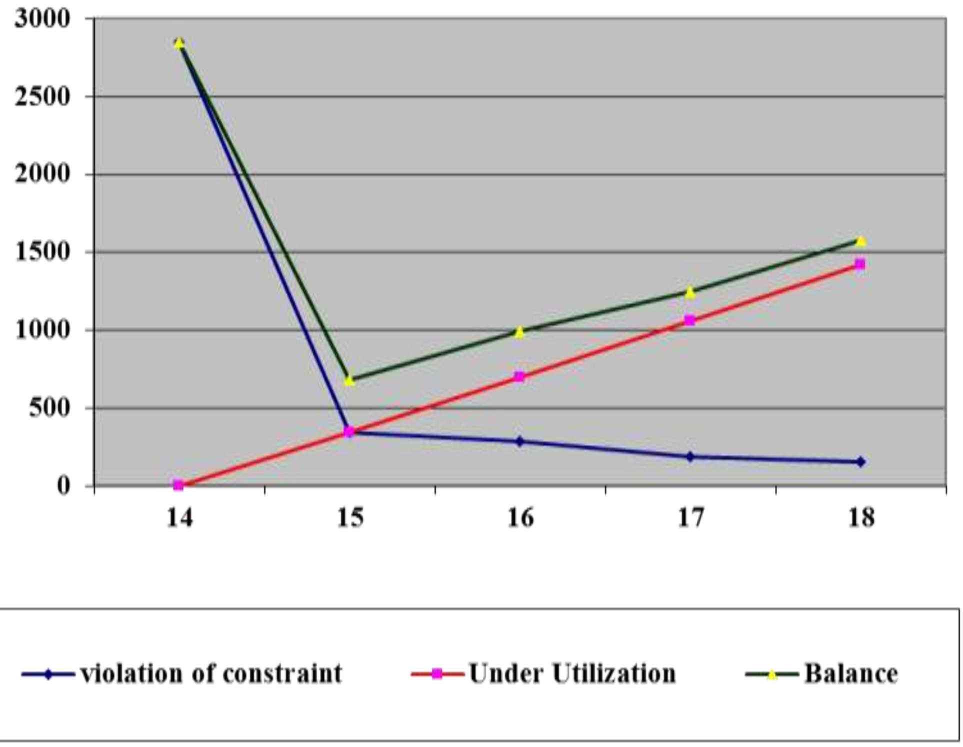
Case 2

Case 2 used ‘8888855’ **D**-shift and the ‘1111111’ N shift cover (TW=50*9+7*8=506 hours).

Table 6.2. Balance of Violation of Soft Constraints and the Under-Utilisation of Nurses for Case 2

TN	TC	U(h/w)	CSC	CU	T(s)
14	504	0	2854	0	761
15	540	34	345	340	501
16	576	70	290	700	204
17	612	106	190	1060	71
18	648	142	160	1420	73

Note: TN=Total number of nurses, TC=Total number of contractual hours, TW=Total number of hours works (in hours), U(h/w)=Under-utilisation of nurses (hours/week), CSC=Cost of violating soft constraint, CU=Cost of under-utilisation, T(s)=Time (in seconds) to execute the software.



Graph 6.2. Balance of constraint costs and cost of under-utilisation for Case 2.

