DOKUZ EYLÜL UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

APPLICATIONS OF OPERATIONS RESEARCH TECHNIQUES FOR OPERATIONAL DECISIONS IN HEALTHCARE INDUSTRY

By

Aykut Melih TURHAN

August, 2018

İZMİR

APPLICATIONS OF OPERATIONS RESEARCH TECHNIQUES FOR OPERATIONAL DECISIONS IN HEALTHCARE INDUSTRY

A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of Dokuz Eylül University
In Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy in Industrial Engineering, Industrial Engineering Program

By
Aykut Melih TURHAN

August, 2018
İZMİR

Ph.D. THESIS EXAMINATION RESULT FORM

We have read the thesis entitled "**APPLICATIONS OF OPERATIONS RESEARCH TECHNIQUES FOR OPERATIONAL DECISIONS IN HEALTHCARE INDUSTRY**" completed by **AYKUT MELİH TURHAN** under supervision of **PROF.DR. BİLGE BİLGEN** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Doctor of Philosophy.

B Bilgen

Prof. Dr. Bilge BİLGEN

Supervisor

Prof. Dr. Hasan SELİM

Thesis Committee Member

Prof. Dr. Sabri ERDEM

Thesis Committee Member

Prof. Dr. Aşkıner GÜNGÖR

Examining Committee Member

Prof. Dr. Deniz TÜRSEL ELİİYİ

Examining Committee Member

L. Salum

Prof. Dr. Latif SALUM

Director

Graduate School of Natural and Applied Sciences

ACKNOWLEDGEMENTS

Undertaking this Ph.D. has truly been a life-changing experience and it would not have been possible without the guidance and support from many individuals.

Foremost, I would like to express my sincere appreciation to my supervisor, Prof. Dr. Bilge BİLGEN, who has been a tremendous mentor. Her invaluable academic guidance, patience, support, and motivation throughout the Ph.D. has made this thesis possible. Her professionalism and encouragement has helped me to grow as a better research scientist.

Apart from my supervisor, I also would like to express my gratitude to my thesis committee members: Prof. Dr. Hasan SELİM and Prof. Dr. Sabri ERDEM. Countless fruitful discussions have led to many different research directions and enriched the context of the study. Their academic and industry experiences have been priceless.

I will not forget to send my special thanks to my friends, colleagues, and family members. Having their support always means a lot to me. I am also thankful to all of the respected faculty members in the Department of Industrial Engineering at Dokuz Eylül University. It has truly been a honor to do this Ph.D. at such a highly established academic institution.

Last but not the least; I am very grateful to my parents, Hediye TURHAN and Hüseyin TURHAN, who have been the major catalyzer since the start of my Ph.D. studies. Words cannot express how thankful I am for their sacrifices. I will always remember our trips to Izmir for semester review presentations. Very special thanks go also to my sister, Ayşe TURHAN CÖMERT, for tirelessly listening my challenges and hardship, and for making me believe I can overcome these difficulties. Finally, words are not enough to show how thankful I am for having my daughter, Elif Beyza TURHAN, and my son, Ahmet TURHAN, in my life.

Aykut Melih TURHAN

APPLICATIONS OF OPERATIONS RESEARCH TECHNIQUES FOR OPERATIONAL DECISIONS IN HEALTHCARE INDUSTRY

ABSTRACT

This thesis addresses two common scheduling problems that are encountered in the health care industry, the patient admission scheduling (PAS) problem and nurse rostering problem (NRP). The PAS automatically assigns elective patients to beds for the duration of their stays considering medical needs and preferences. Both static and dynamic versions are studied in this dissertation. For the static version where patient admissions are known in advance, a mixed integer programming (MIP) based heuristics are proposed. The problem is decomposed into a set of smaller problems and iteratively solved. A similar approach is also proposed for the dynamic version which several real life applications such as existence of the emergency patients, operating room constraints, and patient delays are additionally considered. The approach on the PAS generates schedules within fifteen percent gaps from best known solutions in faster times. The DPAS solution reports six new best-known solutions on test data. The last problem in the thesis, NRP, is a complex scheduling problem in which nurses must be assigned to shifts according to a set of constraints. Two variants of the problem are studied. While one of the versions deals with common constraints such as shift requests and cover needs, the other version extends the previous one with skills and departmental assignments. The standard version is solved via a hybrid of MIP-based heuristics and meta-heuristics approaches to provide powerful schedules. A mat-heuristic algorithm is proposed for the extended version. Computational experiments show that the hybrid algorithm obtains seven new best-known results and the mat-heuristic approach reports six new best-known solutions on instances when a stand-alone IP solver is not able to provide schedules.

Keywords: Operations research in health services, scheduling, patient admission scheduling, nurse rostering, mixed integer programming based heuristics, fix-and-relax, fix-and-optimize, simulated annealing, particle swarm optimization, meta-heuristics, mat-heuristics

SAĞLIK ENDÜSTRİSİNDE OPERASYONEL KARARLAR İÇİN YÖNEYLEM ARAŞTIRMASI TEKNİKLERİ

ÖZ

Bu tez çalışmasında sağlık endüstrisinde yaygın olarak görülen iki tip çizelgeleme problemi üzerinde durulmaktadır, hasta kabul çizelgeleme problemi ve hemşire çizelgeleme problemi. İlk problemde yatan hastalar medikal ihtiyaçlarına ve tercihlerine göre tedavi süresince yataklara atanırlar. Bu tezde bu problemin hem statik hem de dinamik versiyonları çalışılır. Hasta kabul zamanlarının bilindiği statik versiyon için karma tamsayılı programlama önerilir. Bu yöntemde, problem daha küçük problemler kümesine ayrıştırılarak tekrarlı bir şekilde çözülür. Acil hastaların, ameliyathane kısıtlarının ve hasta ertelemelerinin de incelendiği problemin dinamik versiyonu için de çok benzer bir ayrıştırma ve çözüm yaklaşımı tercih edilmiştir. Statik versiyonuna uygulanan yöntem bilinen sonuçlara yüzde on beşten az bir farkla çok daha hızlı hesaplama zamanlarında çizelgeler oluşturabilmektedir. Dinamik versiyon çözümü ise altı yeni en iyi sonuç elde etmektedir. Bu tez çalışmasında yer verilen diğer problem ise hemşirelerin belirli kısıtlara göre vardiyalara atandığı kompleks bir çizelgeleme problemi olan hemşire çizelgelemedir. Araştırmalarda bu problemin iki farklı versiyonu çalışılmıştır, standart ve genişletilmiş versiyonlar. Standart versiyon yaygın olarak bilinen vardiya ve hemşire ihtiyaç kısıtlarının değerlendirildiği temel versiyon iken, genişletişmiş versiyon hemşire yeteneklerini ve departman atamalarını da göz önünde bulundurur. Standart versiyonun çözümünde karma tamsayılı programlama ile ileri sezgisel yaklaşımlar melezlenmiştir. Genişletilmiş versiyon için ise matematik sezgisel bir yöntem önerilmiştir. Melez yöntem literatüre göre yedi tane, matematiksel sezgisel ise çözücülerin yetersiz kaldığı örneklerde altı tane en iyi bilinen sonuç elde etmektedir.

Anahtar Kelimeler: Sağlık endüstrisinde yöneylem araştırması, çizelgeleme, hasta kabul çizelgeleme, hemşire çizelgeleme, karma tamsayılı programlama tabanlı sezgiseller, sabitle-ve-gevşet, sabitle-ve-optimize et, benzetimli tavlama, parçacık sürü optimizasyonu, meta-sezgiseller, matematik-sezgiseller

CONTENTS

Page

LIST OF FIGURES

Page

LIST OF TABLES

Page

CHAPTER ONE
INTRODUCTION

1.1 Background and Motivation

Considering the constant increase in costs in health care due to advancements in health care technology, scarce resources such as doctors, nurses, and operating rooms, and increasing demands by patients have always made this field of research a very promising and strong research area for the Operations Research (OR) community. Inherently, the area is not a new research topic for the community. It has been studied since 1950s and many different models and solution techniques have been proposed to address various problems in the field.

Despite the previous research, technological improvements in computer science and the increased capabilities in the computational power in the last several decades make it possible to adapt many more constraints into health care models and experiment on such models that are close to real life situations. Therefore, models introduced recently are very different than original versions. Moreover, emerging solution techniques in Artificial Intelligence (AI), OR, and machine learning would lead to better and superior results.

Generally speaking, decision making in health care is done at three levels; strategic, tactical or operational. Decision making at the strategic and tactical level such as policy making is outside of the scope of this thesis. Rather, this work focusses on the operational level decision making and some of the models addressing this need.

Needless to say, scheduling is at the core of the operational level decision making in a health institution and there are many resources that need to be scheduled to achieve operational efficiency and reduce costs. Scheduling departments produce schedules for doctors, nurses, operating rooms, etc. to effectively control these scarce resources. Although the advancements in computational science as pointed out in the previous paragraphs, these scheduling activities are still done manually by many health care facilities which results inefficiency.

Additionally, scheduling patients and nurses are two major components of this scheduling effort due to the large volume. While patients seek for care in many different departments, nurses are the first line of the health care provision. From emergency departments to intensive care units, nurses are seen at every level of a health care facility. Thus, these much-needed scheduling models should consider patients and nurses first. In brief, automated decision support systems and solutions that can schedule patients and nurses considering their preferences and medical needs of health care facilities would maximize the patient satisfaction, staff satisfaction, the utilization of scarce resources, and the quality of the health care provision.

All things considered, the motivation behind this Ph.D. thesis has two aspects: new models and new solution techniques. New models which can consider many factors while generating powerful and efficient schedules. These factors for patients could be to consider medical needs of patients, patients' preferences in terms of room sizes, medical equipment in rooms, preferred room features. On the other hand, factors for nurses are to incorporate their skills in the scheduling process, considering shift requests, day on and off requests, last minute changes to schedules, training needs, and vacation plans. And the list goes on. Production of such models would not only improve the overall satisfaction, but also make these models more applicable in the real world systems. In terms of the motivation of the thesis, the latter is to search on new solution techniques and develop such solution approaches and tools that can find the best-known schedules. In general, various algorithms can be embedded to solution frameworks and results can be evaluated. In this thesis, integer programming (IP) and meta-heuristics have received great deal of interest due to several reasons. While IP promises optimal solutions on smaller problems, it may perform poorly on large data sets. On the other hand meta-heuristics can tackle large problems and produce near-optimal results while sacrificing from the optimality. For this main reason, the research on solution techniques has extensively been around the hybridizing of many IP and meta-heuristic based algorithms.

1.2 Research Objectives

The main objective of this Ph.D. research is to develop real-world representing mathematical models and establish solution frameworks and tools that can effectively solve problems of any size generate optimal or near optimal solutions, and be used by end users. The following paragraphs summarize the objectives in detail.

Research objective 1: *Investigation on the trending and emerging research areas in the operational decision making in health care and identifying potential gaps in the literature.*

Literature review sections are incorporated into every chapter of the thesis to better investigate research papers within its own context. Generally, review of the literature on a certain chapter provides details on models examined by researchers and solution techniques developed for these models. Most of the chapters, both sides of the research are presented in tables for better comparison.

Research objective 2: *Identification of emerging mathematical models within the context of health care decision making that can be adapted to real-world systems.*

PAS model is a great example of this objective. The model is relatively new which has only been introduced recently in the literature. It deals with patients who need to stay at least one night at a hospital. Problem assigns these patients to beds considering medical and personal choices. The problem is common in real life. Patients arrive to hospitals with various medical needs. Scheduling unit must assign them to appropriate medical unit. But these patients come with many personal choices. They demand different room sizes, room features, and properties. On the other hand, the scheduling unit needs to consider room availability in terms of medical equipment and associated medical procedure. The studied mathematical model combines all these aspects and assigns patients to rooms and beds to maximize the overall patient satisfaction. Needless to say that manual addressing of these constraints by the scheduling unit would have led to great deal of discomfort.

Research objective 3: *Development of new models to better represent real-life business applications.*

Classical nurse rostering models commonly address the following constraints: limitations of number of shifts a nurse can work per day, restrictions of shift assignments after certain shifts, limitations around number of total shifts in a planning period, minimum and maximum work times, restrictions on consecutive work days and consecutive time offs, limitations on weekends, and vacation constraints. This Ph.D. study extends one of the classical models and incorporates the following hard constraints (HC) and soft constraints (SC) to achieve more realistic business situations: assignment limitation based on required nurse skills, departmental required and preferred skill constraints, last minute day on and day off requests

Research objective 4: *Development of new state-of-the-art solution techniques and frameworks that can solve problems of any size and provide optimal or near-optimal results.*

Several frameworks and solution algorithms are developed as part of the computational experiments of the Ph.D. research. First, MIP-based fix-and-relax (F&R) and fix-and-optimize (F&O) heuristics are utilized for the solution of the PAS problem versions. In general, the F&R heuristic is used to generate initial solutions and the F&O heuristic is utilized for the improvement of these initial solutions. During the initial solution generation phase, problems are decomposed into a set of smaller problems due to their combinatorial nature because IP cannot solve them to optimality. Smaller problems then are solved iteratively until all of them are solved. This property of MIP-based heuristics provides a great deal of flexibility for adaptation to any problem size. Because of the utilization of the strength of the IP, solutions obtained are quite acceptable compared to results from other studies. The improve phase also has a similar decomposition methodology. Smaller problems are re-optimized and results are compared to their existing values and accepted if they yield to superior outcomes.

MIP-based heuristics are further improved for the solution of the NRPs. While initial solutions are still obtained from the F&R heuristics, the F&O heuristic algorithm is hybridized with simulated annealing (SA) approach to benefit from the strength of both IP and meta-heuristic techniques. During the SA iterations, new solutions are obtained by providing new neighborhoods of the current solutions and evaluated in a probabilistic manner to generate better results. When the algorithm reaches to a point where there are no better solutions, the F&O heuristic is inserted into the process. This unique insertion results in far better results in most of the cases. Therefore, it results in intensification of the current search space. Even when no better solutions are found via this insertion, the resulting search space is quite different than the original one which leads to diversification of the search space.

The final solution technique is applied to the extended version of the NRP. In this framework, a mat-heuristic algorithm is developed by hybridizing IP and particle swarm optimization (PSO) algorithms. The IP is used in two ways. During the initial solution generation phase, small nurse models that only consider HCs are dynamically constructed and solved by the IP. The meta-heuristic algorithm then generates new solutions by applying certain encoding/decoding structures. When new structures result infeasible schedules, the IP is used again to repair infeasibilities. Repaired work plans are evaluated by the meta-heuristic whether they provide superior results or not. Better results are captured to update best solutions and this overall framework is followed until a certain termination criterion.

All of the state-of-the-art solution methodologies developed as part of the Ph.D. research mostly provides optimal or near-optimal solutions. MIP-based heuristics on the PAS and DPAS problems provide results that are only 5-15 percent gaps from the best-known solutions. The heuristics are able to provide best-known solutions on some of the test instances. The hybrid solution approaches that are experimented on

NRP problems improve the effectiveness of IP and MIP-based algorithms and outperform the state-of-the-art solutions techniques in most of the test instances. The hybrid methodology on the classical NRP problem report seven new best-known solutions.

1.3 Outline of the Thesis

In the current chapter, motivation, background, outline, and publications resulted from the Ph.D. study are presented. The rest of the chapters are outlined in the following paragraphs.

In Chapter 2, the PAS literature is reviewed and the problem model and notation are introduced. Implementation framework which is a combination of F&R and F&O heuristics is proposed. Computational results are reported and compared against the state-of-the-art solution methodologies in the literature.

In Chapter 3, a dynamic and an extended version of the PAS problem called DPAS is studied. New problem characteristics are presented. Differences between the versions are pointed out. F&O based solution methodology is proposed and the benchmark results are presented among the studies in the field.

In Chapter 4, a concept of nurse scheduling is introduced. An NRP model with its notation is presented with its mathematical representation. MIP-based heuristics are combined with SA meta-heuristic and resulting structure is demonstrated in detail. Publicly available data set is used for the implementation and final solutions are compared against results from other researchers. Concluding remarks are made.

In Chapter 5, a classical NRP problem is extended with new HCs and SCs. A novel model is presented in detail. A mat-heuristic based solution methodology that combines IP and PSO is also developed for the solution of the proposed model. A new data set is generated for computational experimentations. Results are reported and future studies are summarized.

In Chapter 6, conclusions are drawn. Future research directions are discussed. Final remarks are made.

Table 1.1 provides a summary of the problems’ characteristics and solution methods.

Table 1.1 Summary of the research problems and methodologies

	Chapter 2	Chapter 3	Chapter 4	Chapter 5
Problem	PAS Problem	Dynamic Patient Admission Scheduling (DPAS) Problem	NRP	Extended NRP
Problem Definition	Automatically assign patients to beds considering medical and personal needs/preferences	Assign patients to rooms considering PAS HCs and SCs, emergency situations, delays, and operating room utilizations	Assign nurses to shifts considering health care facility’s regulations, nurse needs, and nurses’ preferences such as shift on/off requests	Assign nurses to shifts and departmental units considering standard NRP HCs and SCs, preferred and required skills, and day on/off requests
Objective Function	Minimize costs resulting from SC violations	Minimize costs resulting from SC violations	Minimize costs resulting from SC violations	Minimize costs resulting from SC violations
Decision Variables	Binary integer decision variables	Binary integer decision variables	Binary integer decision variables with integer auxiliary variables	Binary integer decision variables with integer auxiliary variables
Solution Technique	F&R and F&O	F&R	F&R, F&O, and SA	IP and PSO

1.4 Produced Publications

The following research papers have been produced during the course of the Ph.D. research studies. All of the papers aim to address and support the operational decision making in the health care industry. These journals are either published, under review, or in preparation statuses for publications in international journals. Thesis chapters are correlated to the paper publications as noted in the following paragraphs.

Chapter 2:

- Turhan, A.M., & Bilgen, B. (2017). Mixed integer programming based heuristics for the patient admission scheduling problem. *Computers and Operations Research, 80*, 38-49.

Chapter 3:

- Turhan, A.M., & Bilgen, B. (2018). Fix-and-relax heuristic procedure applied to the dynamic patient admission scheduling problem. (Under Review, Submitted to an International Journal).

Chapter 4:

- Turhan, A.M., & Bilgen, B. (2018). A hybrid mixed integer programming based heuristics and simulated annealing approach for solving nurse rostering problems. (Under Review, Submitted to an International Journal).

Chapter 5:

- Turhan, A.M., & Bilgen, B. (2018). A mat-heuristic based solution approach for an extended nurse rostering problem with skills and departments. (In Preparation).

CHAPTER TWO
MIXED INTEGER PROGRAMMING BASED HEURISTICS FOR THE PATIENT ADMISSION SCHEDULING PROBLEM

2.1 Introduction

While healthcare spending costs on average 12 percent of the gross domestic product of the developed countries, the efficient usage of the scarce resources and patient satisfaction have become major tasks that hospital managements need to deal with on a daily basis as a result of increasing costs in this field. Scheduling departments must consider many elements before assigning patients to proper departments and rooms. Assigning patients to rooms related to specialism requirements, considering appropriate medical expertise for the patients, determining room needs with respect to equipment, and including patient preferences in the decision making are some of the considerations that must be made as part of this assignment process. In the end, manual assignments may lead to inefficient utilization of the critical resources and lower patient satisfaction.

Although hospital admission is well studied in the literature, majority of the studies are focused on decision making at the strategic and tactical level. On the other hand, the PAS problem which has been introduced by Demeester et al. (2008) focuses on the operational level and addresses some of the concerns noted by automatically assigning patients to beds for the duration of their stay considering not only the medical necessity but also the patient preferences.

Hospital patients are generally classified as inpatients and outpatients. Outpatient provision is finished during the day, but inpatients stay at hospitals for days. The PAS problem examines the admission of elective inpatients whose admission dates and Length of Stays (LoSs) are known in advance by their physicians. The PAS does not consider the emergency patients since the admission of those patients are not known and random.

The PAS problem has been studied by several researchers to date since its introduction. Bilgin et al. (2008), (2010), and (2012) apply a Hyper Heuristic (HH) to the problem along with a well-known NRP. The numerical results show that the performance of the HH depends on the performance of move acceptance criterion. Demeester et al. (2010) apply Hybrid Tabu Search (TS) algorithm combined with a token-ring and a variable neighborhood descent approaches. Some of the neighborhoods explored by the metaheuristic are moving a patient to another room within same department, moving a patient to another department, swapping two patients within same department, and finally moving the best candidate patient to another department. Ceschia and Schaerf (2011) by using the similar neighborhoods apply SA. Computational experiments report some of the best results known to date as well as lower bounds for the problem. In the same work, dynamic case of the problem where patient admission is not known in advance is also investigated. Range et al. (2014) introduce a new mathematical model and a Column Generation (CG) approach to the PAS and report new best known solutions for five out of thirteen instances. Hammouri and Alrifai (2014) examine the application of the BBO metaheuristic to the problem. The BBO is a metaheuristic inspired by the migration of species between habitats. Authors conclude by stating that the BBO needs further investigation due to obtained computational results. Granja et al. (2014) propose a simulation-based optimization approach to the PAS. Modeling tools and simulation techniques are used in the optimization of a diagnostic imaging department. Kifah and Abdullah (2015) develop an adaptive non-linear great deluge algorithm to tackle the patient admission problems. Chen and Lin (2017) develop patient referral mechanisms integrated with the PSO algorithm. In a recent study, Bolaji et al. (2018) present one point solution- based method named late acceptance hill climbing for solving the PAS problem using Bilgin et al. (2008) and (2012a) dataset consisting of 13 problem instances of different sizes and complexities. Table 3.1 summarizes the current research related to the PAS field along with various problem and model types and solution techniques. Guido et al. (2018) design optimization models for the assignment problem and propose a matheuristic algorithm for the solution. The results improve the best-known bounds. Moosavi and Ebrahimnejad (2018) propose a model to schedule elective patients considering upstream and downstream medical departments and apply MIP-based Local Neighborhood Search (LNS) to solve it. Current research in the PAS field along with problem types and various solution approaches are summarized in Table 2.1. Various solution approaches have been applied to the problem since its inception. SA, CG, and matheuristics are the most promising techniques to date. Although majority of the literature focuses on the deterministic version of the problem. There are also several studies on the dynamic version (Ceschia & Schaerf, 2011; Ceschia & Schaerf, 2012; Ceschia & Schaerf, 2014; Lusby et al., 2016).

Table 2.1 Summary of the PAS research

Reviewed Literature	**Type**	**Model**	**Solution approach**
Demeester et al. (2010)	D	BIP	TS
Ceschia & Schaerf (2011)	D, P	BIP	SA
Bilgin et al. (2012a)	D	BIP	HH
Ceschia & Schaerf (2012)	P	BIP	SA
Range et al. (2014)	D	BIP, DP	CG
Hammouri & Alrifai (2014)	D	BIP	BB
Granja et al. (2014)	D	BIP	SA+S
Ceschia & Schaerf (2016a)	P	BIP	SA
Kifah & Abdullah (2015)	D	BIP	ANGDA
Lusby et al. (2016)	P	BIP	ALNS
Turhan & Bilgen (2017)	D	BIP	FR, FO
Bolaji et al. (2018)	D	BIP	LAHC
Guido et al. (2018)	D	BIP	Matheuristic
Moosavi & Ebrahimnejad (2018)	D	BIP	LNS
Proposed Research	D	BIP	FR & FO

Notes: * *Type*; D – deterministic, P – probabilistic * *Model*; BIP – binary integer programming, DP – dynamic programming * Solution approach; ALNS - adaptive large neighborhood search

In this chapter, we propose MIP-based F&R and F&O heuristics to the PAS problem. Mathematical model of the reformulated problem provided by Ceschia and Schaerf (2011) is used since some of the preprocessing applied to the problem greatly improves the computational time. The approach behind the MIP-based heuristics is that combinatorial problem in hand which is NP hard and computationally intractable to solve to optimality is broken into smaller problems and then solved. There are several ways to decompose a problem into sub-problems. In our study, we employ time and patient decomposition approaches to generate smaller set of problems. In an iterative nature, sub-problems are solved until no other sub-problem is left to be solved. All the publicly available instances on the PAS website (Demeester, 2016) are used to properly assess the quality of the solution and processing times among other published studies. The main contributions of the study are two-fold: our study is the first application of the MIP-based heuristics to the PAS problem and the computational times achieved in our work are superior to the other solution approaches studied in the literature.

Rest of the chapter is organized as follows. In Section 2.2, the related literature on the proposed heuristic methods is reviewed. We present the definition and formulation of the problem in Section 2.3. Section 2.4 describes our solution approach. In Section 2.5, we present computational results and compare to previously reported values. Finally, conclusions and future research opportunities are addressed in Section 2.6.

2.2 Related Work on Heuristics

The F&R and F&O heuristics have successfully been applied to the planning and scheduling activities in production environment and they are receiving significant interest from the OR community for other application areas where the goal is to achieve the optimum in smaller size problems or to obtain near optimum results in larger size problems especially NP-hard ones in faster CPU times with small optimality gaps which also has been one of the motivations for our study.

In this section, we summarize the literature and provide an overview of the various application areas of the heuristics.

2.2.1 F&R Heuristic

F&R was first introduced by Dillenberger et al. (1994) to provide promising solutions to complex problems in deterministic environments. Since then the heuristic has received great interest in planning and scheduling applications especially in production settings. Application areas of the heuristic in the literature till date can generally be categorized as follows: production planning, lot sizing, scheduling, inventory planning, and supply chain management.

Stadtler (2003) applies the heuristic to multi-item multilevel lot sizing problem combined with a time-oriented decomposition where simple submodel is developed and solved with mathematical programming software. Kelly and Mann (2004) employ the heuristic to solve a lot sizing problem with constraint dropping approach. They observe an effective reduction on the processing time to find good solutions.

Alonso-Ayuso et al. (2006) find remarkable reduction in solution value and elapsed time on a supply chain management problem. Absi and Kedad-Sidhoum (2007) also focus on lot sizing while considering production planning and include setup time in their calculations. They apply time decomposition onto planning horizon and report gap analysis and computational time effectiveness on real-world instances. De Araujo et al. (2007) consider back orders and sequence dependent setup costs and conclude that the heuristic performs competitive results. Federgruen et al. (2007) examine joint setup costs also in the multi-item capacitated lot sizing problem.

Pochet and Wolsey (2008) present continuous time formulation for the cyclic scheduling to solve larger problems with a goal of providing good feasible solutions in a short time. Akartunalı and Miller (2009) focus on generating quick high quality solutions by providing a heuristic framework that is developed for production planning problems. Computational results suggest that the framework is effective especially on the most difficult problems.

Ouhimmou et al. (2008) apply the heuristic to supply chain problems. Maritime inventory routing application of the heuristic is studied by Uggen et al. (2013) where the primary purpose of the study is to reduce computation time. Computational findings suggest that the application is advantageous.

Sel and Bilgen (2014) solve a production and distribution planning problem in the soft drink industry. Experiments lead to the optimal solutions and suggest that heuristics are appropriate considering the complex nature of the supply chain problems and large computational times.

Silva et al. (2015) apply the heuristic to surgical scheduling in healthcare considering the skill set of the hospital staff and their availability. Results provide optimal or near optimal solutions. Another application of the heuristic combined with an SA algorithm to the parallel machine scheduling problem is by Xiao et al. (2015) where computational results yield to smaller gaps on the small size problems and outperform on the large size problems.

Liang et al. (2015) study the production planning and facility location problem using the heuristic to provide competitive results. Zhang et al. (2016) experiment the heuristic with a real-world production warehousing case. Computational experiments show that the heuristic can obtain near optimal results in reasonable processing time. Assis and Camponogara (2016) study crude oil transportation problem by proposing rolling-horizon and F&R strategies suggesting smaller gaps from optimum in shorter CPU times.

Roshani et al. (2017) apply the F&R heuristic to dynamic lot sizing problem considering reasonable solution times. The study reports the effectiveness of the algorithm. Qiu et al. (2018) use the F&R heuristic for the solution of production routing problems. The heuristic achieves effective computational results. The literature is summarized in Table 2.2.

Table 2.2 Applications of the F&R heuristic

Reviewed Literature	Production planning	Lot sizing	Scheduling	Other
Dillenberger et al. (1994)	X	X		
Stadtler (2003)		X		
Kelly & Mann (2004)		X		
Alonso-Ayuso et al. (2006)				Supply chain management
Absi & Kedad-Sidhoum (2007)	X	X		
De Araujo et al. (2007)		X	X	
Federgruen et al. (2007)		X		Inventory
Pochet & Wolsey (2008)			X	
Akartunalı & Miller (2009)	X			
Ouhimmou et al. (2008)				Supply chain management
Uggen et al. (2013)				Inventory
Sel & Bilgen (2014)	X			Supply chain management
Silva et al. (2015)			X	
Xiao et al. (2015)		X	X	
Liang et al. (2015)	X			Facility location
Assis & Camponogara (2016)				Supply chain management
Zhang et al. (2017)		X		Facility layout
Roshani et al. (2017)		X		
Qiu et al. (2018)	X			

2.2.2 F&O Heuristic

Very similar approach to solving combinatorial problems is also found in the F&O heuristic. Gintner et al. (2005), Pochet and Wolsey (2006), and Sahling et al. (2009) propose an improvement based heuristic. Pochet and Wolsey (2006) call it exchange heuristic with a goal of finding good feasible solutions and Sahling et al. (2009) introduce the name fix and optimize where large set of binary setup variables are fixed while small set is optimized. Computations show that the results provide high quality solutions with moderate computational efforts.

As in the case of the F&R heuristic, the F&O heuristic has also received considerable interests in applying onto production settings. Helber and Sahling (2010) apply it to a lot sizing problem experimenting different type of decompositions such as product, resource, and process. Results provide high quality solutions. James and Almada-Lobo (2011) use it to solve the same problem with sequence dependent setup times and costs. Goren et al. (2012) propose a hybrid approach of genetic algorithm (GA) and F&O towards the solution of the lot sizing problem achieving promising results. Other applications of hybrid solution approaches to the similar lot sizing problems are seen in Seeanner et al. (2013) and Stadtler and Sahling (2013) resulting high quality solutions in reasonable computational times and Guimaraes et al. (2013) and Xiao et al. (2013) outperforming state of the art solution techniques.

Apart from production environments, the F&O heuristic has also been used to apply in other areas of OR. Ghaderi and Jabalameli (2013) apply the heuristic to a healthcare facility location problem reporting better results than commercial solver. Dorneles et al. (2014) solve the high school timetabling problem with the F&O and the computational experiments yield that the heuristic approach outperforms state of the art algorithms. Camargo et al. (2014) develop a hybrid method including the heuristic and the results support the argument that the approach is a good option to reach better solutions. Toledo et al. (2015) combines F&R and F&O and report that the results are very efficient. Moreira et al. (2015) tackle a real-world freight transportation problem where the heuristic provides good quality solutions in reasonable time. Toledo et al. (2016) study glass container production planning problem where they report that the proposed methods return competitive results for smaller instances and high quality solutions for larger ones. Another production planning problem is studied by Wei et al. (2016) where tactical models are solved employing the time decomposition strategy and F&O heuristics embedding a variable neighborhood search (VNS) resulting good solutions in less CPU time.

Moreno et al. (2016) reports that time decomposition outperforms commercial solver in terms of processing time and optimality gaps. An implementation of the MIP-based heuristics by Aouam et al. (2018) on order acceptance in production planning shows promising results on solution quality and computational time. Vermuyten et al. (2018) introduce a staff scheduling problem in an emergency medical service and develop a Variable Neighborhood Decomposition Search heuristic that utilises the principles of the F&O heuristic. Computational results show that the proposed algorithm improves the current manual scheduling.

Literature is summarized in Table 2.3.

Table 2.3 Applications of the F&O heuristic

Reviewed Literature	Production planning	Lot sizing	Scheduling	Other
Gintner et al. (2005)			X	
Pochet and Wolsey (2006)	X			
Sahling et al. (2009)		X		
Helber and Sahling (2010)		X		
James and Almada-Lobo (2011)		X	X	
Goren et al. (2012)		X		
Seeanner et al. (2013)		X	X	
Stadtler and Sahling (2013)		X	X	
Guimaraes et al. (2013)		X	X	
Xiao et al. (2013)		X	X	
Ghaderi and Jabalameli (2013)				Facility location
Dorneles et al. (2014)			X	
Camargo et al. (2014)		X	X	
Toledo et al. (2015)		X		
Moreira et al. (2015)				Transportation
Toledo et al. (2016)	X	X	X	
Wei et al. (2016)	X			
Moreno et al. (2016)				Facility location
Aouam et al. (2018)	X			
Vermuyten et al. (2018)				Medical service

2.3 Problem Definition

Definition and details of the original problem are provided by Demeester et al. (2008) and (2010) and the preprocessing and the reformulation of the PAS are reported by Ceschia and Schaerf (2011).

PAS problem assigns set of patients to set of beds for the duration of patients' stays. Each patient has an admission date which patient is assigned to a room and a discharge date which patient is released from the medical treatment. Patient stay is the duration between the admission and the discharge dates. During their stay at the hospital, patients seek for medical treatments also referred in the literature as specialisms. Each patient is assigned to a bed and each bed is within in a room where the room is a part of a department. Departments are correlated with the specialisms they offer. Patients must be treated at the departments where the specialism they need is offered. Some of the departments also have age limitations such as pediatrics department. Each room has certain features such as capacity, equipment, specialism it offers, and genders it allows. Patient data comes with age, gender, admission and discharge dates, required specialism, room preference, needed and preferred room properties.

In the PAS problem, the goal is to satisfy all the HCs and to satisfy as many SCs as possible. Assigning patients to available rooms for the duration of their stay without discharging them from the treatment is necessary. Satisfying room age and gender policies as well as assigning patients to rooms with correct specialisms and properties are also major goals for the problem. Furthermore, staying in a preferred room, receiving the treatment in correct department, having preferred equipment in the room, and staying in the same room throughout the treatment are some of the additional needs that might be met.

We also describe here the terminology, notation, mathematical model, and constraints to make the chapter easy to follow.

2.3.1 Terminology

Patient: A person who is in need of a medical treatment. The PAS examines elective inpatients where a patient stays at least a night at hospital with a known admission and a known discharge date.

Night: The smallest unit of time. LoS is expressed in nights.

Planned patient: A patient who has a scheduled admission date and known LoS.

Admitted patient: A patient who is assigned to a room.

Admission Date: The date on which an inpatient admission occurs.

Discharge Date: The date on which a patient is medically ready to be discharged from hospital.

LoS: A period of stay which is known for each pathology.

Bed/Room/Department: A room may have one or more beds and every room belongs to a specific department. Capacity of a room is determined by number of beds in it. General practice is not to assign same gender to same room.

Specialism: Departments are specialized in treating various type of pathology (pediatric, intensive care unit, general surgery, neurology, etc.). Departments in the problem are categorized into major and minor specialism. Patients mainly prefer to be treated in departments with major specialisms correspond with their medical needs.

Room feature: Rooms have various features to treat patients (telemetry, oxygen, nitrogen, etc.).

Room gender policy: Every room has its own gender policy. Some rooms only accept male or female patients while other rooms allow both genders.

Room age policy: Some of rooms are specialized to treat patients within specific age limit. Ex. A room in pediatric department accepts only to a certain age.

Transfer: Moving a patient from one room to another during period of stay.

2.3.2 Constraints and Notation

Constraints are categorized into soft and hard constraints. HCs are those that have to be satisfied in order to provide a feasible solution. SCs are considered in the objective function and increasing number of satisfied SCs would lead to better solutions. HCs 5 – 8 originally reported by Demeester et al. (2010) are considered as SCs in our study since they are evaluated in the objective function and calculated and added to the objective value as penalties.

HC 1: During the planning period, room must be available.

HC 2: Admission and discharge dates cannot be changed.

HC 3: Patient LoS is continuous. Patients do not leave treatment before the discharge date.

HC 4: Two patients cannot be assigned to the same bed for the same night.

SC 1: Room gender policy should be satisfied.

SC 2: Room age policy should be satisfied.

SC 3: Some of the treatments require rooms to have specific equipment. These are called mandatory room properties.

SC 4: Some patients must be placed to single rooms as a result of special treatments (quarantine, etc.).

SC 5: Patient may ask for a single, twin, or ward during the stay.

SC 6: Patient may be treated in a department with correct specialism.

SC 7: Patient may be treated in a room with correct specialism.

SC 8: Some of the treatments suggest rooms to have specific equipment. These are called preferred room properties.

SC 9: Patients prefer to stay in the same room without any transfer to another room unless it is medically necessary.

For consistency, we use the same constraint weights reported by Demeester et al. (2010). Table 2.4 presents constraints and their corresponding weights.

Table 2.4 Weights of the constraints

Constraint	**Weight**
SC1	5.0
SC2	10.0
SC3	5.0
SC4	10.0
SC5	0.8
SC6	1.0
SC7	1.0
SC8	2.0
SC9	11.0

The notation used in the problem formulation is presented in Table 2.5.

Table 2.5 Notation used for the PAS model

Sets/Parameters	Definition
$p \in P$	Set of patients. P_M represents set of male patients and P_F represents set of female patients
$d \in D$	Set of days
$r \in R$	Set of rooms
c_r	Capacity of the room r
AD_p	Admission date of the patient p
DD_p	Discharge date of the patient p
$C_{p,r}$	The penalty of assigning patient p to room r. All the room penalties are combined into this value except SC1 (gender policy) and SC9 (transfer).
w_{RG}	Weight of room gender policy constraint (SC1).
w_{Tr}	Weight of transfer constraint (SC9).

Variables	Definition
$x_{p,r,d}$	1 if patient p is assigned to room r in day d, 0 otherwise
$t_{p,r,d}$	1 if patient p is transferred from room r in day d, 0 otherwise
$f_{r,d}$	1 if there is at least one female patient in room r in day d, 0 otherwise
$m_{r,d}$	1 if there is at least one male patient in room r in day d, 0 otherwise
$b_{r,d}$	1 if there are both male and female patients in room r in day d, 0 otherwise

2.3.3 Mathematical Model

The reformulation of the PAS reported by Ceschia and Schaerf (2011) is summarized in this section to make the chapter easy to follow. Objective function denoted with (2.1) is to minimize total penalties associated with assigning patients to rooms for the duration of their stay. First part of the objective function refers to the cost of assigning patients to rooms with a combined penalty of constraints SC2, SC3, SC4, SC5, SC6, SC7, and SC8. While second part of the function addresses the cost of violating the room gender policy, the last part of the function describes the cost associated with a transfer.

Minimize;

$$\sum_{p\in P,r\in R,d\in D} C_{p,r}.x_{p,r,d} + \sum_{r\in R,d\in D} w_{RG}.b_{r,d} + \sum_{p\in P,r\in R,d\in D} w_{Tr}.t_{p,r,d} \tag{2.1}$$

It is subject to the following constraints:

$$\sum_{r\in R} x_{p,r,d} = 1, \qquad \forall p \in P, d \in D_p \tag{2.2}$$

$$\sum_{p\in P} x_{p,r,d} \leq c_r, \qquad \forall r \in R, d \in D \tag{2.3}$$

$$f_{r,d} \geq x_{p,r,d}, \qquad \forall p \in P_F, r \in R, d \in D \tag{2.4}$$

$$m_{r,d} \geq x_{p,r,d}, \qquad \forall p \in P_M, r \in R, d \in D \tag{2.5}$$

$$b_{r,d} \geq \ m_{r,d} + f_{r,d} - 1, \qquad \forall r \in R, d \in D \tag{2.6}$$

$$t_{p,r,d} \geq x_{p,r,d} - x_{p,r,d+1}, \qquad \forall p \in P, r \in R, d \in D \tag{2.7}$$

Constraint (2.2) enforces every patient to be assigned to a room between admission and discharge dates. Constraint (2.3) ensures number of patients assigned to a room for a specific day cannot exceed the capacity of the room. Variables $b_{r,d}, m_{r,d}, f_{r,d}, and\ t_{p,r,d}$ are all dependent on the different circumstances of the x variables which describe the actual search space. If there is a female in a room, Constraint (2.4) forces the auxiliary variable $f_{r,d}$ to be equal to 1 to reflect the female existence in that room. Similar approach is taken for the Constraint (2.5) to reflect that there is a male in a room. If both genders exist in a room, Constraint (2.6) ensures that $b_{r,d}$ becomes 1 and gender penalty in the objective value is reflected accordingly. Finally, Constraint (2.7) ensures if patient has a room change between two consecutive days. In case patient room is changed, auxiliary variable $t_{p,r,d}$ becomes 1 and the transfer penalty is calculated in the objective value.

2.4 Solution Approach

In this chapter, we employ MIP-based F&R and F&O heuristics for the solution of the problem. To the best of our knowledge, this study is the first on applying the MIP-based heuristics in healthcare and also to the PAS problem in particular.

The main idea behind the heuristics is that the NP-Hard problem is broken into smaller set of problems and iteratively solved until the whole set is solved. Dividing the main problem into smaller set of problems is called decomposition and this property is changeable based on the structure of the problem. In the PAS problem, we utilize time and patient decomposition approaches to obtain smaller sets.

2.4.1 F&R Heuristic

F&R partitions the main problem into sub-problems and solves the smaller problems in an iterative nature until all the sub-problems are solved. Decomposition of the complex problem plays an important role in the application of the heuristic. Figure 2.1 shows an example flow for time decomposition. In this example, the whole horizon is divided into n intervals. In the first step, while decision variables in intervals 2 to n are relaxed, only interval 1 called optimization window is optimized. When optimization is finished, optimization window is increased to examine the next interval. In step 2, all the decision variables in interval 1 is fixed to their optimized values and interval 2 is optimized. Rest of the intervals in the planning horizon is non-fixed and continuous. Iterations are carried as represented in step 2 until the algorithm reaches to the last optimization window which is interval n. Algorithm proceeds to the last step where all the previously optimized decision variables from interval 1 to interval n-1 are fixed and interval n is optimized. Upon optimization of the last interval, the solution is presented.

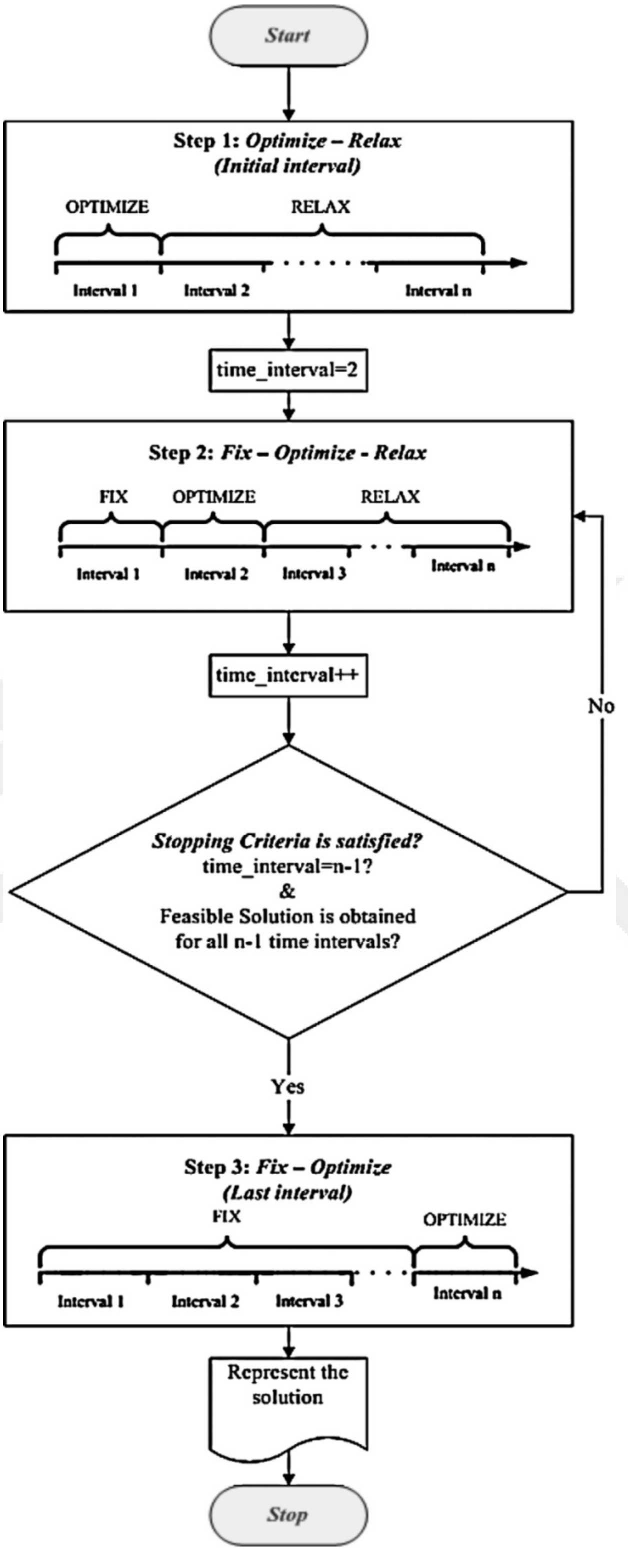


Figure 2.1 F&R flow by (Sel & Bilgen, 2014)

Overall algorithm for the F&R Heuristic is described in Figure 2.2.

Start and end of the optimization window are initialized and all the sub-problems are solved in the inner loop. When the end of optimization window becomes greater than the end of the planning horizon, inner loop is finished. The last sub problem is solved in line 9 and the result is returned.

```
Algorithm: F&R
1  :  Initialize startOfTheInterval and endOfTheInterval
2  :  while endOfTheInterval < endOfThePlanningHorizon do
3  :      Solve the sub-problem
4  :      Increase the start and end of intervals
5  :      if endOfTheInterval > endOfThePlanningHorizon then
6  :          endOfTheInterval = endOfThePlanningHorizon
7  :      end if
8  :  end while
9  :  Solve the last sub-problem
10:  Return solution
```

Figure 2.2 Pseudo-code of the F&R heuristic

2.4.2 F&O Heuristic

While the understanding of decomposing the problem into smaller sub-problems remains the same in the F&O, it additionally requires an initial solution to start and evaluates the new solution against the old solution before accepting.

Initial solution is received as an input to the algorithm; then it is being cross checked with the objective values received from the optimization. Whichever result provides a better objective value, it is accepted as a new solution for the corresponding sub-problem.

Time decomposition example is given by Sel and Bilgen (2014) in Figure 2.3. Initial solution is set as the best solution and the objective value associated with the initial solution is set as the objective value of the best solution. Planning horizon is divided into n intervals.

While end of the optimization window is less than the planning horizon, the algorithm loops through the steps. In step 1, the first interval is optimized and the rest of the intervals from 2 to n are fixed at their best known results.

Objective value obtained as the result of the optimization in step 1 is evaluated and accepted or rejected based on the value.

In step 2, interval 1 is fixed based on step 1, interval 2 is optimized, and the rest is fixed based on the initial solution. This approach is carried until the last interval.

In the last step, interval n is optimized while the rest of the binary decision variables are set to their integer values. After the objective value is evaluated, the result is presented.

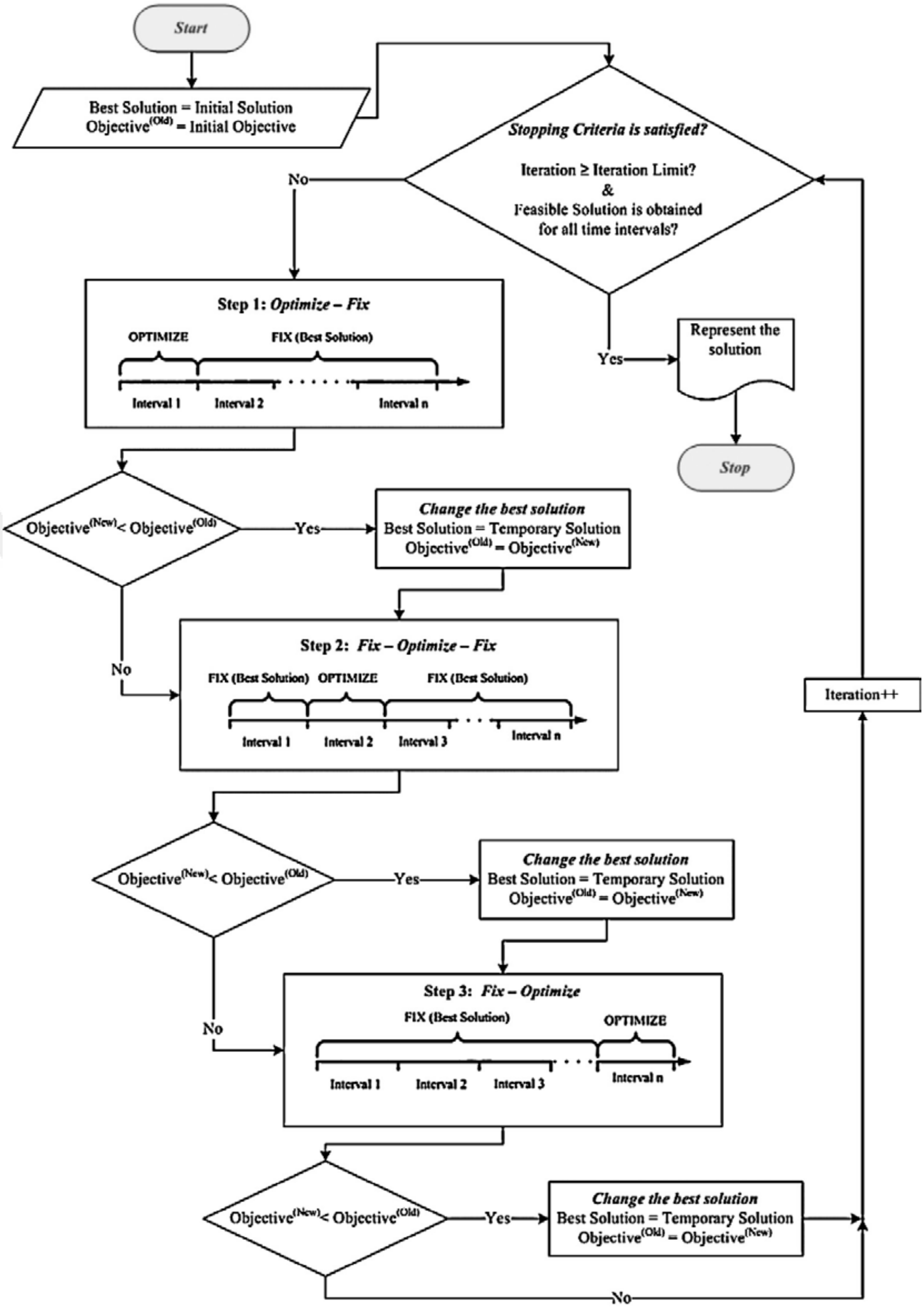

Start
Best Solution = Initial Solution
Objective$^{(Old)}$ = Initial Objective
Stopping Criteria is satisfied?
Iteration ≥ Iteration Limit?
&
Feasible Solution is obtained for all time intervals?
No
Yes
Represent the solution
Stop
Step 1: Optimize – Fix
OPTIMIZE
FIX (Best Solution)
Interval 1
Interval 2
Interval n
Objective$^{(New)}$< Objective$^{(Old)}$
Change the best solution
Best Solution = Temporary Solution
Objective$^{(Old)}$ = Objective$^{(New)}$
Step 2: Fix – Optimize – Fix
Interval 3
Step 3: Fix – Optimize
Iteration++

Figure 2.3 F&O flow by (Sel & Bilgen, 2014)

Figure 2.4 describes the algorithm of the F&O Heuristic. Initial solution is set as the best solution and the best cost is also the initial cost. Inner loop examines all the intervals except the last interval and evaluates each solution against the initial solution values. Last sub problem is solved and the same acceptance/rejection criterion is applied and the results are reported.

```
Algorithm: F&O
bestSolution = initialSolution
bestCost = initialCost
Initialize startOfTheInterval and endOfTheInterval
while endOfTheInterval < endOfThePlanningHorizon do
    if newCost < oldCost then
        bestSolution = newSolution
        bestCost = newCost
    end if
    Solve the sub-problem
    Increase the start and end of intervals
    if endOfTheInterval > endOfThePlanningHorizon then
        endOfTheInterval = endOfThePlanningHorizon
    end if
end while
Solve the last sub-problem
if newCost < oldCost then
    bestSolution = newSolution
    bestCost = newCost
end if
Return solution
```

Figure 2.4 Pseudo-code of the F&O heuristic

2.4.3 Implementation to the PAS Problem

Figure 2.5 and Figure 2.6 present a toy instance of the problem and the implementation of the proposed heuristics to the PAS problem. Patients are ranked based on their preferences as seen in Table 2.6. For instance, patient 5 looks for a twin room and he needs telemetry and oxygen in the room. Therefore two equipment need that is worth 5 points each (SC3) and twin room preference that is worth 0.8 points (SC5) result 10.8 points. Since the patient needs to stay 2 nights, the total cost of that patient is reflected as 21.6 points on the patient ranking column.

Table 2.7 reflects the details about the room. For simplicity of the toy instance, we choose to examine only SC 3 and 5. While Room 1 has a capacity of 4 and oxygen as available equipment, Room 2 has more equipment and it is more desirable for patients looking to stay in less crowded places.

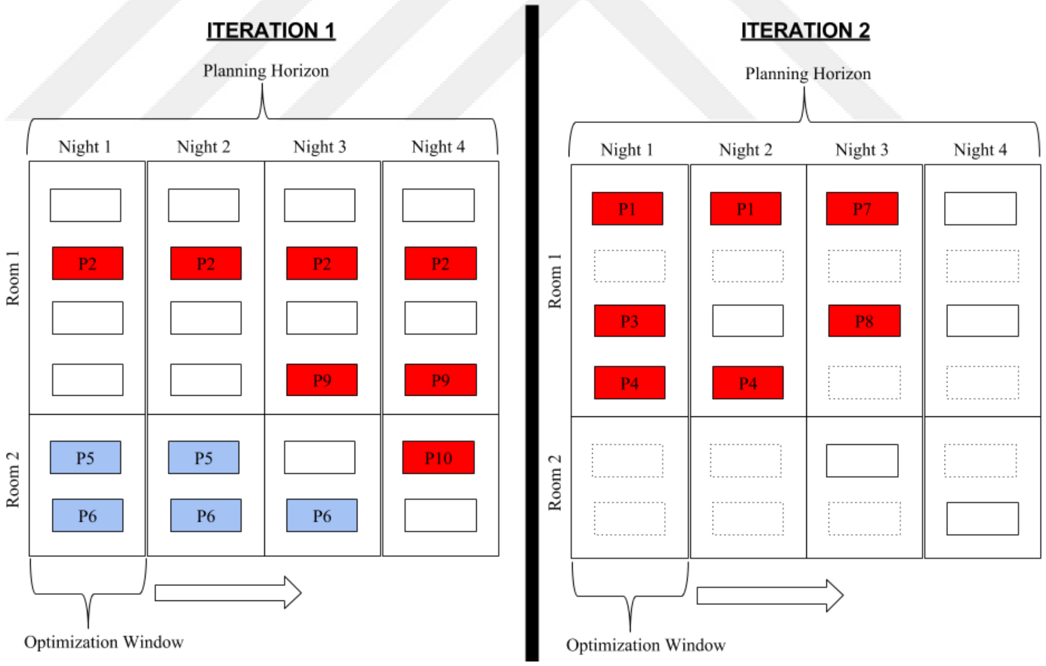
Table 2.6 Patient details for the toy instance

Patient ID	Gender	Admission Date	Discharge Date	Room Choice	Required Equipment	Patient Ranking
P1	Female	0	2	Ward	Oxygen	(0+5) * 2 = 10.0
P2	Female	0	4	Ward	Oxygen	(0+5) * 4 = 20.0
P3	Female	0	1	Ward	Oxygen	(0+5) * 1 = 5.0
P4	Female	0	2	Ward	Oxygen	(0+5) * 2 = 10.0
P5	Male	0	2	Twin	Telemetry and Oxygen	(0.8+10) * 2 = 21.6
P6	Male	0	3	Twin	Telemetry and Oxygen	(0.8+10) * 3 = 32.4
P7	Female	2	3	Ward	Oxygen	(0+5) * 1 = 5.0
P8	Female	2	3	Ward	Oxygen	(0+5) * 1 = 5.0
P9	Female	2	4	Twin	Oxygen	(0.8+5) * 2 = 11.6
P10	Female	3	4	Twin	Telemetry and Oxygen	(0.8+10) * 1 = 10.8

Table 2.7 Room details for the toy instance

Room ID	Room Capacity	Available Equipment
Room 1	4	Oxygen
Room 2	2	Telemetry and Oxygen

After patients are ranked based on the constraints and their LoS, certain threshold is selected to decompose patients into smaller groups. For our example, we choose to decompose patients based on a cost value of 10.5. Therefore we have patients 2, 5, 6, 9, and 10 to optimize in the first iteration whose cost values are greater than the threshold value as shown in Figure 2.5 (a). Optimization window length in our example is selected as only one night and planning horizon is decomposed into four separate nights. In the first iteration, only patients 2, 5, and 6 are evaluated. Night 1 is optimized while the integrality of the variables in the rest of the nights is relaxed. Then Night 1 is fixed and the optimization window is increased by 1 to inspect the next night. Upon evaluation of the whole planning horizon by applying this approach, iteration 1 is finished and the second iteration starts. In iteration 2, rest of the patients whose ranking values are less than 10.5 are evaluated keeping in mind that the beds already been assigned in the previous iteration are not available any more. For that reason, these beds are reflected with dotted borders. In the last iteration as shown in Figure 2.5 (b), patients 1, 3, 4, 7, and 8 are evaluated based on time decomposition. Only patients 1, 3, and 4 are optimized on the first night and the decision variables for the rest of the patients are relaxed. Rest of the planning horizon is optimized in this iterative nature until all the decision variables are optimized and fixed. The final schedule for the toy instance is presented in Figure 2.6.



(a) (b)

Figure 2.5 Patient and time decompositions for the toy instance of the problem

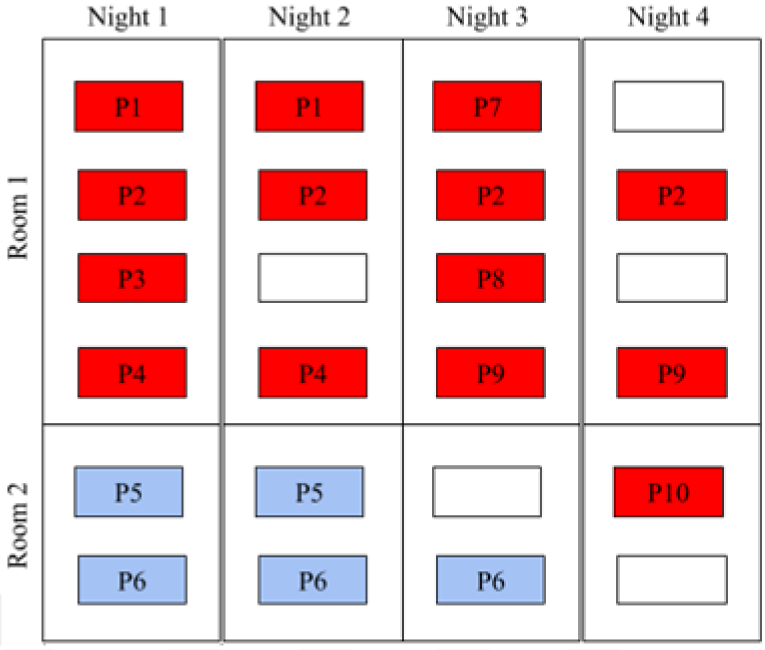


Figure 2.6 Final schedule for the toy instance of the problem

Figure 2.7 shows the details of the relationship between the F&R and F&O heuristics and its implementation to the PAS problem. Patients are decomposed and ranked first in the decomposition stage as represented in Table 2.6. This property provides a way to differentiate high cost patients from the low cost ones. Then starting from the first night, optimization window is decided. Depending on the length of the optimization window, room costs for all the patients staying within the period are calculated, CPLEX matrix structure is constructed and optimized. Finally, patients are assigned to room and capacities and gender policies are updated. This process is repeated until all the nights within the planning horizon are investigated. Quick generation of an initial solution by F&R is then used as an input to the F&O heuristic. Similar procedure is followed. Planning horizon is decomposed based on patient ranking and time and each sub problem is evaluated against the cost values obtained from the F&R. Better solutions and values are accepted and this loop is carried over until the whole set of sub problems are evaluated. When there are high volume of patients to be optimized exists in a certain optimization window even after the categorization, patients are randomly selected within their categories. Finally the best schedule and the best solution are reported.

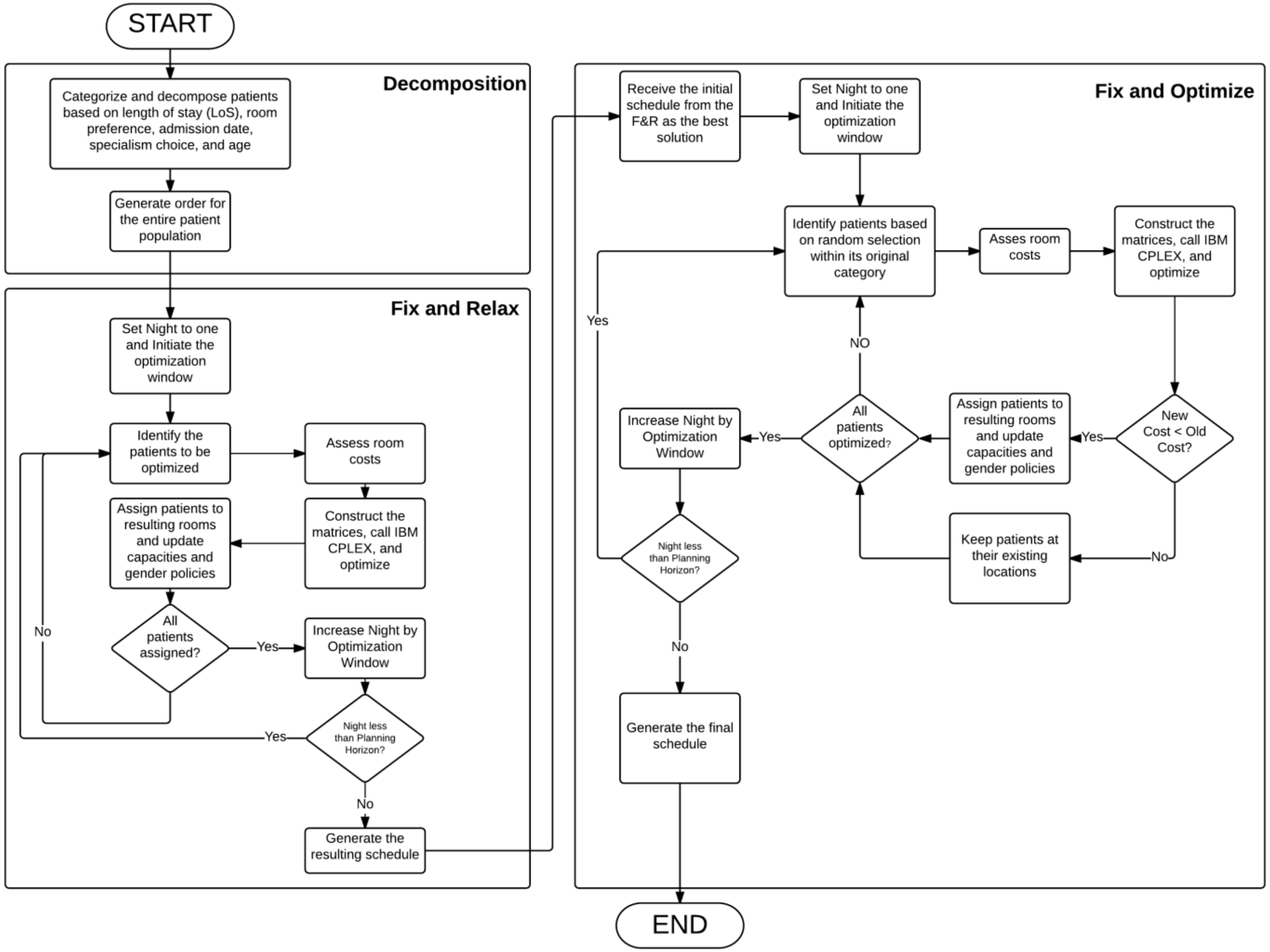

START
Decomposition
Categorize and decompose patients based on length of stay (LoS), room preference, admission date, specialism choice, and age
Generate order for the entire patient population
Fix and Relax
Set Night to one and Initiate the optimization window
Identify the patients to be optimized
Assess room costs
Construct the matrices, call IBM CPLEX, and optimize
Assign patients to resulting rooms and update capacities and gender policies
All patients assigned?
No
Yes
Increase Night by Optimization Window
Night less than Planning Horizon?
Yes
No
Generate the resulting schedule
Fix and Optimize
Receive the initial schedule from the F&R as the best solution
Set Night to one and Initiate the optimization window
Identify patients based on random selection within its original category
Asses room costs
Construct the matrices, call IBM CPLEX, and optimize
New Cost < Old Cost?
Yes
Assign patients to resulting rooms and update capacities and gender policies
No
Keep patients at their existing locations
All patients optimized?
NO
Yes
Increase Night by Optimization Window
Night less than Planning Horizon?
Yes
No
Generate the final schedule
END

Figure 2.7 PAS Implementation

2.5 Computational Experiments

In this section, we discuss the results of our proposed solution approach on instances provided by Demeester et al. (2008) to the state of the art solution techniques previously reported in the literature.