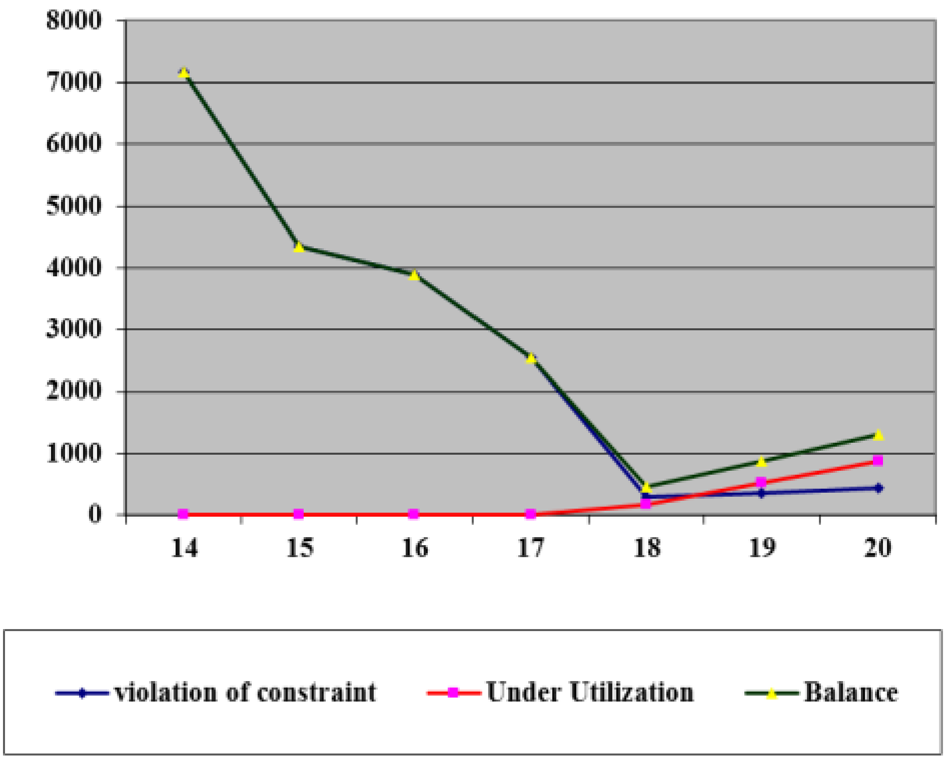
Case 3

Case 3 used ‘101010101077’**D**-shift and the ‘1111111’ N shift cover (TW=64*9+7*8=632 hours).

Table 6.3. Balance of Violation of Soft Constraints and the Under-Utilisation of Nurses for Case 3

TN	TC	U(h/w)	CSC	CU	T(s)
14	504	0	7175	0	863
15	540	0	4350	0	809
16	576	0	3900	0	790
17	612	0	2550	0	779
18	648	16	300	160	504
19	684	52	355	520	515
20	720	88	430	880	641

Note: TN=Total number of nurses, TC=Total number of contractual hours, TW=Total number of hours works (in hours), U(h/w)=Under-utilisation of nurses (hours/week), CSC=Cost of violating soft constraint, CU=Cost of under-utilisation, T(s)=Time (in seconds) to execute the software.



Graph 6.3. Balance of constraint cost and cost of under-utilisation for Case 3.

6.5 Discussion

This chapter has presented a combined investigation of nurse scheduling and staffing level decisions. The findings quantify how the constraints associated with the scheduling problem influence the cost- effectiveness of employing additional staff. Under-utilisation is undesirable, as the cost of nurses increased with their idleness. The investigation was conducted using a representative set of three scenarios, with total number of contractual hours per week and total number of hours worked by all nurses.

The results indicate that for the original problem demand of ‘999966’ **D**-shift, 16 nurses are required (see Graph 6.1). With fewer than 16 nurses, the clinical cover requirement could not be satisfied, while larger numbers of nurses resulted in an unnecessarily high employment cost. The ideal numbers of nurses for the alternative cover‘8888855’was15 nurses (see Graph 6.2), while 18 nurses were required for cover ‘101010101077’ (see Graph 6.3). It was also found that the number of contractual hours was equivalent to two-thirds of the possible number of hours the nurses could work. This finding may ensure that fewer nurses are under-utilised. Similarly, as found here, the cost is higher for over-utilised nurses. A balance is thus important. The result presented here can help hospitals in addressing any instant issues implied.

6.6 Conclusion

The NSP considered at the level of detailed time constraints and different types of dayshifts represents a very significant computational challenge. In this chapter, we proposed an unusual set of demands such as ‘8888855’ or ‘101010101077’ for use in hospitals by using domain transformation. Investigation was also undertaken of the efficiency of identifying feasible schedules for varying combinations of cover demand and nurse availability. Nurse scheduling is a difficult, time-consuming managerial problem and there are many types of NSP. Automating the solution of the NSP can reduce the effort and time required for scheduling. It can also increase nurses’ satisfaction and long-term retention. Many constraints can affect total labour cost. Applying some ‘tactical’ scheduling analysis can ensure the satisfaction of all constraints and give a rapid valuation of schedules.