Table 2.8 shows the details of the instances. The number of elective patients is less than the number of total patients. This is due to some of the patients staying only during the day. For this reason, it is not required to schedule these outpatients to a room for any night. As discussed earlier, the PAS problem only deals with the scheduling of the elective patients.

Table 2.8 Details of the test instances

Test Data	Number of Rooms	Number of Beds	Number of Patients	Number of Elective Patients	Planning Horizon
1	98	286	693	652	14
2	151	465	778	755	14
3	131	395	757	708	14
4	155	471	782	746	14
5	102	325	631	587	14
6	104	313	726	685	14
7	162	472	770	519	14
8	148	441	895	895	21
9	105	310	1400	1400	28
10	104	308	1575	1575	56
11	107	318	2514	2514	91
12	105	310	2750	2750	84
13	125	368	907	907	28

Planning horizon is 14 days for the instances 1-7. As a result of number of elective patients and the planning horizon, these instances are smaller compared to the instances 8-13 where the planning horizon is between 21 days to 91 days. The studies in the literature as shown in Table 2.9 are mainly focused on the smaller instances. Solving the instances 8–13 is computationally more intractable. Best known results are reported by Range et al. (2014) on the instances 1-6 except 4 via applying CG and by Ceschia and Schaerf (2011) on the instances 4 and 7-13 via SA. As far as computational time is concerned, it is around 10-20 minutes for the CG approach on the smaller instances whereas it takes 18 hours on average for the SA approach on the bigger instances. This property motivates us for our study where we provide high quality solutions in a matter of seconds.

Employing the solution approach introduced earlier, the PAS problem has been re-implemented. The software is written in Matlab R2010a using the IBM ILOG CPLEX Optimization Studio 12.6 Application Programming Interface and run in a Windows 7 PC with Intel Core i3 2.27 GHz processor and 3 GB of RAM. All of our results have been validated using the Java program provided by the PAS website (Demeester, 2016). We limit our study only for the instances 1-12 and opt out the implementation of the solution on the instance 13 where multi-spec patients exist.

Results of the computational studies are summarized in Table 2.10. The processing times in our study are lower than the elapsed times reported in the previous studies in Table 2.9. Our first experiment is to apply only time decomposition and choose the optimization window as 1 day. The algorithm runs within 19 seconds on average for all the instances and as low as 11 seconds for some of the instances and provides cost values within 23 percent gap from the best known values. For some of the instances such as test 1 and test 12, the cost is even within 20 percent gap. And for test 5, algorithm is able to provide cost values even within 10 percent gap. But for the larger instances, the gap is around 25 percent.

The second set of experiments involves increasing the optimization window to 2 days and observing the change in the cost values and their gap without changing the approach on decomposition. Average run time for the algorithm is 66 seconds in this case overall for the instances and results are within 15-18 percent gap and as low as 9 percent for the test data 5. Another improvement is seen in the larger instances as well. 25 percent gap for the 1 day decomposition is improved to 20 percent on average when the optimization window is 2 days. We do not choose to report the results for 2 of the test data here since our goal is to provide good feasible solutions within a run time range around 120 seconds.

In the next step we choose to examine time decomposition where the optimization window is greater than 2 days. For smaller instances, even sub problems become computationally intractable as a result of the increase in the optimization window. But for the larger test data, we still observe improvement on the cost values where on average 18 percent gap is achieved within 104 seconds.

The results for the last set of experiments as shown on the last column in Table 2.10 are obtained by applying time and patient decompositions simultaneously. In this experiment, the computational time performance of the algorithm is as high as the times reported in the previous experiments. Without adding additional seconds to the run time, greater gap values are achieved. 10 percent gap is obtained within 104 seconds run time on average. Some of the gap values are within even 5 percent range. The most important property in this type of decomposition approach is to examine the cut off values when all the patients are ranked. If patients are divided into very small sets, the optimization window can be increased and vice versa. The cut off value and the optimization window width for each test data are used defined parameters and extensive computational experiments must be carried out to identify the best parameter values that lead to the best cost values.

The results reported here are the average values reached. Computational experiments show that the computational times in our algorithm are extremely lower than values reported in other studies.

Table 2.9 Previously reported computation times and costs.

	Demeester et al. (2010)		**Ceschia and Schaerf (2011)**		**Bilgin et al. (2012a)**		**Range et al. (2014)**		**Hammouri and Alrifai (2014)**	
Test Data	**Time (s)**	**Cost**	**Time (s)**	**Cost**	**Time (s)**	**Cost**	**Time (s)**	**Cost**	**Time (s)**	**Cost**
1	3000.00	671.20	13957.80	655.60	3000.00	830.36	595.40	**654.40**	686.30	1233.40
2	3000.00	1210.40	47465.50	1137.20	3000.00	1382.28	832.96	**1130.40**	945.70	2027.00
3	3000.00	827.00	23574.20	773.60	3000.00	923.16	709.67	**768.20**	790.00	1385.20
4	3000.00	1283.00	71291.70	**1172.20**	3000.00	1608.68	5347.90	1179.00	902.50	2211.00
5	3000.00	638.40	34718.90	625.60	3000.00	661.52	252.53	**624.00**	592.60	800.80
6	3000.00	828.80	14473.90	798.00	3000.00	955.04	446.66	**792.60**	684.70	1283.20
7	3000.00	1331.20	1363.10	**1193.00**	-	-	-	-	-	-
8	3000.00	4682.00	46287.40	**4149.80**	-	-	-	-	-	-
9	3000.00	22221.80	>24 hours	**21501.80**	-	-	-	-	-	-
10	3000.00	9806.60	57402.30	**8036.20**	-	-	-	-	-	-
11	3000.00	16025.60	>24 hours	**11811.80**	-	-	-	-	-	-
12	3000.00	28553.40	>24 hours	**23344.20**	-	-	-	-	-	-
13	3000.00	10277.60	16396.90	**9340.80**	-	-	-	-	-	-

Notes: Dashes represent that no results are reported by the studies. Bold and underlined values are the best known costs.

Table 2.10 Our study

	Time Decomposition - Optimization Window 1 night			Time Decomposition - Optimization Window 2 nights			Time Decomposition - Optimization Window > 2 nights			Time and Patient Decomposition		
Test Data	Time (s)	Cost	Gap	Time (s)	Cost	Gap	Time (s)	Cost	Gap	Time (s)	Cost	Gap
1	11.84	820.00	20.20%	43.94	775.20	15.58%	*	*	*	101.74	693.00	5.57%
2	24.77	1431.60	21.04%	*	*	*	*	*	*	121.76	1300.00	13.05%
3	13.48	1054.80	27.17%	114.05	934.00	17.75%	*	*	*	84.26	901.40	14.78%
4	21.34	1593.00	26.42%	137.23	1464.80	19.98%	*	*	*	109.34	1434.80	18.30%
5	12.40	693.80	10.06%	47.88	691.20	9.72%	*	*	*	46.48	657.60	5.11%
6	11.26	1038.40	23.67%	66.17	914.40	13.32%	*	*	*	71.00	899.00	11.84%
7	12.31	1730.40	31.06%	27.39	1620.60	26.39%	64.79	1566.00	23.82%	122.65	1426.60	16.37%
8	13.97	5460.20	24.00%	26.29	4968.20	16.47%	54.30	4763.80	12.89%	174.15	4673.40	11.20%
9	19.96	27305.80	21.26%	*	*	*	*	*	*	-	-	-
10	20.63	10904.80	26.31%	56.95	10800.80	25.60%	158.71	9850.00	18.41%	-	-	-
11	32.82	17115.40	30.99%	63.34	15173.40	22.15%	141.33	14743.40	19.88%	-	-	-
12	35.90	29125.60	19.85%	76.76	27878.60	16.26%	-	-	-	-	-	-
13	-	-	-	-	-	-	-	-	-	-	-	-

Notes: Asterisks represent that no results are reported due to problem size becoming large to solve. Dashes represent that no results are reported due to computation time being greater than three minutes. Instance 13 is not examined.

2.6 Conclusion

PAS is an important planning activity that hospital managers must consider on a daily basis. While resource utilization is still quite critical in healthcare, increasing patient demand forces planners to account for other areas such as patient satisfaction. Making this complex decision without a computerized solution does not seem to be feasible. Software solution that can quickly produce optimal or near optimal results can be very beneficial for the planning departments.

In this chapter, we have applied MIP-based heuristics called F&R and F&O to the PAS problem defined by Demeester et al. (2008). The F&R Heuristic provides feasible solutions in short calculation times and the F&O Heuristic improves the initial solution received by the F&R in an iterative nature. Overall, patients are decomposed based on their preferences and LoSs. During the heuristic phase, planning horizon is decomposed into optimization windows and different optimization window sizes are evaluated as well as different patient classification thresholds.

In conclusion, our solution approach provides high quality solutions in less than 3 minutes when they are compared to other state of the art solution techniques. Our computational results show that most of the objective values are within 5 to 15 percent gap from the best known solutions and overall results are averaging 14 percent gap. While most of the studies are focused on the smaller instances, our approach can still produce good feasible solutions even for the larger instances. For future research, we could investigate possible applications of metaheuristics to improve the existing solutions. Dynamic version of the PAS introduced by Ceschia and Schaerf (2012) is also another promising area of research towards adapting the problem into real world situations such as no-shows, service delays, and resource availabilities.

In the next chapter, we extend our work to the dynamic version of the PAS problem. Current structure of the problem does not account for delays and emergency situations. But it is very common in real-life that there are always unforeseeable changes in this health care process at all times. The DPAS problem additionally considers these dynamic situations as well as accounts for the usage of operating rooms.

CHAPTER THREE
FIX-AND-RELAX HEURISTIC PROCEDURE APPLIED TO THE DYNAMIC PATIENT ADMISSION SCHEDULING PROBLEM

3.1 Introduction

Health care provision in hospitals is a complex and expensive process that is managed at the strategic, tactical, and operational levels. Strategic planning consists of long-term planning activities to address supply and demand gaps between hospital resources and patients. The tactical planning focuses on medium-term activities and take actions based on strategic plans. And the operational planning considers the daily or weekly allocation of critical resources to meet needs. As described also in the previous chapter, it is at the operational level where a problem arises: patients who need to stay in a hospital overnight must be assigned to certain rooms based on medical needs, patient preferences, room and department availabilities, and resource constraints. This chapter extends the research in the previous chapter by additionally considering probabilities and new constraints to achieve real-world practices.

Ceschia and Schaerf (2011) introduce the concept of uncertainty for the first time to Demeester et al. (2008)'s original PAS problem. As in the PAS problem, the DPAS problem also deals with generating patient and bed assignment schedules that include many HCs and SCs. The final schedules are to be used by hospital scheduling departments for better allocation of resources. The first formulation on the DPAS problem is by Ceschia and Schaerf (2012). In this DPAS model, risk of rooms being overcrowded because of delays is studied. Later, Ceschia and Schaerf (2016a) extend the model by considering emergency patients and operating room availabilities, making the previously formulated model closer to real-life scenarios. The major difference between the PAS and the DPAS problems is that while the planned admission date and actual admission date of patients to a hospital are the same in the PAS problem, they can be different in the DPAS problems due to the possibility of delays in health care provision for other patients, unavailability of operating rooms, or the existence of emergency situations.

A comprehensive review of the PAS problem literature is provided in the previous chapter. In this section, the literature around the dynamic version of the PAS problem is discussed. Ceschia and Schaerf (2011) use the SA algorithm to solve the dynamic case, in which a patient's admission is not known in advance. The authors use similar neighborhoods as in the deterministic problem. In a subsequent work, Ceschia and Schaerf (2012) further explore the dynamic case by experimenting with patient admission delays and uncertainty in the length of patients' stays using a similar SA procedure. Computational results show that the algorithm can be utilized as a legitimate solution approach for dynamic environments. Same researchers in Ceschia and Schaerf (2016b) reformulate the DPAS problem to introduce new constraints, including delays, operating room availability, and emergency situations, thereby advancing the earlier version of the DPAS problem closer to actual hospital operations. The outcome of the work shows that operating rooms can be effectively incorporated into patient scheduling problems. Lusby et al. (2016) develop an ALNS procedure to solve the previous version of the DPAS problem by Ceschia and Schaerf (2012) that focuses on improving the solution quality and provides solutions in faster calculation times. In spite of longer solution times on large instances, the proposed ALNS approach finds more feasible solutions than the techniques in the literature.

Table 3.1 summarizes the current research related to the DPAS field along with solution techniques.

Table 3.1 Summary of the DPAS research

Reviewed Literature	**Type**	**Model**	**Solution approach**
Ceschia & Schaerf (2011)	D, P	BIP	SA
Ceschia & Schaerf (2012)	P	BIP	SA
Ceschia & Schaerf (2016a)	P	BIP	SA
Lusby et al. (2016)	P	BIP	ALNS
Proposed Research	P	BIP	FR

In this chapter, we extend our previous work to the DPAS problem which is described by Ceschia and Schaerf (2016a). We first report a full mathematical model of the problem which has been missing in the literature previously. Then, we solve the model using the F&R heuristic procedure. In this heuristic, the NP hard problem is divided into a set of smaller problems because an optimal solution of the original problem is not computationally likely. Obtaining the set makes it possible to optimize each smaller problem in short computational times, and each of the sub-problems is optimized until there are none left to be solved. Publicly available instances of the DPAS problem on the DPAS website (Ceschia & Schaerf, 2017) are solved using the proposed procedure, and computational findings are compared to the state-of-the-art solution techniques.

This chapter contributes to the literature in three ways: it is the first implementation of the F&R procedure to the DPAS problem, we outperform some of the previously reported values, and the processing times in our procedure are much shorter than those in other studies.

The remainder of this chapter is organized as follows. In Section 3.2, the details, notation, and formulation of the DPAS problem are defined. Section 3.3 describes the proposed solution approach with a sample toy instance. In Section 3.4, we present the computational experiments and compare them to previously reported results. Finally, in Section 3.5, we address our conclusions and possible future research opportunities.

3.2 Problem Definition

In the DPAS problem, first formulated by Ceschia and Schaerf (2012), the length of patients' hospital stays are uncertain, delays are likely, and rooms may become overcrowded.

The original deterministic PAS problem becomes dynamic when the idea that patients' planned room assignments might be different than the actual assignments is taken into account. While a patient can be assigned to a room weeks before their treatment in the PAS problem, they can only be assigned on the previous night or actual day of the treatment in the DPAS problem.

In a subsequent study, Ceschia and Schaerf (2016a) further enhance the problem by adding operating-room-utilization constraints and emergency situations. We prefer studying this more complex problem because it offers more-realistic real-world scenarios.

Over time, the PAS and DPAS problems have evolved from assigning patients to beds to assigning patients to rooms. Each room has certain capacity limitations, gender policies, and features (i.e. infusion pumps, nitrogen tanks, televisions). Rooms are specific to departments and medical specialties, such as endocrinology or internal medicine. Operating rooms have daily total capacities and designated capacities for each specialism. Treatments are only offered in certain specialisms. A patient who seeks a certain treatment needs to be assigned to a room and department which offer the needed treatment and specialism.

One of the unique characteristics of the PAS and DPAS problems is that they do not only deal with medical necessities: they also try to incorporate patient preferences. For instance, a patient may want a single occupancy room or expect a room with a television.

The DPAS problem considers constraints in two categories. HCs must be met, while SCs are not required to be fulfilled – although failure to meet them results in penalties. For example, one HC is that a patient cannot be released from a room while a treatment is in progress. On the other side, although it may result in a penalty, it may still be feasible to assign patients to rooms that lack properties they desire.

3.2.1 Terminology and Notation

Below, we describe the terminology, notation, mathematical model, and constraints that will be used throughout the remainder of this chapter.

Patient: A patient is a person who receives a certain treatment. Due to the fact that the DPAS problem focuses on the assignment of patients who stay at least a day at a hospital, only elective patients are considered.

Day: Every elective patient stays at least a day at a hospital. The number of days between the admission and discharge dates defines the LoS of each patient. Once patients are admitted to a hospital, they receive treatment continuously during their stay. Patients have planned and actual dates of acceptances to a hospital. *Registration day* is the day a patient becomes a treatment-seeker in the hospital system, and *admission date* is the day a patient is planned to be assigned to a room. Some patients may have required admission dates due to the severity or urgency of their medical problems. Similarly, a patient's *discharge date* is a planned date that may change in practice depending on the actual admission date.

Room: A patient is assigned to a room based on not only the bed availability, but also the room features such as equipment and size. Some rooms may have gender or age restrictions, such as a room located in a pediatrics department. Once patients are assigned to rooms, moving them to other rooms will cause discomfort.

Department/Specialism/Treatment: Each room is equipped with features specific to its department's specialism. A single department may include multiple specialisms, and each specialism offers multiple treatments. Some departments also include secondary specialisms. For example, general medicine performs internal medicine as its main specialty, but it can also perform critical care functions if necessary. It may be required to assign a patient to a specific room because it has certain medical equipment.

The notation used in the problem formulation is presented in Table 3.2.

Table 3.2 Notation for the DPAS model

Symbol	Definition	Symbol	Definition
$p \in P$	Set of patients	c_r	Capacity of the room r
P_M	Set of male patients	c^{OR}	Capacity of the operating room (OR)
P_F	Set of female patients	$c^{OR}{}_s$	Capacity of the OR for specialty s
P_d	Set of patients present on day d	l_o	Length of operation
$P_d{}^+$	Set of patients potentially present on day d	$A_{p,r}$	Boolean valued matrix that decides if patient p can be assigned to room r
$P_d{}^A$	Set of patients who may be assigned on day d	$Z_{p,d}$	Boolean valued matrix that decides if patient p can be delayed or not on day d
$P^O{}_d$	Set of patients who has operation on day d	$C_{p,r}$	Patient-room combined weight
$P^O{}_{s,d}$	Set of patients who has operation in specialty s on day d	w_{RG}	Weight for room gender cost
$d \in D$	Set of days	w_{OR}	Weight for overcrowding risk
D_p	Set of days that patient p is present	w_{DE}	Weight for delay
$D^A{}_p$	Set of days that patient p may be assigned	w_{IR}	Weight for idle room capacity
$r \in R$	Set of rooms	w_{IOR}	Weight for idle operating room
R_{SG}	Set of rooms with same gender policy	w_{ORO}	Weight for specialism overtime
$s \in S$	Set of specialties	w_{ORTO}	Weight for total overtime

$x_{p,r,d}$	1 if patient p is assigned to room r on day d, 0 otherwise
$z_{p,d}$	1 if patient p is delayed on day d, 0 otherwise
$y_{r,d}$	1 if the room r risks being overcrowded in day d, 0 otherwise
$f_{r,d}$	1 if there is at least one female patient in room r in day d, 0 otherwise
$m_{r,d}$	1 if there is at least one male patient in room r in day d, 0 otherwise
$b_{r,d}$	1 if there are both male and female patients in room r in day d, 0 otherwise

3.2.2 Hard and Soft Constraints

As in previous versions of the PAS problem, the DPAS problem contains hard and soft constraints. HC must be satisfied; otherwise the problem solution would be infeasible. Failure to satisfy SC only affects the solution quality. Better solutions are obtained as the number of satisfactory SCs increases. Since the HCs and SCs in the DPAS problem are slightly different than those introduced in the earlier PAS problems, we define them here.

HC 1: Capacity of a room is known and cannot be violated.

HC 2: Once a patient is accepted and assigned to a room, the patient must continuously stay under a treatment until the patient is fully discharged from the hospital.

HC 3: When a patient is assigned to a room, the patient can no longer be delayed.

HC 4: A patient cannot be assigned to a department where the needed treatment is not offered at all.

HC 5: A patient cannot be assigned to a room in which equipment used to perform a required treatment is not present.

HC 6: A male or female patient cannot be assigned to a room with a policy respectively enforcing the acceptance of female or male patients only.

HC 7: A patient cannot be assigned to a room with an age policy that restricts admittance due to a patient’s age.

SC 1: If a room has a gender policy, it should be satisfied.

SC 2: A department should have the specialism patient needs as its main specialism.

SC 3: A room should have the features a patient prefers.

SC 4: The size of a room should be less than or equal to the size a patient wants (i.e. single, twin, or ward with capacity of four).

SC 5: Transferring a patient to another room from the current room should be minimized.

SC 6: Delaying the admission of a patient should be minimized.

SC 7: The risk of overcrowding rooms due to delays and overstays should be minimized.

SC 8: Rooms should not be idle.

SC 9: Operating rooms should not be idle.

SC 10: Operating rooms should not be over utilized per specialty or overall.

The same constraint weights used by Ceschia and Schaerf (2016a) are reported here to properly assess the quality of the solution procedure and for consistency with the literature. Constraints, default values, and the calculation procedure are presented in Table 3.3.

Table 3.3 Weights of the constraints

Constraint	Weight	Calculation
SC1 – Gender Policy	50	Per day, per misplaced patient
SC2 – Primary Specialism	20	Per day, per patient
SC3 – Room feature	20	Per day, per patient
SC4 – Room size	10	Per day, per patient
SC5 – Transfer	100	Per patient
SC6 – Delay	5	Per day, per patient, per priority
SC7 – Overcrowd risk	1	Per patient
SC8 – Idle room	20	Per day, per bed
SC9 – Idle operating room	10	Per minute
SC10 – Overtime	3	Per minute

3.2.3 Model

We provide here the full mathematical model which is partially formulated in Ceschia and Schaerf (2016a). The overall objective of the DPAS model is to minimize the number of violations that result in penalties due to unsatisfied SCs. The objective function in Equation (3.1) can be denoted as follows:

Minimize

$$\mathrm{F} = \mathrm{F}_{\mathrm{RoomCost}} + \mathrm{F}_{\mathrm{Gender}} + \mathrm{F}_{\mathrm{Overcrowd}} + \mathrm{F}_{\mathrm{Delay}} + \mathrm{F}_{\mathrm{IdleRoom}} + \mathrm{F}_{\mathrm{IdleOR}} + \mathrm{F}_{\mathrm{ORO}} + \mathrm{F}_{\mathrm{ORTO}} \tag{3.1}$$

Function F is the sum of penalties from room costs, violations of the room's gender policy, the room being overcrowded for a certain day as a result of patients with overstay risks, the cost of delays, idle rooms and operating rooms, and the overutilization of operating rooms by certain specialties.

$$\mathrm{F}_{\mathrm{RoomCost}} = \sum_{\mathrm{d}\in \mathrm{D}_p, \mathrm{r}\in\mathrm{R}, \mathrm{p}\in\mathrm{P}} \mathrm{C}_{\mathrm{p,r}}.\, \mathrm{x}_{\mathrm{p,r,d}} \tag{3.2}$$

$$\mathrm{F}_{\mathrm{Gender}} = \sum_{\mathrm{d}\in\mathrm{D}, r\in R_{SG}} w_{RG}.\, b_{r,d} \tag{3.3}$$

$$\mathrm{F}_{\mathrm{Overcrowd}} = \sum_{\mathrm{d}\in\mathrm{D}, r\in R} w_{OR}.\, y_{r,d} \tag{3.4}$$

$$\mathrm{F}_{\mathrm{Delay}} = \sum_{d\in D_p^A, p\in P} w_{DE}.\, z_{p,d} \tag{3.5}$$

$$\mathrm{F}_{\mathrm{IdleRoom}} = \sum_{d\in D} \Big(\sum_{r\in R} c_r - \sum_{r\in R, p\in P} x_{p,r,d} \Big).\, w_{IR} \tag{3.6}$$

$$\mathrm{F}_{\mathrm{IdleOR}} = \sum_{d\in D} \Big(c^{OR} - \sum_{r\in R, p\in P_d^O} x_{p,r,d}.\, l_o \Big).\, w_{IOR} \tag{3.7}$$

$$\mathrm{F}_{\mathrm{ORO}} = \sum_{d\in D} \Big(\sum_{s\in S} \Big(\sum_{r\in R, p\in P_{s,d}^O} x_{p,r,d}.\, l_o - c_s^{OR} \Big) \Big).\, w_{ORO} \tag{3.8}$$

$$\mathrm{F}_{\mathrm{ORTO}} = \sum_{d\in D} \Big(\sum_{r\in R, p\in P_d^O} x_{p,r,d}.\, l_o - c^{OR} \Big).\, w_{ORTO} \tag{3.9}$$

Equation (3.2) accounts for all the penalties related to assigning a patient to a room. The $\mathrm{C}_{\mathrm{p,r}}$ matrix is calculated as part of the pre-processing of penalties because patient room preferences will not need to be changed during the optimization procedure. The only penalty added while optimization occurs is the transfer penalty, wherein moving a patient from a previously assigned room to any other room adds costs to other rooms.

Equation (3.3) deals with gender violations, should a room have mixed genders on a given day. Equation (3.4) accounts for the risk of overcrowded rooms. When patients risk staying beyond their discharge dates, their rooms risk becoming overcrowded. Equation (3.5) addresses a delay in the patient's planned admission date. Equation (3.6) and Equation (3.7) calculate penalties when rooms or operating rooms are underused. Finally, the overutilization of operating rooms per specialism and per room is taken into account in Equation (3.8) and Equation (3.9), respectively.

The objective function is subject to the following constraints:

$$\sum_{\mathrm{r \in R}} \mathrm{x_{p,r,d}} + \mathrm{z_{p,d}} = 1, \qquad \forall\, \mathrm{p} \in P_d^A, \mathrm{d} \in \mathrm{D} \tag{3.10}$$

$$\sum_{\mathrm{p \in P_d}} \mathrm{x_{p,r,d}} \leq \mathrm{c_r}, \qquad \forall\, \mathrm{r \in R, d \in D}, \tag{3.11}$$

$$\mathrm{x_{p,r,d}} \leq \mathrm{A_{p,r}}, \qquad \forall\, \mathrm{p \in P, r \in R, d \in D} \tag{3.12}$$

$$\sum_{\mathrm{p \in P_d^+}} (1 - \mathrm{x_{p,r,d}}) \geq (|\mathrm{P_d^+}| - \mathrm{c_r}) \cdot (1 - \mathrm{y_{r,d}}), \qquad \forall\, \mathrm{r \in R, d \in D} \tag{3.13}$$

$$\mathrm{f_{r,d}} \geq \mathrm{x_{p,r,d}}, \qquad \forall\, \mathrm{p \in P_F, r \in R, d \in D_p} \tag{3.14}$$

$$\mathrm{m_{r,d}} \geq \mathrm{x_{p,r,d}}, \qquad \forall\, \mathrm{p \in P_M, r \in R, d \in D_p} \tag{3.15}$$

$$\mathrm{b}_{\mathrm{r,d}} \geq \mathrm{m}_{\mathrm{r,d}} + \mathrm{f}_{\mathrm{r,d}} - 1, \quad \forall\, \mathrm{r} \in \mathrm{R}, \mathrm{d} \in \mathrm{D} \tag{3.16}$$

$$\mathrm{z}_{\mathrm{p,d}} \leq \mathrm{Z}_{\mathrm{p,d}}, \quad \forall\, \mathrm{p} \in \mathrm{P}, \mathrm{d} \in \mathrm{D} \tag{3.17}$$

Constraint (3.10) enforces the rule that a patient can be assigned to a room or delayed admission for that day. Constraint (3.11) limits the number of patients in a room to be less than or equal to the room’s capacity.

Matrix A is calculated in advance to determine whether or not a patient can be assigned to a room. This Boolean-valued matrix takes into account some of the HCs, including Equations (3.4), (3.5), and (3.6). For instance, if a room does not have the medical equipment for the treatment a patient requires, the patient must not be assigned to the room. Constraint (3.12) guarantees this property.

Constraint (3.13) calculates if there is an overcrowding risk for a certain room on a given day. Rooms having at least one female or male are calculated via Constraint (3.14) and Constraint (3.15), respectively. Nevertheless, when both genders are present in a room on the same day, auxiliary gender variables force the same gender variable to be 1 in Constraint (3.16).

Finally, Constraint (3.17) addresses delays. Matrix Z is a Boolean-valued matrix which controls if patients can be delayed or not. Once a patient is assigned to a room for a given day, Matrix Z must ensure that the patient cannot be delayed anymore for the rest of the days in the planning horizon. In addition, the matrix must validate the maximum admission criteria when a patient can no longer be delayed.

3.3 Solution Technique

Processing time efficiency in our previous study on the deterministic PAS problem, Turhan and Bilgen (2017), motivated us to extend our work to the dynamic version. Dividing an NP-hard problem into smaller sub-problems and solving each step-by-step makes MIP-based heuristic techniques a good alternative to meta-heuristic solution procedures.

In this section, we briefly introduce literature on the heuristic and its applications and discuss its implementation on the DPAS problem in depth.

3.3.1 F&R Heuristic

As reviewed in the previous chapter in detail, the F&R heuristic identified by Dillenberger et al. (1994) has been applied to a wide range of planning and scheduling problems, especially in production environments. The main reason is that mathematical models dealing with planning activities in real-life often result in computationally intractable systems to solve to optimality. This leads researchers to heuristic solution techniques to obtain near-optimal results. The main advantage of the MIP-based heuristics comes into the picture at this point. Because the heuristics utilize decomposition approaches to break models into smaller problems. With the successful decomposition implementation of the F&R previously, similar decomposition technique can also be employed in the DPAS model for obtaining sets of smaller problems to achieve less-complex sub-problem sets.

Figure 3.1 explains a sample procedure that can be applied to any planning or scheduling setting where day decomposition is possible. In this example, the planning horizon is seven days. The goal is to decompose the horizon into seven small problems and solve each in an iterative process. As the first step, the first day of the horizon is optimized, and the rest of the days in linear programming are relaxed. Once the optimization of Day 1 is finished, the algorithm fixes the variables in Day 1 and proceeds to Day 2. Days between 2 and 6 are optimized and fixed day-by-day until Day 7 is reached. In the last step of the procedure, Day 7 is optimized, and the final schedule is reported. Depending on the complexity of the smaller problems, the optimization window approach can also be employed such that the planning horizon can be examined by a two-day window. In the first step, Days 1 and 2 can be optimized together, followed by Days 3 and 4, etc. While the computational time may be affected by the change in the optimization window, it may provide better computational results.

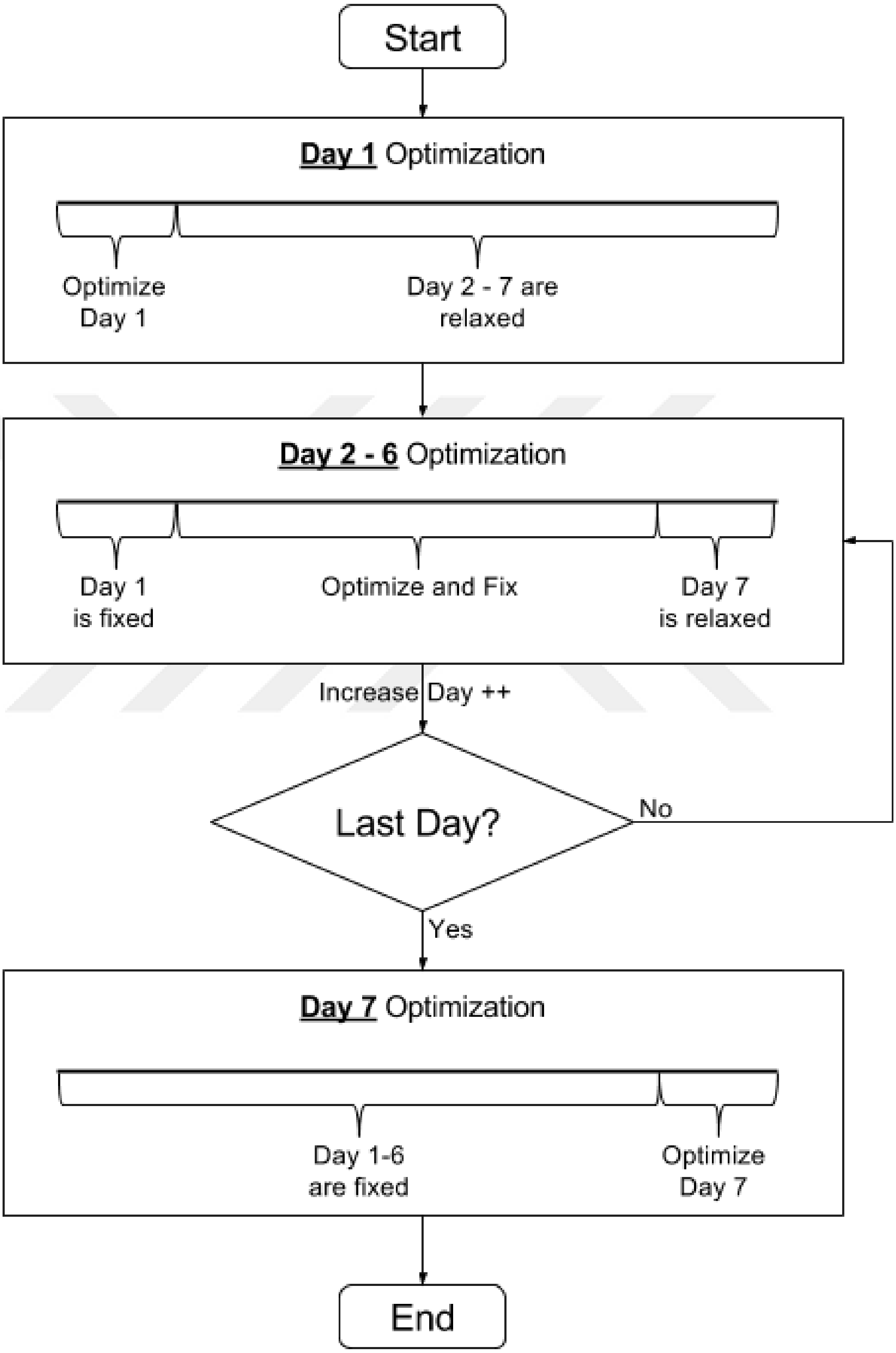


Figure 3.1 Sample F&R process with day decomposition

The F&R algorithm can be summarized in a pseudo-code as in Figure 3.2. Problem size and the optimization window are defined first. Then, the starting window is initiated. Until the end of the planning horizon, each window is optimized step-by-step. The last window is not optimized in this iterative procedure. Instead, it is optimized as the last step of the algorithm where the results are reported.

```
Algorithm: F&R
Start
    Initialize Problem Size $k_{max}$ and Optimization Window $i_{int}$;
    Initialize the starting point $i$;
    while $i + i_{int} < k_{max}$
        Set window start, $i_{start} = i$;
        Set window end, $i_{end} = i + i_{int}$;
        Optimize the window;
        $i = i + i_{int}$;
    end
    Solve the last window;
    Return solution;
End
```

Figure 3.2 Pseudo-code

3.3.2 The DPAS Problem Adaptation

One of the most significant differences between the DPAS problem and the PAS problem is that in the DPAS problem, planning must be done on admission day or the day before. This results in scheduling one day at a time. In the previous chapter, we decomposed the planning horizon into days and then decomposed patients within each day to obtain smaller, solvable sub-problems. In the DPAS problem, there is no need for a day decomposition because the problem is based on daily optimization. Therefore, we only utilize patient decomposition to obtain smaller problems.

To further elaborate on the F&R application, we introduce a toy instance of the problem in this section. Table 3.4 represents a set of patients, their admission and discharge dates, and their needs and preferences. Earlier versions of the PAS problem consider needed equipment as an SC, but the DPAS problem considers it as part of the medical treatment and therefore as an HC. If a patient requires certain medical equipment, the patient must be assigned to a room with that equipment. As seen in Table 3.4, patients are decomposed into groups. In our previous work, Turhan and Bilgen (2017), we provide a detailed explanation of patient decomposition. In this toy instance, we only report which group each patient belongs to. Group 1 is composed of patients who have a higher cost to the objective value due to their medical or personal needs. Group 2 patients are secondary in the algorithm because their costs are considerably lower than the first group.

Table 3.4 Patient details for the toy problem

Patient	Gender	Admission Date	Discharge Date	Max-Admission Date	Room Choice	Needed Equipment	Optional Equipment	Group
1	M	1	4	-	3	O2	TV	1
2	F	5	7	-	1	-	-	2
3	M	2	5	2	3	-	-	2
4	F	1	2	-	2	TM	O2, TV	2
5	M	5	9	-	1	O2, TM	TV	1
6	F	5	6	-	3	-	-	2
7	F	2	4	-	1	O2, TM	-	1
8	M	3	4	3	3	O2	-	2
9	F	1	4	1	1	O2, TM	TV	1
10	F	4	5	-	1	O2, TM	TV	2
11	M	1	5	-	3	O2	TV	1
12	F	5	10	-	2	TM	O2	2
13	M	4	5	-	1	-	-	1
14	M	3	4	3	3	TM	-	2
15	M	1	2	1	2	O2	TV	2

Table 3.5 lists the details of the rooms.

Table 3.5 Room details for the toy problem

Room	Capacity	Equipment
1	4	Oxygen (O2)
2	2	Oxygen, Telemetry (TM)
3	1	Oxygen, Telemetry, and TV

Unlike the PAS problem, the DPAS problem considers empty rooms and operating rooms for each day and penalizes for underutilization. If a room is left idle at the end of a day, it results in a penalty.

The procedure starts with Day 1 as shown in Figure 3.3. Patients 1, 9, and 11 are Group 1 patients, and are considered first. The F&R process assigns these three patients to appropriate rooms based on their HCs and SCs.

Patient 9 is an emergency patient and must be assigned to a room without delay; the patient is also admitted to the operating room for surgery, utilizing four slots of operating room time. In the next iteration, Group 2 patients are considered for optimization while Group 1 patients are fixed to their assigned rooms.

Patient 4 takes Room 2, and Patient 15 is assigned to Room 1. Patient 15 is also an emergency patient and needs same-day surgery, but requires a shorter procedure time than Patient 9.

In the toy instance emergency patients are indicated with their maximum admission dates. If these dates are equal to their admission dates, it is implied that the patients are emergency patients.

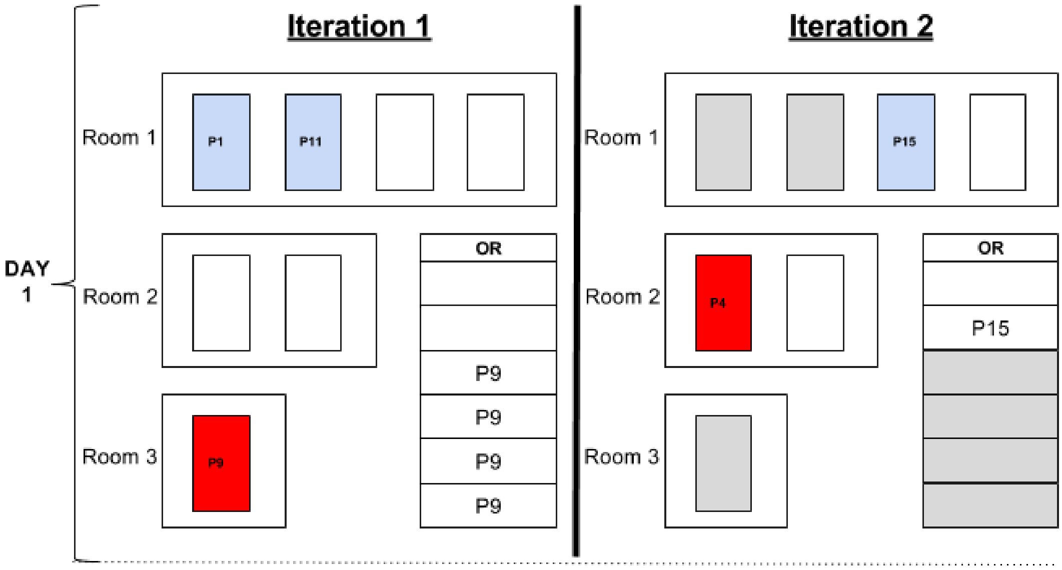


Figure 3.3 Iterations for the first day

Another difference between the PAS and DPAS problems in regards to the implementation of the F&R heuristic is that in the DPAS, the algorithm runs again for the next day and includes the previously optimized patients as represented in Figure 3.4.

Day 2 optimization starts with previously optimized and fixed patients who also stay on Day 2 as well as new patients such as Patients 3 and 7, the latter of which is the only addition to Group 1.

The algorithm not only evaluates the previous patients again, but also considers the cost of transferring patients to different rooms. For instance, removing Patients 1 and 11 from their Day 1 assignments would generate transfer costs. Therefore, they are placed in their original rooms. Because room transfer is not an HC, patients can be moved from their previously assigned rooms to better rooms.

After patients are optimized and fixed, Patient 3 is optimized in the second iteration of Day 2 and assigned to the operating room for a procedure.

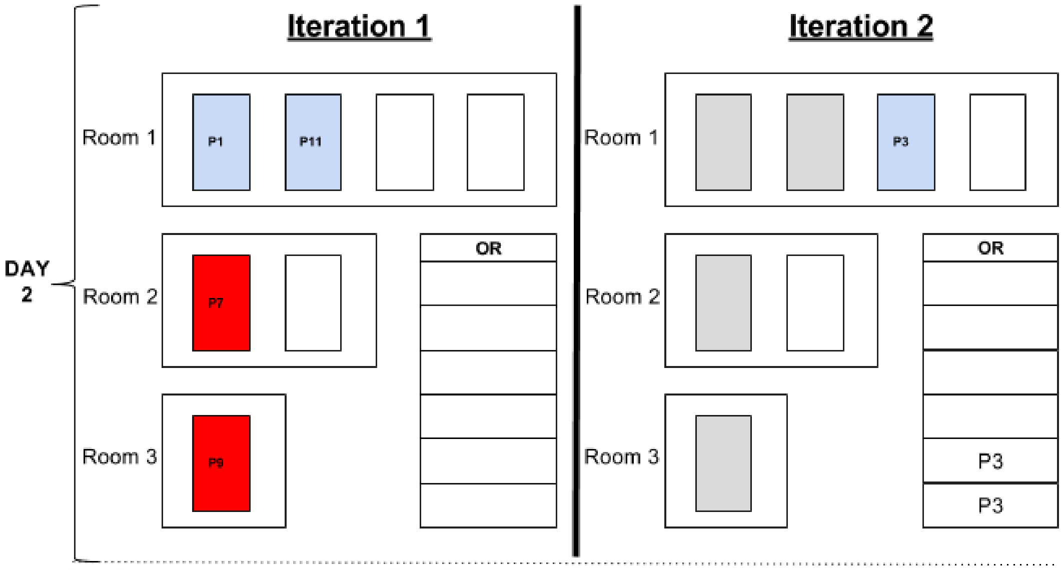


Figure 3.4 Iterations for the second day

In Day 3, a similar procedure is applied, and patients are optimized and fixed. Note that Room 2 must host two genders on the same day because the required medical equipment is not available in any other room on that day.

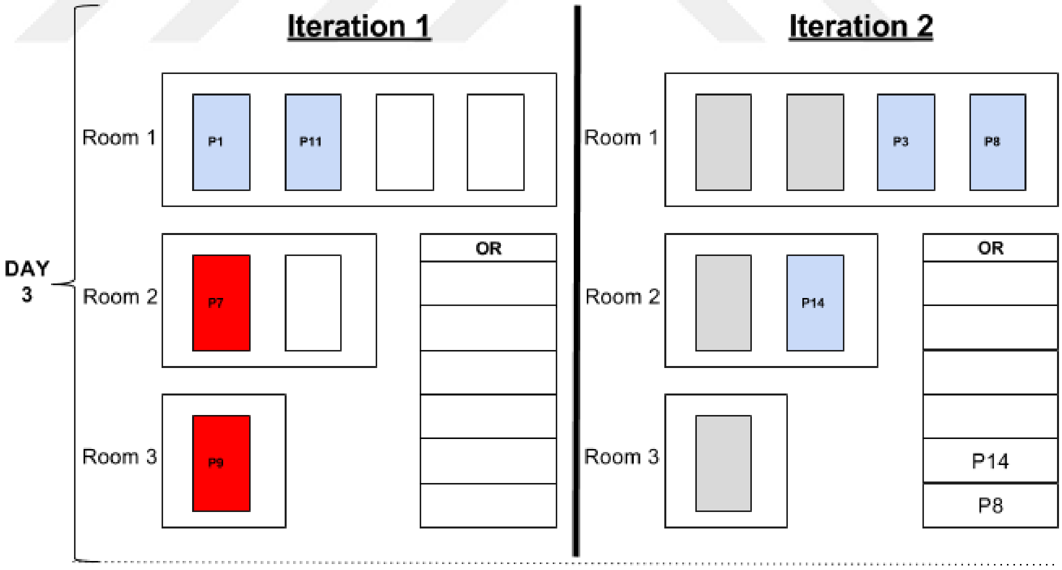
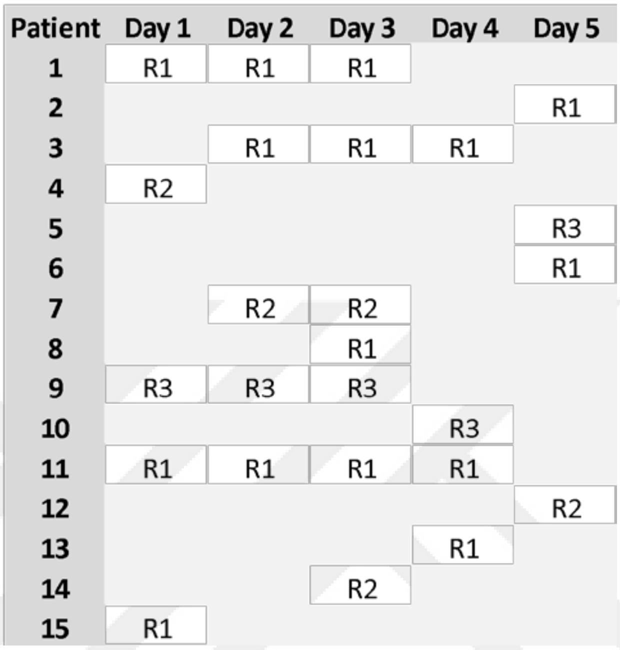


Figure 3.5 Iterations for day 3

The rest of the days are calculated via the same procedure until the entire planning horizon is optimized as depicted in Figure 3.5. The overall cost is calculated based on the optimized schedule of each day. Day 1 results in patient satisfaction related penalties as well as two beds and one operating room slot being idle, while a violation of the one-gender constraint occurs on Day 3.

Figure 3.6 represents the sample final schedule for the toy instance.



Patient	Day 1	Day 2	Day 3	Day 4	Day 5
1	R1	R1	R1		
2					R1
3		R1	R1	R1	
4	R2				
5					R3
6					R1
7		R2	R2		
8			R1		
9	R3	R3	R3		
10				R3	
11	R1	R1	R1	R1	
12					R2
13				R1	
14			R2		
15	R1				

Figure 3.6 Final schedule for the toy problem

3.4 Experimental Results

In this section, we report our computational results and compare them against results previously reported in the literature. Problem instances are publicly available on the DPAS website (Ceschia & Schaerf, 2017). The researchers also have another website (Ceschia & Schaerf, 2016b) for PASU where an earlier version of the dynamic PAS is studied. However, we prefer to experiment with the DPAS problem because the problem itself includes more real-world scenarios.

A summary of the main features of the DPAS instances is displayed in Table 3.6. There are 30 instances in total are grouped in six categories. The short group has a two-week planning horizon while the long group has a four-week horizon. Within each group, there are three subcategories. Category 1 has two departments, Category

2 has four, and Category 3 has six. Each subcategory has five instances with variable patient sizes.

Table 3.6 Main features of the problem instances on the DPAS website (Ceschia & Schaerf, 2017)

Group	**Rooms**	**Departments**	**Operating Rooms**	**Specialisms**	**Treatments**	**Patients**
Short1	25	2	2	9	15	391-439
Short2	50	4	4	18	25	574-644
Short3	75	6	5	23	35	821-925
Long1	25	2	2	9	15	693-762
Long2	50	4	4	18	25	1089-1169
Long3	75	6	5	23	35	1488-1602

Table 3.7 reports the average and best objective results from Ceschia and Schaerf (2016a).

Table 3.7 Previously reported values by Ceschia and Schaerf (2016a)

Data	**Group**	**Instance name**	**Avg**	**Dev**	**Med**	**Best**
1	Short1	or_pas_dept2_short00	84743.13	434.93	84697.00	83750.00
2	Short1	or_pas_dept2_short01	76421.17	672.82	76413.00	75185.00
3	Short1	or_pas_dept2_short02	85644.67	451.86	85614.00	84743.00
4	Short1	or_pas_dept2_short03	85578.17	694.82	85734.00	84041.00
5	Short1	or_pas_dept2_short04	80065.43	542.35	80110.00	78979.00
6	Short2	or_pas_dept4_short00	162431.33	736.19	162445.00	160908.00
7	Short2	or_pas_dept4_short01	131807.20	829.73	131874.00	130417.00
8	Short2	or_pas_dept4_short02	130755.77	794.25	130598.00	129365.00
9	Short2	or_pas_dept4_short03	132371.20	1066.50	132471.00	130044.00
10	Short2	or_pas_dept4_short04	118835.60	809.96	118975.00	116781.00
11	Short3	or_pas_dept6_short00	211538.80	1186.79	211655.00	209289.00
12	Short3	or_pas_dept6_short01	231615.80	1395.02	232004.00	229258.00
13	Short3	or_pas_dept6_short02	227800.30	1732.82	227598.00	223933.00
14	Short3	or_pas_dept6_short03	189188.20	1033.87	189196.00	186491.00
15	Short3	or_pas_dept6_short04	211278.67	906.75	211612.00	208996.00
16	Long1	or_pas_dept2_long00	175608.50	1778.51	175958.00	172398.00
17	Long1	or_pas_dept2_long01	185516.37	1351.85	185570.00	183134.00
18	Long1	or_pas_dept2_long02	162102.70	1237.24	162007.00	159459.00
19	Long1	or_pas_dept2_long03	141373.47	1221.96	141613.00	139098.00

Table 3.7 continues

Data	Group	Instance name	Avg	Dev	Med	Best
20	Long1	or_pas_dept2_long04	181073.43	861.35	181237.00	179153.00
21	Long2	or_pas_dept4_long00	229260.90	1605.24	229615.00	225738.00
22	Long2	or_pas_dept4_long01	256394.27	1802.09	256616.00	253143.00
23	Long2	or_pas_dept4_long02	283645.53	1917.39	283635.00	279354.00
24	Long2	or_pas_dept4_long03	390986.00	2366.04	391276.00	385649.00
25	Long2	or_pas_dept4_long04	303337.23	2711.70	303173.00	298670.00
26	Long3	or_pas_dept6_long00	521369.50	4009.90	521248.00	514810.00
27	Long3	or_pas_dept6_long01	551616.71	5718.19	553391.00	541950.00
28	Long3	or_pas_dept6_long02	455385.47	3625.85	455504.00	447008.00
29	Long3	or_pas_dept6_long03	470707.24	2898.82	470723.00	465630.00
30	Long3	or_pas_dept6_long04	340457.53	2140.44	340116.00	336353.00

In this chapter, the F&R heuristic procedure has been applied to the problem instances utilizing the previously discussed approaches. The software is written in Java and the sub-problems are optimized using the IBM ILOG CPLEX Optimization Studio 12.6 Application Programming Interface with standard parameters. The code is executed in the Java Runtime Environment 7 and run on a Windows 7 PC with Intel Core i3 2.27 GHz processor and 3 GB of RAM.

Table 3.8 summarizes the results of the computational studies. We report six new best-known results out of the thirty results in this study. Our solution procedure outperforms previously reported values by an average of 5 percent in Instances 1, 4, 24, 25, and 26. For the rest of the test data, we report mostly 40 percent faster processing times on average. The average gap to the best-known results is 15 percent. In some instances, the gap values are as low as 3 percent. The capacity of the hardware used in our study is lower than that used in Ceschia and Schaerf (2016a). Higher CPU speeds would lead to much better processing times.

Table 3.9 displays the comparable average time and cost values grouped by the instance classifications.

Short Group 1 not only has a very minor gap between the average costs, but also has a significant average processing time advantage. Our algorithm performs competitively in this group.

Short Group 2, in which there are four departments and considerably more patients, has higher gap values, but still has shorter processing times, with an average of 159 seconds. The gap value is higher in this group due to the performance of the F&R algorithm on Test Data 7 and 10. The algorithm performance is negatively affected when the number of patients potentially present on certain days is significantly high, resulting in higher room utilization and increasing the complexity of the overall sub-problems.

Lastly, Short Group 3 performs within an 8 percent gap in our study. Although the average processing time seems to be less than in Ceschia and Schaerf (2016a), this is due to Test Data 11. It is a known issue that combinatorial integer problems might be more challenging to CPLEX for getting good performance. There are performance improvement strategies already available on this matter for CPLEX, but because we prefer using the default CPLEX settings on all the test datasets, this specific instance takes more time than others.

As discussed earlier, the long group has a four-week planning period which leads to more patient involvement. For the first long group, the F&R heuristic achieves gap values within 12 percent of those in Ceschia and Schaerf (2016a) and within 28 percent faster processing time. Apart from Instance 16, the instances in this group perform well, with an average 7 percent gap. We report two new best results in the second long group which is consistent with the experimental studies in our previous study, Turhan and Bilgen (2017), where the MIP-based heuristics provide promising feasible solutions with shorter processing times. The procedure provides an even better solution in our current study. The last long group on average has a 6 percent gap from the best-known results within almost 40 percent faster processing times. We also report two new best results in this group. Test Data 26 and 27 have better cost values in shorter computational times.

Table 3.8 Comparison between Ceschia and Schaerf (2016) and our study

			Ceschia and Schaerf (2016a)		Our Study		
Data	**Group**	**Instance name**	**Time(s)**	**Best**	**Time(s)**	**Best**	**Gap**
1	Short1	or_pas_dept2_short00	123.51	83750	**45.74**	**77695**	**-2.59%**
2	Short1	or_pas_dept2_short01	127.45	75185	11.30	76889	2.27%
3	Short1	or_pas_dept2_short02	111.02	84743	21.16	62040	2.49%
4	Short1	or_pas_dept2_short03	127.58	84041	**39.08**	**75440**	**-5.75%**
5	Short1	or_pas_dept2_short04	120.32	78979	67.08	64709	6.51%
6	Short2	or_pas_dept4_short00	276.15	160908	240.95	156235	1.95%
7	Short2	or_pas_dept4_short01	288.07	130417	114.57	175674	34.70%
8	Short2	or_pas_dept4_short02	280.17	129365	123.66	149040	15.21%
9	Short2	or_pas_dept4_short03	263.48	130044	151.28	155265	19.39%
10	Short2	or_pas_dept4_short04	265.02	116781	166.82	151030	29.33%
11	Short3	or_pas_dept6_short00	476.31	209289	956.52	201029	0.86%
12	Short3	or_pas_dept6_short01	460.02	229258	498.37	221329	1.37%
13	Short3	or_pas_dept6_short02	465.21	223933	417.73	239159	6.80%
14	Short3	or_pas_dept6_short03	473.59	186491	245.12	227469	21.97%
15	Short3	or_pas_dept6_short04	451.99	208996	429.99	227455	8.83%
16	Long1	or_pas_dept2_long00	288.22	172398	65.99	227682	32.07%
17	Long1	or_pas_dept2_long01	280.06	183134	167.11	151530	7.57%
18	Long1	or_pas_dept2_long02	269.04	159459	66.79	167654	5.14%
19	Long1	or_pas_dept2_long03	279.93	139098	342.95	152764	9.82%
20	Long1	or_pas_dept2_long04	274.97	179153	349.83	146072	6.00%
21	Long2	or_pas_dept4_long00	584.43	225738	234.53	341690	51.37%
22	Long2	or_pas_dept4_long01	585.02	253143	216.92	276449	9.21%
23	Long2	or_pas_dept4_long02	611.60	279354	186.47	352709	26.26%
24	Long2	or_pas_dept4_long03	651.52	385649	**260.05**	**351189**	**-4.38%**
25	Long2	or_pas_dept4_long04	640.30	298670	**327.34**	**266884**	**-6.17%**
26	Long3	or_pas_dept6_long00	1004.86	514810	**754.63**	**463449**	**-5.48%**
27	Long3	or_pas_dept6_long01	1055.35	541950	**492.58**	**476409**	**-7.70%**
28	Long3	or_pas_dept6_long02	1037.30	447008	780.61	473249	5.87%
29	Long3	or_pas_dept6_long03	1146.85	465630	530.17	515924	10.80%
30	Long3	or_pas_dept6_long04	845.84	336353	649.65	431299	28.23%

Notes: Bolded data represents new best scores.

Table 3.9 Comparison of groups between Ceschia and Schaerf (2016a) and our study

		Ceschia and Schaerf (2016a)		Our Study		
Group	**Departments**	**Average Time(s)**	**Average Cost**	**Average Time(s)**	**Average Cost**	**Gap**
Short1	2	121.98	81339.60	36.87	82500.58	0.59%
Short2	4	274.58	133503.00	159.46	165321.24	20.12%
Short3	6	465.42	211593.40	509.55	234452.61	7.97%
Long1	2	278.44	166648.40	198.53	192477.52	12.12%
Long2	4	614.57	288510.80	245.06	333673.41	15.25%
Long3	6	1,018.04	461150.20	641.53	495669.30	6.35%

3.5 Conclusions and Future Studies

Patient scheduling is a critical activity in hospitals that affects many resources. Effective patient scheduling will lead to better allocation of scarce resources such as medical staff and operating rooms. The PAS problem formulated by Demeester et al. (2008) and the DPAS problem defined by Ceschia and Schaerf (2016a) are important steps in addressing this issue.

In this chapter, we report a full mathematical model of the DPAS problem which has previously been missing in the literature. Also we have proposed and implemented the F&R heuristic, a MIP-based procedure, in solving the problem.

The F&R heuristic is a good alternative to meta-heuristic solution approaches because it provides quick, feasible, and high-quality solutions. The F&R heuristic divides a combinatorial problem into smaller sub-problems and iteratively solves them to optimality. While different decomposition strategies can be employed, we choose to decompose the problem based on patients.

We report here six new best-known results using the publicly available data instances on the DPAS website (Ceschia & Schaerf, 2017). Overall, the F&R approach provides high-quality solutions within a 10 percent gap from the cost values reported in Ceschia and Schaerf (2016a) in 35 percent faster processing times on average. Most of the new best results we obtain are on the larger problems, findings which are also consistent with results in our previous work, Turhan and Bilgen (2017), where we see a strong performance of the MIP-based heuristics on larger instances.

The next steps in future studies include expanding the DPAS problem into new constraints, especially those related to medical staff, like the utilization of nurses and doctors.

CHAPTER FOUR
A HYBRID MIXED INTEGER PROGRAMMING BASED HEURISTICS AND SIMULATED ANNEALING APPROACH FOR SOLVING NURSE ROSTERING PROBLEMS

4.1 Introduction

In the previous chapters, two novel health care problems, the PAS and DPAS, are studied to support the operational level decision making. In this chapter, another important operational level OR application, the NRP, is introduced and a new solution approach is proposed.

Personnel scheduling problems have been studied by the OR community since the 1950s. Scheduling today is notably different than it was that time, as many new features have been introduced to the process (Bergh et al., 2013). For example, employee satisfaction has become an important part of the scheduling effort. Employees are offered part-time and full-time opportunities as well as flexible working hours. Their preferences (i.e. desired shifts, days-off, planned vacations) are taken into consideration when a work schedule is developed.

According to Baker (1976), personnel scheduling problems are generally classified in three groups: time-of-day scheduling, day-of-week scheduling, and a combination of both. In time-of-day scheduling, shift start and end times are scheduled in a daily planning horizon. Weekly planning horizon is used in day-of-week scheduling in which a facility’s operating week may not match employees’ working week. Typical scenario is when employees work five days a week while a facility operates seven days a week. In the last group, scheduling must be done on both a daily and weekly basis. Hospitals are an example of this combined group. They must operate at all times, and employees have a variety of shifts in a day.

In hospitals, nurses are one of many scarce resources. Hospitals produce work schedules that ensure nurses are on staff 24 hours a day, seven days a week. The quality of nurse schedules affects the quality of health care (Oldenkamp, 1996), but scheduling is generally done manually. Head nurses or hospital scheduling departments spend significant time constructing schedules that satisfy many constraints. Factors such as preset shifts, nurse requests, and last minute changes make the task tedious and time consuming (Cheang et al., 2003). Therefore, nurse scheduling has attracted the attention of the OR community, which has developed the NRP—also known as the Nurse Scheduling Problem (NSP)—to address these concerns.

The NRP generally has two types of constraints. HCs are those that must be satisfied in order to generate feasible rosters. SCs are not necessary conditions for feasible schedules, but violating them causes a penalty. For example, one HC is that a nurse cannot be assigned to a morning shift directly following a late shift because there must be rest time after a shift. An example of an SC would be a nurse asking to be assigned to a certain shift on a certain day. If this request is not fulfilled, a feasible roster can still be created, although it would result in a penalty because of the nurse's dissatisfaction.

In the literature, solution techniques for the NRP can be divided into three categories: exact methods, heuristics, and hybrid solutions. The exact methods are IP and constraint programming (CP). These methods are able to find optimal solutions, but computational times increase drastically as problem sizes increase. The heuristics category can provide high-quality solutions in faster processing times, but the solutions may not be optimal. Heuristics include many solution approaches such as VNS, TS, SA, GA, ant colony optimization (ACO), electromagnetic algorithm (EM), scatter search (SS), memetic algorithm (MA), tailor-made heuristics, estimation of distribution algorithms (EDA), and case-based reasoning (CBR). The final category of solutions techniques, hybrid solutions, is relatively a new area of study. It combines different solution techniques to achieve greater strength and flexibility.

The NRP has received a significant attention in the literature. For comprehensive surveys, the reader is referred to literature by Cheang et al. (2003), Burke et al. (2004), and Bergh et al. (2013).

Table 4.1 represents a comprehensive summary of studies in the NRP field in the last several decades.

Table 4.1 Summary of the NRP research

Study	Solution category/approach
Warner & Prawda (1972)	Exact/IP
Brusco & Jacobs (1993)	Heuristics/SA
Darmoni et al. (1995)	Exact/CP
Weil et al. (1995)	Exact/CP
Brusco & Jacobs (1995)	Heuristics/SA
Berrada & Ferland (1996)	Exact/IP
Beaumont (1997)	Exact/IP
Dowsland (1998)	Heuristics/TS
Burke et al. (1998)	Heuristics/TS
Easton & Mansour (1999)	Heuristics/GA
Dowsland & Thompson (2000)	Exact/IP
Meyer auf’m Hofe (2000)	Exact/CP
Aickelin & Dowsland (2000)	Heuristics/GA
Cai & Li (2000)	Heuristics/GA
Kawanaka et al. (2001)	Heuristics/GA
Burke et al. (2001)	Heuristics/MA
Ikegami & Niwa (2003)	Exact/IP
Bourdais et al. (2003)	Exact/CP
Burke et al. (2003)	Heuristics/VNS
Aickelin & Dowsland (2004)	Heuristics/GA
Bard & Purnomo (2005a)	Exact/IP
Azaiez & Al Sharif (2005)	Exact/IP
Cipriano et al. (2006)	Exact/CP
Gutjahr & Rauner (2007)	Heuristics/ACO
Maenhout & Vanhoucke (2007)	Heuristics/EM
Aickelin & Li (2007)	Heuristics/EDA
Beddoe & Petrovic (2007)	Heuristics/CBR

Table 4.1 continues

Study	**Solution category/approach**
Burke et al. (2008)	Heuristics/VNS
Qu & He (2008)	Hybrid/CP and VNS
Tsai & Li (2009)	Heuristics/GA
Glass & Knight (2010)	Exact/IP
Maenhout & Vanhoucke (2010)	Exact/IP
Burke et al. (2010a)	Hybrid/IP and VNS
Burke et al. (2010b)	Heuristics/SS
Girbea et al. (2011)	Exact/CP
Stølevik et al. (2011)	Heuristics/VNS
Bilgin et al. (2012b)	Heuristics/VNS
Lu & Hao (2012)	Heuristics/Tailor-made
Valouxis et al. (2012)	Heuristics/Tailor-made
M’Hallah & Alkhabbaz (2013)	Exact/IP
Soto et al. (2013)	Exact/CP
Curtois & Qu (2014)	Heuristics/EC
Della Croce & Salassa (2014)	Heuristics/VNS
Burke & Curtois (2014)	Exact/IP
Tassopoulos et al. (2015)	Heuristics/VNS
Awadallah et al. (2015)	Heuristics/ACO
Santos et al. (2016)	Exact/IP
Rahimian et al. (2017a)	Hybrid/IP and VNS
Rahimian et al. (2017b)	Hybrid/IP and CP
The Proposed Study	Hybrid/IP and SA

Two of these studies are particularly notable because they use similar NRP models and experiment on the same data sets as in our work.