The results suggest that hospital management can significantly reduce annual nursing labour costs by setting less through demand requirements. The calculation of the number of nurses required per shift is also important in solving the NSP; this can overcome the problems of over-staffing (i.e., increased labour costs) and under-staffing (i.e., reduced quality of care or service). Further, using a non-optimal demand can consume costly managerial time and effort. Determining optimal demand is also important to avoid downgrading. In the case of shortages, schedulers may consider downgrading, whereby higher skilled nurses are assigned to tasks that lower skilled nurses are capable of performing. However, as the reverse is not possible, determining the best demand with the best set of nurses is important to ensure the best possible utilisation of all nurses' available work hours.

Appendix D: Investigator's Agreement with Hospital Kajang

Versi 2.0 Tarikh: 15 Feb

INVESTIGATOR'S AGREEMENT, HEAD OF DEPARTMENT'S AND INSTITUTIONAL APPROVAL

PERSETUJUAN PENYELIDIK, PENGESAHAN KETUA JABATAN DAN INSTITUSI

This document is intended for online submission for purpose of formal research review and approval. It is to be used in lieu of other equivalent manually printed document such as Borang JTP/KKM 1-2 and Borang JTP/KKM 3. After completing the form below and obtaining the required signatures, please scan this document and submit online.

Dokumen ini adalah untuk penghantaraan atas talian (online) mengikut prosedur rasmi semakan dan persetujuan penyelidikkan. Borang ini dikeluarkan sebagai gantian dokumen kebenaran manual yang serupa seperti Borang JTP/KKM 1-2 dan Borang JTP/KKM 3. Selepas melengkapkan borang di bawah dan mendapatkan tanda tangan yang diperlukan, sila imbaskan dokumen ini dan hantar atas talian.

Unique Research ID : *[Nombor Pendaftaran]*	14765
Research Title : *[Tajuk]*	Domain Transformation Approach to Solve Nurse Rostering Problem at Hospital Kajang
Protocol Number if available : *[Nombor Protokol jika ada]*	

Investigator agreement ***[Persetujuan penyelidik]***

I have understood the above titled proposed research and I agree to participate in the research as an investigator.

Saya faham cadangan penyelidikan yang bertajuk di atas dan saya bersetuju mengambil bahagian dalam projek tersebut sebagai penyelidik.

Name of Investigator : *[Nama Penyelidik]*	GEETHA BASKARAN
IC number : *[Nombor KP]*	790420045440
Site Institution : *[Institusi]*	Kajang Hospital
Signature & Official stamp : *[Tandatangan dan Cop Rasmi]*	FACULTY OF SCIENCE THE UNIVERSITY OF NOTTINGHAM MALAYSIA CAMPUS (Company No. 473520-K) JALAN BROGA, 43500 SEMENYIH SELANGOR DARUL EHSAN, MALAYSIA. TEL: 03-8924 8000 FAX: 03-8924 8018
Date : *[Tarikh]*	

Head of Department Agreement ***[Persetujuan Ketua Jabatan]***

I agree to allow the above named investigator to conduct or to participate in the above titled research.

Saya membenarkan pegawai yang bernama di atas untuk menjadi penyelidik dalam projek penyelidikan tersebut di atas.

Name of Head : *[Nama Ketua]*	ZALIHA BT KASIM KETUA PENYELIA JURURAWAT U42 HOSPITAL KAJANG
Name of Department and Institution *[Jabatan dan Institusi]*	
Signature & Official stamp : *[Tandatangan dan Cop Rasmi]*	
Date : *[Tarikh]*	

Institutional approval ***[Pengesahan Institusi]***

This section maybe omitted if one of the NIH institute is authorized to approve on behalf of institution. Refer NIH for details.

[Bahagian ini tidak perlu jika salah satu daripada institusi NIH diberi kuasa pengesahan bagi pihak institusi tersebut. Rujuk NIH untuk maklumat lanjut]

I agree to allow the investigator(s) named above to conduct or to participate in the above titled research. Where applicable, I further agree to allow my institution to be one of the sites participating in the research.

Saya membenarkan pegawai yang bernama di atas menjalankan penyelidikan selaku penyelidik dalam projek penyelidikan tersebut. Jika berkenaan, saya juga membenarkan institusi ini mengambil bahagian dalam projek tersebut.