Curtois and Qu (2014) use ejection chain (EC) and branch and price (B&P) algorithms, and the Gurobi optimizer to solve the test data. Their results indicate that the B&P method is effective on smaller instances but it is inadequate on larger test data, and it runs out of memory at times. Conversely, the EC metaheuristic finds good solutions on the larger data sets but is outperformed by the B&P method on smaller sets.

The second study experimenting on the same instances as our work is by Rahimian et al. (2017a), who hybridize the VNS algorithm using IP to solve the NRP. In the study, initial solutions are generated using a greedy heuristic and improved using the hybrid method. Five different neighborhoods are applied to improve schedules during the VNS phase and an IP based ruin-and-recreate framework is embedded into the process to further improve solutions. The study reports new best-known results and compares findings with studies in the literature. It also generates new test data and makes the instances publicly available for other researchers for benchmarking.

In this chapter, we propose a hybrid approach to solve the NRP. Our approach integrates MIP-based F&R and F&O heuristics with SA. In MIP-based heuristics, a problem is decomposed into a set of sub-problems, and then each sub-problem is optimized. This process continues iteratively until all the sub-problems are solved.

In our proposed NRP implementation framework, we use F&R heuristic as a starting point to find high-quality initial solutions. The NRP is decomposed based on the number of nurses and weeks. The initial solution obtained using the F&R heuristic is then used in the SA part of the framework. Many neighborhood structures are applied during the subsequent iterations to improve the initial solution. When solutions can no longer be improved, the F&O heuristic is injected into the process as a vehicle to diversify the search space. Low cost days are fixed to their existing values and other days are optimized. This interaction between the SA and F&O algorithms often results in better solutions and leads to intensification. Even when the F&O heuristic produces worse solutions, the diversification of the search space improves the performance of the SA algorithm and the neighborhoods.

Results are reported when a termination criterion is met in the SA algorithm. Finally, we use data instances previously reported in the literature to assess the performance of the proposed solution and compare our results to other solution techniques.

The main contributions of the chapter to the literature are two-fold: our study is the first implementation of the MIP-based heuristics to the NRP problem and computational results outperform many of the state-of-the-art solution techniques studied in the literature.

The remainder of this chapter is organized as follows. In Section 4.2, the description, notation, and the mathematical programming formulation of the NRP are defined. Section 4.3 describes the proposed algorithm and illustrates the implementation with an example. In Section 4.4, the computational experiments are presented and compared to results previous reported in the literature. Finally, in Section 4.5, conclusions are drawn and possible future research opportunities are discussed.

4.2 Problem Description

Before describing the problem in detail, some of the useful terms and essential definitions of the NRP literature are provided in the following paragraphs.

Nurse: Person who has completed a program of nursing education and is authorized to care for patients.

Shift: Time period during which nurses perform their duties. There can be multiple shifts in a day (i.e. early shift, day shift, night shift). Shifts have pre-defined start and end times.

Cover: Minimum number of nurses per shift needed to provide sufficient health care services.

Schedule: Work plan that defines assigned shifts for nurses for each day in a timetable.

Planning horizon: Length of one schedule. Each planning horizon is defined by its start and end dates.

Contract: Employment agreement between a nurse and the management of a hospital. Contracts are defined in part by laws and regulations. For example, there is a mandated minimum period of rest following a shift. Thus, hospitals cannot assign a day shift to a nurse directly after a night shift. The law also defines the maximum number of consecutive days that nurses can work without time off as well as their minimum amount of time off. Generally, nurses work five days consecutively and receive at least two days off. Contracts also specify the maximum and minimum number of hours nurses must work in a planning horizon.

Request: Appeal from a nurse that could results in a schedule change. For instance, a nurse may request to be assigned to a morning shift instead of a night shift on a given day.

Fixed assignment: Assignment that is pre-defined before a schedule is generated. Vacations are considered in this group.

The NRP automatically assigns nurses to shifts according to the cover requirements, contractual agreements, nurses' requests, and fixed assignments. Contract requirements and fixed assignments are generally considered HCs, and therefore must be met.

In the NRP, cover requirements and requests are SCs; violating them results in costs in the objective function of the problem. The quality of a schedule increases as the number of satisfied SCs increases.

A general formulation of the NRP problem studied in this chapter is originally provided by Curtois and Qu (2014). A detailed description is presented in this section to make the chapter self-contained and easy to follow.

4.2.1 Notation

Table 4.2 describes the notation of the model.

Table 4.2 Notation of the model

Sets	**Definition**
I	Set of nurses (i = 1, 2,…, I)
h	Number of days in the planning horizon
D	Set of days in the planning horizon = {1…h} (d = 1, 2,…, D)
W	Set of weekends in the planning horizon = {1...h/7} (w = 1, 2,…, W)
T	Set of shift types (t = 1, 2,…, T)
Rt	Set of shift types that cannot be assigned immediately after shift type t
Ni	Set of days that nurse i cannot be assigned a shift on
lt	Length of shift type t in minutes
s	Number of restricted days in a pattern
Parameters	**Definition**
m_{it}^{max}	Maximum number of shifts of type t that can be assigned to nurse i
b_{i}^{min}	Minimum number of minutes that nurse i must be assigned
b_{i}^{max}	Maximum number of minutes that nurse i can be assigned
c_{i}^{min}	Minimum number of consecutive shifts that nurse i must work
c_{i}^{max}	Maximum number of consecutive shifts that nurse i can work
o_{i}^{min}	Minimum number of consecutive days off nurse i can be assigned
a_{i}^{max}	Maximum number of weekends that nurse i can work
q_{idt}	Penalty if shift type t is not assigned to nurse i on day d
p_{idt}	Penalty if shift type t is assigned to nurse i on day d
u_{dt}	Preferred total number of nurses assigned shift type t on day d
v_{dt}^{min}	Penalty weight when below the preferred cover for shift type t on day d
v_{dt}^{max}	Penalty weight when exceeding the preferred cover for shift type t on day d
Decision Variables	**Definition**
x_{idt}	1 if nurse i is assigned shift type t on day d, 0 otherwise
k_{iw}	1 if nurse i works on weekend w, 0 otherwise
y_{dt}	Total below the preferred cover for shift type t on day d
z_{dt}	Total above the preferred cover for shift type t on day d
x_{idt}	1 if nurse i is assigned shift type t on day d, 0 otherwise

4.2.2 Constraints

The HCs of the mathematical model are as follows:

$$\sum_{t \in T} x_{idt} \leq 1, \qquad \forall i \in I, d \in D \tag{4.1}$$

$$x_{idt} + \, x_{i(d+1)u} \leq 1, \qquad \forall i \in I, d \in \{1 \dots h-1\}, t \in T, u \in R_t \tag{4.2}$$

$$\sum_{d \in D} x_{idt} \leq m_{it}^{max}, \qquad \forall i \in I, t \in T \tag{4.3}$$

$$b_i^{min} \leq \sum_{d \in D} \sum_{t \in T} l_t x_{idt} \, \leq b_i^{max}, \qquad \forall i \in I \tag{4.4}$$

$$\sum_{j=d}^{d+c_i^{max}} \sum_{t \in T} x_{ijt} \leq \, c_i^{max}, \qquad \forall i \in I, d \in \{1 \dots h - c_i^{max}\} \tag{4.5}$$

$$\sum_{t \in T} x_{idt} + \left(s - \sum_{j=d+1}^{d+s} \sum_{t \in T} x_{ijt} \right) + \sum_{t \in T} x_{i(d+s+1)t} > 0, \;\; \forall i \in I, s \in \left\{1 \dots c_i^{min} - 1\right\}, d \in \{1 \dots h - (s+1)\} \tag{4.6}$$

$$\left(1 - \sum_{t \in T} x_{idt} \right) + \sum_{j=d+1}^{d+s} \sum_{t \in T} x_{ijt} + \left(1 - \sum_{t \in T} x_{i(d+s+1)t} \right) > 0, \;\; \forall i \in I, s \in \left\{1 \dots o_i^{min} - 1\right\}, d \in \{1 \dots h - (s+1)\} \tag{4.7}$$

$$k_{iw} \;\leq \sum_{t \in T} x_{i(7w-1)t} + \sum_{t \in T} x_{i(7w)t} \;\leq\; 2k_{iw}, \qquad \forall i \in I, w \in W \tag{4.8}$$

$$\sum_{w \in W} k_{iw} \;\leq\; a_i^{max}, \qquad \forall i \in I \tag{4.9}$$

$$x_{idt} = 0, \qquad \forall i \in I, d \in N_i, t \in T \tag{4.10}$$

$$\sum_{i \in I} x_{idt} - z_{dt} + y_{dt} = u_{dt}, \qquad \forall d \in D, t \in T \tag{4.11}$$

Constraints (4.1) ensure that a maximum of one shift per day can be assigned to each nurse. Because of minimum rest-time regulations, certain shifts cannot follow others. As mentioned earlier, an early shift assignment cannot follow a late shift. This requirement is satisfied by Constraints (4.2).

Constraints (4.3) limit the number of shift assignments. The total number of assignments of a shift in the entire planning horizon must be less than or equal to its maximum allowed limit. Constraints (4.4) control total minutes worked in a planning horizon, which must be within the thresholds defined in the contracts. There can be differences in time limits based on nurses' employment statuses. For example, part-time nurses do not work as much as full-time nurses.

Constraints (4.5) enforce the maximum number of consecutive shifts a nurse can be assigned. When the limit is reached, a day off must be granted for the next day. Hospitals also want to ensure that the minimum number of consecutive work shifts is achieved. Therefore, when nurses are assigned to shifts, they should be scheduled for successive days. Constraints (4.6) ensure that the total number of work days is greater than this pre-defined requirement. These constraints are modeled by preventing every pattern below the minimum provision. For example, if a nurse must work at least three days before taking a day off, "off-on-off" and "off-on-on-off" patterns cannot be allowed. The same approach is utilized for modeling Constraints (4.7), which require that each nurse receives the minimum number of days off after working consecutively. If a nurse must take off for a minimum of three days, these constraints will prevent "on-off-on" and "on-off-off-on" patterns.

Because hospitals operate continuously, having adequate staff on weekends is a critical consideration. Constraints (4.8) and (4.9) ensure the maximum number of weekends nurses work to be less than the limit defined in their contracts. Any Saturday or Sunday shift is defined weekend work. Vacations are considered in Constraints (4.10). A nurse cannot be assigned a shift while on vacation.

Finally, Constraints (4.11) relate y and z variables to x to identify the gap between the number of nurses assigned and daily cover requirements.

4.2.3 Objective Function

$$\sum_{i\in I}\sum_{d\in D}\sum_{t\in T} q_{idt}(1-x_{idt}) + \sum_{i\in I}\sum_{d\in D}\sum_{t\in T} p_{idt}x_{idt} + \sum_{d\in D}\sum_{t\in T} v_{dt}^{min}y_{dt} + \sum_{d\in D}\sum_{t\in T} v_{dt}^{max}z_{dt} \qquad (4.12)$$

The first two parts of the objective function are considered under SC1. The first expression of the objective function deals with nurses' shift-on requests. The second one addresses shift off requests. SC1 considers scenarios in which a nurse asks to be assigned to or excused from a certain shift on a certain day.

The third part of the objective function (SC2) penalizes the number of staff below the needed number of nurses for a day and for a shift.

The last part (SC3) accounts for over staffing for a given shift on a given day. While overstaffing (SC3) is not an optimal situation, understaffing (SC2) causes more significant problems in health care coverage. Therefore, the cost of understaffing is weighted more than any other cost in the scheduling process.

Table 4.3 presents the weights of the SCs.

Table 4.3 Weights of the constraints

Constraint	Description	Weight
SC1	Shift on/off requests	1 – 3
SC2	Understaffing	100
SC3	Overstaffing	1

4.3 Solution Methodology

This chapter proposes a hybrid solution methodology to solve the NRP. It is a combination of the MIP-based F&R and F&O heuristics and SA. As stated earlier, the F&R heuristic, which is first introduced by Dillenberger et al. (1994), finds quick, high-quality initial solutions. During this algorithm, the problem is decomposed into a set of smaller sub-problems. The method of decomposition is based on the size of the problem. For smaller problem sizes, we choose to decompose by weeks (WD). For the larger instances, we decompose by nurses (ND). Table 4.4 provides the notation used to describe the algorithms.

Table 4.4 Notation of the algorithms

Notation	Definition
P	Set of sub-problems
P_{WD}, P_{DD}, P_{ND}	Set of sub-problems with week, day, and nurse decomposition
x_{idt}^{O}	Set of x variables that are optimized
x_{idt}^{T}	Set of x variables that are to be optimized in future iterations
x_{idt}^{F}	Set of x variables that are fixed
x_{idt}^{R}	Set of x variables that are relaxed continuously
S	Solution
S’	Current solution
S*	Best solution
T	Temperature
T_0	Initial temperature
T_{min}	Final temperature
β	Cooling rate
N_k	Neighborhood k
I_{max}	Number of maximum iterations
J_{max}	Number of maximum iterations allowed without an improvement

A pseudo-code of the F&R algorithm using week decomposition is described in Figure 4.1. The algorithm first uses the Decompose() function to obtain a set of sub-problems based on a chosen decomposition input.

The RelaxVariables() block relaxes entire x variables continuously. For every sub-problem in the set, the algorithm first clears all the fixations and relaxations for the variables ready to be optimized. Then, the problem is solved with an IP solver.

If a feasible solution is found, optimized variables are fixed before the next iteration starts in the Update() function. This iterative process continues until all the sub-problems in the set are solved, and a final feasible solution is reported.

```
Input: Problem instance, week decomposition
Output: Status, Feasible solution if found
$P_{WD}$ ← Decompose(P);
RelaxVariables($x_{idt}$);
foreach p in $P_{WD}$ do
    Clear($x_{idt}^{T}$);
    (status, solution) ← Solve(p);
    if status == notFeasible then
        return(status, 0);
    end if
    $x_{idt}^{F}$ ← Update($x_{idt}^{O}$);
end
return(Feasible, solution);
```

Figure 4.1 Pseudo-code of the F&R heuristic

When the final solution is reported from the F&R algorithm, it is treated as an initial solution and is passed to the next part of the algorithm, the SA metaheuristic algorithm, which originates in the physical annealing process (Kirkpatrick et al., 1983; Černý, 1985). It has many variants, but the version used in the second part of the hybrid framework uses probabilistic acceptance and geometric cooling. In this

SA algorithm, SA parameters are put in place and the starting solution-related parameters are set based on the solution received from the F&R algorithm.

At each temperature level, a new solution is generated using different neighborhood structures which are defined by the Generate() function. Neighborhoods are selected randomly. Then, the cost of the new feasible solution is calculated. The new cost is immediately accepted if it is lower than the cost of the current schedule. The new cost is accepted in a probabilistic manner even if it is higher than the current cost. This property of the SA algorithm enables the search to expand to global optima.

When a certain criteria is met or there is no improvement at a temperature level, the current schedule is forwarded to the F&O heuristic of the hybrid algorithm by the FixAndOptimize() block.

There are couples of benefits of the F&O algorithm in this context which are worth to mention. First of all, the algorithm works as a way to intensify the structure when it finds a better solution. Secondly, and more importantly, the heuristic helps for diversification of the search structure even when no better solution is found due to the parameters of the IP solver, such as reaching the maximum time limit.

In any case, diversifying the search space significantly impacts solution quality.

Figure 4.2 provides a pseudo-code of the SA algorithm. It also details how the SA and F&O algorithms are connected.

```
Input: Initial solution
Output: Final solution
Initialize $T_0$, $T_{min}$, β, I;
T = $T_0$; S* = S; j = 0;
while $T_{min}$ < T do
    i = 0;
    while i < $I_{max}$ do
        k = Random();
        S’ = Generate(S, $N_k$);
        if Cost(S’) < Cost(S) then
            S = S’;
       else
           Δ = Cost(S’) – Cost(S);
           if Random(0,1) < exp(-1*(Δ/T)) then
               S = S’;
           end if
       end if
       if S’ < S* then
           S* = S’;
       else
           j = j + 1;
       end if
       i = i + 1;
       if j < $J_{max}$
           S = FixAndOptimize(S);
           j = 0;
       end if
    end while
    T = T * β;
end while
return S*;
```

Figure 4.2 Pseudo-code of the SA algorithm

The final part of the hybrid algorithm is the F&O heuristic, which is discussed in several studies (Gintner et al., 2005; Pochet & Wolsey, 2006; Sahling et al., 2009; Helber & Sahling, 2010).

Unlike the F&R heuristic, the F&O is an improvement heuristic. An initial solution is provided to the algorithm with a rule of decomposition, and the F&O heuristic improves the initial solution.

In the hybrid implementation, the day decomposition is utilized. First, low-cost days of the solution received from the SA algorithm are identified and fixed to their current values.

The rest of the days in the planning horizon are randomly selected. However, the selection is not entirely random: it accounts for the costs of days by using weighted averages. For example, if the cost of a Monday is four times higher than the cost of a Tuesday for a certain week, the random number generator is more likely to pick Monday.

When the cost of a new solution is lower than that of the initial solution provided to the algorithm, it is accepted and the optimized variables are added to the set of fixed variables.

If the new solution is not feasible, the Restore() function brings the previous feasible decision variables back into effect.

Figure 4.3 displays a pseudo-code of the F&O heuristic implemented to solve the NRP.

```
Input: Initial solution, day decomposition
Output: Final solution
S* = S;
$P_{DD}$ ← Decompose(P);
$x^{F}_{idt}$← Update($x_{idt}$);
foreach p in $P_{DD}$ do
    Clear($x^{T}_{idt}$);
    (status, S’) ← Solve(p);
    if status == feasible then
        if Cost(S’) < Cost(S) then
            S* = S’;
            $x^{F}_{idt}$← Update($x^{O}_{idt}$);
        end if
    else
        Restore($x^{T}_{idt}$);
    end if
end
return S*;
```

Figure 4.3 Pseudo-code of the F&O heuristic

The flow of the hybrid algorithm is summarized in Figure 4.4, which shows the relationships and connection points between each technique.

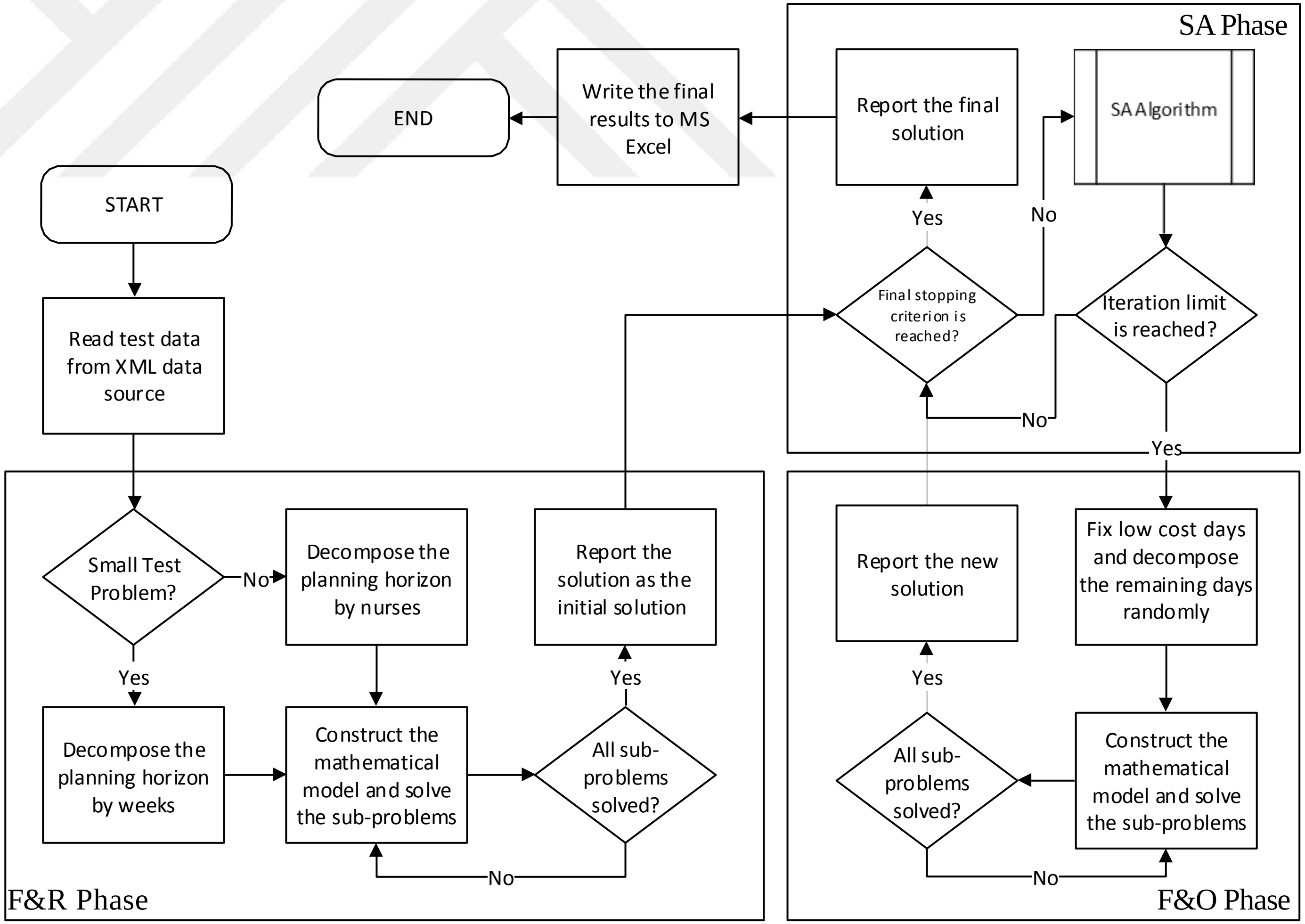

SA Phase
END
Write the final results to MS Excel
Report the final solution
SA Algorithm
START
Yes
No
Final stopping criterion is reached?
Iteration limit is reached?
Read test data from XML data source
No
Yes
Small Test Problem?
No
Decompose the planning horizon by nurses
Report the solution as the initial solution
Report the new solution
Fix low cost days and decompose the remaining days randomly
Yes
Yes
Yes
Decompose the planning horizon by weeks
Construct the mathematical model and solve the sub-problems
All sub-problems solved?
All sub-problems solved?
Construct the mathematical model and solve the sub-problems
No
No
F&R Phase
F&O Phase

Figure 4.4 The overall flow of the hybrid algorithm

4.3.1 Neighborhoods

In this sub-section, we describe the neighborhoods used during the neighborhood generation phase of the SA algorithm (Generate()). These neighborhoods are commonly used in the literature (Burke et al., 2010a; Stølevik et al., 2011; Rahimian et al., 2017a), apart from the last neighborhood, which is a new structure proposed in this thesis.

2-Exchange: Shifts between two randomly selected nurses are exchanged on a randomly selected day

3-Exchange: Shifts between three randomly selected nurses are exchanged on a randomly selected day

Double-Exchange: Shifts between two randomly selected nurses are exchanged on two randomly selected consecutive days

Multi-Exchange: Shifts between two randomly selected nurses are exchanged on three to six randomly selected days. Days are not necessarily consecutive.

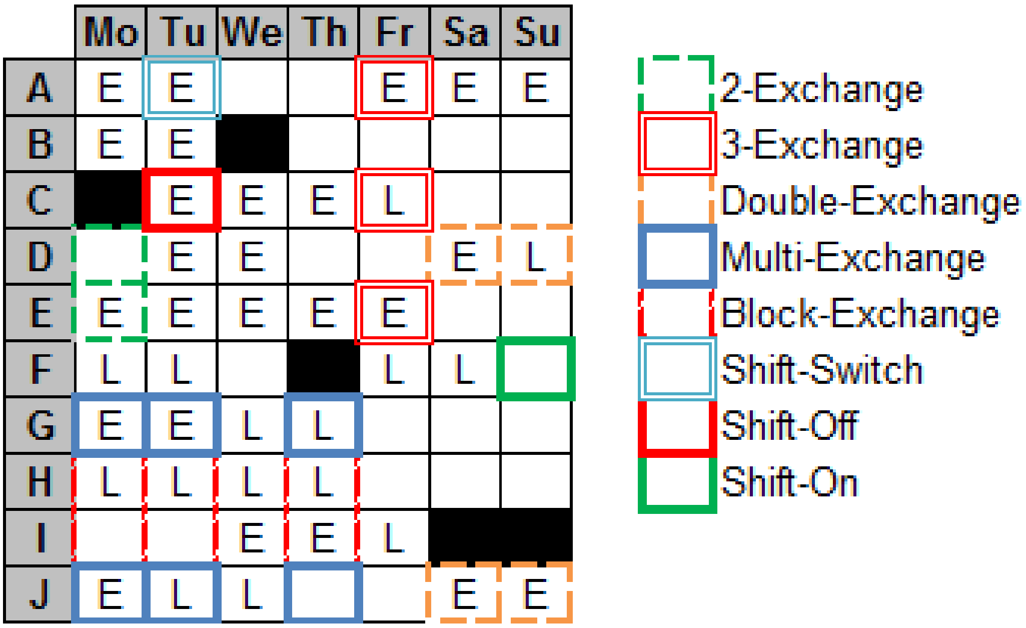
Block-Exchange: Shifts between two randomly selected nurses are exchanged on three to six randomly selected consecutive days.

Shift-Switch: An existing shift of a randomly selected nurse on a randomly selected day is switched to another shift.

Shift-Off: An existing shift of a randomly selected nurse on a randomly selected day is removed.

Shift-On: A new shift is assigned to a free slot of a randomly selected nurse on a randomly selected day.

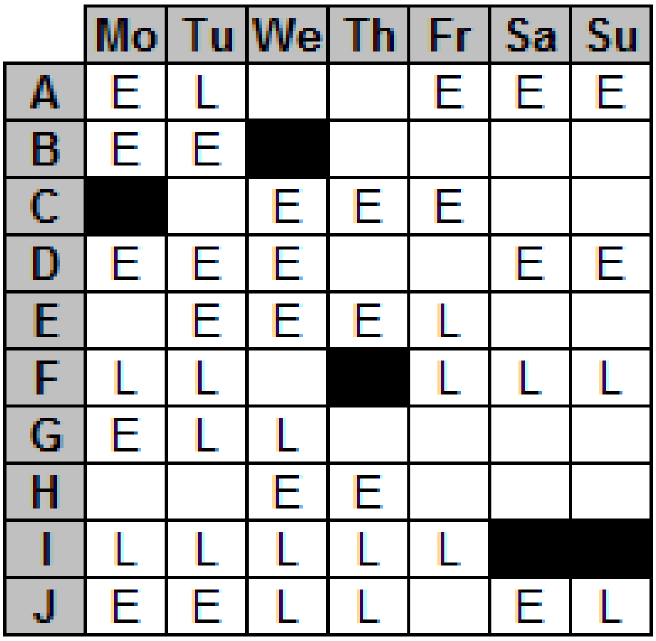
Figure 4.5 gives examples of various neighborhood applications. For instance, Nurses A, C, and E exchange their shifts on Friday as part of the 3-Exchange neighborhood. Nurses G and J multi-exchange their shifts on Monday, Tuesday, and Thursday. While Nurse A switches from shift E to shift L on Tuesday, Nurse C receives a shift off on the same day. Although Nurse F did not have any shifts assigned on Sunday initially, the nurse is assigned to shift L as part of the Shift-On neighborhood.



	Mo	Tu	We	Th	Fr	Sa	Su
A	E	E			E	E	E
B	E	E	■				
C	■	E	E	E	L		
D		E	E			E	L
E	E	E	E	E	E		
F	L	L		■	L	L	
G	E	E	L	L			
H	L	L	L	L			
I			E	E	L	■	■
J	E	L	L			E	E

Figure 4.5 An illustration of the neighborhoods

The resulting schedule after the applications of the neighborhoods described above is shown in Figure 4.6.



	Mo	Tu	We	Th	Fr	Sa	Su
A	E	L			E	E	E
B	E	E	■				
C	■		E	E	E		
D	E	E	E			E	E
E		E	E	E	L		
F	L	L		■	L	L	L
G	E	L	L				
H			E	E			
I	L	L	L	L	L	■	■
J	E	E	L	L		E	L

Figure 4.6 The new schedule after the neighborhood application

4.4 Computational Results

The proposed algorithm is tested on the data instances that have recently been introduced by Curtois and Qu (2014). Extensive computational experiments have been carried out on this data. The instances are of different sizes and complexities, which allows a more comprehensive assessment of the efficiency and strength of the proposed solution.

Table 4.5 summarizes the primary characteristics of the instances. There is a direct correlation between the complexity of the data and number of nurse requests.

The planning horizon on instances 20–24 are for six months or a full year. They are not included in our experiments because the data is not practical in real world situations. In reality, the planning horizon of nurse schedules is generally set to be bi-weekly or monthly.

Table 4.5 Summary of the test data (Curtois & Qu, 2014)

Instance	Days	Nurses	Shifts	Day-off Requests	Shift-on/off Requests
Instance 01	14	8	1	8	26
Instance 02	14	14	2	14	62
Instance 03	14	20	3	20	64
Instance 04	28	10	2	20	71
Instance 05	28	16	2	32	106
Instance 06	28	18	3	36	135
Instance 07	28	20	3	40	168
Instance 08	28	30	4	60	225
Instance 09	28	36	4	72	232
Instance 10	28	40	5	80	284
Instance 11	28	50	6	100	336
Instance 12	28	60	10	120	422
Instance 13	28	120	18	240	841
Instance 14	42	32	4	128	359
Instance 15	42	45	6	180	490
Instance 16	56	20	3	120	280
Instance17	56	32	4	160	480
Instance18	84	22	3	176	414
Instance19	84	40	5	320	834
Instance20	182	50	6	900	2318
Instance21	182	100	8	1800	4702
Instance22	364	50	10	1800	4638
Instance23	364	100	16	3600	9410
Instance24	364	150	32	5400	13,809

The software of the proposed hybrid algorithm is written in Java and the sub-problems are optimized using the IBM ILOG CPLEX Optimization Studio 12.6 Application Programming Interface with default parameters. The code is executed in the Java Runtime Environment 7 and run on a Windows 7 PC with an Intel Core i3 2.27 GHz processor and 3 GB of RAM.

The parameters of the SA algorithm are set to more-common values in the literature to determine the hybrid algorithm’s generic performance in solving scheduling problems. Therefore, the starting temperature is set to 10.0, the cooling rate to 0.99, and the stopping temperature to 0.00001 for all the instances.

Computational results for a 10-minute runtime are presented in Table 4.6. The initial solution column shows the results of the F&R heuristic phase, and the Final solution column shows the results of the hybrid method.

The overall 10-minute runtime is divided into two parts to achieve the best outcome of the hybrid approach. Thus, 10 percent of the runtime (60 seconds) is allocated to the initial solution phase and the rest of the time is used by the SA and F&O algorithms.

The algorithm runtime allocation is controlled by the IP solver by limiting optimization of each run. For example, in order to comply with the 60 seconds time limit to the F&R algorithm, each week has to be optimized in 15 seconds if the planning horizon is four weeks. The algorithm terminates when it reaches the iteration time limit.

The performance of the hybrid algorithm on Instance 8 is shown in Figure 4.7. This instance neither is too small nor too large and it has a good representation of the improvement progress of the solution technique. The F&R heuristic provides an initial solution of 4208 in 60 seconds. Then, objective values are reported every 30 seconds until the runtime limit. Lines representing sharp cost declines on the graph are generally as a results of the F&O heuristic. The SA algorithm often results in continuous and steady improvements.

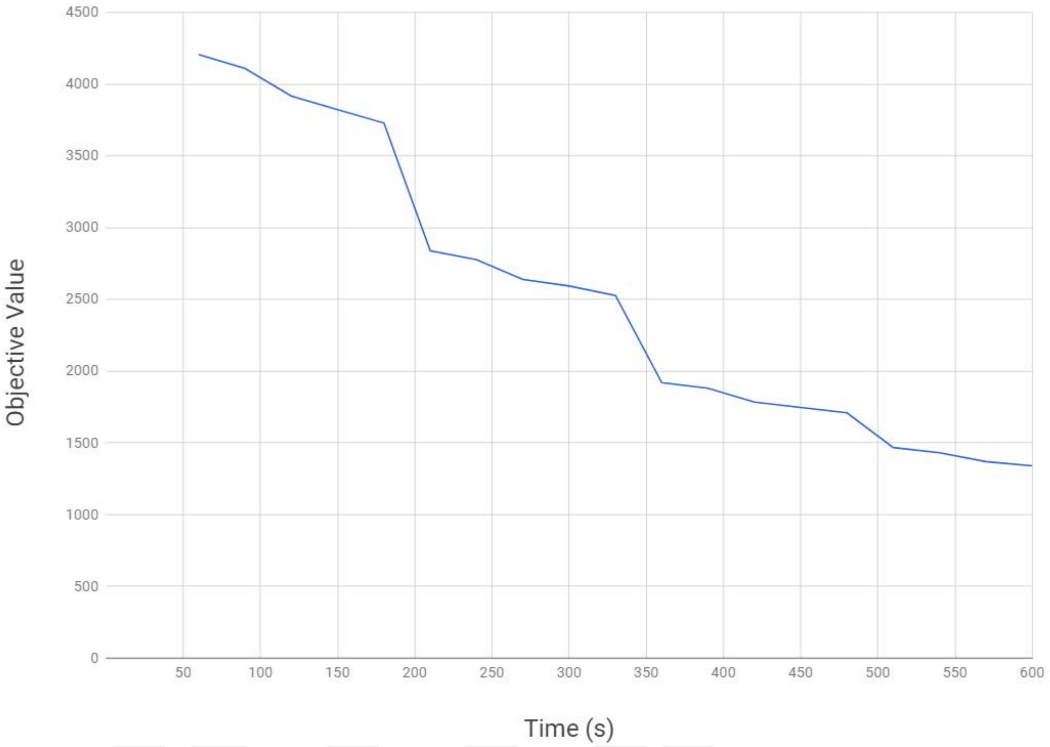


Figure 4.7 Performance of the hybrid algorithm on instance 8 over time

Gap values as shown in Table 4.6 are relatively smaller for Instances 1–8, where week decomposition is mainly used in the F&R heuristic. This enables the SA part of the algorithm to start from a very high-quality initial solution. Thus, it quickly reaches optimality. As problem sizes increase, the week decomposition often fails to provide a feasible solution. Instances 9–19 are more complex and use large data sets and planning horizons range from four weeks to three months. Applying the week decomposition is not the best choice in this case. First of all, week decomposition requires the optimization to start from the first week and to iteratively solve the rest of the weeks. This would result in a biased calculation toward the first several weeks and reduce the quality of the remaining weeks in the planning horizon. As the algorithm proceeds to the SA block, the last part of the horizon is selected more than the first part because the random selection accounts for the cost of the days in the schedule as discussed earlier. Using nurse decomposition for these larger problems remedies this issue. Assigning nurses by preferences provides more-randomly distributed schedules because the main cost component for a schedule is the violation of cover needs. As a result of choosing the nurse decomposition for these instances, the gap values from the Final Solution increase and they range from 80 to 95 percent.

Table 4.6 Computational results on 10-minute runtime

Instance	Initial Solution	% Gap	Final Solution
Instance 01	706	14.02	607
Instance 02	1226	32.46	828
Instance 03	1208	17.14	1,001
Instance 04	1931	11.13	1,716
Instance 05	1652	30.81	1,143
Instance 06	3267	40.31	1,950
Instance 07	2070	48.99	1,056
Instance 08	4208	68.13	1,341
Instance 09	2750	84.04	439
Instance 10	22066	79.01	4,631
Instance 11	27188	87.34	3,443
Instance 12	26960	85.00	4,044
Instance 13	35560	91.00	3,200
Instance 14	23988	94.60	1,295
Instance 15	34988	87.37	4,420
Instance 16	15809	79.42	3,253
Instance 17	34388	82.15	6,138
Instance 18	28731	82.60	5,000
Instance 19	49661	92.33	3,809

Table 4.7 presents a comparison of the studies in the literature applied to the instances to date. Curtois and Qu (2014) report results of an ejection chain heuristic, Gurobi, and B&P. Rahmian et al., 2017 publish findings of an IP&VNS hybrid.

The comparison of the algorithms is based on a time limit on the central processing unit (CPU). Experiments are completed in 10–60 minute runtimes. New best-known results reported in the study are marked in bold print. In general, the proposed hybrid is very efficient. It achieves optimality for smaller problems and performs better than the EC, Gurobi, and B&P optimizations in most cases. The proposed solution also outperforms the powerful and efficient IP&VNS Hybrid in seven test instances.

Table 4.7 Comparison between the studies in the literature

	10-minute runtime				**60-minute runtime**				
	IP&VNS Hybrid (Rahimian et al., 2017a)	**Ejection Chain (Curtois & Qu, 2014)**	**Gurobi (Curtois & Qu, 2014)**	**Our Study**	**IP&VNS Hybrid (Rahimian et al., 2017a)**	**Ejection Chain (Curtois & Qu, 2014)**	**Gurobi (Curtois & Qu, 2014)**	**B&P (Curtois & Qu, 2014)**	**Our Study**
Instance 01	607	607	607	607	607	607	607	607	607
Instance 02	828	923	828	828	828	837	828	828	828
Instance 03	1,001	1,003	1,001	1,001	1,001	1,003	1,001	1,001	1,001
Instance 04	1,716	1,719	1,716	1,716	1,716	1,718	1,716	1,716	1,716
Instance 05	1,143	1,439	1,143	1,143	1,143	1,358	1,143	1,160	1,143
Instance 06	1,950	2,344	1,950	1,950	1,950	2,258	1,950	1,952	1,950
Instance 07	1,056	1,284	1,056	1,056	1,056	1,269	1,056	1,058	1,056
Instance 08	1,364	2,529	8,995	**1,341**	1,344	2,260	1,323	1,308	**1,322**
Instance 09	439	474	439	439	439	463	439	439	439
Instance 10	4,631	4,999	4,631	4,631	4,631	4,797	4,631	4,631	4,631
Instance 11	3,443	3,967	3,443	3,443	3,443	3,661	3,443	3,443	3,443
Instance 12	4,042	5,611	4,045	4,044	4,040	5,211	4,040	4,046	4,040
Instance 13	3,109	8,707	500,410	3,200	1,905	3,037	3,109	–	2,900
Instance 14	1,281	2,542	1,482	1,295	1,279	1,847	1,280	–	1,280
Instance 15	4,144	6,049	78,144	4,420	3,928	5,935	4,964	–	4,190
Instance 16	3,306	4,343	3,521	**3,253**	3,225	4,048	3,233	3,323	3,225
Instance 17	5,760	7,835	6,149	6,138	5,750	7,835	5,851	–	5,848
Instance 18	5,049	6,404	7,950	**5,000**	4,662	6,404	4,760	–	**4,650**
Instance 19	3,974	6,522	29,968	**3,809**	3,224	5,531	5,420	–	**3,218**

The solutions of the hybrid algorithm presented in the study are also available along with additional NRP materials on the following website https://github.com/ORProjects/NRP.

4.5 Conclusion

The efficient usage of scarce resources, especially nurses, in health care is an extremely important task. Yet many scheduling departments and head nurses still create schedules manually. Automated scheduling solutions will improve the utilization of and fairness among nurses. The NRP problem addresses this critical need by automating the assignment of nurses to shifts and days according to hospital needs and nurses’ preferences.

In this chapter, we propose a hybrid solution for the NRP by integrating MIP-based heuristics and SA. The F&R heuristic generates an initial solution utilizing nurse or week decompositions and feeds it to the SA algorithm, which then applies a variety of neighborhoods to improve the solution. When a solution can no longer be improved, it is routed to another MIP-based heuristic. The F&O heuristic freezes low-cost days and optimizes only the remaining days.

Integrating the F&O algorithm into the SA algorithm has several benefits. First of all, the IP solver generally provides better solutions, and therefore intensifies the search. Sometimes, the F&O may not provide better solutions due to a time limit or other parameters, but it still diversifies the search space, which leads to better performance of the neighborhoods in the SA algorithm.

This hybrid framework is tested using the publicly available problems. Seven new best-known results are reported, and the results are compared to the state-of-the-art solution techniques in the literature. The algorithm outperforms most of the solution techniques.

Extending the NRP model with couple of real-world scenarios such as incorporations of skills could be a promising future research direction. Another interesting experiment could be to investigate the performance of mat-heuristics on the NRP instances as this technique is gaining a momentum. Additionally, testing the proposed hybrid framework on other personnel scheduling problems such as high school timetabling could yield valuable results.

CHAPTER FIVE
A MAT-HEURISTIC BASED SOLUTION APPROACH FOR AN EXTENDED NURSE ROSTERING PROBLEM WITH SKILLS AND DEPARTMENTS

5.1 Introduction

The NRP problem is introduced in the previous chapter and a new solution methodology is proposed. This chapter aims to extend the previously introduced model characteristics and introduce new constraints and objectives to the model. Moreover, a new solution methodology is also proposed for the solution of the extended version.

When work assignments are made, skills must always be a part of the decision making due to the fact that not all employees can cover every single task. In health care scheduling problems for nurse workloads, experiences, capabilities, and skills of nurses are not formed directly in mathematical models. Rather, titles or grades are used to represent nurses' experiences. The grading system which is generally built upon education and experience is set by national standards (Beddoe et al., 2009). This results in a classification of skills hierarchically. For instance, in a typical health care setting, the following types of titles are seen among nurses: head nurse, regular nurse, junior nurse, ambulance driver, caretaker, cleaner, etc. (Berghe, 2002; Smet et al., 2014). As the result, skills of a nurse who has a higher seniority in the organizational hierarchy can cover for nurses with lower job titles.

When it comes to the personnel scheduling literature, there are fair amount of studies considering skill incorporations to staffing models. In general, skills are incorporated to these staffing models in two ways. Most of the studies consider skills as HCs and therefore skill violations are not permitted. The other alternative of incorporating skills is to consider them as SCs and penalize. In this case, skills are part of objective functions. The biggest advantage of the skill consideration in the objective functions is that these models better represent the real-world situations because of their flexibility. No need to say that this increases the complexity of the scheduling models. We refer the reader to excellent surveys and reviews on the general workforce scheduling with skill incorporations provided by Bruecker et al. (2015). The review paper covers across a variety of industries. There are also studies like Ağralı et al. (2017) which consider skills in service industries.

The review papers on the NRP considering skills are limited. Designing NRP models with skills on both sides of the constraints has recently received more attention. To the best of our knowledge there are only a few studies in the literature on NRP including skill characteristics modeled as SC and HC at the same time. Most of the previous studies directly address skills as HCs and do prevent violations. But there are still several studies to mention which cover skill constraints in their models as in our study. Burke et al. (2006) approaches to skills differently unlike other nurse models. Skill categories in this research are also not hierarchically overlapping as in other models. But in addition to main skill categories of nurses, they also have alternative skills. This enables a regular nurse to temporarily fill in for a head nurse when there is a shortage. Burke et al. (2010b) not only use the similar approach to skills but also introduce a concept of primary and secondary skills. In this new structure, nurses still cannot be assigned to shifts when they lack the required skills. However, assigning a nurse to a shift where the needed skill of the shift is a secondary skill of the nurse would be permissible with a cost of a penalty. The SS algorithm is used as a solution approach. The algorithm is tested on a publicly available real-world data instances. The computational results show the effectiveness of the proposed solution. A similar incorporation of skills is also seen in Bilgin et al. (2012b). The authors introduce a new model, propose a VNS and ALNS based solution approach, and experiment on real-world data. Results show that new neighborhoods proposed in the model perform better than the traditional neighborhoods seen in the literature. Apart from the skill types, Martin et al. (2013) not only consider skills both in HCs and SCs, but also aim to achieve fair schedules among nurses. An agent-based framework for cooperative meta-heuristic search is proposed in the study. Experiments conducted on real-world benchmark problems demonstrate the success of the algorithm in generating fairer schedules for nurses.

Table 5.1 provides a comprehensive summary of studies in the literature that consider skills as part of their models. Interested readers are referred to general NRP review papers (Cheang et al., 2003; Burke et al., 2004; Bergh et al. 2013).

Table 5.1 Summary of the NRP research with skill constraints

Study	**Skill constraints**		**Solution technique**
	HC	**SC**	
Aickelin & White (2004)	✓		GA
Aickelin & Dowsland (2004)	✓		GA
Bard & Purnomo (2005a)	✓		CG
Bard & Purnomo (2005b)	✓		CG & IP
Burke et al. (2006)	✓	✓	TS
Bester et al. (2007)	✓		TS
Gutjahr & Rauner (2007)	✓		ACO
Beddoe et al. (2009)	✓		CBR
Goodman et al. (2009)	✓		GRASP
Bai et al. (2010)	✓		SA-HH
Brucker et al. (2010)	✓	✓	AC
Burke et al. (2010b)	✓	✓	SS
Bilgin et al. (2012b)	✓	✓	VNS-ALNS
Maenhout & Vanhoucke (2013)	✓		CG
Wright & Mahar (2013)	✓		Heuristic
Martin et al. (2013)	✓	✓	Agent-based
Smet et al. (2014)	✓		HH
Awadallah et al. (2015)	✓	✓	ACO
Wu et al. (2015)	✓		PSO
Xiang et al. (2015)	✓		ACO
Tassopoulos et al. (2015)		✓	VNS
Liang & Turkcan (2016)	✓		MOA
Lim et al. (2016)	✓		CG
Farasat & Nikolaev (2016)	✓		SSSO
Martinelly & Meskens (2017)	✓		MOA
Zheng et al. (2017)		✓	VNS
Jaradat et al. (2018)		✓	Elitist-AS
Dumrongsiri & Chongphaisal (2018)	✓		MOA
Proposed study	✓	✓	Mat-heuristic

GRASP - Greedy randomized adaptive search procedure, MOA – multi-objective approach, AS – ant system, SSSO - Signed social structure optimization

In brief, the earlier model lacks this important property. In addition to that, it does not properly reveal final nurse assignments to departments. Resulting schedules provide details about shifts, but they fail to include departmental unit assignments. For instance, a nurse would know which shift he/she will be working on in the planning period, but he/she would not have information on the assigned departments. Incorporating this information would lead to better and clearer schedules and help real-world users. To our best knowledge, an NRP model considering skill features in both sides of the constraints as well as a centralized scheduling of nurses that provides medical unit level assignment details has not been addressed in the literature.

In this chapter, our contribution to the NRP OR literature is two-fold. First of all, we propose a new NRP model that incorporates many new elements to the scheduling procedure. The new model extends existing models in three ways. It considers skills both as HCs and SCs. It penalizes the assignment of nurses to shifts when these shifts require more experienced nurses. Moreover, the model also accounts for last minute day on and off requests. These types of requests are different than planned vacations. A nurse may ask to be off on a specific day for various reasons. The last extension to the model is that final schedules provide unit assignments. The second contribution of the chapter to the literature is that we develop a new solution technique that combines the strength of IP and the flexibility of meta-heuristics. Finding initial solutions via the IP algorithm and improving them within a PSO framework that utilizes the IP method when new solutions are infeasible are the unique features of the proposed technique.

The new matheuristic is tested on data instances generated from real-world data sets. The computational results show that the approach provides powerful schedules with 2 percent gaps compared to stand-alone IP solutions for smaller instances and report the best-known results on large test data.

In the next couple of sections, details around the extended model and the new solution methodology is provided. Section 5.2 provides model details. Solution technique is introduced in Section 5.3. Computational experiments are presented in Section 5.4. And finally, Section 5.5 draws the conclusion and future research opportunities.

5.2 New NRP Model with Departments and Skills

In addition to hard and soft constraints introduced in the earlier NRP model, this new model considers departmental unit assignments, required and preferred nurse skills, and day on/off requests from nurses.

One of the unique features of this new model is the presentation of the unit index in decision variables. Resulting schedules provide nurse assignments to shifts along with assigned medical departments.

The main importance of this unit information is in the practical usage of it in real-world applications. For example, when there are multiple departments in a health care facility, head nurses or scheduling departments are required to enter all nurse data, data related to contracts, policies and regulations into these scheduling softwares every time they run the optimization under the model presented in the previous chapter. The new model regards the facility holistically and eliminates the need for multiple optimization runs. Entire data is entered at one time and the resulting schedules provide work plans for every nurse in every unit.

In the following sub-sections, the mathematical model details are provided.

5.2.1 Notation

Table 5.2 below describes the notation used in the model.

Table 5.2 Notation of the mathematical model

Sets	Definition
N	Set of nurses (n = 1, 2,…, N)
h	Number of days in the planning horizon
D	Set of days in the planning horizon = {1…h} (d = 1, 2,…, D)
W	Set of weekends in the planning horizon = {1...h/7} (w = 1, 2,…, W)
S	Set of shift types (s = 1, 2,…, S)
$S_s^{Restricted}$	Set of shifts that cannot be assigned immediately after shift s
$D_n^{Vacation}$	Set of days that nurse n cannot be assigned a shift
l_s	Length of shift s in minutes
p	Number of restricted days in a pattern
$N_u^{Required}$	Set of nurses who have the required skill group in unit u
$N_u^{Preferred}$	Set of nurses who have the preferred skill group in unit u
Parameters	**Definition**
t_{ns}^{max}	Total maximum number of shifts of s that can be assigned to nurse n
a_n^{min}	Minimum number of work time in the planning horizon
a_n^{max}	Maximum number of work time in the planning horizon
c_n^{min}	Minimum number of consecutive shift assignments
c_n^{max}	Maximum number of consecutive shift assignments
f_n^{min}	Minimum number of consecutive days off
v_n^{max}	Maximum number of weekend assignments
n_{dsu}	Needed total number of nurses in unit u on shift s on day d
$p_{nds}^{s\_off}$	Penalty if shift s is assigned to nurse n on day d
$p_{nds}^{s\_on}$	Penalty if shift s is not assigned to nurse n on day d
p_{dsu}^{under}	Penalty for understaffing in unit u on shift s on day d
p_{dsu}^{over}	Penalty for overstaffing in unit u on shift s on day d
p^{skill}	Penalty for missing preferred skill
$p_{nd}^{d\_off}$	Penalty if day off request is not granted to nurse n on day d
$p_{nd}^{d\_on}$	Penalty if day on request is not granted to nurse n on day d
Decision Variables	**Definition**
x_{ndsu}	1 if nurse n is assigned in unit u to shift s on day d, 0 otherwise
k_{nw}	1 if nurse n works on weekend w, 0 otherwise
u_{dsu}	Total nurses understaffed in unit u on shift s on day d
o_{dsu}	Total nurses overstaffed in unit u on shift s on day d
m_{dsu}	Total nurses missing preferred skills in unit u on shift s on day d

5.2.2 Constraints

The following HCs are enforced as part of the model.

$$\sum_{s\in S}\sum_{u\in U} x_{ndsu} \leq 1, \qquad \forall n \in N, d \in D \tag{5.1}$$

$$\sum_{u\in U} x_{ndsu} + \sum_{u\in U} x_{n(d+1)ru} \leq 1, \quad \forall n \in N, d \in \{1 \ldots h-1\}, s \in S, r \in S_s^{Restricted} \tag{5.2}$$

$$\sum_{d\in D}\sum_{u\in U} x_{ndsu} \leq t_{ns}^{max}, \qquad \forall n \in N, s \in S \tag{5.3}$$

$$a_n^{min} \leq \sum_{d\in D}\sum_{s\in S}\sum_{u\in U} l_s x_{ndsu} \leq a_n^{max}, \qquad \forall n \in N \tag{5.4}$$

$$\sum_{j=d}^{d+c_n^{max}} \sum_{s\in S}\sum_{u\in U} x_{njsu} \leq c_n^{max}, \qquad \forall n \in N, d \in \{1 \ldots h - c_n^{max}\} \tag{5.5}$$

$$\sum_{s\in S}\sum_{u\in U} x_{ndsu} + \left(p - \sum_{j=d+1}^{d+p}\sum_{s\in S}\sum_{u\in U} x_{ndsu}\right) + \sum_{s\in S}\sum_{u\in U} x_{n(d+p+1)su} > 0, \ \ \forall n \in N, p \in \left\{1 \ldots c_n^{min} - 1\right\}, d \in \{1 \ldots h-(p+1)\} \tag{5.6}$$

$$\left(1 - \sum_{s\in S}\sum_{u\in U} x_{ndsu}\right) + \sum_{j=d+1}^{d+p}\sum_{s\in S}\sum_{u\in U} x_{njsu} + \left(1 - \sum_{s\in S}\sum_{u\in U} x_{n(d+p+1)su}\right) > 0, \ \ \forall n \in N, p \in \left\{1 \ldots f_n^{min} - 1\right\}, d \in \{1 \ldots h-(p+1)\} \tag{5.7}$$

$$k_{nw} \leq \sum_{s \in S} \sum_{u \in U} x_{n(7w-1)su} + \sum_{s \in S} \sum_{u \in U} x_{n(7w)su} \leq 2k_{nw}, \quad \forall n \in N, w \in W \tag{5.8}$$

$$\sum_{w \in W} k_{nw} \leq v_n^{max}, \qquad \forall n \in N \tag{5.9}$$

$$x_{ndsu} = 0, \qquad \forall n \in N, d \in D_n^{Vacation}, s \in S, u \in U \tag{5.10}$$

$$\sum_{n \in N} x_{ndsu} - o_{dsu} + u_{dsu} = n_{dsu}, \qquad \forall d \in D, s \in S, u \in U \tag{5.11}$$

$$x_{ndsu} = 0, \qquad \forall n \notin N_u^{Required}, d \in D, s \in S, u \in U \tag{5.12}$$

$$\sum_{n \in N} x_{ndsu} = \sum_{n \in N_u^{Preferred}} x_{ndsu} + m_{dsu}, \qquad \forall d \in D, s \in S, u \in U \tag{5.13}$$

Constraints (5.1) enforce that only one assignment can be made per day per employee for all shifts and all units.

Because of the minimum rest after a shift, certain shifts cannot follow others. For example, late shift cannot follow an early shift. This requirement is ensured by Constraints (5.2). Constraints (5.3) require that total assignments of shift s in the horizon have to be less than or equal to the maximum allowed limit for that shift. Total work time in the planning horizon must be between limits. This is assured by Constraints (5.4).

Constraints (5.5) verify if maximum consecutive work assignment must be less than a predefined limit. Constraints (5.6) address minimum consecutive work time.

Nurses must work at least a certain number of days before taking a day off. If a nurse has to work at least three days without a day off, the following pattern must not be allowed: “off-on-off” and “off-on-on-off”. Similar pattern validation is in Constraints (5.7). Nurses must rest at least a certain number of days before they are assigned to shifts again. If an employee has to take minimum two days off, the following patterns must not be permitted: “on-off-on”.

Constraints (5.8) and Constraints (5.9) deal with weekend work allocation. Maximum number of weekends must be less than the limit. Any shift on Saturday or Sunday is accounted as the nurse works over the weekend.

No shift assignment can be made while a nurse is on vacation. This is assured by Constraints (5.10). Constraints (5.11) identify the number of nurses assigned below or above the daily desired nurse need of departmental units. Constraints (5.12) enforce that nurses who do not have required skills to work in units cannot be assigned to these units.

Preferred skills are accounted in Constraints (5.13). They capture the number of nurses who lack preferred skills in departmental units. If a unit needs 3 regular nurses but the assignment results in only 2 regular and 1 junior. The junior nurse assignment must be captured and penalized.

5.2.3 Objective Function

The goal of the model is to minimize the sum of all the SC penalties resulting from requests and staffing discrepancies.

$$\boldsymbol{minimize}\; F = F_s + F_a + F_m + F_d \tag{5.14}$$

F_s part of the objective value accounts for shift on and off requests from nurses. When a nurse does not desire to be assigned to a shift on a day, but the resulting schedule has the nurse assigned to that undesired shift, it will result in a discomfort.

The same principle is also true for when he/she wants to be assigned, but the request is not granted. The following equation calculates total penalties for these shift related request violations.

$$
\begin{aligned}
F_s = {} & \sum_{n\in N}\sum_{d\in D}\sum_{s\in S} p_{nds}^{s\_on}\left(1-\sum_{u\in U} x_{ndsu}\right) \\
& + \sum_{n\in N}\sum_{d\in D}\sum_{s\in S} p_{nds}^{s\_off}\sum_{u\in U} x_{ndsu}
\end{aligned}
\tag{5.15}
$$

The next part of the objective function, F_a, accounts for over and under staffing of nurses based on needed number for that departmental unit, shift, and the day. While over staffing is not desired, understaffing always causes greater problems in the healthcare provision.

$$
F_a = \sum_{d\in D}\sum_{s\in S}\sum_{u\in U} p_{dsu}^{under} u_{dsu} + \sum_{d\in D}\sum_{s\in S}\sum_{u\in U} p_{dsu}^{over} o_{dsu} \tag{5.16}
$$

The third part of the objective function, F_m, addresses the assignment of less skilled nurses to units when the preferences is to have experienced ones.

$$
F_m = \sum_{d\in D}\sum_{s\in S}\sum_{u\in U} p^{skill} m_{dsu} \tag{5.17}
$$

The last part, F_d, penalizes situations when day on and day off requests are violated. To the best of our knowledge, no mathematical model exists in the NRP literature that accounts for this SC.

$$
F_d = \sum_{n\in N}\sum_{d\in D} p_{nd}^{d\_on}\left(1-\sum_{s\in S}\sum_{u\in U} x_{ndsu}\right) + \sum_{n\in N}\sum_{d\in D} p_{nd}^{d\_off}\sum_{s\in S}\sum_{u\in U} x_{ndsu} \tag{5.18}
$$

List of penalties and their weight are presented in Table 5.3.

Table 5.3 Penalties

Penalty	Description	Weight
$p_{nds}^{s\_off}$	Shift off request violation	1 – 3
$p_{nds}^{s\_on}$	Shift on request violation	1 – 3
p_{dsu}^{under}	Understaffing penalty	100
p_{dsu}^{over}	Overstaffing penalty	1
p^{skill}	Preferred skill violation	5
$p_{nd}^{d\_off}$	Day off request violation	9
$p_{nd}^{d\_on}$	Day on request violation	6

5.3 Solution Methodology

The solution methodology proposed in this study is a mat-heuristic based approach. Mat-heuristics are heuristic algorithms developed by the interoperation of meta-heuristics and mathematical programming techniques, in other words, exploiting mathematical programming techniques in meta-heuristic frameworks (Boschetti et al. 2009; Caserta & Voß, 2009). In the NRP literature, a few researchers have focused on combining mathematical programming techniques and metahuristics to utilize their complementary strengths to solve the real size NRP problems (Della Croce and Salassa, 2014; Rahimian et al., 2017a).

The proposed mat-heuristic algorithm in this chapter combines PSO and IP to generate powerful schedules. The PSO algorithm introduced by Eberhart and Kennedy (1995) is one of the well-known swarm based evolutionary approaches. The algorithm is inspired by social behavior of bird flocks and fish schooling. Potential solutions, particles, fly through the solution space by communicating to and following each other. Particles' positions are changed based on each particle's own experience as well as the experience of the swarm. Unlike the other evolutionary algorithms, the PSO algorithm does not use crossover or mutation to generate new solutions. In contrast, it utilizes velocities for obtaining new particle structures. In the proposed mat-heuristic framework, initial schedules are generated by the IP and are improved by the coordination of PSO-IP hybrid iteratively. Figure 5.1 presents the flow in detail.

In the initial solution generation phase of the approach, a group of nurses are randomly selected and an IP model is constructed by feeding associated nurse data into the model. IP model is then solved and nurses are placed to their related sequence in the schedule. This loop is carried over until all the nurses are optimized and the resulting schedule is reported to the PSO-IP part of the flow.

In the second part of the flow, initial solutions are immediately accepted as the current positions of particles and objective values are calculated accordingly and stored. Moreover, initial solutions are also set as best positions of particles and evaluated among themselves to identify the best global schedule and fitness. When all particles in the swarm are evaluated, the algorithm proceeds to the improvement phase.

In the improvement phase, a day is randomly selected and encoded as a two dimensional binary 0-1 matrix for all units where one dimension representing nurses and the other one representing shifts. For instance, if nurse n is assigned to shift s on day d, the resulting encoding would be 1 for that day. Otherwise, the matrix would have value 0. The encoding structure is reviewed in detail in Section 5.3.1.1. As the next step in the flow, new positions are calculated. At first, new velocities are calculated and applied to current positions to capture the new positions. Section 5.3.1.2 addresses the algorithm behind the new position calculation which includes probabilistic decision making for binary values. The binary approach so often results in infeasible schedule. Thus, a repair algorithm is needed to obtain feasible solutions. This is achieved via IP. Group of nurses are randomly selected and a full mathematical model is generated including all the hard and soft constraints. Decision variables on the selected day are fixed for randomly selected nurses and all other variables of these nurses are relaxed in the schedule. Rest of the nurses and their prior assignments are kept in place untouched. Infeasibilities on the selected nurses are repaired and remaining nurses are iteratively selected and the schedule is fixed until all the infeasibilities are addressed. The schedule is decoded and the new fitness values are calculated.

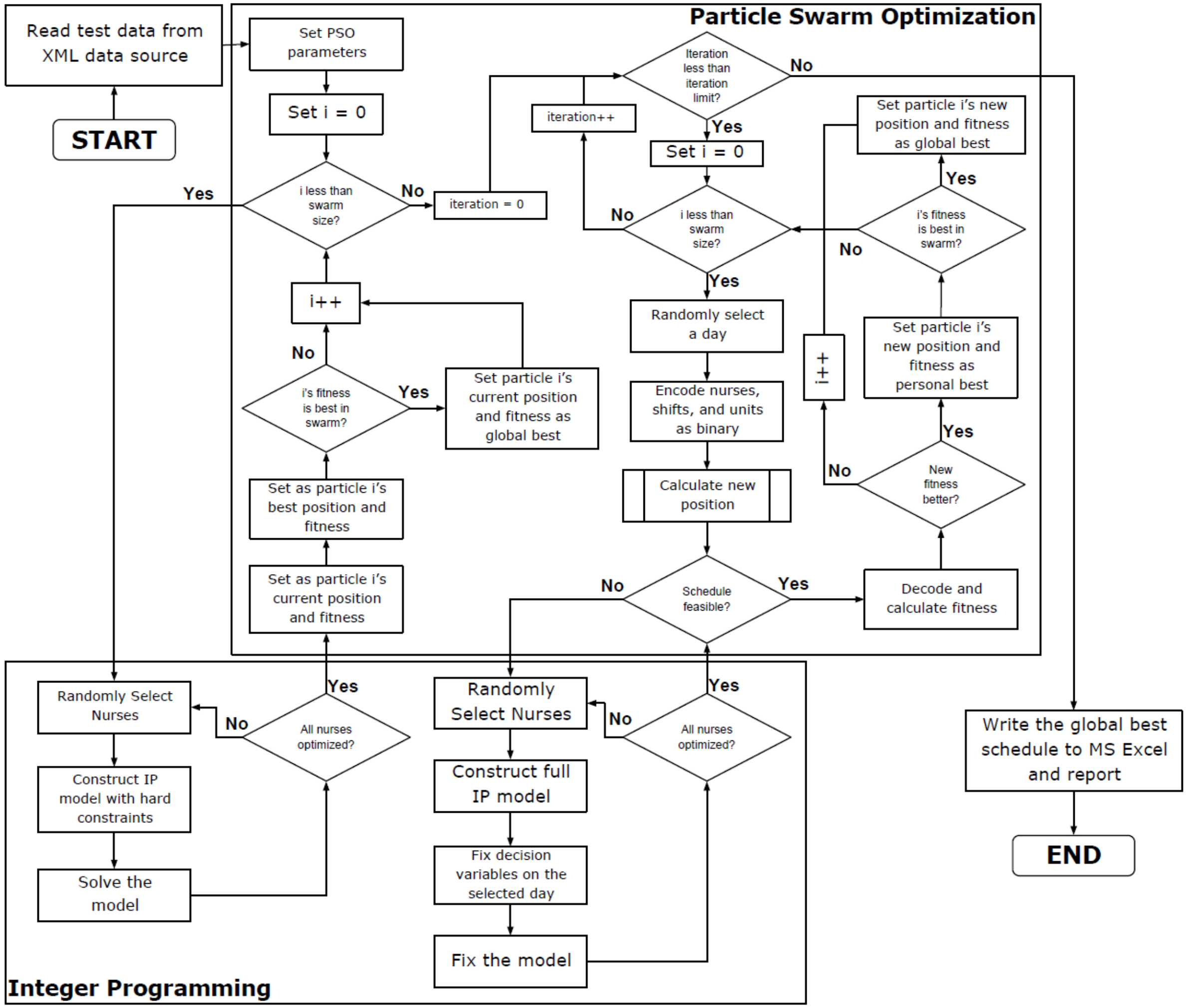

Particle Swarm Optimization
Read test data from XML data source
START
Set PSO parameters
Set i = 0
i less than swarm size?
Yes
No
iteration = 0
i++
No
i's fitness is best in swarm?
Yes
Set particle i's current position and fitness as global best
Set as particle i's best position and fitness
Set as particle i's current position and fitness
Iteration less than iteration limit?
No
iteration++
Yes
Set i = 0
i less than swarm size?
No
Yes
Randomly select a day
Encode nurses, shifts, and units as binary
Calculate new position
Schedule feasible?
No
Yes
Decode and calculate fitness
New fitness better?
No
Yes
Set particle i's new position and fitness as personal best
i's fitness is best in swarm?
No
Yes
Set particle i's new position and fitness as global best
i++
Write the global best schedule to MS Excel and report
END
Randomly Select Nurses
Construct IP model with hard constraints
Solve the model
All nurses optimized?
No
Yes
Randomly Select Nurses
Construct full IP model
Fix decision variables on the selected day
Fix the model
All nurses optimized?
No
Yes
Integer Programming

Figure 5.1 The flow of the algorithm

When the new fitness value is better than the previous position's fitness, the new position is captured as the new personal best for the current particle. If the value is even better than the current global, the current global is also updated. This process continues until the termination criterion is satisfied and then the final global best schedule and value are reported.

5.3.1 Discrete PSO

The classical PSO approach is restricted with real numbers. But many optimization and especially scheduling problems have to deal with discrete notions in variables. To address this need, Kennedy and Eberhart (1997) propose a discrete version of the PSO algorithm. In this approach, particles are encoded as sets of binary variables and their velocities are calculated by the change of probabilities. Table 5.4 presents the notation used for describing the discrete PSO.

Table 5.4 Notation of the discrete PSO

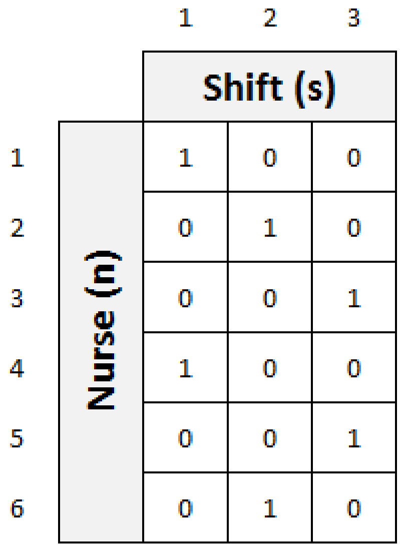
Symbol	Definition
N_p	Number of particles in the swarm population
t	Iteration t
D	Dimension
X_i^t	Particle i in iteration t. $X_i^t = (x_{i1}^t, x_{i2}^t, \dots, x_{iD}^t,), x_{id}^t \in \{0, 1\}$
V_i^t	Velocity of particle i in iteration t. $V_i^t = (v_{i1}^t, v_{i2}^t, \dots, v_{iD}^t,), v_{id}^t \in R$
V_{max}	Maximum velocity
P_i^t	Particle i's best position in iteration t. $P_i^t = (p_{i1}^t, p_{i2}^t, \dots, p_{iD}^t,)$
P_g^t	Global best position in iteration t. $P_p^t = \left(p_{g1}^t, p_{g2}^t, \dots, p_{gD}^t,\right)$
r_1, r_2	Uniformly distributed random numbers in [0, 1]
c_1	Cognition learning factor. A.k.a. personal experience
c_2	Social learning factor. A.k.a. population's experience
w	Inertia weight
$s(v_{id}^t)$	Probability of x_{id}^t taking value 1

Details of the encoding structure and new position calculation are addressed in the following sub-sections.

5.3.1.1 Encoding Approach

In the discrete PSO, current particle representation must be encoded into a binary structure to properly apply the algorithm. This study chooses to encode based on nurse-shift pairs on a selected day for all units. There are several reasons for this decision.

First of all, calculated new positions must be in a structure where the IP algorithm can repair. There are units where nurse can never be assigned to due to the skill related HCs and there are days in which nurse assignment is not permissible because of vacations. On the other hand, assignments can be made without limits between nurses and shifts and new structures can be repaired by the IP. Another consideration is that, if all days are accounted for the encoding at once, it would computationally be extremely expensive to repair infeasible schedules. Because fixing multiple days to their values during optimization would likely to generate additional infeasible structures and the IP part of the algorithm would take considerable amount of time. Figure 5.2 below shows the encoding mechanism for particle X_i^t in day d for all units. 1s in the figure represent assignments of nurses to related shifts and 0s correspond to no assignment. As the general HC, a nurse can only be assigned to one shift in a day. For this reason, every row will have to have only one assignment.



		1	2	3
		Shift (s)		
1	Nurse (n)	1	0	0
2		0	1	0
3		0	0	1
4		1	0	0
5		0	0	1
6		0	1	0

Figure 5.2 Nurse-shift encoding in day d for all units

5.3.1.2 *New Position Calculation*

New positions are first generated by calculating velocities. The following velocity function is used to obtain new velocities.

$$v_{id}^{t} = wv_{id}^{t-1} + c_1 r_1 (p_{id}^{t} - x_{id}^{t}) + c_2 r_2 \left(p_{gd}^{t} - x_{id}^{t}\right), v_{id}^{t} \in [-V_{max}, +V_{max}] \tag{5.19}$$

Generally V_{max} is set to 4 as proposed by Kennedy et al. (2001). w is the inertia weight proposed by Shi and Eberhart (1998) and it is set to 1.2 in the literature. Generally the sum of the cognition and social learning factors is equal to 4.

Because direct addition of velocities to current particle positions is not possible in discrete PSO, the following function is used to obtain the changes of probabilities.

$$s(v_{id}^{t}) = \frac{1}{1 + \exp(-v_{id}^{t})} \tag{5.20}$$

Finally, the new position, x_{id}^{t+1}, is calculated by generating a random number between [0, 1] and comparing the value against $s(v_{id}^{t})$. If the random number is less than $s(v_{id}^{t})$, then x_{id}^{t+1} is set to 1, otherwise x_{id}^{t+1} takes the value 0.

5.3.1.3 *Discrete PSO Algorithm*

Figure 5.3 presents a pseudo-code for the algorithm. For each particle, initial solutions of particles are evaluated. First, each particle’s best positions are captured by looping all the dimensions if current solutions are better than previous best ones. Then, current solutions are evaluated against the population’s best positions and stored accordingly.

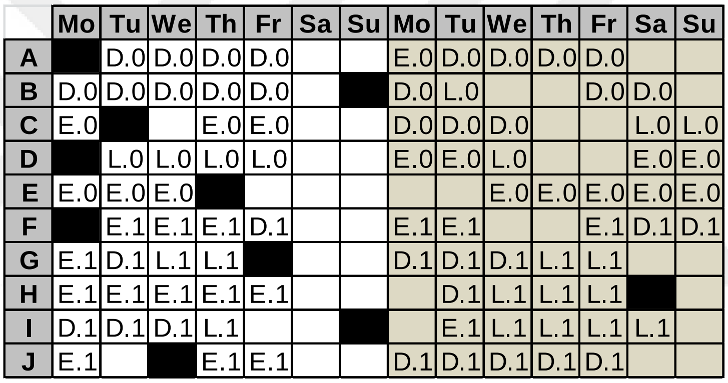
Finally for each particle, new velocities and positions are calculated based on a random number generation. If it is less than the change in probabilities, an assignment is made; otherwise there would not be an assignment.

```
Algorithm 1: Discrete PSO
Input: Initial Solution
Output: Final Global Best Solution
foreach i in $N_p$
    if CalculateObjective($X_i^t$) < CalculateObjective($P_i^t$)
        foreach d in D
            $p_{id}^t = x_{id}^t;$
        end for
    end if
    if CalculateObjective($X_i^t$) < CalculateObjective($P_g^t$)
        g = i;
        $P_g^t = X_i^t;$
    end if
    foreach d in D
        $v_{id}^t = wv_{id}^{t-1} + c_1r_1(p_{id}^t - x_{id}^t) + c_2r_2(p_{gd}^t - x_{id}^t);$
        if $v_{id}^t < -V_{max}$
            $v_{id}^t = -V_{max};$
        elseif $v_{id}^t > +V_{max}$
            $v_{id}^t = +V_{max};$
        end if
        $s(v_{id}^t) = \frac{1}{1+\exp(-v_{id}^t)};$
        if Random(0,1) < $s(v_{id}^t)$
            $x_{id}^{t+1} = 1;$
        else
            $x_{id}^{t+1} = 0;$
        end if
    end for
end for
return(Global Best Solution);
```

Figure 5.3 Pseudo-code of the discrete PSO

5.4 Computational Experiments

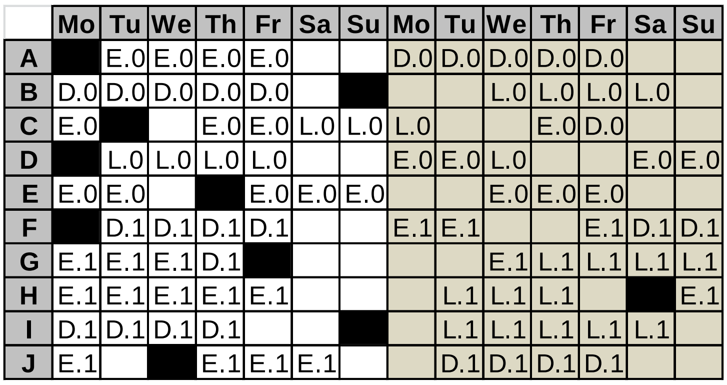
The following couple of figures explain the implementation of the mat-heuristic on a small scale problem. Figure 5.4 shows a particle that was generated via the IP by randomly selecting group of nurses and generating initial schedules for them. First character of the assignment addresses the shift which nurses are assigned to, and the last character is for the departmental unit. For example, nurse F on the Friday of the first week is assigned to shift D and unit 1. In this sample problem, we have 3 shifts (E – Early, D – Day, and L - Late), 2 departmental units (Unit 1 – Critical Care, Unit 2 – Internal Medicine), 10 nurses (Nurses A, B, C, D, E, F, G, H, I, and J), and 2 weeks for the planning horizon.



	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su
A	■	D.0	D.0	D.0	D.0			E.0	D.0	D.0	D.0	D.0		
B	D.0	D.0	D.0	D.0	D.0		■	D.0	L.0			D.0	D.0	
C	E.0	■		E.0	E.0			D.0	D.0	D.0			L.0	L.0
D	■	L.0	L.0	L.0	L.0			E.0	E.0	L.0			E.0	E.0
E	E.0	E.0	E.0	■						E.0	E.0	E.0	E.0	E.0
F	■	E.1	E.1	E.1	D.1			E.1	E.1			E.1	D.1	D.1
G	E.1	D.1	L.1	L.1	■			D.1	D.1	D.1	L.1	L.1		
H	E.1	E.1	E.1	E.1	E.1				D.1	L.1	L.1	L.1	■	
I	D.1	D.1	D.1	L.1			■		E.1	L.1	L.1	L.1	L.1	
J	E.1		■	E.1	E.1			D.1	D.1	D.1	D.1	D.1		

Figure 5.4 First particle in the first iteration with cost 4623

Figure 5.5 represents the best schedule in the population. While the first particle's cost value is 4623, the best global schedule has 4017.



	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su
A	■	E.0	E.0	E.0	E.0			D.0	D.0	D.0	D.0	D.0		
B	D.0	D.0	D.0	D.0	D.0		■			L.0	L.0	L.0	L.0	
C	E.0	■		E.0	E.0	L.0	L.0	L.0			E.0	D.0		
D	■	L.0	L.0	L.0	L.0			E.0	E.0	L.0			E.0	E.0
E	E.0	E.0		■	E.0	E.0	E.0			E.0	E.0	E.0		
F	■	D.1	D.1	D.1	D.1			E.1	E.1			E.1	D.1	D.1
G	E.1	E.1	E.1	D.1	■					E.1	L.1	L.1	L.1	L.1
H	E.1	E.1	E.1	E.1	E.1				L.1	L.1	L.1		■	E.1
I	D.1	D.1	D.1	D.1			■		L.1	L.1	L.1	L.1	L.1	
J	E.1		■	E.1	E.1	E.1			D.1	D.1	D.1	D.1		

Figure 5.5 Best global schedule in the first iteration with cost 4017

When Thursday of the first week – Day 4 is randomly selected as seen in Figure 5.6, that day needs to be encoded into a binary 0-1 matrix for all positions and velocities where the rows represent nurses and the columns represent shifts.

		1	2	3	4	5	6	7	8	9	10	11	12	13	14
		Mo	**Tu**	**We**	**Th**	**Fr**	**Sa**	**Su**	**Mo**	**Tu**	**We**	**Th**	**Fr**	**Sa**	**Su**
1	**A**	■	D.0	D.0	D.0	D.0			E.0	D.0	D.0	D.0	D.0		
2	**B**	D.0	D.0	D.0	D.0	D.0		■	D.0	L.0			D.0	D.0	
3	**C**	E.0	■		E.0	E.0			D.0	D.0	D.0			L.0	L.0
4	**D**	■	L.0	L.0	L.0	L.0			E.0	E.0	L.0			E.0	E.0
5	**E**	E.0	E.0	E.0	■						E.0	E.0	E.0	E.0	E.0
6	**F**	■	E.1	E.1	E.1	D.1			E.1	E.1			E.1	D.1	D.1
7	**G**	E.1	D.1	L.1	L.1	■			D.1	D.1	D.1	L.1	L.1		
8	**H**	E.1	E.1	E.1	E.1	E.1				D.1	L.1	L.1	L.1	■	
9	**I**	D.1	D.1	D.1	L.1			■		E.1	L.1	L.1	L.1	L.1	
10	**J**	E.1		■	E.1	E.1			D.1	D.1	D.1	D.1	D.1		

Figure 5.6 Randomly selected day: 4 – thursday for the first particle

As seen in Figure 5.7, current position of the first particle in the swarm gets value 1 for nurse A (the first nurse) and shift D (the second shift) on Day 4. Because the current iteration is the first iteration, current best solution of the particle is same as the current position. Starting velocities are set to value 0 for the first iteration. Best global position is the encoding of the best global schedule's Thursday of the first week. The below binary position matrices take value 1 when nurses are assigned to respected shifts, and take value 0 otherwise.

Current Positions

		1	2	3
		Shift (s)		
1	**Nurse (n)**	0	1	0
2		0	1	0
3		1	0	0
4		0	0	1
5		0	0	0
6		1	0	0
7		0	0	1
8		1	0	0
9		0	0	1
10		1	0	0

Current Best Positions

		1	2	3
		Shift (s)		
1	**Nurse (n)**	0	1	0
2		0	1	0
3		1	0	0
4		0	0	1
5		0	0	0
6		1	0	0
7		0	0	1
8		1	0	0
9		0	0	1
10		1	0	0

Current Velocities

		1	2	3
		Shift (s)		
1	**Nurse (n)**	0	0	0
2		0	0	0
3		0	0	0
4		0	0	0
5		0	0	0
6		0	0	0
7		0	0	0
8		0	0	0
9		0	0	0
10		0	0	0

Best Global Position

		1	2	3
		Shift (s)		
1	**Nurse (n)**	1	0	0
2		0	1	0
3		1	0	0
4		0	0	1
5		0	0	0
6		0	1	0
7		0	1	0
8		1	0	0
9		0	1	0
10		1	0	0

Figure 5.7 Encoded nurse-shift positions and velocities for day 4 – thursday

When below equations are applied to nurse 1 and shift 1 for the selected day, the resulting velocities are seen in Figure 5.8.

$$v_{i11}^{t} = wv_{i11}^{t-1} + c_1 r_1 (p_{i11}^{t} - x_{i11}^{t}) + c_2 r_2 \left(p_{g11}^{t} - x_{i11}^{t}\right), i = 1, t = 1 \qquad (5.21)$$

In this application, the cognition part of the learning factor yields to 0. Because the personal best positions in the first iteration and current positions are the same. Therefore, the above equation is only derived from the social part. A random number of 0.29 and social learning factor of 2 results in 0.58 for velocity.

$$v_{i11}^{t} = 1.2 * 0 + 2 * 0.63 * (0 - 0) + 2 * 0.29 * (1 - 0) = 0.58 \qquad (5.22)$$

All the dimensions are taken into account by the above formula and new velocities are calculated. Unlike the continuous PSO, the discrete PSO requires calculation of the changes of probabilities by the below formula to further identify the new positions. For the same nurse 1 and the same shift 1, velocity of 0.58 yields 0.641 as a change of probability. High probability increases the chance of shifts to be assigned to nurses.

$$s(v_{i11}^{t}) = \frac{1}{1 + \exp(-v_{i11}^{t})} = \frac{1}{1 + \exp(-0.58)} = 0.641 \qquad (5.23)$$

Final step is that a random number is generated against the changes of probabilities and the nurse is assigned to that shift if the change of probability is greater than the random number.

New Velocities

		Shift (s)		
		1	2	3
Nurse (n)	1	0.58	-0.58	0
	2	0	0	0
	3	0	0	0
	4	0	0	0
	5	0	0	0
	6	-0.58	0.58	0
	7	0	0.58	-0.58
	8	0	0	0
	9	0	0.58	-0.58
	10	0	0	0

Changes of Probabilities

		Shift (s)		
		1	2	3
Nurse (n)	1	0.641	0.359	0.5
	2	0.5	0.5	0.5
	3	0.5	0.5	0.5
	4	0.5	0.5	0.5
	5	0.5	0.5	0.5
	6	0.359	0.641	0.5
	7	0.5	0.641	0.359
	8	0.5	0.5	0.5
	9	0.5	0.641	0.359
	10	0.5	0.5	0.5

New Positions

		Shift (s)		
		1	2	3
Nurse (n)	1	1	0	0
	2	0	1	0
	3	1	0	0
	4	0	0	0
	5	0	0	0
	6	0	1	0
	7	0	1	0
	8	1	0	0
	9	0	1	0
	10	0	0	1

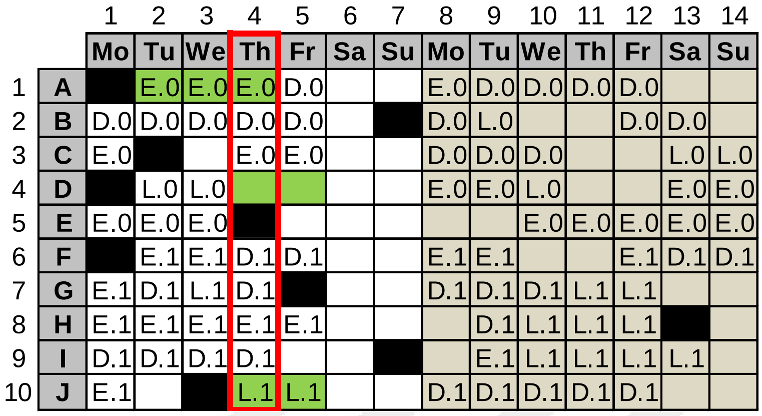
Figure 5.8 New velocities, probabilities and positions for day 4 – thursday

When the decoding algorithm is applied to the first particle’s new position, nurse-shift assignments on Thursday become as presented in Figure 5.9. This new schedule results in infeasibilities for three different nurses. First of all, Nurse A cannot be assigned to an early shift right after a day shift. Nurse D has to take at least two days off after any work day. And finally, Nurse J cannot work on an early shift immediately after a late shift. Therefore, all of these infeasibilities need to be repaired. The IP algorithm is used for this repair purpose. The decision variables on day 4 are set to their existing values, and the rest are relaxed. New IP model is constructed considering all the HCs and SCs and optimized.

		1	2	3	4	5	6	7	8	9	10	11	12	13	14
		Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su
1	A	■	D.0	D.0	E.0	D.0			E.0	D.0	D.0	D.0	D.0		
2	B	D.0	D.0	D.0	D.0	D.0		■	D.0	L.0			D.0	D.0	
3	C	E.0	■		E.0	E.0			D.0	D.0	D.0			L.0	L.0
4	D	■	L.0	L.0		L.0			E.0	E.0	L.0			E.0	E.0
5	E	E.0	E.0	E.0	■						E.0	E.0	E.0	E.0	E.0
6	F	■	E.1	E.1	D.1	D.1			E.1	E.1			E.1	D.1	D.1
7	G	E.1	D.1	L.1	D.1	■			D.1	D.1	D.1	L.1	L.1		
8	H	E.1	E.1	E.1	E.1	E.1				D.1	L.1	L.1	L.1	■	
9	I	D.1	D.1	D.1	D.1			■		E.1	L.1	L.1	L.1	L.1	
10	J	E.1		■	L.1	E.1			D.1	D.1	D.1	D.1	D.1		

Figure 5.9 New schedule for day 4 – thursday

Finally, the repaired model as depicted in Figure 5.10 is evaluated among all particles to set particle’s best and global best values.



		1	2	3	4	5	6	7	8	9	10	11	12	13	14
		Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su
1	A	■	E.0	E.0	E.0	D.0			E.0	D.0	D.0	D.0	D.0		
2	B	D.0	D.0	D.0	D.0	D.0		■	D.0	L.0			D.0	D.0	
3	C	E.0	■		E.0	E.0			D.0	D.0	D.0			L.0	L.0
4	D	■	L.0	L.0	🟩	🟩			E.0	E.0	L.0			E.0	E.0
5	E	E.0	E.0	E.0	■						E.0	E.0	E.0	E.0	E.0
6	F	■	E.1	E.1	D.1	D.1			E.1	E.1			E.1	D.1	D.1
7	G	E.1	D.1	L.1	D.1	■			D.1	D.1	D.1	L.1	L.1		
8	H	E.1	E.1	E.1	E.1	E.1				D.1	L.1	L.1	L.1	■	
9	I	D.1	D.1	D.1	D.1			■		E.1	L.1	L.1	L.1	L.1	
10	J	E.1		■	L.1	L.1			D.1	D.1	D.1	D.1	D.1		

Figure 5.10 New feasible schedule after IP

The software of the proposed mat-heuristic algorithm is written in Java and mathematical models are optimized using the IBM ILOG CPLEX Optimization Studio 12.6 Application Programming Interface with default parameters. The code is executed in the Java Runtime Environment 7 and run on a Windows 7 PC with an Intel Core i3 2.27 GHz processor and 3 GB of RAM.

New experimental data set is generated based on the previously published real-world data instances and industry XML standards to properly assess the performance and effectiveness of the proposed algorithm.

Table 5.5 demonstrates the general characteristics of the data sets. Instances 1-3 are relatively small test data. Therefore, any optimization software can solve them to optimality without having computational difficulties. The rest of the instances are rather intractable. In this experimental design, we choose to only experiment with 2-4 weeks planning horizons unlike in the NRP problem presented in the previous chapter. Generating schedules beyond a month is not practical in real-world applications. At any rate, there are still various numbers of skills and medical units in instances. Major complexity arises when numbers of skills, day on/off requests, and shift on/off requests expand. Hence, numbers of SCs in a test instance are great predictors for the problem ramification.

Table 5.5 Summary of the test data

Instance	Nurses	Days	Shifts	Skills	Units	Vacation Requests	Day-on/off Requests	Shift-on/off Requests
Instance 01	8	14	1	2	2	8	2	26
Instance 02	10	14	3	2	2	10	5	32
Instance 03	20	14	3	2	2	20	10	64
Instance 04	18	28	3	2	2	36	36	135
Instance 05	20	28	3	4	2	40	40	138
Instance 06	30	28	4	6	4	60	60	225
Instance 07	36	28	4	8	4	72	72	232
Instance 08	40	28	5	12	6	80	80	284
Instance 09	50	28	6	18	6	100	100	336
Instance 10	60	28	10	24	6	120	120	422

Table 5.6 summarizes computational results on the instances. The mat-heuristic algorithm is very efficient to find powerful schedules. As discussed earlier, initial solutions are generated by the IP algorithm. There is a direct correlation between number of nurses selected for each iteration during the initial solution generation phase and the quality of initial solutions. When more nurses are evaluated at the same time, the gap values between initial solutions and final solutions decrease drastically. Generally, the number of nurses selected at each iteration is set around one fourth of total number of nurses for a better analysis of the algorithm performance. Therefore, the gap values between initial solutions and final solutions are around 50 percent.

To be able to compare between studies, the first four instances are also solved by a stand-alone IP solver. The IP solver reports optimum solutions for the first three data sets. The fourth data set is not solved to optimality due to the fact that the computational time is limited to two minutes. Overall, the proposed algorithm reaches out to near optimal results with only two percent gaps from the best known results on the first four data sets. As the complexity of an instance increases due to the increase on number of requests, the IP solver fails to provide solutions. But the proposed algorithm can still generate solutions. Best known solutions on instances five to ten are by the mat-heuristic. Optimum results reported in the table are marked in bold print.

Table 5.6 Computational results for 2 minute run time

Instance	Initial Solution	% Gap	Final Solution	% Gap	Best Known Solution
Instance 01	1217	49.14	619	-	**619**
Instance 02	4017	35.05	2609	0.19	**2604**
Instance 03	6140	49.64	3092	2.30	**3021**
Instance 04	17052	63.13	6286	2.34	6139
Instance 05	16240	50.94	7967	-	7967
Instance 06	17519	56.29	7658	-	7658
Instance 07	18127	53.48	8432	-	8432
Instance 08	21287	54.53	9679	-	9679
Instance 09	16984	45.57	9245	-	9245
Instance 10	16812	36.92	10605	-	10605

5.5 Conclusion

An extension to the NRP problem is examined and studied in this chapter. A new mathematical model is developed that additionally accounts for required/ preferred skills, departmental units, and day on/off constraints. Moreover, a new solution methodology is proposed for the solution of the new model.

The proposed solution is a mat-heuristic that combines IP and discrete PSO. Initial solutions generated by IP are fed to the PSO part of the algorithm and a certain encoding/decoding schema is applied to generate new positions and improve the solutions. When infeasible schedules are generated as part of the encoding structure, IP is called again to repair infeasibilities. Schedules are iteratively improved until the final stopping criterion and the best global schedule is reported. Algorithm results compared to stand alone IP solutions are very promising towards solving NP hard problems providing near optimal solutions.

For future studies, new constraints can be added to the problem to make the problem even closer to the real world situations. For instance; not only generating schedules that address cover needs and requests but also provide fair work among nurses. As an example, providing a schedule that equally balances night shifts. Another consideration would be to account for nurse pairings. For example, some nurses may prefer to work with certain peers or some nurses must be assigned together due to training needs.

CHAPTER SIX
CONCLUSION

6.1 Summary

Health care provision is a costly process. Facility, staff, equipment, medicine, service etc. play significant roles and contribute substantially to the cost. Manual handling of this complex process even today in many health care institutions would not only result in additional costs but also reduce satisfaction of care givers and takers. On the other hand, there are countless numbers of OR, AI, and Machine Learning techniques that can be applied to this process for a better utilization of resources, to improve efficiency, and to reduce costs. This dissertation focuses on two crucial operational level decision making problems in healthcare industry: PAS and nurse scheduling. The aim behind choosing these two problems is that both of the partners of health care do exist in any health care setting. Effective scheduling of patients' admissions and nurses' work plans would lead to better health care provision in this very precious field of research.

Therefore, four chapters have been designated to tackle these valuable scheduling problems. Chapters 2 and 3 are devoted to the PAS problem and Chapters 4 and 5 are devoted to the NRP.

In Chapter 2, the PAS literature is reviewed in detail. Introduction to the patient admission field is also presented. The mathematical model of the PAS problem with its notation is elaborated in great detail. Simply, the PAS problem assigns patients to rooms and beds considering many elements. During this assignment process, patients' preferences as well as their medical needs are considered. HCs must be satisfied, but SCs can be violated. Overall goal of the problem is to minimize number of SC violations. Then, a new solution framework is proposed. The framework is a combination of MIP-based heuristics namely F&R and F&O. F&R algorithm is used as a tool to generate initial solutions and the F&O is for the improve phase of the framework. The basic idea behind the MIP-based heuristics is that a computationally intractable combinatorial optimization problem is decomposed into a set of smaller problems and each problem is solved until all the sub-problems are solved. While this technique gives away the optimality, it is as strong and powerful as meta-heuristic models, many times better, and can generate strong results. Computational experiments are done on publicly available data instances and results show that proposed framework can produce schedules in faster computational times than the state-of-the-art solution techniques and gap values between the results and the best known solutions are within 5-15 percent.

In Chapter 3, the extended version of the PAS problem called DPAS is studied. In the DPAS, several real-life applications such as the existence of emergency patients, operating room constraints, and patient delays are additionally considered in automatically assigning patients to rooms to achieve resource-scheduling efficiency, patient satisfaction, and treatment success. There are several new SCs in the DPAS problem that add to the overall cost of the objective value. Delaying of patients' admission days would lead to a discomfort that would result in an additional cost. Keeping rooms and operating theatres idle are also penalized in the DPAS problem. Similar to the idle penalty, using operating theatres more than the designated time also results in costs. Lastly, risk of overcrowding in rooms due to overstays of some of patients are also penalized in the problem. Similar to the PAS problem, the goal of the DPAS problem is to minimize number of violations of the SCs. To solve the problem, MIP-based F&R heuristic is used. Because of the dynamic nature of the problem, all the assignments are evaluated daily. This helps the F&R heuristic to generate schedules with minimal decompositions which leads to better solutions. Public data is used to evaluate the efficiency and strength of the solution method. Six new best-known results are reported. Results on the rest of the data instances show that 35 percent faster processing times are achieved and overall gap from the best-known results is around 10 percent.

In Chapter 4, the research efforts are spanned over to the field of nurse scheduling. The NRP is a complex scheduling problem where nurses must be assigned to shifts considering a set of constraints. Similar to the PAS and DPAS, the

NRP also has the HCs and SCs. Total work time in a planning horizon, minimum rest time, minimum and maximum consecutive work time, weekend restrictions, vacation plans are all considered under the HCs. Therefore, they must be satisfied. Violations are allowed for SCs. When nurses' shift preferences are violated, penalties are incurred. On the other hand, when there are over or under staffing problems for a given day, these types of violations are also reflected in the objective function. In short, schedule quality improves as more and more SCs are satisfied. The problem is difficult to solve to optimality due to its combinatorial structure. Therefore, a hybrid solution framework is proposed. The hybrid technique combines the strength of IP based algorithm and the flexibility of metaheuristics algorithms. MIP-based F&R generates initial solutions by decomposing planning horizons based on weeks or nurses depending on the problem complexity. For smaller size problems, week decomposition is used. Considering all the nurses in a week generates high quality initial solutions. As the result, the improve phase performs relatively better. But for larger problems where week decomposition is not possible because of the size, planning horizons are decomposed based on nurses. Thus, the quality of these initial solutions is worse. At any rate, the improve phase is based on SA and F&O algorithms. SA applies various neighborhood structures to generate new schedules and evaluates the results in a probabilistic manner. F&O is injected to the process in such a unique way. When SA algorithm can no longer improve the search space based on the neighborhoods, the solution is forwarded to the F&O heuristic. Most of the cases, the F&O improves the current schedule and results in exploitation of the search space. Even when the F&O algorithm cannot provide better solutions, the new schedule is significantly different than the previous one resulting in exploration of the search space. Both cases lead to better performance of the neighborhoods and the SA algorithm. To assess the quality and efficiency of the hybrid approach, 24 publicly available test instances recently introduced in the literature are used. Computational results show that the hybrid method outperforms the state-of-the-art solution techniques in most of the test data and report seven new best-known results.

In Chapter 5, an extension is made to the previous NRP model to include more real-world features. The new model is quite different than the classical model. Apart from the existing HCs and SCs, it also incorporates departmental unit assignments, required and preferred nurse skills, and day on and day off requests. The mathematical model of the new NRP and notation are presented and discussed. The model is elaborated. The objective function is a combination of four components. The first component penalizes shift on and off violations. The second component reflects under and over staffing violations. The third part is for the preferred skill penalties. And the last component is for last minute day on and off request violations. For the solution of the problem, MIP-based heuristics are not used for the first time. But the idea of leveraging the power of IP and decomposition of problems into smaller problems is still employed. A novel solution methodology combining IP and PSO is proposed. This new solution methodology is a matheuristic based approach. After data is read from a data source, the PSO specific parameters are set and initial solutions are generated for each particle. Just to note that each particle represents a feasible solution. The IP is used to generate a feasible initial solution for each particle by only addressing HCs. All of the initial solutions are set as particles' personal best solutions and the best solution in the swarm is set as the global best solution and schedule. When the PSO iterations start, a new solution is obtained by applying a certain encoding and decoding structure. The encoding is originated from the discrete PSO. The classical PSO is restricted to real numbers. But most of the scheduling problems like the NRP have to deal with discrete notations. Therefore, particles are encoded as sets of binary variables and their velocities are calculated by the change of probabilities. When new positions are calculated, they often result in infeasible schedules. To overcome this hurdle, the IP is utilized again as a repair tool. Fixed schedules then are evaluated against personal and global best values and accepted or rejected accordingly. The iterations end when a designated termination criterion is met. Besides, a new data set is generated referencing real-world data instances for the computational experiments and the computational results show that the mat-heuristic algorithm is able to generate schedules that have less than 1 percent gap from the optimal solutions. In addition to this solution quality, the algorithm is very flexible for solving problems of any size.

6.2 Contributions

This dissertation contributes to the literature in many ways. The original contributions are summarized in the following paragraphs.

First of all, the MIP-based F&R and F&O heuristics combination is applied to the scheduling literature in health care for the first time to the best of our knowledge. This implementation proves that complex health care scheduling problems can be solved in much shorter processing times than the ones in classical IP techniques. Additionally, it also demonstrates that near optimal solutions are possible even for larger data sets.

Secondly, the hybrid solution technique which combines the MIP-based heuristics and SA together in such a unique way is the first of its kind implementation to the health care scheduling. It carries over the strength of IP based exact solution techniques and takes advantage of the flexibility of meta-heuristics and neighborhood exploration approaches. This unmatched combination achieves diversification and intensification in the search space, generates new schedules, and even improves the performance of the neighborhoods. Even the mat-heuristic model that hybridizes the IP with the discrete PSO has a similar working mechanism. Moreover, both of the solution methodologies are very flexible for applications to other scheduling problems and even to any combinatorial optimization problems of any size. By all means, the solution techniques developed as part of the Ph.D. studies are stand-alone solution tools for optimization.

The last contribution to the literature is that a new scheduling model is proposed for nurses’ work plans. This new model accounts for many more real-world features in addition to the ones that are in the earlier versions. It is a valuable step towards truly representing real life business environments. Nurse skills, last minute day on and day off requests, and departmental assignments are considered as part of the model.

6.3 Future Research

There are still many outstanding research directions from applying the solutions techniques to other models to extending existing models with more constraints and objectives. The following several paragraphs provide more details.

The hybrid MIP-based heuristics and SA methodology can be applied to the static version of the PAS problem to observe the performance. Most of the best known results for this problem are achieved by a plain SA based approach. This implementation would compromise from the sole meta-heuristic implementation, but it would add the strength of the IP.

Another future research can be done on the DPAS problem. The problem can be extended to include new constraints, especially those related to medical staff, like the utilization of nurses and doctors. Every room assignment is closely related to medical staff because nurses and doctors are the ones providing the care. This new constraint would guarantee a balance between patients' satisfaction and the staff utilization.

Along these lines, new constraints can also be added to the NRP problems to make them even much closer to the real world situations. For instance; fair work assignment among nurses are not considered in the current models. This can be a valuable adjustment and would lead to better schedules. Generating schedules that equally balance early and night shifts would also contribute to the staff satisfaction. Another extension to the NRP would be to account for nurse pairings. For example, some nurses may prefer to work with certain peers or some nurses must be assigned together due to training needs. Providing same work schedules for these cases would satisfy requests and needs.

Finally, as scheduling in health care is not limited to the scheduling of patients and nurses, the proposed solution techniques can be tested on other scheduling problems such as operating room scheduling. Furthermore, the proposed solutions are as flexible as they get to be applied beyond health care. Experimenting on other scheduling problems such as high school timetabling would be interesting future research direction.

6.4 Acknowledgements

The methodologies developed as part of this Ph.D. study would not be tested and examined properly without the existence of real-world data. But data gathering in health care has always been a troublesome task due to its sensitivity. There are several researchers who provide instances on publicly available repositories. For this reason, we would like to mention their names to recognize efforts. We thank Peter Demeester and Greet Vanden Berghe for providing all the instances, explaining the details, and making the java validator available to use on a website. We are also thankful to Sara Ceschia and Andrea Schaerf for publicly providing the test instances, problem description, instance generator and validator, and their experimental solutions. Finally, to Tim Curtois and Rong Qu for providing the datasets, explanations, and related links to proper sites in a single website.

REFERENCES

Absi, N., & Kedad-Sidhoum, S. (2007). MIP-based heuristics for multi-item capacitated lot-sizing problem with setup times and shortage costs. *RAIRO - Operations Research, 41*(2), 171–192.

Ağralı, S., Taşkın, Z.C., & Ünal, A.T. (2017). Employee scheduling in service industries with flexible employee availability and demand. *Omega, 66*, 159-169.

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 & Operations Research, 31*(5), 761–778.

Aickelin, U., & White, P. (2004). Building better nurse scheduling algorithms. *Annals of Operations Research, 128*(1–4), 159–177.

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

Akartunalı, K., & Miller, A.J. (2009). A heuristic approach for big bucket multi-level production planning problems. *European Journal of Operational Research, 193*(2), 396-411.

Alonso-Ayuso, A., Escudero, L., Pizarro, C., Romeijn, H., & Morales, D.R. (2006). On solving the multi period single-sourcing problem under uncertainty. *Computational Management Science, 3*(1), 29-53.

Aouam, T., Geryl, K., Kumar, K., & Brahimi, N. (2018). Production planning with order acceptance and demand uncertainty. *Computers & Operations Research, 91,* 145-159.

Assis, L.S.D., & Camponogara, E. (2016). A MILP model for planning the trips of dynamic positioned tankers with variable travel time. *Transportation Research Part E, 93*, 372-388.

Awadallah, M.A., Bolaji, A.L., & Al-Betar, M.A. (2015). A hybrid artificial bee colony for a nurse rostering problem. *Applied Soft Computing, 35,* 726-739.

Azaiez, M.N., & Al Sharif, S.S. (2005). A 0–1 goal programming model for nurse scheduling. *Computers & Operations Research, 32*(3), 491–507.

Bagheri, M., Devin, A.G., & Izanloo, A. (2016). An application of stochastic programming method for nurse scheduling problem in real world hospital. *Computers & Industrial Engineering, 96,* 192-200.

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

Baker, K.R. (1976). Workforce allocation in cyclical scheduling problems: a survey. *Operational Research Quarterly, 27*, 155-167.

Bard, J.F., & Purnomo, H.W. (2005a). A column generation-based approach to solve the preference scheduling problem for nurses with downgrading. *Socio-Economic Planning Sciences, 39*(3), 193–213.

Bard, J.F., & Purnomo, H.W. (2005b). Preference scheduling for nurses using column generation. *European Journal of Operational Research, 164*(2), 510–534.

Beaumont, N. (1997). Scheduling staff using mixed integer programming. *European Journal of Operational Research, 98*(3), 473–484.

Beddoe, G.R., & Petrovic, S. (2007). Enhancing case-based reasoning for personnel rostering with selected tabu search concepts. *Journal of the Operational Research Society, 58*(12), 1586–1598.

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

Bergh, J.V.D., Beliën, J., Bruecker, P.D., Demeulemeester, E., & Boeck, L.D. (2013). Personnel scheduling: A literature review. *European Journal of Operational Research, 226*(3), 367-385.

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

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

Bilgin, B., Demeester, P., & Berghe, G.V. (2008). A hyper heuristic approach to the patient admission scheduling problem. *Techical Report, KaHo Sint-Lieven, Gent.*

Bilgin, B., Demeester, P., Mısır, M., Vancroonenburg, W., & Berghe, G.V. (2010). One hyperheuristic approach to two timetabling problems in health care. *Techical Report, KaHo Sint-Lieven, Gent.*

Bilgin, B., Demeester, P., Misir, M., Vancroonenburg, W., & Berghe, G.V. (2012a). One hyper-heuristic approach to two timetabling problems in health care. *Journal of Heuristics, 18,* 401-434.

Bilgin, B., De Causmaecker, P., Rossie, B., & Vanden Berghe, G. (2012b). Local search neighbourhoods for dealing with a novel nurse rostering model. *Annals of Operations Research, 194*(1), 33–57.

Bolaji, A.L., Bamigbola, A.F., & Shola, P.B. (2018). Late acceptance hill climbing algorithm for solving patient admission scheduling problem. *Knowledge-Based Systems, 145*, 197–206.

Boschetti, M.A., Maniezzo, V., Roffilli, M., Bolufé Röhler, A. (2009). Matheuristics: Optimization, Simulation and Control. In Blesa M.J., Blum C., Di Gaspero L., Roli A., Sampels M., Schaerf A. (Eds). *Hybrid Metaheuristics - Lecture Notes in Computer Science, 5818,* 171-177. Germany: Springer-Verlag.

Bourdais, S., Galinier, P., & Pesant, G. (2003). HIBISCUS: A constraint programming application to staff scheduling in health care. In F. Rossi (Eds.). *Principles and practice of constraint programming - Lecture Notes in Computer Science, 2833*, 153–167. Germany: Springer-Verlag.

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

Bruecker, P.D., Van den Bergh, J., Beliën, J., & Demeulemeester, E. (2015). Workforce planning incorporating skills: state of the art. *European Journal of Operational Research, 243*, 1–16.

Brusco, M.J., & Jacobs, L.W. (1993). A simulated annealing approach to the cyclic staff-scheduling problem. *Naval Research Logistics, 40*(1), 69–84.

Brusco, M.J., & Jacobs, L.W. (1995). Cost analysis of alternative formulations for personnel scheduling in continuously operating organizations. *European Journal of Operational Research, 86*(2), 249–261.

Burke, E.K., De Causmaecker, P., & Vanden Berghe, G. (1999). A hybrid tabu search algorithm for the nurse rostering problem. In: McKay B., Yao X., Newton C.S., Kim JH., Furuhashi T. (Eds.). *Simulated Evolution and Learning - Lecture Notes in Computer Science, 1585*. Germany: Springer-Verlag.

Burke, E.K., Cowling, P., De Causmaecker, P., & Vanden Berghe, G. (2001). A memetic approach to the nurse rostering problem. *Applied Intelligence, 15*(3), 199–214.

Burke, E.K., De Causmaecker, P., Petrovic, S., Vanden Berghe, G. (2003). Variable neighborhood search for nurse rostering problems. *In: Metaheuristics: Computer Decision-Making. Applied Optimization, 86*, 153-172. USA: Springer.

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

Burke, E.K., De Causmaecker, P., Petrovic, S., & Vanden Berghe, G. (2006). Metaheuristics for handling time interval coverage constraints in nurse scheduling. *Applied Artificial Intelligence, 20*(9), 743–766.

Burke, E.K., Curtois, T., Qu, R., & Berghe, G.V. (2007). *A Time pre-defined variable depth search for nurse rostering*. Technical Report NOTTCS-TR-2007-6. School of Computer Science and Information Technology, University of Nottingham.

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

Burke, E.K., Li, J., & Qu, R. (2010a). A hybrid model of integer programming and variable neighbourhood search for highly-constrained nurse rostering problems. *European Journal of Operational Research, 203*(2), 484–493.

Burke, E.K., Curtois, T., Qu, R., & Vanden Berghe, G. (2010b). A scatter search methodology for the nurse rostering problem. *Journal of the Operational Research Society, 61*(11), 1667–1679.

Burke, E.K., & Curtois, T. (2014). New approaches to nurse rostering benchmark instances. *European Journal of Operational Research, 237*, 71-81.

Cai, X., & Li, K. (2000). A genetic algorithm for scheduling staff of mixed skills under multi-criteria. *European Journal of Operational Research, 125*(2), 359–369.

Camargo, V.C.B., Toledo, F.M.B., & Almada-Lobo, B. (2014). HOPS – hamming-oriented partition search for production planning in the spinning industry. *European Journal of Operational Research, 234*, 266-277.

Caserta, M., & Voß, S. (2009). Metaheuristics: Intelligent Problem Solving. In Maniezzo V., Stützle T., Voß S. (Eds). *Matheuristics. Annals of Information Systems, 10*, 1-32. USA: Springer.

Černý, V. (1985). Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications, 45*(1), 41-51.

Ceschia, S., & Schaerf, A. (2011). Local search and lower bounds for the patient admission scheduling problem. *Computers & Operations Research, 38*, 1452-1463.

Ceschia, S., & Schaerf, A. (2012). Modeling and solving the dynamic patient admission scheduling problem under uncertainty. *Artificial Intelligence in Medicine, 56*, 199-205.

Ceschia, S., & Schaerf, A. (2016a). Dynamic patient admission scheduling with operating room constraints, flexible horizons, and patient delays. *Journal of Scheduling,* 19(4), 377–389.

Ceschia, S., & Schaerf, A. (2016b). *Patient admission scheduling problem under uncertainty website.* Retrieved December 12, 2016, from http://satt.diegm.uniud.it/%20index.php?page=pasu.

Ceschia, S., & Schaerf, A. (2017). *Dynamic patient admission scheduling website*. Retrieved January 31, 2017, from https://bitbucket.org/satt/or-pas.

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

Chen, P.S., & Lin, M.H. (2017). Development of simulation optimization methods for solving patient referral problems in the hospital-collaboration environment, *Journal of Biomedical Informatics, 73,* 148-158.

Cipriano, R., Gaspero, L.D., & Dovier, A. (2006). Hybrid approaches for rostering: A case study in the integration of constraint programming and local search. In: Almeida F. et al. (Eds.). *Hybrid Metaheuristics - Lecture Notes in Computer Science, 4030*. Germany: Springer-Verlag.

Curtois, T., & Qu, R. (2014). *Computational results on new staffscheduling bench-mark instances*. Technical Report 06-Oct-2014. ASAP Research Group, School of Computer Science, University of Nottingham.

Curtois, T., & Qu, R. (2018). *Employee shift scheduling benchmark data sets*. Retrieved February 27, 2018, from http://www.schedulingbenchmarks.org.

Darmoni, S.J., Fajner, A., Mahé, N., Leforestier, A., Vondracek, M., Stelian, O., & Baldenweck, M. (1995). Horoplan: computer-assisted nurse scheduling using constraint-based programming. *Journal of the Society for Health Systems, 5*(1), 41–54.

De Araujo, S.A., Arenales, M.N., & Clark, A.R. (2007). Joint rolling-horizon scheduling of materials processing and lot-sizing with sequence-dependent setups. *Journal of Heuristics, 13*(4), 337-358.

Della Croce, F., & Salassa, F. (2014). A variable neighborhood search based matheuristic for nurse rostering problems. *Annals of Operations Research, 218*(1), 185–199.

Demeester, P., Berghe, G.V., & De Causmaecker, P. (2008). Applying a local search algorithm to automatically assign patients to beds. *Proceedings of the 22nd conference on quantitative methods for decision making* (pp. 35-36).

Demeester, P., Souffriau, W., De Causmaecker, P., & Berghe, G.V. (2010). A hybrid tabu search algorithm for automatically assigning patients to beds. *Artificial Intelligence in Medicine, 48*(1), 61-70.

Demeester, P. (2016). *Patient admission scheduling website.* Retrieved July 31, 2016, from https://people.cs.kuleuven.be/~wim.vancroonenburg/pas.

Dillenberger, C., Escudero, L.F., Wollensak, A., & Zhang W. (1994). On practical resource allocation for production planning and scheduling with period overlapping setups. *European Journal of Operational Research, 75*(2), 275–86.

Dorneles, A.P., De Araujo, O.C.B., & Buriol, L.S. (2014). A fix-and-optimize heuristic for the high school timetabling problem. *Computers & Operations Research, 52*(A), 29–38.

Dowsland, K.A. (1998). Nurse scheduling with tabu search and strategic oscillation. *European Journal of Operational Research, 106*(2), 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*(7), 825–833.

Dumrongsiri, A. & Chongphaisal, P. (2018). Nurse scheduling in a hospital emergency department: A case study at a Thai university hospital. *Songklanakarin Journal of Science and Technology, 40*(1), 187-196.

Easton, F.F., & Mansour, N. (1999). A distributed genetic algorithm for deterministic and stochastic labor scheduling problems. *European Journal of Operational Research, 118*(3), 505–523.

Eberhart, R.C., & Kennedy, J. (1995). A new optimizer using particle swarm theory. *In Proceedings of the Sixth International Symposium on Micro Machine and Human Science,* US, 4-6 October 1995, 39–43.

Farasat, A. & Nikolaev, A.G. (2016). Signed social structure optimization for shift assignment in the nurse scheduling problem. *Socio-Economic Planning Sciences, 56,* 3-13.

Federgruen, A., Meissner, J., & Tzur, M. (2007). Progressive interval heuristics for multi-item capacitated lot-sizing problems. *Operations Research, 55*(3), 490–502.

Ghaderi, A., & Jabalameli, M.S. (2013). Modeling the budget-constrained dynamic incapacitated facility location-network design problem and solving it via two efficient heuristics: a case study of health care. *Mathematical and Computer Modelling, 57*(3-4), 382-400.

Giglio, D., Paolucci, M., & Roshani, A. (2017). Integrated lot sizing and energy-efficient job shop scheduling problem in manufacturing/remanufacturing systems. *Journal of Cleaner Production*, 148, 624-641.

Gintner, V., Kliewer, N., & Suhl, L. (2005). Solving large multiple-depot multiple-vehicle–type bus scheduling problems in practice. *OR Spectrum, 27*(4), 507–523.

Girbea, A., Suciu, C., & Sisak, F. (2011). Design and implementation of a fully automated planner-scheduler constraint satisfaction problem. *In Proceedings of 6th IEEE International Symposium on Applied Computational Intelligence and Informatics*, Romania, 19-21 May 2011, 477-482.

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

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.

Goren, H.G., Tunali, S., & Jans, R. (2012). A hybrid approach for the capacitated lot sizing problem with setup carryover. *International Journal of Production Research, 50*(6), 1582–1597.

Granja, C., Almada-Lobo, B., Janela, F., Seabra, J., & Mendes, A. (2014). An optimization based on simulation approach to the patient admission scheduling problem using a linear programming algorithm. *Journal of Biomedical Informatics, 52*, 427–437.

Guido, R., Groccia, M.C., & Conforti, D. (2018). An efficient matheuristic for offline patient-to-bed assignment problems. *European Journal of Operational Research, 268*(2), 486-503.

Guimaraes, L., Klabjan, D., & Almada-Lobo, B. (2013). Pricing, relaxing and fixing under lot sizing and scheduling. *European Journal of Operational Research, 230*(2), 399-411.

Gutjahr, W.J., & Rauner, M.S. (2007). An ACO algorithm for a dynamic regional nurse-scheduling problem in Austria. *Computers & Operations Research, 34*(3), 642–666.

Hammouri, A.I., & Alrifai, B., (2014). Investigating biogeography based optimisation for patient admission scheduling problems. *Journal of Theoretical and Applied Information Technology, 70*(3).

Helber, S., & Sahling, F. (2010). A fix-and-optimize approach for the multi-level capacitated lot sizing problem. *International Journal of Production Economics, 123*(2), 247–256.

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

James, R.J., & Almada-Lobo, B. (2011). Single and parallel machine capacitated lot sizing and scheduling: new iterative MIP-based neighborhood search heuristics. *Computers and Operations Research, 38*(12), 1816-1825.

Jaradat, G.M., Al-Badareen, A., Ayob, M., Al-Smadi, M., Al-Marashdeh, I., Ash-Shuqran, M., & Al-Odat, E. (2018). Hybrid elitist-ant system for nurse rostering problem. Journal of King Saud University - Computer and Information Sciences, In Press.

Kawanaka, H., Yamamoto, K., Yoshikawa, T., Shinigi, T., & Tsuruoka, S. (2001). Genetic algorithm with the constraints for nurse scheduling problem. *In Proceedings of Congress on Evolutionary Computation*, South Korea, 27-30 May 2001, 1123–1130.

Kelly, J.D., & Mann, J.L. (2004). Flowsheet decomposition heuristic for scheduling: a relax-and-fix method. *Computers & Chemical Engineering, 28*(11), 2193-2200.

Kennedy, J., & Eberhart, R.C. (1997). A discrete binary version of the particle swarm algorithm. *In Proceedings of the world multiconference on systemics, cybernetics and informatics*, US, 12-15 October 1997, 4104-4108.

Kennedy, J., Eberhart, R.C., & Shi, Y. (2001). Swarm intelligence. *In The Morgan Kaufmann Series in Artificial Intelligence*, San Francisco: Morgan Kaufmann.

Kifah, S., & Abdullah, S. (2015). An adaptive non-linear great deluge algorithm for the patient-admission problem. *Information Sciences, 295*, 573–585.

Kirkpatrick, S., Gelatt Jr., C.D., & Vecchi, M.P. (1983). Optimization by simulated annealing. *Science, 220*(4598), 671–80.

Liang, Z., He, Y., Wu, T., & Zhang, C. (2015). An informative column generation and decomposition method for a production planning and facility location problem. *International Journal of Production Economics, 170*, 88-96.

Liang, B., & Turkcan, A. (2016). Acuity-based nurse assignment and patient scheduling in oncology clinics. *Health Care Management Science, 19,* 207-226.

Lim, G.J., Mobasher, A., Bard, J.F., & Najjargashi, A. (2016). Nurse scheduling with lunch break assignments in operating suites. *Operations Research in Health Care, 10,* 35-48.

Lu, Z., & Hao, J.K. (2012). Adaptive neighborhood search for nurse rostering. *European Journal of Operational Research, 218*(3), 865–876.

Lusby, R.M., Schwierz, M., Range, T.M., & Larsen, J. (2016). An adaptive large neighborhood search procedure applied to the dynamic patient admission scheduling problem. *Artificial Intelligence in Medicine, 74*, 21-31.

M'Hallah, R., & Alkhabbaz, A. (2013). Scheduling of nurses: a case study of a Kuwaiti health care unit. *Operations Research for Health Care, 2*(1-2), 1–19.

Maenhout, B., & Vanhoucke, M. (2007). An electromagnetic meta-heuristic for the nurse scheduling problem. *Journal of Heuristics, 13*(4), 359–385.

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 longer-term nursing staff allocation problems. *Omega, 41*(2), 485–499.

Martin, S., Ouelhadj, D., Smet, P., Berghe, G.V., & Özcan, E. (2013). Cooperative search for fair nurse rosters. *Expert Systems with Applications, 40*, 6674-6683.

Martinelly, C.D., & Meskens, N. (2017). A bi-objective integrated approach to building surgical teams and nurse schedule rosters to maximise surgical team affinities and minimise nurses' idle time. *International Journal of Production Economics, 191*, 323-334.

Meyer auf'm Hofe, H. (2000). Solving rostering tasks as constraint optimization. In E. K. Burke & W. Erben (Eds.). *Selected papers from the third international* *conference on practice and theory of automated timetabling - Lecture Notes in Computer Science, 2079*, 191–212. Germany: Springer-Verlag.

Moosavi, A. & Ebrahimnejad, S. (2018). Scheduling of elective patients considering upstream and downstream units and emergency demand using robust optimization. *Computers & Industrial Engineering, 120*, 216-233.

Moreira, F.N., Amorim, P., Guimaraes, P., & Lobo, B.A. (2015). A long-haul freight transportation problem: Synchronizing resources to deliver requests passing through multiple transshipment locations. *European Journal of Operational Research, 248*, 487-506.

Moreno, A., Alem, D., & Ferreira, D. (2016). Heuristic approaches for the multiperiod location-transportation problem with reuse of vehicles in emergency logistics. *Computers & Operations Research, 69*, 79-96.

Oldenkamp, J.H. (1996). Quality in fives: on the analysis, operationalization and application of nursing schedule quality. *University of Groningen Doctor of Philosophy Dissertation*, 217.

Ouhimmou, M., D’Amours, S., Beauregard, R., Ait-Kadi, D., & Chauhan, S.S. (2008). Furniture supply chain tactical planning optimization using a time decomposition approach. *European Journal of Operational Research, 189*(3), 952-970.

Pochet, Y., & Wolsey, L. (2006). Mixed integer programming algorithms. In: Production planning by mixed integer programming. *Springer Series in Operations Research and Financial Engineering*, 77-113. USA: Springer.

Pochet, Y., & Warichet, F. (2008). A tighter continuous time formulation for the cyclic scheduling of a mixed plant. *Computers & Chemical Engineering, 32*(11), 2723-2744.

Qiu, Y., Wang, L., Xu, X., Fang, X., & Pardalos, P.M. (2018). Formulations and branch-and-cut algorithms for multi-product multi-vehicle production routing problems with startup cost. *Expert Systems with Applications, 98*, 1-10.

Qu, R., & He, F. (2008). A hybrid constraint programming approach for nurse rostering problems. In T. Allen, R. Ellis, M. Petridis, M. (Eds.). *Applications and Innovations in Intelligent Systems XVI*, 211–224. UK: Springer-Verlag.

Rahimian, E., Akartunali, K., & Levine, J. (2017a). A hybrid integer programming and variable neighbourhood search algorithm to solve nurse rostering problems. *European Journal of Operational Research, 258*, 411-423.

Rahimian, E., Akartunali, K., & Levine, J. (2017b). A hybrid integer and constraint programming approach to solve nurse rostering problems. *Computers & Operations Research, 82*, 83-94.

Range, T.M., Lusby, R.M., & Larsen, J. (2014). A column generation approach for solving the patient admission scheduling problem. *European Journal of Operational Research, 235*, 252-262.

Roshani, A., Giglio, D., & Paolucci, M. (2017). A relax-and-fix heuristic approach for the capacitated dynamic lot sizing problem in integrated manufacturing/remanufacturing systems. *IFAC-PapersOnLine, 50*(1), 9008-9013.

Sahling, F., Buschkühl, L., Tempelmeier, H., & Helber, S. (2009). Solving a multi-level capacitated lot sizing problem with multi-period setup carry-over via a fix-and-optimize heuristic. *Computers & Operations Research, 36*(9), 2546–2553.

Santos, H.G., Toffolo, T.A.M., Gomes, R.A.M., & Ribas, S. (2016). Integer programming techniques for the nurse rostering problem. *Annals of Operations Research, 239*, 225-251.

Seeanner, F., Almada-Lobo, B., & Meyr, H. (2013). Combining the principles of variable neighborhood decomposition search and the fix & optimize heuristic to solve multi-level lot-sizing and scheduling problems. *Computers and Operations Research, 40*(1), 303-317.

Sel, C., & Bilgen, B. (2014). Hybrid simulation and MIP based heuristic algorithm for the production and distribution planning in the soft drink industry. *Journal of Manufacturing Systems, 33*, 385–399.

Shi, Y., Eberhart, R.C. (1998). A modified particle swarm optimizer. *In Proceedings of the IEEE congress on evolutionary computation*, US, 12-15 October 1998, 69–173.

Silva, T.A.O., Souza, M.C.D., Saldanha, R.R., & Burke, E.K. (2015). Surgical scheduling with simultaneous employment of specialised human resources. *European Journal of Operational Research, 245*, 719-730.

Smet, P., Bilgin, B., De Causmaecker, P., & Vanden Berghe, G. (2014). Modelling and evaluation issues in nurse rostering. *Annals of Operations Research, 218*(1), 303–326.

Soto, R., Crawford, B., Bertrand, R., & Monfroy, E. (2013). Modeling NRPs with soft and reified constraints. *AASRI Procedia, 4*, 202–205.

Stadtler, H. (2003). Multilevel lot sizing with setup times and multiple constrained resources: internally rolling schedules with lot-sizing windows. *Operations Research 200351:3*, 487-502.

Stadtler, H., & Sahling, F. (2013). A lot-sizing and scheduling model for multi-stage flow lines with zero lead times. *European Journal of Operational Research, 225*(3), 404-419.

Stølevik, M. , Nordlander, T.E. , Riise, A. , & Frøyseth, H. (2011). A hybrid approach for solving real-world nurse rostering problems. In: Lee J. (Eds.). *Principles and Practice of Constraint Programming – Lecture Notes in Computer Science, 6876*. Germany: Springer-Verlag.

Tassopoulos, I.X., Solos, I.P., & Beligiannis, G.N. (2015). A two-phase adaptive veriable neighborhood approach for nurse rostering. *Computers & Operations Research, 60,* 150-169.

Toledo, C.F.M., Arantes, M.D.S., Hossomi, M.Y.B., Franca, P.M., & Akartunali, K. (2015). A relax-and-fix with fix-and-optimize heuristics applied to multi-level lot-sizing problems. *Journal of Heuristics, 21*, 687-717.

Toledo, C.F.M., Arantes, M.D.S., Hossomi, M.Y.B., & Lobo, B.A. (2016). Mathematical programming-based approaches for multi-facility glass container production planning. *Computers & Operations Research, 74*, 92-107.

Tsai, C.-C., & Li, S.H. (2009). A two-stage modeling with genetic algorithms for the nurse scheduling problem. *Expert Systems with Applications, 36*(5), 9506–9512.

Turhan, A.M., & Bilgen, B. (2017). Mixed integer programming based heuristics for the patient admission scheduling problem. *Computers and Operations Research, 80,* 38-49.

Uggen, K.T., Fodstad, M., & Nørstebø, V.S. (2013). Using and extending fix-and-relax to solve maritime inventory routing problems. *TOP*, *21*(2), 355-377.

Valouxis, C., Gogos, C., Goulas, G., Alefragis, P., & Housos, E. (2012). A systematic two phase approach for the nurse rostering problem. *European Journal of Operational Research, 219*(2), 425–433.

Vanden Berghe, G. (2002). *An advanced model and novel meta-heuristic solution methods to personnel scheduling in healthcare*. Phd Thesis, KU Leuven, Leuven.

Vermuyten, H., Rosa, J.N., Marques, I., Beliën, J., Barbosa-Póvoa, A. (2018). Integrated Staff Scheduling at a Medical Emergency Service: An Optimisation Approach. *Expert Systems with Applications*, 112, 62-76.

Warner, M., & Prawda, J. (1972). A mathematical programming model for scheduling nursing personnel in a hospital. *Management Science, 19*(4), 411–422.

Wei, W., Guimaraes, L., Amorim, P., & Lobo, B.A. (2016). Tactical production and distribution planning with dependency issues on the production process. *Omega*, 67, 99-114.

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.

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

Wu, T.H., Yeh, J.Y., & Lee, Y.M. (2015). A particle swarm optimization approach with refinement procedure for nurse rostering problem. *Computers & Operations Research, 54,* 52-63.

Xiang, W., Yin, J., & Lim, G. (2015). A short-term operating room surgery scheduling problem integrating multiple nurse roster constraints. *Artificial Intelligence in Medicine, 63,* 91-106.

Xiao, J., Zhang, C., Zheng, L., & Gupta, J.N. (2013). MIP-based fix-and-optimize algorithms for the parallel machine capacitated lot-sizing and scheduling problem. *International Journal of Production Research, 51*(16), 2011-2028.

Xiao, J., Yang, H., Zhang, C., Zheng, L., & Gupta, J.N.D. (2015). A hybrid Lagrangian-simulated annealing-based heuristic for the parallel-machine capacitated lot-sizing and scheduling problem with sequence-dependent setup times. *Computers & Operations Research*, *63*, 72-82.

Zhang, G., Nishi, T., Turner, S.D.O., Oga, K., & Li, X. (2017). An integrated strategy for a production planning and warehouse layout problem: Modeling and solution approaches. *Omega*, *68*, 85-94.

Zheng, Z., Liu, X., & Gong, X. (2017). A simple randomized variable neighbourhood search for nurse rostering. *Computers & Industrial Engineering*, *110*, 165-174.