Name of Director : *[Nama Pengarah]*	DR. KULDIP KAUR A/P PREM SINGH
Name of Institution *[Institusi]*	No. Pendaftaran Penuh MPM : [illegible] Pengarah Hospital Hospital Kajang
Signature & Official stamp : *[Tandatangan dan Cop Rasmi]*	
Date : *[Tarikh]*	23/1/2013

23/1/13

ZALIHA BT KASIM
KETUA PENYELIA JURURAWAT U42
HOSPITAL KAJANG

Appendix E: Cover Letter

Cover Letter

January 23, 2013
National Institute of Health (NIH)
Ministry of Health Malaysia
Malaysia Research Ethic Committee
Jalan Rumah Sakit Bangsar,
59000 Kuala Lumpur.

RE: Request for Expedited Review status for "DTA-NRPKH"

Research Title: *Domain Transformation Approach to Solve Nurse Rostering Problem at Kajang Hospital*(Research ID : **14765**)

Honourable Members of the MREC Executive Board:

My name is Geetha Baskaran, IC No. 790420045440 and I am a PhD student, researcher from University of Nottingham Malaysia Campus and a member of Automated Scheduling, Optimisation and Planning Group (ASAP). For my PhD studies and research writings, I am examining on nurse rostering problem.

Nurse Rostering problem (NRP) represents a subclass of scheduling problems and is one of the NP-hard problems that are difficult to solve for optimality. Nurse Rostering Problem (NRP) concerns about producing a high quality workable duty roster for the available staff nurses. The aim of this study is to present a novel approach to solving the nurse rostering problem by simplifying it through information granulation. It deals with assigning shifts to staff nurses' subject to satisfying required workload and other constraints. The constraints are classified into hard constraints (compulsory) and soft constraints (should be satisfied as much as possible). A feasible solution is a solution that satisfies all hard constraints. However, the eminence of the duty roster is considered based on satisfying the soft constraints. . This study is an effort to solve a real world situation from Kajang Hospital.

I am requesting for a quick approval to conduct the research in Kajang Hospital. This will actually help me to complete my PhD studies and also to publish my approach by using our Malaysian hospital nurse rostering problem as a base. Together with this letter I am attaching the

a) Study Proposal
b) Questionnaire
c) Research Agreement

Thank you for your consideration of this request.

Sincerely,

30/1/2013

X

Geetha Baskaran
Coordinating Investigator

Appendix F: Letter of Approval to Conduct Research

JAWATANKUASA ETIKA & PENYELIDIKAN PERUBATAN
(Medical Research & Ethics Committee)
KEMENTERIAN KESIHATAN MALAYSIA
d/a Institut Pengurusan Kesihatan
Jalan Rumah Sakit, Bangsar
59000 Kuala Lumpur

Tel. : 03 2282 9082/03 2282 9085
03 2287 4032/03 2282 0491
Faks : 03 2287 4030

Ruj. Kami : (2) dlm.KKM/NIHSEC/800-2/2/2 Jld2.P13-466
Tarikh : 5 Julai 2013

Geetha Baskaran
Fakulti Sains
The University of Nottingham Malaysia Campus

Puan,

NMRR-13-294-14765
DOMAIN TRANSFORMATION APPROACH TO SOLVE NURSE ROSTERING PROBLEM AT KAJANG HOSPITAL

Lokasi Projek : Hospital Kajang

Dengan hormatnya perkara di atas adalah dirujuk.

2. Jawatankuasa Etika & Penyelidikan Perubatan (JEPP), Kementerian Kesihatan Malaysia (KKM) mengambil maklum bahawa projek tersebut adalah untuk memenuhi keperluan akademik Program PhD Sains Kesihatan, *The University of Nottingham Malaysia Campus*.

3. Sehubungan dengan ini, dimaklumkan bahawa pihak JEPP KKM tiada halangan, dari segi etika, ke atas pelaksanaan projek tersebut. JEPP mengambil maklum bahawa kajian ini tidak melibatkan sebarang intervensi dan hanya menggunakan borang soalselidik sahaja untuk mengumpul data kajian. Segala rekod dan data adalah SULIT dan hanya digunakan untuk tujuan kajian dan semua isu serta prosedur mengenai *data confidentiality* mesti dipatuhi. Kebenaran daripada Pengarah Hospital di mana kajian akan dijalankan mesti diperolehi terlebih dahulu sebelum kajian dijalankan. Puan perlu akur dan mematuhi keputusan tersebut.

4. Adalah dimaklumkan bahawa kelulusan ini adalah sah sehingga 5 Julai 2014. Puan perlu menghantar *'Continuing Review Form'* (Lampiran 1) selewat-lewatnya 2 bulan sebelum tamat tempoh kelulusan ini bagi memperbaharui kelulusan etika. Pihak Puan juga perlu mengemukakan laporan tamat kajian dan juga laporan mengenai "*All adverse events, both serious and unexpected*" kepada Jawatankuasa Etika & Penyelidikan Perubatan, KKM.

Sekian terima kasih.

BERKHIDMAT UNTUK NEGARA

Saya yang menurut perintah,

(DATO' DR CHANG KIAN MENG)
Pengerusi
Jawatankuasa Etika & Penyelidikan Perubatan
Kementerian Kesihatan Malaysia

Appendix G: Hospital Kajang Sample Schedule

JADUAL TUGAS JURURAWAT U29 WAD 2 CCU JABATAN PERUBATAN
HOSPITAL KAJANG 2012

BIL	JADUAL DARI	12/11/2012 HINGGA 25/11/2012													
	HARI	ISNIN	SELASA	RABU	KHAMIS	JUM	SABTU	AHAD	ISNIN	SELASA	RABU	KHAMIS	JUM	SABTU	AHAD
	TARIKH	12-Nov	13-Nov	14-Nov	15-Nov	16-Nov	17-Nov	18-Nov	19-Nov	20-Nov	21-Nov	22-Nov	23-Nov	24-Nov	25-Nov
	JURURAWAT U29 CCU														
1	JT SITI ZUBAIDAH MAT HUSSAIN	PG 5	PG 5	HK	PT 1	PG 1	PG 1	PT 5	PG 1	PG 5	HK	PT 1	PT 1	PG 1	PG 1
2	JT MASLINA NGADEMIN	CR	[illegible]	WP/CRP	[illegible]	WP/CRP	HKS	HK	CR	WP/CRP				HKS	HK
3	JT ZANARIAH MD ISA	HK	[illegible]	HK	[illegible]	PT 1	PT 1	PT 1	PG 5	PG	PG 1	PG 1	HK	PT 1	PT 1
4	JT ROSHAYATI ISMAIL	[illegible]	KM	KM	[illegible]	[illegible]	[illegible]	HK	PT 1	PT 1	PT 1	PT 1	PG 1	PG 1	HK
5	JT AZLINA ABDULLAH	PG 1	[illegible]	[illegible]	[illegible]	KM	KM	HK	PT 1	PT 5	PG 1	HK	PT 1	PT 1	PG 1
6	JT ROSNANI A. LATIP	PT 2	PT 1	PT 1	HK	[illegible]	[illegible]	[illegible]	KM	KM	[illegible]	[illegible]	CR	CR	HK
7	JTSHARINA MOHD REDZUAN	PT 5	PT 2	PT 1	HK	PT 1	PT 5	PG 1	[illegible]	[illegible]	[illegible]	KM	KM	[illegible]	HK
8	JT JUNAINAH OMAR	PG 1	PG 1	PG 1	HK	PG 1	PG 5	PG 5	PG 1	PG 1	HK	[illegible]	[illegible]	KM	KM
9	JT MUHABSAH MUKHTARUDDIN	PT 1	PT 5	PG 1	PG 1	[illegible]	[illegible]	[illegible]	[illegible]	PT 1	PT 1	PG 1	PG 1	HK	[illegible]
	JUMLAH	3+3	1+4	2+2	2+1	2+2	2+2	2+2	3+2	3+3	2+2	2+2	2+2	2+2	3+1

ASSIGNMENT INCHARGE	KENYATAAN
1- INCHARGE CCU	~~09/11/12@2.30PM- WARD MEETING~~ [illegible]
2- I/C CUBICLE 1 + 2	
3- I/C CUBICLE 3	
4- I/C CUBICLE 4 +5	
5- I/C UBAT	
6- RUNNER	

DISEDIAKAN:
[illegible] 9/11/12
NOORSIAH BINTI HARUN
KETUA JURURAWAT U32
HOSPITAL KAJANG

DISEMAK:

JADUAL TUGAS JURURAWAT U29 WAD 2 PERUBATAN LELAKI
HOSPITAL KAJANG 2012

BIL	JADUAL DARI	12/11/2012 HINGGA 25/11/2012													
	HARI	ISNIN	SELASA	RABU	KHAMIS	JUM	SABTU	AHAD	ISNIN	SELASA	RABU	KHAMIS	JUM	SABTU	AHAD
	TARIKH	12-Nov	13-Nov	14-Nov	15-Nov	16-Nov	17-Nov	18-Nov	19-Nov	20-Nov	21-Nov	22-Nov	23-Nov	24-Nov	25-Nov
1	KJ KUP NORZILA MINHAT	PG 2	PG 5	CR	[illegible]	CR	[illegible]	HK	[illegible]	[illegible]	[illegible]	HT	KM	HK	PG 5
	JURURAWAT U29 PERUBATAN														
1	JT NOREMY SAHIZA SHAARI	PG 3	[illegible]	HK	[illegible]	[illegible]	[illegible]	[illegible]	KM	KM	HK	[illegible]	CR	PT 2	PT 2
2	JT ROZITA MOHD SHAWAL	HK	PT 3	PG 4	[illegible]	[illegible]	[illegible]	[illegible]	KM	KM	[illegible]	[illegible]	CR	CR	HK
3	JT NUR FATIN MOHD YUSOFF	[illegible]	[illegible]	KMO1/11	HK	[illegible]	[illegible]	[illegible]	KM	KM	HK	PT 4	PT 3	PT 5	PG 3
4	JT ROBIYATI ABD GHANI	PT 3	[illegible]	PG 5	PT 5	PG 3	[illegible]	HT	[illegible]	[illegible]	[illegible]	KM	KM	HK	[illegible]
5	JT INTAN DIANA MOHD ALI	HK	[illegible]	PT 5	PT 3	PT 3	PG 2	PG 2	[illegible]	[illegible]	[illegible]	KM	KM	HK	PT 4
6	JTNORULASHIKIN SHAHARUDDIN	HK	[illegible]	PT 3	PT 1	PT 4	[illegible]	PG 4	HK	PG 3	PG 5	[illegible]	[illegible]	[illegible]	KM
7	JT JUNAIDAH DAUD	HK	PG 4	PG 2	PG 2	PG 5	[illegible]	[illegible]	PT 2	PT 2	PG 3	[illegible]	[illegible]	[illegible]	KM
8	JT ZALEHA ROSLI	[illegible]	[illegible]	PT 4	PT 4	PT 2	[illegible]	PG 3	PG 2	PG 2	HK	HT	[illegible]	[illegible]	KM
9	JT ROSEZANA AZILA MUSLAN	[illegible]	[illegible]	PT 2	[illegible]	PG 4	[illegible]	[illegible]	PT 4	PT 4	PG 2	PG 4	PG 5	HK	[illegible]
10	JT NOR HAZWANI A. HALIM	[illegible]	KM	KM	HK	[illegible]	[illegible]	[illegible]	PT 3	PG 4	HK	PT 5	PT 4	PG 3	[illegible]
11	JT ROSMALIANA ALWI	[illegible]	KM	KM	HK	[illegible]	[illegible]	[illegible]	[illegible]	HK	PT 2	PG 3	PG 3	PG 4	[illegible]
12	JT NOOR ESLIZA ALI	[illegible]	KM	KM	[illegible]	CR	[illegible]	HK	PT 5	PT 3	PG 4	PG 2	PG 2	HK	PT 1
13	JT SYARIFAH MUNIRAH WAFA	PT 4	[illegible]	[illegible]	[illegible]	KM	[illegible]	HK	[illegible]	[illegible]	HK	CR	CR	PT 2	PT 3
14	JT WAN ZAIDATUL AKMA	HK	[illegible]	[illegible]	[illegible]	KM	[illegible]	PG 5	PG 3	HK	PT 4	PG 5	PG 4	PT 4	PT 5
15	JT ROHAYA MAT ZIN	PG 4	[illegible]	[illegible]	[illegible]	KM	[illegible]	[illegible]	[illegible]	HK	PT 3	PT 3	PT 5	PG 2	PG 2
16	JT NOR ASIDA BADRUDDIN	HK	PT 4	PG 3	PG 5	PG 2	PT 2	PT 4	PG 4	HK	PT 5	PT 2	PT 2	PG 5	PG 4
17	JT NUR AMALINA RAMLI	BERTUGAS DI HOSPITAL SERDANG													
18	JT SHAZWANI ABD GHANI	BERTUGAS DI HOSPITAL SERDANG													
	JUMLAH	3+3	4+2	4+4	5+4	4+3	3+3	3+3	3+4	3+3	4+4	4+4	4+4	4+4	4+5

ASSIGNMENT INCHARGE	KENYATAAN
1- INCHARGE CCU	
2- I/C CUBICLE 1 + 2	
3- I/C CUBICLE 3	
4- I/C CUBICLE 4 +5	
5- I/C UBAT	
6- RUNNER	

DISEDIAKAN:
[illegible] 9/11/12
NOORSIAH BINTI HARUN
KETUA JURURAWAT U32
HOSPITAL KAJANG

DISEMAK:

Appendix H: Initial Questionnaire

The University of Nottingham

UNITED KINGDOM · CHINA · MALAYSIA

GEETHA BASKARAN

The Nottingham University of Malaysia Campus
Faculty of Science, JalanBroga, 45300 Semenyih, Kajang
Tel: 603 8924 8129
E-mail: Geetha.Baskaran@nottingham.edu.my

NURSE SCHEDULING INTERVIEW QUESTIONNAIRE

Q1: What wards are you in charge on?

Q2: How do you generate schedules for the wards that you are in charge?

Q3: How do you deal with ad- hoc request for any changes in the schedule?

Q4: Would it be useful to be able to generate schedules more quickly?

Q5: What is the schedule horizon (fixed one week or moving windows)?

Q6: What are the shifts that you practice in this hospital?

Q7: How many nurses you need for each shift?

Q8: What are the regulatory constraints involved in your scheduling?

- The number of night shifts
- The number of day shifts
- The number of consecutive shifts
- Can the nurse have more than one shift a day
- The number of rest day
- Maximum length of the shift
- Other constraints

Appendix I: Certificate of Excellent Presentation Award

Appendix J: Certificate of Best Paper Award


IPN Network

IPN.org International Postgraduate Network

BEST PAPER AWARD

ASSISTANT PROFESSOR GEETHA BASKARAN

UNIVERSITY OF NOTTINGHAM MALAYSIA

SEEING THE BIGGER PICTURE: DOMAIN TRANSFORMATION APPROACH TO THE NURSE SCHEDULING PROBLEM

IPN - IWNEST Conferences Jakarta 2014

12-13 December 2014

Jakarta, Indonesia

Prof. Dr. Abdel Rahman Al-Tawaha

President

Islamic World Network for Environmental Science and Technology

Honorary Advisor International Postgraduate Network

Appendix K: Certificate of Keynote Speaker

IIE Int'l Conference
www.iieng.org

Keynote Speaker

This Certificate is awarded to
Asst. Prof. Geetha A/P Baskaran
The University Of Nottingham Malaysia Campus, Malaysia
for Presenting Excellent Keynote Speech
Domain Transformation in Nurse Scheduling
at
International Conference on Artificial Intelligence, Energy and Manufacturing Engineering (ICAEME'2015)

IIE Chair

Jan. 7-8, 2015
Dubai (UAE)

Appendix L: Certificate of Award: Gold Medal

Certificate of Award

Organised By:

MINISTRY OF EDUCATION MALAYSIA

POLITEKNIK MALAYSIA
SULTAN ABDUL HALIM MU'ADZAM SHAH

This is to certify that

GEETHA BASKARAN

Has been awarded the **Gold Medal**

For the invention/innovation ***DOMAIN TRANSFORMATION IN NURSE SCHEDULING PROBLEM USING INTEGER PROGRAMMING***

In **iCompEx'15**

NATIONAL INNOVATION AND INVENTION COMPETITION THROUGH EXHIBITION 2015

Held On ***24, 25 & 26 March 2015***

At ***Politeknik Sultan Abdul Halim Mu'adzam Shah***

FACULTY OF SCIENCE
THE UNIVERSITY OF NOTTINGHAM MALAYSIA CAMPUS
JALAN BROGA, 43500 SEMENYIH
SELANGOR DARUL EHSAN, MALAYSIA.

MEJAR (K) ASMARA BINTI SULONG
Pengarah
Politeknik Sultan Abdul Halim Mu'adzam Shah

LT. KOL (K) DATUK HJ MOHLIS BIN JAAFAR
Ketua Pengarah
Jabatan Pengajian Politeknik

Supported By: