Nurse Scheduling with Lunch Break Assignments in Operating Suites

Gino J. Lim* Arezou Mobasher † Jonathan F. Bard ‡ Amirhossein Najjarbashi §

Accepted for publication: July 2, 2016

Abstract

Motivated by the need to make frequent changes in operating suites, this paper presents a highly scalable and efficient solution framework for scheduling nurses in operating suites over the day. This framework consists of two core optimization models that are necessary for scheduling OR nurses in the clinic. The first model addresses the multi-objective optimization problem of assigning nurses to upcoming surgery cases based on their specialties and competency levels. The second model is designed to generate lunch break assignments for the nurses once their caseloads are determined. The latter problem has been largely overlooked by the research community despite its importance. Because the multi-objective model is too large to solve using commercial software, we developed both a column generation algorithm and a two-phase swapping heuristic to find feasible assignments in a fast manner. For both approaches, initial solutions are obtained with a restricted model and lunch breaks are scheduled in a post-processing step. Experiments were conducted to determine the value of the models and the performance of the algorithms using real data provided by MD Anderson Cancer Center in Houston, Texas. The results show that the two approaches can produce implementable daily schedules in a matter of minutes for instances with over 100 nurses, 50 surgery cases and 33 operating rooms.

Key words: nurse scheduling; lunch breaks; multi-objective programming; column generation; improvement heuristics

1 Introduction

The ability of healthcare systems to deliver high-quality, cost-effective care to an aging population is under assault by a worldwide shortage of nurses [29]. As the population ages, the demand for surgery has grown. Not having enough skilled nurses in clinical settings can have a significant negative impact on nurse retention rates, patient safety and healthcare outcomes [5, 6, 11]. Given the current situation, hospital managers are in dire need to maximize the utilization and retention of their nursing staff without jeopardizing job satisfaction.

*Department of Industrial Engineering, University of Houston, 4800 Calhoun Road, Houston, TX 77204, Email: ginolim@uh.edu

†Optym, Inc., Gainesville, FL

‡Graduate Program in Operations Research & Industrial Engineering, University of Texas - Austin

§Department of Industrial Engineering, University of Houston

Assigning each available nurse to the right place at the right time to do the right job is a major concern for healthcare organizations. Such organizations are typically divided into specialized units that house numerous job positions, each requiring a specific set of skills. This leads to a large number of possible work schedules when coupled with demand and case variability. To determine "optimal" schedules, one must consider nurse availability by skill level, their shift preferences, patient demand classifications and uncertainty (e.g., demand, case durations, and resource capacity). Additional considerations include regulatory and union requirements, working contract options, overtime, and break times during a shift, to name a few. Moreover, each unit in the hospital may have a host of individual rules and policies that play a role in staffing decisions.

Motivated by this need as well as the desire to avoid a heavy computational burden when generating solutions, the purpose of this paper is to present a series of models to support the timely construction of daily schedules for the nursing staff at surgery-centered hospitals. The work was done in consultation with MD Anderson Cancer Center in Houston, Texas, one of the largest cancer treatment facilities in the U.S. The primary model produces a daily roster that specifies the assignment of nurses to shifts in accordance with their skills and planned cases. A second model is used to adjust the corresponding schedules to allow for lunch breaks without disrupting the surgeries underway. As in most realistic situations, there are multiple objectives that must be weighed in the rostering process. The most prominent include the minimization of overtime and idle time, the minimization of changes in assignments during the day to accommodate breaks, and the maximization of case demand satisfaction in light of nurse competency levels and specialties. Solutions are constrained by shift options, contract details, and nurse availabilities. In particular, nurses are assigned to cases based on how closely their specialty and procedure competency match the nature of the case and the procedure requirements.

In previous work [22], we developed a solution pool method (SPM) and a modified goal programming method (MGPM) to produce daily schedules. However, we found both methods computationally challenging as they required the solution of large-scale MIPs at intermediate steps. Knowing that staff availability can change at any time during the day, the main contribution of our work centers on the computational efficiency of the proposed methodology. We developed two independent algorithms, both starting with the same feasible schedule derived from a third model, which is a restricted version of the original model. The first algorithm is based on column generation, and the second is a two-phase swapping heuristic that iteratively works towards reducing staff shortages, overtime and idle time. Updated schedules can be obtained in less than a few minutes as case lengths and staffing needs change over the day. After a solution is obtained, our lunch break model is called to ensure that lunch breaks are provided to all eligible nurses. The modeling of this problem has been largely overlooked by the research community and represents the second contribution of the paper.

The remainder of the paper is organized as follows. In Section 2, we review the most recent research on nurse scheduling. Section 3 introduces our optimization models for assigning nurses to different surgery cases and assuring that each nurse is given a lunch break when required. The two solution algorithms are discussed in Section 4 and partially illustrated with examples. Numerical results are presented in Section 5 for six data sets obtained from MD Anderson. Conclusions are drawn in Section 6.

2 Literature Review

Given the benefits that can be achieved with more efficient use of staffing resources, there has been a great deal of work directed at solving general shift and tour scheduling problems (e.g., see [15]). With respect to nurse scheduling and rostering problems, researchers have published surveys that cover the period from 1965 to 2004 [7, 10]. Since that time, dozens of additional studies have appeared in the literature presenting new models and solution methodologies for tackling a variety of related problems. Much of this work has centered on integer programming-based methods with the objective of either minimizing cost or maximizing nurse preferences. Planning horizons considered can be as short as a shift or as long as a year [1, 3, 8, 26].

In contrast to the short-term problem addressed in this paper, most of the work on nurse scheduling has focused on monthly (mid-term) scheduling. Some relevant papers can be found in [20, 23, 26]. One of the few studies that considered daily scheduling was undertaken in [2], where the authors developed a reactive planning model for dealing with staff shortages for the 24 hours. Taking a hospital-wide view, the model was aimed at minimizing the costs of covering all shifts for the current day by considering the use of overtime, agency nurses, pools, and canceling days off. Solutions were found with a branch and price algorithm in conjunction with mixed-integer rounding cuts to tighten up the relaxed feasible region of the master problem.

As mentioned in Section 1, due to specific nurse restrictions and the complexity of surgery procedures, scheduling nurses in operating suites should be considered separately from scheduling nurses in other areas. The surgery scheduling process of elective cases can be classified into four planning phases [9]. First, one determines how much operating room time is assigned to the different surgeons or surgical groups. This phase is often referred to as case mix planning and is viewed as a strategic consideration. The second phase, which is tactically oriented, concerns the development of a master surgery schedule, i.e., defining the number and type of operating rooms available, the hours that rooms will be open, and the surgeons or surgical groups to whom the operating room time is assigned. In the third phase, individual patients or cases are scheduled on a daily base. In the fourth phase, the surgery schedule is monitored online and rescheduling is considered when the current schedule is disrupted due to uncertainties. The nurse scheduling problem is present in the first three phases on strategic, tactical and operational levels. This paper deals with the nurse scheduling problem on an operational level, i.e., daily assignment of nurses to surgery cases.

Beliën and Demeulemeester [4] tackled an integrated nurse and surgery scheduling problem using integer programming. They enumerated all possible ways of assigning operating blocks to different surgeons subject to individual preferences, surgery demand and capacity restrictions. Solutions were found with a column generation algorithm. To generate columns, they implemented two types of pricing algorithms: the first generates a new roster line using a dynamic programming recursion and the second generates a new surgery schedule using a mixed-integer programming (MIP) scheme. Van Huele and Vanhoucke [31] combined three types of constructive heuristics with two priority rule classes to solve an integrated physician and surgery scheduling problem. They proposed a goal programming model for the problem when open scheduling strategy is used. The objectives were to balance the physicians' workload while satisfying their preferences subject to constraints on breaks between shifts, skill levels, and on-call nurses. Xiang et al. [35] investigated an integrated daily surgery and nurse scheduling problem using a mixed-integer nonlinear programming model. The model considered a variety of nurse constraints such as role, specialty, qualification and availability.

To find solutions, they developed a modified ant colony optimization (ACO) algorithm with a two-level ant graph.

The above mentioned papers have mainly focused on the surgery scheduling problem while incorporating nurse scheduling constraints into the model. Wong et al. [33] studied a nurse scheduling problem for an emergency department in which seniority, qualifications, preferences, and legal regulations were taken into account. As is the norm, each case required a proper mix of manpower with different skill sets and proficiency levels. A two-stage approach combining a shift assignment heuristic and sequential local search was developed to find feasible solutions with the objective of minimizing the violations of soft constraints. Mobasher et al. [22] proposed a multi-objective MIP model for the daily scheduling of nurses in operating suites. The overall goal was to assign the nurses to surgery cases based on their specialties and competency levels, subject to a series of hard and soft constraints related to nurse satisfaction, idle time, overtime, and role changes during a shift. They developed a solution pool method as well as a modified goal programming approach to find solutions.

Uncertainty is an inherent characteristic of patient care problems. Van den Bergh at al. [30] proposed three main classes for the uncertainty in personnel scheduling problems: uncertainty of demand, arrivals, and capacity. Uncertainty of process times (e.g., surgery durations) can also be considered as another main class. In fact, provider time with the patient is a prevalent source of uncertainty in planning and scheduling problems in healthcare. Gutjahr and Rauner [17] proposed an ACO algorithm to solve a dynamic nurse scheduling problem for a group of 15 hospitals in Vienna, Austria. They considered uncertainty of demand and arrivals, along with a variety of constraints related to working patterns, nurse qualifications and preferences, management preferences, and the cost of resources. They simulated the working environment over a four-week period to compare the performance of the proposed ACO with a simulated annealing (SA) algorithm. Also, several authors have used stochastic programming methods to solve nurse staffing and scheduling problems characterized by demand uncertainty [18, 25, 27].

Scheduling breaks to help staff maintain their concentration while working is a common concern in many areas such as air traffic control, security checking, assembly lines, and healthcare delivery. Despite its critical importance, especially pertaining to breaks for nurses, break scheduling in healthcare has been largely ignored by the research community. In contrast, there has been a moderate amount of work in break scheduling for supervisory personnel. Widl and Musliu [32] in their work, for example, considered several personnel-related constraints, such as legal requirements and ergonomic limitations, and developed two variations of a memetic algorithm with genetic operators to find solutions.

Although a vast amount of literature exists on nurse scheduling and operating room scheduling, the combined problem has not been well addressed. The absence of an integrated approach ignores the critical influence that nurses have on operating room efficiency and patient outcomes. To the best of our knowledge, no models exist that consider the assignment of nurses to surgery cases taking into account the lunch break requirement. In the next section, we present both a nurse assignment model and a lunch break model to fill this void.

3 Nurse Scheduling Problem in an Operating Suite

A number of individual and systemic factors must be taken into account when assigning nurses to an operating suite. As in most healthcare settings, unless the interaction among all procedural and personal factors are considered, the resultant schedules may not be practical or make the most efficient use of the staffing resources. For our problem, the major factors include case specialties, procedure complexities, nurse skill levels, and lunch breaks. Each is discussed below.

Surgery case. The building blocks of an OR schedule are the surgery cases, where each is defined as a series of surgical procedures performed on one patient in one operating room in a day. Elective cases are scheduled in advance while emergency cases occur on an as-needed basis. We define *surgery duration* as the time required to finish a case starting with the arrival of the patient to the operating room, performing the surgery, and finishing with the transfer of the patient to a post-anesthesia care unit. We define *surgery demand* as the number of nurses required for each case during each time period of the day in each role. The demand is a function of the case complexity and the service performing the operation. Procedures can be classified as 'simple,' 'moderate' or 'complex'.

Nurse categories. Nurses can be categorized in different ways based on their skill level, experience, education, knowledge, and certification. The most commonly recognized roles are *circulator* and *scrub* [22]. Nurses are assigned to surgeries of different complexities based on their skill level and experience. It is assumed that nurses who are qualified to work on harder procedures can also work on easier ones.

Shift limitations. In all hospitals, the day is nominally divided into shifts that can span anywhere from 8 to 12 hours. Each shift has its own regulations such as break hours, overtime rules, and on-call obligations. Operating suites generally have their own staffing and shift restrictions, along with regulatory and union requirements that circumscribe nurse schedules. Moreover, nurses cannot leave a surgery to take a break unless the case is finished or someone is available to relieve them. These restrictions add another layer of difficulty when trying to generate implementable schedules for nurses.

3.1 Nurse Assignment Model

In this paper, we assume that the decision maker has complete information on the number of available nurses, their specialties and procedure competencies, their shift assignments, their role abilities, surgery schedules, surgery durations, surgery specialties and procedural complexities, break hours, and contract specifications. These parameters are defined as model inputs. Some hospitals utilize computer tools to estimate surgery durations based on historical data as we did in this paper [34], whereas others rely on surgeon's experience to generate time estimates.

At the hospital that served as a backdrop for our study, there is no single objective that guides the process of generating schedules. Ideally, nurse assignments should minimize overtime and idle time as well as meet surgery demand. In practice, though, it is not mandatory to satisfy all demand and to finish all cases on time so in our model, violations of these goals are penalized.

This has the effect of increasing the total cost of the schedule. The nurse assignment model (NAM) presented below is designed to produce daily nurse schedules that match their skill levels with case requirements.

3.1.1 Notation and assumptions

We assume that each working day can be divided into equal time intervals (e.g. 30-minute or 1 hour). All shifts include regular shift hours and authorized overtime. The latter are additional hours that a nurse can be assigned to a surgery case when there is no other means of satisfying the demand. This may occur when a case is not finished by the end of the regular shift and there is no available nurse to relieve the working nurse, or when the demand for an in-progress case is not satisfied. We also assume that in the operating suites being modeled, only circulators and scrubs are required. These roles are typically filled by registered nurses (RNs) and scrub techs, respectively, but RNs can perform both if necessary. Each nurse is limited to cases that match his or her skill level and specialty. The competency level of a nurse must be at least as high as the complexity level of the case that is assigned. Nurses with a higher competency can perform surgery procedures with lower complexity but not vice versa; that is, nurse skill levels follow a hierarchical pattern and are subject to downgrading [14]. Numerous articles in workforce planning have considered the notion of hierarchical skills [13, 21].

In developing our NAM, we make use of the following notation.

Indices

$\mathcal{I}$ set of available nurses

$\mathcal{J}$ set of available ORs

$\mathcal{K}$ set of roles that are required for each surgery case (1: RN, 2: scrub tech)

$\mathcal{Q}$ set of specialties

$\mathcal{C}$ set of cases scheduled for surgery on the current day

$\mathcal{S}$ set of available shifts

$\mathcal{P}$ set of competency/complexity levels

$\mathcal{H}$ time intervals in a working day

Parameters

P^1_{is} 1 if nurse $i \in \mathcal{I}$ is working in shift $s \in \mathcal{S}$, 0 otherwise

P^2_{ikqp} 1 if nurse $i \in \mathcal{I}$ can perform role $k \in \mathcal{K}$ for specialty $q \in \mathcal{Q}$ with competency level $p \in \mathcal{P}$, 0 otherwise

P^3_{cj} 1 if case $c \in \mathcal{C}$ is scheduled for surgery in OR $j \in \mathcal{J}$, 0 otherwise

P^4_{cqph} 1 if case $c \in \mathcal{C}$ requires specialty $q \in \mathcal{Q}$ and has procedural complexity $p \in \mathcal{P}$ in time interval $h \in \mathcal{H}$, 0 otherwise

P^5_{ckh} required number of nurses for case $c \in \mathcal{C}$ who can perform role $k \in \mathcal{K}$ in time interval $h \in \mathcal{H}$

P^6_{ch} 1 if case $c \in \mathcal{C}$ is in progress during time interval $h \in \mathcal{H}$, 0 otherwise

P^7_{c} case $c \in \mathcal{C}$ duration (length of surgery)

P^8_{sh}	1 if shift $s \in \mathcal{S}$ contains time interval $h \in \mathcal{H}$ as regular working hours, 0 otherwise
P^9_{sh}	1 if shift $s \in \mathcal{S}$ contains time interval $h \in \mathcal{H}$ as authorized overtime, 0 otherwise
M	sufficiently large number

Decision variables Our aim is to determine which nurse should be assigned to which surgery cases, during which time intervals, and which role they will perform. Accordingly, the decision variables are defined as:

y_{ickh}	1 if nurse $i \in \mathcal{I}$ is assigned to case $c \in \mathcal{C}$ to perform role $k \in \mathcal{K}$ in time interval $h \in \mathcal{H}$, 0 otherwise
x_{ick}	1 if nurse $i \in \mathcal{I}$ is assigned to case $c \in \mathcal{C}$ to perform role $k \in \mathcal{K}$, 0 otherwise

3.1.2 Constraints

Our model contains both hard and soft constraints. Each set is applicable for all nurses.
Hard constraints cannot be violated under any circumstances. Examples include shift regulations and nurse skill requirements.

$$\sum_{c \in \mathcal{C}} \sum_{k \in \mathcal{K}} y_{ickh} \leq 1, \qquad \forall i \in \mathcal{I}, h \in \mathcal{H} \tag{1}$$

$$y_{ickh} \leq \sum_{s \in S} \left(P^1_{is} \cdot (P^8_{sh} + P^9_{sh})\right), \qquad \forall i \in \mathcal{I}, c \in \mathcal{C}, k \in \mathcal{K}, h \in \mathcal{H} \tag{2}$$

$$\sum_{c \in \mathcal{C}} \sum_{k \in \mathcal{K}} \sum_{h \in \mathcal{H}} y_{ickh} \leq \sum_{s \in \mathcal{S}} \sum_{h \in \mathcal{H}} \left(P^1_{is} \cdot (P^8_{sh} + P^9_{sh})\right), \qquad \forall i \in \mathcal{I} \tag{3}$$

$$y_{ickh} \leq P^6_{ch} \cdot \sum_{q \in \mathcal{Q},} \sum_{p \in \mathcal{P}} \left(P^4_{cqph} \cdot P^2_{ikqp}\right), \qquad \forall i \in \mathcal{I}, c \in \mathcal{C}, k \in \mathcal{K}, h \in \mathcal{H} \tag{4}$$

$$\sum_{i \in \mathcal{I}} y_{ickh} \geq P^6_{ch}, \qquad \forall c \in \mathcal{C}, k \in \mathcal{K}, h \in \mathcal{H} \tag{5}$$

$$\sum_{h \in \mathcal{H}} y_{ickh} \leq M \cdot x_{ick}, \qquad \forall i \in \mathcal{I}, c \in \mathcal{C}, k \in \mathcal{K} \tag{6}$$

$$\sum_{k \in \mathcal{K}} x_{ick} \leq 1, \qquad \forall i \in \mathcal{I}, c \in \mathcal{C} \tag{7}$$

Constraints (1) ensure that each nurse is assigned to at most one case in each time interval and performs a single role. Constraints (2) and (3) ensure that in each shift, cases will be assigned to the nurses who are working during their regular or authorized overtime hours. In addition, the total working hours for a nurse each day must be less than his or her total regular and overtime working hours. Each nurse is allowed to work at most 4 hours overtime and 12 hours in total. Constraints (4) and (5) allow nurses to be assigned to a case only if their skill level is high enough to handle the specialty requirements, and they have sufficient competency to deal with its procedural complexities. Moreover, if case c is not in progress during time interval h, then no nurse will be assigned to it. Nurses must perform the same role for the entire duration of a case rather than rotating from one role to another. This is enforced with constraints (6) and (7).

Soft constraints are those that we would like to satisfy, but may not be able to without creating an infeasible problem. To formulate the soft constraints, we define a deviation variable and an auxillary variable for each. As indicated in the next subsection, the objective function is designed to collectively minimize all deviations.

de_{ckh} demand undercoverage during case $c \in \mathcal{C}$ for role $k \in \mathcal{K}$ in time interval $h \in \mathcal{H}$, 0 otherwise

$\mathcal{DE}$ maximum staff shortage for any case $c \in \mathcal{C}$ and role $k \in \mathcal{K}$, i.e., $max_{c,k}\{\sum_h de_{ckh}\}$

dev^1_{ih} 1 if nurse $i \in \mathcal{I}$ is idle in time interval $h \in \mathcal{H}$, 0 otherwise

$\mathcal{DS}$ maximum number of non-consecutive idle intervals for any nurse $i \in \mathcal{I}$

dev^2_{ih} 1 if nurse $i \in \mathcal{I}$ is assigned overtime in time interval $h \in \mathcal{H}$, 0 otherwise

$\mathcal{DF}$ maximum amount of overtime assigned to any nurse $i \in \mathcal{I}$

X_{ij} 1 if nurse $i \in \mathcal{I}$ is assigned to room $j \in \mathcal{J}$, 0 otherwise

$\mathcal{X}$ maximum number of room assignments for any nurse $i \in \mathcal{I}$

nc_{ic} 1 if nurse $i \in \mathcal{I}$ is assigned to case $c \in \mathcal{C}$, 0 otherwise

$\mathcal{NCT}$ maximum number of case assignments given to any nurse $i \in \mathcal{I}$

cd_{ic} 1 if the assignment of nurse $i \in \mathcal{I}$ to case $c \in \mathcal{C}$ is broken, 0 otherwise. An assignment is broken if the nurse is assigned to another case prior to the termination of the current case.

$\mathcal{CDT}$ maximum number of times an individual assignment to a case can be broken for any nurse $i \in \mathcal{I}$

The soft constraints are formulated as follows:

$$\sum_{i\in\mathcal{I}} y_{ickh} + de_{ckh} \geq P^5_{ckh} \cdot P^6_{ch}, \qquad \forall c \in \mathcal{C}, k \in \mathcal{K}, h \in \mathcal{H} \tag{8}$$

$$\mathcal{DE} \geq \sum_{h\in\mathcal{H}} de_{ckh}, \qquad \forall c \in \mathcal{C}, k \in \mathcal{K} \tag{9}$$

$$-\, dev^1_{ih} \leq \sum_{c\in\mathcal{C}}\sum_{k\in\mathcal{K}} y_{ick(h+1)} - \sum_{c\in\mathcal{C}}\sum_{k\in\mathcal{K}} y_{ickh} \leq dev^1_{ih}, \qquad \forall i \in \mathcal{I}, h \in \mathcal{H} \tag{10}$$

$$\mathcal{DS} \geq \sum_{h\in\mathcal{H}} dev^1_{ih}, \qquad \forall i \in \mathcal{I} \tag{11}$$

$$\sum_{c\in\mathcal{C}}\sum_{k\in\mathcal{K}}\sum_{h\in\mathcal{H}} \left(y_{ickh} \cdot P^3_{cj}\right) \leq M \cdot X_{ij}, \qquad \forall i \in \mathcal{I}, j \in \mathcal{J} \tag{12}$$

$$\sum_{j\in\mathcal{J}} X_{ij} \leq \mathcal{X}, \qquad \forall i \in \mathcal{I} \tag{13}$$

$$\sum_{s\in\mathcal{S}} P^1_{is} \cdot P^9_{sh} \cdot \sum_{c\in\mathcal{C}}\sum_{k\in\mathcal{K}} y_{ickh} \leq dev^2_{ih}, \qquad \forall i \in \mathcal{I}, h \in \mathcal{H} \tag{14}$$

$$\mathcal{DF} \geq \sum_{h\in\mathcal{H}} dev^2_{ih}, \qquad \forall i \in \mathcal{I} \tag{15}$$

$$\sum_{k\in\mathcal{K}}\sum_{h\in\mathcal{H}} y_{ickh} - P^7_c + M \cdot cd_{ic} + M \cdot (1 - nc_{ic}) \geq 0, \qquad \forall i \in \mathcal{I}, c \in \mathcal{C} \tag{16}$$

$$\sum_{k\in\mathcal{K}}\sum_{h\in\mathcal{H}} y_{ickh} \leq M \cdot nc_{ic}, \qquad \forall i \in \mathcal{I}, c \in \mathcal{C} \tag{17}$$

$$\sum_{c\in\mathcal{C}} nc_{ic} \leq \mathcal{NCT}, \qquad \forall i \in \mathcal{I} \tag{18}$$

$$\sum_{c\in\mathcal{C}} cd_{ic} \leq \mathcal{CDT}, \qquad \forall i \in \mathcal{I} \tag{19}$$

Although all demand should be met for each case in each time interval, this may not be possible due to a shortage of qualified nurses. In such circumstances, constraints (8) and (9) permit undercoverage. Moreover, it is preferred that nurses work continuously during their regular hours rather than having idle periods. Constraints (10) and (11) avoid these idle periods. By minimizing $\mathcal{DS}$ in the objective function, we assure that the maximum number of times that nurses work non-consecutive hours will be minimized along with idle time hours during the shift. Constraints (12) and (13) account for the preference that nurses work continuously in one operating room rather than moving around. By minimizing $\mathcal{X}$, we assure that the maximum number of ORs to which a nurse is assigned is reduced as much as possible. Constraints (14) and (15) record overtime. By minimizing $\mathcal{DF}$, we assure that the maximum number of times a nurse is working during overtime hours is as low as possible. Constraints (16)–(19) ensure that if a nurse is assigned to a surgery case, (s)he will stay for the entire duration of the case unless there is a greater need for that nurse elsewhere. Such a need may arise, for example, when limits on undercoverage or room assignments cannot be maintained without reassigning the nurse to another case prior to the termination of his or her current case. Also, by minimizing the maximum number of cases that a nurse can work, $\mathcal{NCT}$, we limit the movement of nurses between cases.

3.1.3 Objective function

As explained previously, soft constraints related to staff shortages, overtime, idle time, room changes, number of assignments, and broken assignments may be violated. The objective of the NAM is to minimize each violation (i.e., deviation) variable introduced in Section 3.1.2. In abbreviated terms, this can be written as

$$\text{Minimize } \{\mathcal{DE}, \mathcal{DF}, \mathcal{DS}, \mathcal{X}, \mathcal{NCT}, \mathcal{CDT}\} \tag{20}$$

Although trying to minimize a weighted sum of the violations in (20) is a common approach to dealing with multi-objective problems (e.g., see [22]), solving the corresponding MIP with commercial software proved to be too difficult. Instead, we developed an efficient algorithm that involved generating an initial feasible solution and improving each violation with an iterative procedure. The details are provided in Sections 4.1 and 4.2.

3.1.4 Strengthening the mixed-integer programming formulation

A common question when a big M is included in a constraint is what is the best value to use. Although an arbitrarily large value will ensure feasibility of the constraint, it may slow down the computations due to an unnecessarily large feasible region. To tighten the feasible region, we set M in constraints (6), (16), and (17) to $P^7_c \sum_{q,p,h} \left(P^4_{cqph} \cdot P^2_{ikqp}\right)$, P^7_c and $P^7_c \sum_{k,q,p,h} \left(P^4_{cqph} \cdot P^2_{ikqp}\right)$ respectively. For each case $c \in \mathcal{C}$ that *can be assigned* to a nurse $i \in \mathcal{I}$, (s)he cannot work more than the duration of case. The M in constraint (12) is also replaced by $\sum_{c,k,q,p,h} \left(P^4_{cqph} \cdot P^2_{ikqp} \cdot P^3_{cj} \cdot P^7_c\right)$.

3.2 Nurse Lunch Model

This model is applicable for nurses who are working during the lunch hour interval and require a break. A nurse is not allowed to leave the OR until the case on which she is working is finished or another nurse with the appropriate skill and competency is available to relieve her. Ideally, shift assignments and break assignments would be made together. However, this is not practical because NAM instances of realistic size are already too large to be solved exactly. Instead, we have developed a nurse lunch model (NLM) that takes as input the solution of the NAM and adjusts it to accommodate breaks.

3.2.1 Notation and assumptions

In general, nurses whose shifts start later in the day can first fill in for their colleagues who start early in the morning, and then start their cases. In practice, there are usually enough nurses starting the next shift to relieve all working nurses who need lunch break. Alternatively, nurses who are assigned to short cases can fill in for those working long cases when a lunch break is due. Those nurses who cannot be given a break during lunch period, will have their lunch break as soon as the cases they are assigned to are finished and they become idle. When staffing is tight, some nurses may not receive a break.

Indices

$\hat{\mathcal{I}}$ set of nurses who are working during the lunch period; $\hat{\mathcal{I}} \subset \mathcal{I}$

$\mathcal{I}'$ set of nurses who can provide an hour of relief during the lunch period; $\mathcal{I}' \subset \mathcal{I}$

$\mathcal{H}'$ set of hours during which lunch should be taken; $\mathcal{H}' \subset \mathcal{H}$

$\mathcal{L}$ lunch break periods; $l \in \mathcal{L}$

Parameters Several of the parameters introduced in Section 3.1.1 for the NAM are used in this model. In addition, we need to know who is assigned to which case during the lunch hours. This information is obtained from y_{ickh} after solving the NAM.

Decision variables The following binary variables are used to determine which nurse relieves nurse i and in which period l the break occurs.

$\zeta_{ii'l}$ 1 if nurse $i \in \mathcal{I}$ is relieved by nurse $i' \in \mathcal{I}'$ in lunch break period $l \in \mathcal{L}$, 0 otherwise

3.2.2 Constraints

Nurse i can be relieved during lunch break period l only when an available nurse i' with the same specialty and competency level is available to fill in for her in the OR. Constraints (21) ensure that this requirement is met for any substitution. Constraints (22) enforce the restriction that nurse i can only be relieved by nurse i' in one and only one of the lunch break periods. Finally, constraints (23) limit nurse i' to providing relief for at most one nurse during each of the lunch break periods.

$$\zeta_{ii'l} \leq \left(\sum_{q \in \mathcal{Q}, p \in \mathcal{P}} P^4_{cqph} \cdot P^2_{i'kqp} \right) \cdot y_{ickh}, \qquad \forall l \in \mathcal{L}, h \in \mathcal{H}', k \in \mathcal{K}, i' \in \mathcal{I}', i \in \hat{\mathcal{I}}, c \in \mathcal{C} \tag{21}$$

$$\sum_{\substack{i' \in \mathcal{I}' \\ l \in \mathcal{L}}} \zeta_{ii'l} \leq 1, \qquad \forall i \in \hat{\mathcal{I}} \tag{22}$$

$$\sum_{i \in \hat{\mathcal{I}}} \zeta_{ii'a} \leq 1, \qquad \forall i' \in I', l \in \mathcal{L} \tag{23}$$

3.2.3 Objective function

Our goal is to maximize the total number of nurses who can be given a break during the lunch hours.

$$\text{Maximize } z' = \sum_{i \in \hat{\mathcal{I}}, i' \in \mathcal{I}', l \in \mathcal{L}} \zeta_{ii'l} \tag{24}$$

4 Solution Algorithms

The nurse assignment model introduced in Section 3.1 is a multi-objective MIP. To find solutions, it is necessary to decide how each objective function component is to be treated with respect to the others. In previous work, Mobasher et al. [22] first optimized each term separately and then minimized the weighted sum of the "deviation" from the optimal value of each. The weights were derived using the analytic hierarchy process (AHP) with input obtained from interviews with nurse managers. A comparison index was then used to compare the results provided by their solution pool method (SPM) and modified goal programming method (MGPM). However, they found both methods computationally challenging as they required the solution of large-scale MIPs at intermediate steps. As an alternative, we have developed a column generation scheme (CGS) and a two-phase heuristic (SWAP), which finds good schedules relative to SPM and MGPM and strikes a balance between runtime and solution quality.

4.1 Column Generation Scheme

In this section, we describe our new approach to the NAM based on column generation. We begin by creating an equivalent formulation of the NAM from a subset of the original constraints which serves as the *master problem*. Each column in the master problem represents a feasible assignment of nurses to a particular case over the planning horizon; that is, the specification of the y_{ickh} variables for case c. We start with a small set of columns and generate more as needed. This is done by solving one *subproblem* for each surgery case at each iteration of the algorithm.

The basic steps of our column generation scheme (CGS) are as follows. The master problem is initialized with a solution to a simplified version of the NAM presented below. We then solve its LP relaxation to get the dual prices for each constraint containing the y_{ickh} variables. Once these values are available, we determine the reduced cost for each case c and solve the corresponding subproblem. If the optimal objective function value is negative, then a new column is identified and added to the master problem. By re-solving the master problem, new values for the dual variables are found and the scheme is repeated. The process continues until either no negative reduced cost is found or a stopping criterion (e.g. time limit, iterations) is reached. Nurse-related applications of column generation are addressed in [4, 20, 26].

The specifics of our approach are now presented. Let $\Omega(c)$ be the set of columns available for surgery case c and define the decision variables for the master problem as $Z_c^\rho = 1$ if column ρ is chosen for surgery case $c \in \mathcal{C}$; 0 otherwise.

The master problem is created from constraints (1), (3), (10) – (15), (18) and (19). For each case c, column $\rho \in \Omega(c)$ is constructed from the values of the y variables obtained from the solution of subproblem c, such that $y_{ickh}^\rho = 1$ if nurse i performs role k in hour h, and 0 otherwise. For presentation purposes, we use the NAM objective function given in Eq. (20). The linear master problem (LMP) is as follows.

$$\text{Minimize } \{\mathcal{DE}, \mathcal{DF}, \mathcal{DS}, \mathcal{X}, \mathcal{NCT}, \mathcal{CDT}\} \tag{25}$$

subject to

$$\sum_{c\in\mathcal{C},k\in\mathcal{K}} \sum_{\rho\in\Omega(c)} y^{\rho}_{ickh} \cdot Z^{\rho}_{c} \leq 1, \qquad \forall i \in \mathcal{I}, h \in \mathcal{H} \tag{26}$$

$$\sum_{c\in\mathcal{C},k\in\mathcal{K},h\in\mathcal{H}} \sum_{\rho\in\Omega(c)} y^{\rho}_{ickh} \cdot Z^{\rho}_{c} \leq \sum_{s\in\mathcal{S},h\in\mathcal{H}} P^{1}_{is} \cdot (P^{8}_{sh} + P^{9}_{sh}), \qquad \forall i \in \mathcal{I} \tag{27}$$

$$\sum_{i\in\mathcal{I}} \sum_{\rho\in\Omega(c)} y^{\rho}_{ickh} \cdot Z^{\rho}_{c} + de_{ckh} \geq P^{5}_{ckh} \cdot P^{6}_{ch}, \qquad \forall c \in \mathcal{C}, k \in \mathcal{K}, h \in \mathcal{H} \tag{28}$$

$$\sum_{h\in\mathcal{H}} de_{ckh} - \mathcal{DE} \leq 0, \qquad \forall c \in \mathcal{C}, k \in \mathcal{K} \tag{29}$$

$$-dev^{1}_{ih} \leq \sum_{c\in\mathcal{C},k\in\mathcal{K}} \sum_{\rho\in\Omega(c)} y^{\rho}_{ick(h+1)} \cdot Z^{\rho}_{c} - \sum_{c\in\mathcal{C},k\in\mathcal{K}} \sum_{\rho\in\Omega(c)} y^{\rho}_{ickh} \cdot Z^{\rho}_{c} \leq dev^{1}_{ih}, \qquad \forall i \in \mathcal{I}, h \in \mathcal{H} \tag{30}$$

$$\sum_{h\in\mathcal{H}} dev^{1}_{ih} - \mathcal{DS} \leq 0, \qquad \forall i \in \mathcal{I} \tag{31}$$

$$\sum_{c\in\mathcal{C},k\in\mathcal{K},h\in\mathcal{H}} \sum_{\rho\in\Omega(c)} y^{\rho}_{ickh} \cdot Z^{\rho}_{c} \cdot P^{3}_{cj} \leq \sum_{c,k,q,p,h} \left(P^{4}_{cqph} \cdot P^{2}_{ikqp} \cdot P^{3}_{cj} \cdot P^{7}_{c}\right) \cdot X_{ij}, \qquad \forall i \in \mathcal{I}, j \in \mathcal{J} \tag{32}$$

$$\sum_{j\in\mathcal{J}} X_{ij} - \mathcal{X} \leq 0, \qquad \forall i \in \mathcal{I} \tag{33}$$

$$dev^{2}_{ih} - \sum_{s\in\mathcal{S}} P^{1}_{is} \cdot P^{9}_{sh} \cdot \sum_{c\in\mathcal{C},k\in\mathcal{K}} \sum_{\rho\in\Omega(c)} y^{\rho}_{ickh} \cdot Z^{\rho}_{c} \geq 0, \qquad \forall i \in \mathcal{I}, h \in \mathcal{H} \tag{34}$$

$$\sum_{h\in\mathcal{H}} dev^{2}_{ih} - \mathcal{DF} \leq 0, \qquad \forall i \in \mathcal{I} \tag{35}$$

$$\sum_{c\in\mathcal{C}} \sum_{\rho\in\Omega(c)} nc^{\rho}_{ic} - \mathcal{NCT} \leq 0, \qquad \forall i \in \mathcal{I}, c \in \mathcal{C} \tag{36}$$

$$\sum_{c\in\mathcal{C}} \sum_{\rho\in\Omega(c)} cd^{\rho}_{ic} - \mathcal{CDT} \leq 0, \qquad \forall i \in \mathcal{I}, c \in \mathcal{C} \tag{37}$$

$$\sum_{\rho\in\Omega(c)} Z^{\rho}_{c} \leq 1, \qquad \forall c \in \mathcal{C} \tag{38}$$

$$\mathcal{DE}, \mathcal{DF}, \mathcal{DS}, \mathcal{X}, \mathcal{NCT}, \mathcal{CDT}, de_{ckh}, dev^{1}_{ih}, dev^{2}_{ih}, nc^{\rho}_{ic}, cd^{\rho}_{ic} \geq 0$$

$$X_{ij} \in [0,1],\ Z^{\rho}_{c} \in [0,1], \qquad \forall i \in \mathcal{I}, j \in \mathcal{J}, c \in \mathcal{C}, k \in \mathcal{K}, h \in \mathcal{H}, \rho \in \Omega(c) \tag{39}$$

where all the values of $y^{\rho}_{ickh}, nc^{\rho}_{ic}$ and cd^{ρ}_{ic} are known.

In order to solve the LMP, the objective function in Eq. (25) must be linearized. Following the approach in [22], we form the cumulative weighted index (CWI) by taking the weighted sum of the deviation variables; that is,

$$CWI = w^{1}\mathcal{DE} + w^{2}\mathcal{DF} + w^{3}\mathcal{DS} + w^{4}\mathcal{X} + w^{5}\mathcal{NCT} + w^{6}\mathcal{CDT} \tag{40}$$

The weights included in Equation (40) are calculated using the AHP. Table 2 in Section 5.2 shows the pairwise comparisons of the six objectives and their relative weights $w^{j}, j = 1, ..., 6$.

To generate new columns, we first need to determine the reduced cost of variable Z_c^ρ; call it $\overline{RC}_c^\rho$. This term serves as the objective function for subproblem $c \in \mathcal{C}$. Also, The feasible region of each subproblem consists of the remaining hard and soft constraints from the NAM that were not included in the LMP. The procedure of calculating the reduced cost and the mathematical formulation of the subproblem for case c are explained in Appendix A.

4.1.1 Initial solutions

Before we can begin solving the LMP, we need to provide an initial feasible solution. Experience has shown that the quality of this solution can have a dramatic effect on the convergence speed of heuristics [16, 24]. To derive good feasible solutions, we propose solving a simplified version of the NAM.

Accordingly, we temporarily assume that the demand for nurses to perform role $k \in \mathcal{K}$ on case $c \in \mathcal{C}$ is one. We also assume that the deviations mentioned in Section 3.1.2 are no longer permitted so all soft constraints must be satisfied. As an example, the assignment of a nurse to a case can no longer be broken while the surgery is in progress even if overtime is required.

The *notation and parameters* used in the formulation of the initial solution model (ISM) are based on the NAM, and are as follows:

P^{10}_{ick} — 1 if nurse $i \in \mathcal{I}$ can do role $k \in \mathcal{K}$ for case $c \in \mathcal{C}$, 0 otherwise

ST_c — starting time of case $c \in \mathcal{C}$

ET_c — ending time of case $c \in \mathcal{C}$

$P^{11}_{cc\prime}$ — 1 if case $c \in \mathcal{C}$ and case $c\prime \in \mathcal{C}$ overlap, 0 otherwise

The *decision variables* , x_{ick}, determine the nurse-to-case assignments as in the NAM.

Now, considering the above assumptions, we can formulate the ISM as follows.

$$\text{Maximize} \quad \sum_{i\in\mathcal{I},c\in\mathcal{C},k\in\mathcal{K}} x_{ick}, \tag{41}$$

$$\text{subject to} \quad x_{ick} \le P^{10}_{ick}, \qquad \forall i \in \mathcal{I}, c \in \mathcal{C}, k \in \mathcal{K} \tag{42}$$

$$\sum_{i\in\mathcal{I}} x_{ick} = 1, \qquad \forall c \in \mathcal{C}, k \in \mathcal{K} \tag{43}$$

$$\sum_{k\in\mathcal{K}} x_{ick} \le 1, \qquad \forall i \in \mathcal{I}, c \in \mathcal{C} \tag{44}$$

$$\sum_{c\in\mathcal{C},k\in\mathcal{K}} x_{ick} \le \eta, \qquad \forall i \in \mathcal{I} \tag{45}$$

$$P^{11}_{cc\prime} \cdot \left(\sum_{k\in\mathcal{K}} x_{ick} + \sum_{k\in\mathcal{K}} x_{ic\prime k} \right) \le 1, \qquad \forall i \in \mathcal{I}, c \ne c' \in \mathcal{C} \tag{46}$$

$$x_{ick} \in \{0,1\}, \qquad \forall i \in \mathcal{I}, c \in \mathcal{C}, k \in \mathcal{K} \tag{47}$$

The objective function (41) is aimed at maximizing the total number of nurses assigned to surgery cases. Constraints (42) guarantee that a nurse will only be assigned to a case if the qualifications are met to work on it (i.e., $P^{10}_{ick} = 1$). Constraints (43) and (44) ensure that exactly one nurse is assigned to each case c to perform role k, and that no nurse can have more than one role on a case, respectively. Constraints (45) state that nurse i can be assigned to at most η cases (i.e., a pre-specified number) for the day. The final restriction (46) prevents a nurse from being assigned to more than one case at a time, and breaking an assignment in the middle of a surgery has been ruled out by assumption.

Proposition 1. *A feasible solution of the NAM can be constructed from an optimal solution of the ISM.*

Proof. See Appendix B. □

Corollary 2. *The optimal solution of the ISM provides an upper bound for the NAM.*

Proof. According to Proposition 1, an optimal solution of the ISM is also a feasible solution to the NAM. Because any feasible solution provides a primal bound on the objective function value, the optimal solution of the ISM provides an upper bound for the NAM. □

Corollary 3. *The optimal solution of the ISM provides a feasible solution for the LMP.*

Proof. After removing some of the hard constraints, the LMP becomes a relaxation of the NAM . Therefore, any feasible solution to the NAM is also feasible to the LMP. From Proposition 1, we conclude that the optimal solution of the ISM provides a feasible solution to the LMP. □

The results obtained from solving the ISM can also be used to improve the quality of the columns generated by the subproblems. Since the optimal solution of the ISM is an upper bound of the NAM, we can use the staff shortages associated with the initial solution as an upper bound for the staff shortages introduced in the LMP in Eqs. (28) and (29). Therefore, if we assume that DE_c^{IS} is the amount of staff shortage for surgery case $c \in \mathcal{C}$ based on the ISM optimal solution, then $\mathcal{DE} \leq DE_c^{ISM}$ in the subproblem associated with surgery case $c \in \mathcal{C}$. A pseudocode of the column generation scheme is presented in *Algorithm 1.*

4.2 Swapping Heuristic (SWAP)

In this section, we present a swapping heuristic (SWAP) that is designed to provide good feasible schedules in a few seconds. Swapping techniques are the mainstay of heuristics (e.g., tabu search, bee colony algorithm, genetic algorithm) designed to solve scheduling problems [19, 28]. SWAP is a two-phase procedure. In the first phase (Construction), the ISM in Section 4.1 is solved to obtain an initial solution; in the second phase (Improvement), three exchange procedures are called to try to improve the schedule by reducing the violation of soft constraints. They are executed sequentially to reduce staff shortage, overtime, and idle time, respectively. Using construction and improvement heuristics is a well documented approach for solving tightly constrained integer programing problems [12, 33].

The improvement phase consists of three heuristics designed to reduce violations of soft constraints. However, these constraints are not of equal importance or priority from the point of view of those responsible for staffing the ORs. Based on discussions with the nurse managers at MD Anderson Cancer Center provided in Table 2, we determined that demand satisfaction, overtime and idle time are the three most critical soft constraints, in that order. Accordingly, the improvement phase consists of three successive groups of heuristics to reduce staff shortage, overtime, and idle time.

Algorithm 1 Column generation scheme for NAM

Step 1: Initialization
• Solve the ISM to obtain an initial solution to the NAM, and initialize the master problem.
• Add constraint $\mathcal{DE} \leq DE_{ck}^{ISM}$ to the subproblems.
• Let n = number of solutions obtained from each subproblem using the solution pool feature; set $t = 0$.
while $t \leq K$ **do**
  Step 2: Dual variable generation
  • Solve the LP-relaxation of the master problem optimally, and obtain dual variables associated with the master problem constraints.
  Step 3: New columns generation
  for each surgery case $c \in \mathcal{C}$ **do**
    • Obtain the reduced cost using Eq. (48)
    if reduced cost < 0 **then**
      • Obtain n alternate solutions from the subproblem to serve as new columns for case c. (Here, we try to generate m feasible solutions such that $m \geq n$, and then choose the n among them with the lowest reduced costs.)
      • If the column is not redundant, add it to the master problem.
    end if
  end for
  • Put $t \leftarrow t + 1$;
  • Go to Step 2.
end while
• Output CWI^*, $\mathcal{DE}^*$, $\mathcal{DF}^*$, $\mathcal{DS}^*$, $\mathcal{X}^*$, $\mathcal{NCT}^*$,$\mathcal{CDT}^*$, y^*_{ickh}, x^*_{ick}, de^*_{ckh}, dev^{1*}_{ih}, dev^{2*}_{ih}, X^*_{ij}, nc^*_{ic}, cd^*_{ic}.

Reducing staff shortages The first stage in the improvement phase is minimizing staff shortages. The first step is to identify surgery cases with demand shortage (set SD) based on the initial solution obtained from the ISM. The cases are then sorted in descending order of staff shortage. The procedure for assigning idle nurses to cases in SD is based on the availability of nurses who can perform the required role on the case (See the flowchart in Appendix D). Algorithm 2 and Figure 1 indicate how to find the best nurse to assign to a case with a shortage. In the worst case, the "for" loop in Algorithm 2 has to be executed for all cases and nurses, so its complexity is $O(mnlog(n))$, with m being the number of cases, and n being the number of nurses.

If there are no idle nurses who can work on the case under consideration for at least a portion of its duration, two procedures are called. In each, an idle nurse is swapped with a nurse who can work on the case but who is already assigned to another case. Figure 2 illustrates the logic. The procedure also runs in $O(mnlog(n))$ time.

Reducing Overtime. Starting from the updated solution from the staff shortage reduction stage, we now try to reduce the overtime assignments. The process is illustrated in the flowchart in Appendix D. The first step is to find the nurses who work overtime (set OD) and sort them in descending order based on their total overtime. For each nurse i in OD, we search for another nurse i' who is idle during his/her regular working hours, and the corresponding overtime hours of i. If such a nurse i' is found and can work on the case, we relieve the overtime nurse from his assignment and assign the available nurse to that case. This exchange procedure has $O(n^2log(n))$ complexity in the worst case, where n is the number of nurses.

Algorithm 2 Reducing staff shortages

Objective: Assign available nurses to case $c \in SD$ for all or part of its remaining duration
for each surgery case $c \in SD$ **do**
- Define set ID as the set of nurses who can work on case c from time ST_{ic} to time ET_{ic}.

for each nurse $i \in \mathcal{I}$ **do**
- Check if nurse i has not been assigned to any cases throughout the day.
- If nurse i can do role k on case c for duration $DB = ET_{ic} - ST_{ic} + 1$, add nurse i to set ID.

end for
- Sort ID based on DB in descending order.

if set ID is empty **then**
- Break.

else
- Assign the first nurse $i \in ID$ to case c from time ST_{ic} to time ET_{ic}.

end if
- Update set ID.
- Update staff shortage for case $c \in SD$.

end for

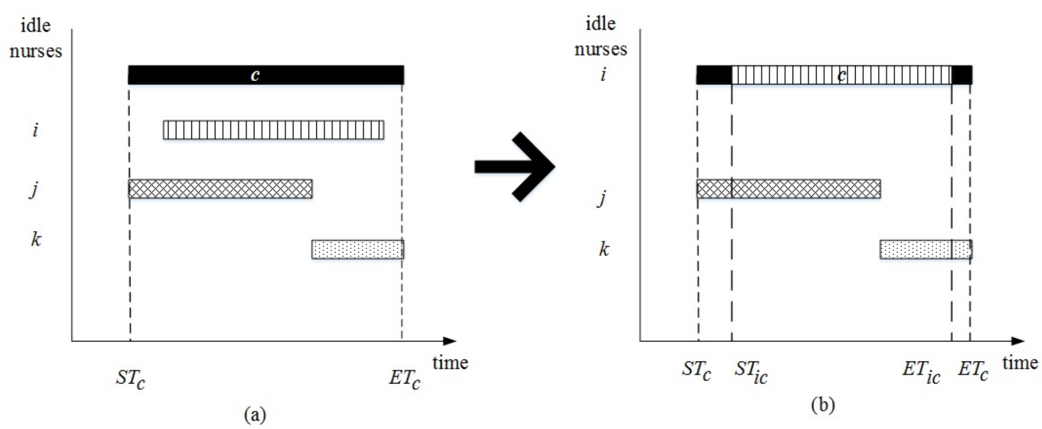


Figure 1: The procedure for assigning idle nurses to cases in SD is based on the availability of nurses who can perform the required role on the case. The horizontal bar next to each nurse depict the amount of time (s)he can work on case $c \in \mathcal{C}$. (a) Nurses i, j, $k \in \mathcal{I}$ are idle nurses who can work on case $c \in \mathcal{C}$, (b) The nurse with the greatest available time to work on the case is selected.

Reducing Idle Time To reduce idle time as well as to improve the efficiency and experience of the nursing staff, it is desirable to assign nurses who are idle to surgery cases with the highest complexity levels that match their competency level. In some situations, it is also desirable to assign nurses to cases outside of, but related to, their specialty in order for them to gain experience in other specialties. Nurses in this category are called learning fellows. Cases with demand shortages are given a higher priority in this process. The process is illustrated in the flowchart in Appendix D. The procedure has $O(mnlog(n))$ complexity in the worst case, with m being the number of cases and n the number of nurses.

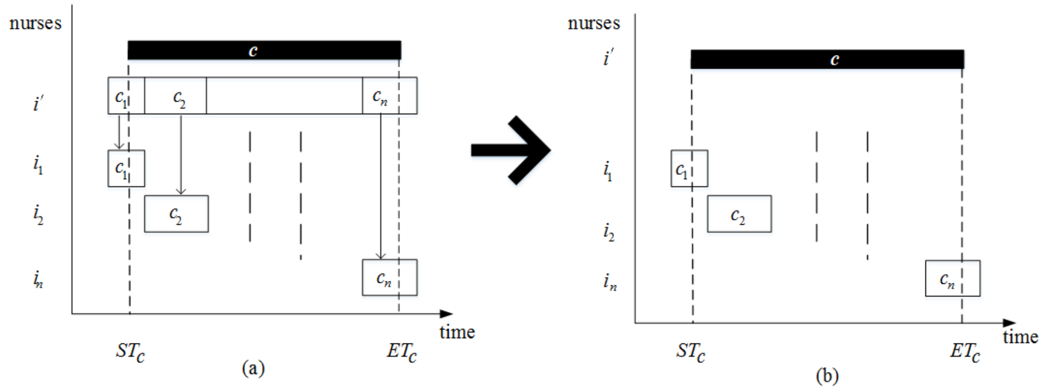


Figure 2: (a) The indices i_1, i_2,...,$i_n \in \mathcal{I}$ present idle nurses who cannot work on case c, while i' can work on case c but is already assigned to other cases; (b) i' is relieved from his assignments by i_1, i_2, ..., i_n and assigned to work on case c. Nurses do not need to be available for the entire duration of the case in order to be swapped to the case, but preference is given to nurses who are available for the entire duration.

5 Computational Experience

The case study presented in this section is based on data gathered from the University of Texas MD Anderson Cancer Center, Houston, TX. The preoperative enterprise at MD Anderson handles an estimated 1500 surgeries per month. There are 148 nurses and 140 surgeons on the staff who perform these surgeries. Scheduling nurses based on their abilities and shift preferences is a complex task that has always been done manually in the main operating suite.

5.1 Test Problems Setup

Data were collected on nurse attributes and daily surgery schedules in the main operating suite, which comprises 33 operating rooms. Most of the ORs are multi-functional and run five days a week with each day being scheduled separately. We assume that surgery durations are deterministic and known based on the surgeons' estimates. On average, 100 nurses (RNs and scrub techs) are available for the different shifts every day. Shifts are 8, 10 and 12 hours in length. The combination of regular shift hours and authorized overtime hours cannot exceed 12 hours for each nurse. Nurses are categorized in 11 different specialties, (e.g., head and neck, plastic, oncology) and three competency levels (simple, moderate, and complex) based on their experience and certifications. Nurses are assigned to work on different cases based on the surgery sheet provided by the scheduling department. These sheets contain information on the surgeries scheduled for that day, the surgeon assigned to each case, his or her instrument preferences, the operating room, the estimated duration and procedural complexity, as well as the surgery demand. A 30-minute time interval ($h = 30$ min) is used for developing a schedule.

We generated six data sets based on actual data from MD Anderson. Table 1 shows the characteristics of each set. The optimization models were implemented in a C++ environment and solved using CPLEX 12.2 on a 3GHz workstation with 16GB memory running RedHat Server 2008.

Table 1: Data sets for different operating suites

Data set no.	No. of RNs	No. of scrub techs	No. of surgery cases ($\|\mathcal{C}\|$)	No. of shifts ($\|\mathcal{S}\|$)	No. of ORs ($\|\mathcal{J}\|$)
1	4	2	3	1	2
2	19	6	14	2	10
3	28	9	28	5	17
4	39	11	28	5	17
5	44	38	40	6	26
6	56	56	53	6	33

5.2 Results for the Nurse Assignment Model

For the sake of brevity, we only report the results for data sets 1, 2 and 6 as shown in Tables 3, 4 and 5, respectively. The results for data sets 3, 4 and 5 are reported in Appendix C. In these tables, performance of the Nurse Assignment Model (NAM), Solution Pool Method (SPM), Modified Goal Programming Method (MGPM), Column Generation Scheme (CGS), Initial Solution Model (ISM) and the swapping heuristic (SWAP) in generating nurse schedules are evaluated and compared.

For the NAM, we used objective function $z = \mathcal{DE} + \mathcal{DF} + \mathcal{DS} + \mathcal{X} + \mathcal{NCT} + \mathcal{CDT}$; for the remaining models, we determined the weights for CWI in Eq. (40) with the AHP based on the actual opinions of nurse managers at MD Anderson. Table 2 presents the pairwise comparisons used to compute the six weights, $w^j, j = 1, ..., 6$.

Table 2: Pairwise comparisons of objectives and their relative importance weights for NAM

	j	(1)	(2)	(3)	(4)	(5)	(6)	w^j
(1)	$\mathcal{DE}$	1	3	7	5	9	9	0.53
(2)	$\mathcal{DF}$	1/3	1	7/3	5/3	3	3	0.18
(3)	$\mathcal{DS}$	1/7	3/7	1	5/7	9/7	9/7	0.08
(4)	$\mathcal{X}$	1/5	3/5	7/5	1	9/5	9/5	0.11
(5)	$\mathcal{NCT}$	1/9	1/3	7/9	5/9	1	1	0.06
(6)	$\mathcal{CDT}$	1/9	1/3	7/9	5/9	1	1	0.06

Table 3: Numerical results for data set 1

Solution method	Staff shortage ($\mathcal{DE}$)	Overtime assignments ($\mathcal{DF}$)	Idle times ($\mathcal{DS}$)	Time (sec)	CWI
NAM	0	1	8	14.21	1.22
SPM	0	1	10	4	1.21
MGPM	0	1	8	1	1.05
CGS	0	1	8	0.54	1.05
ISM	0	1	10	0.03	1.21
SWAP	0	1	7	0.03	0.97

From Table 3 we see that all algorithms provide solutions that satisfy the surgery demand (i.e., $\mathcal{DE} = 0$) and perform almost equally well in minimizing all deviations. However, NAM, SPM and MGPM require more computation time compared to CGS and SWAP, which converge in less than a second. Looking deeper into the results, SWAP provides the least number of idle intervals ($\mathcal{DS}$), and the smallest CWI value, 0.97. On balance, SWAP can be considered as the best solution

Table 4: Numerical results for data set 2

Solution method	Staff shortage ($\mathcal{DE}$)	Overtime assignments ($\mathcal{DF}$)	Idle times ($\mathcal{DS}$)	Time (sec)	CWI
NAM	14	5	23	18000	10.85
SPM	22	5	15	18364	14.22
MGPM	22	5	16	1674	14.07
CGS	22	7	18	4.52	14.7
ISM	22	7	19	0.1	14.78
SWAP	22	1	18	0.29	13.85

Table 5: Numerical results for data set 6

Solution method	Staff shortage ($\mathcal{DE}$)	Overtime assignments ($\mathcal{DF}$)	Idle times ($\mathcal{DS}$)	Time (sec)	CWI
Actual	22	7	16	N/A	14.6
NAM	22	6	19	54080	17.26
SPM	2	5	17	54076	6.2
MGPM	9	1	19	39800	7.51
CGS	22	7	22	65.84	15.19
ISM	22	7	22	1.24	15.19
SWAP	14	4	13	3.6	9.75

method for data set 1.

For data set 2, the NAM provides less staff shortage in comparison with the other methods but required five hours of runtime. SPM also required excess runtime while yielding an inferior solution compared to the NAM and so can be ruled out from further consideration. MGPM and CGS perform equally but CGS is dominant with respect to runtime by a factor of 40. SWAP provides the least overtime ($\mathcal{DF}$) among all methods by a factor between 5 and 7, and less idle time in comparison with NAM that has the smallest CWI value. The main advantage of SWAP is its computational efficiency. In comparison to CGS that found its best solution in 5 sec, SWAP needed less than 10% of that time to obtain an even better solution.

Data set 6 represents an actual nurse scheduling problem at MD Anderson. The second row in Table 5 reports the corresponding results. From the table we see that the NAM is dominated in every measure, with respect to the actual results, and the results provided by all its competitors. SPM does the best in meeting demand and MGPM yields the least amount of overtime. However, extremely large solution times make these methods impractical. Solving the problem using SPM and MGPM takes 15 and 11 hours, respectively. CGS can solve the problem in one minute which is reasonable for practical instances.

SWAP overcomes the computational disadvantages of SPM and MGPM, while providing better solutions than CGS. It reduces the demand shortage in the actual schedule by 36% and obtains a solution with maximum staff utilization (i.e., minimum idle times). It also solves the problem in only 3.6 seconds which is less than 0.01% of the runtimes for SPM and MGPM, and 6% of the required time for CGS.

Table 6 highlights performance of the column generation algorithm, which was seen to be computationally efficient and able to provide good feasible solutions. The third column indicates that very few iterations were needed to converge at the root node of the implied search tree. At that point, the LMP was solved as an IP with $Z_c^{\rho} \in \{0, 1\}$ to arrive at a feasible solution. An advantage of the column generation algorithm as compared to the SWAP is that the quality of the obtained results by this algorithm can be measured by checking the optimality gap between the LMP solution and the integer solution.

Table 6: Column generation results

Data set	Gap*(%)	CG Iterations	CG time (sec)	IP time (sec)
1	14.00	3	0.53	0.01
2	11.86	2	4.48	0.04
3	15.27	2	11.28	0.03
4	15.27	2	28.10	0.04
5	16.52	2	35.53	0.12
6	14.63	2	65.74	0.10

* $100 \times (obj_{(IP)} - obj_{(LMP)})/obj_{(IP)}$

For large data sets, we found that the NAM was not able to find a feasible solution, and SPM was the worst approach with respect to computational efficiency. MGPM ran more quickly and produced a smaller CWI than did NAM and SPM for all data sets. However, none of these methods were seen to be computationally efficient, especially for large instances. When it is desirable or necessary to explore different scenarios or deal with nurse shortages, one of the faster methods would be preferred. Operationally speaking, one must be able to obtain a good feasible solution in a matter of minutes.

The column generation approach was developed to accommodate this situation. It solved quickly for all data sets. Embedding it in a branch-and-price scheme, or at least allowing for some branching might notably improve its results and reduce the optimality gap. For large data sets, if we need to re-solve the scheduling problem to reflect any last minute cancellations or call-ins, we will need a faster approach. As the results show, SWAP can improve the initial solutions in seconds and provides good solutions at convergence across all criteria.

5.3 Results for the Nurse Lunch Model

For the early shift, three 1-hour lunch break intervals are available starting from 10:30 AM and going to 1:30 PM. Nurses whose shift starts at 10:30 AM typically begin by relieving nurses who started their shift at 6:30 AM, work in the relief role until 1:30 PM, and then spend the remainder of the day on other cases. We used the same data sets introduced in Section 5.2 for evaluating our lunch model. Given the nurse assignments, we can easily solve the NLM in seconds for each data set. For illustrative purposes, only the results for data set 6 (i.e., the actual scheduling problem) are presented in Table 7. This was the most computationally challenging instance. Recall that our objective in making the lunch break assignments is to maximize the total number of nurses who can be relieved during the lunch break periods between 10:30 AM and 1:30 PM. In all, there are 70 nurses who start at 6:30 AM and 42 relief nurses who start at 10:30 AM.

Table 7 shows that the nurse lunch model in conjunction with one of the solution algorithms can provide a complete set of schedules that efficiently meets case demand and lunch break requirements. For those nurses working the second shift and who need to be relieved during their lunch break interval, the NLM can be run again but with the nurses working the third shift filling in for those on the second shift.

6 Summary and Conclusions

Knowing that nurse availability can change in real time, we have developed a fast solution framework for the daily nurse scheduling problem in operating suites. A unique feature of our methodology is that it provides both a surgery and lunch break schedule for the nursing staff. The daily assignments are designed to satisfy a range of requirements related to case specialty, procedure complexities, and skill levels. The problem is formulated as a mixed integer program with the objective of minimizing demand shortage, overtime and idle time. Two new heuristics were developed to solve the problem: a column generation scheme and a two-phase heuristic. Using six instances derived from data provided by the University of Texas MD Anderson Cancer Center in Houston, we observed that both CGS and SWAP can solve the NAM and NLM in a couple of minutes, which is the main contribution of this paper. Compared to our previous heuristics including the solution pool approach and a goal programming model, the results showed reductions in solution times of up to 99%, especially, for the large instances.

SWAP seems to be the best alternative if one wishes to have a higher quality of the schedule, and needs a scheduling tool to make frequent changes to the nurse assignments. Considering stochastic durations for surgical procedures and studying their effect on nurse assignments would be an interesting topic for future research.

Table 7: Daily lunch assignments provided by the NLM based on data set 6 using SWAP

Working nurse #	Relieving nurse #			Working nurse #	Relieving nurse #		
	10:30-11:30	11:30-12:30	12:30-1:30		10:30-11:30	11:30-12:30	12:30-1:30
1		29		57			28
2		28		58		26	
3		27		59	76		
9	112			60			75
11		112		61		73	
37	78			62		74	
38		24		67	29		
41			78	84		75	
42	73			85		78	
43			74	86			29
44			112	88			27
45	26			90	74		
46		76		92			76
47	1			93	75		
48		10		97		25	
49			25	98	25		
50			24	104			26
51			73	108		77	
53	24			109	28		
55			10	110	27		
56			77	111	10		

References

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

[2] J. Bard and H. Purnomo. Hospital-wide reactive scheduling of nurses with preference considerations. *IIE Transactions*, 37(7):589–608, 2005.

[3] J. Bard and H. Purnomo. Cyclic preference scheduling of nurses using a Lagrangian-based heuristic. *Journal of Scheduling*, 10(1):5–23, 2007.

[4] J. Beliën and E. Demeulemeester. A branch-and-price approach for integrating nurse and surgery scheduling. *European Journal of Operational Research*, 189(3):652–668, 2008.

[5] M. Bester, I. Nieuwoudt, and J. Van Vuuren. Finding good nurse duty schedules: a case study. *Journal of Scheduling*, 10(6):387–405, 2007.

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

[7] E. Burke, P. De Causmaecker, and H. Van Landeghem. The state of the art of nurse rostering. *Journal of Scheduling*, 7(6):441–499, 2004.

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

[9] B. Cardoen, E. Demeulemeester, and J. Beliën. Sequencing surgical cases in a day-care environment: an exact branch-and-price approach. *Computers & Operations Research*, 36(9): 2660–2669, 2009.

[10] B. Cheang, H. Li, A. Lim, and B. Rodrigues. Nurse rostering problems, a bibliographic survey. *European Journal of Operational Research*, 151(3):447–460, 2003.

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

[12] A. Constantino, D. Landa-Silva, E. de Melo, C. de Mendonça, D. Rizzato, and W. Romão. A heuristic algorithm based on multi-assignment procedures for nurse scheduling. *Annals of Operations Research*, 218(1):165–183, 2014.

[13] A. Corominas, A. Lusa, and J. Olivella. A detailed workforce planning model including nonlinear dependence of capacity on the size of the staff and cash management. *European Journal of Operational Research*, 216(2):445–458, 2012.

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

[15] A. Ernst, H. Jiang, M. Krishnamoorthy, B. Owens, and D. Sier. An annotated bibliography of personnel scheduling and rostering. *Annals of Operations Research*, 127(1):21–144, 2004.

[16] J. W. Escobar, R. Linfati, M. G. Baldoquin, and P. Toth. A Granular Variable Tabu Neighborhood Search for the capacitated location-routing problem. *Transportation Research Part B: Methodological*, 67.

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

[18] K. Kim and S. Mehrotra. A two-stage stochastic integer programming approach to integrated staffing and scheduling with application to nurse management. *Operations Research*, 63(6): 1431–1451, 2015.

[19] Y. F. Liu and S. Y. Liu. A hybrid discrete artificial bee colony algorithm for permutation flowshop scheduling problem. *Applied Soft Computing*, 13(3):1459–1463, 2013.

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

[21] B. Maenhout and M. Vanhoucke. An integrated nurse staffing and scheduling analysis for longer-term nursing staff allocation problems. *Omega*, 41(2):485–499, 2013.

[22] A. Mobasher, G. Lim, J. Bard, and V. Jordan. Daily scheduling of nurses in operating suites. *IIE Transactions on Healthcare Systems Engineering*, 1(4):232–246, 2011.

[23] J. Oddoye, M. Yaghoobi, M. Tamiz, D. Jones, and P. Schmidt. A multi-objective model to determine efficient resource levels in a medical assessment unit. *Journal of the Operational Research Society*, 58(12):1563–1573, 2006.

[24] P. H. V. Penna, A. Subramanian, and L. S. Ochi. An iterated local search heuristic for the heterogeneous fleet vehicle routing problem. *Journal of Heuristics*, 19(2):201–232, 2013.

[25] P. Punnakitikashem, J. M. Rosenberber, and D. F. Buckley-Behan. A stochastic programming approach for integrated nurse staffing and assignment. *IIE Transactions*, 45(10):1059–1076, 2013.

[26] H. Purnomo and J. Bard. Cyclic preference scheduling for nurses using branch and price. *Naval Research Logistics*, 54(2):200–220, 2006.

[27] S. J. Sadjadi, R. Soltani, M. Izadkhah, F. Saberian, and M. Darayi. A new nonlinear stochastic staff scheduling model. *Scientia Iranica*, 18(3):699–710, 2011.

[28] A. Subramanian, M. Battarra, and C. N. Potts. An Iterated Local Search heuristic for the single machine total weighted tardiness scheduling problem with sequence-dependent setup times. *International Journal of Production Research*, 52(9):2729–2742, 2014.

[29] C. Ulrich, G. Wallen, C. Grady, M. Foley, A. Rosenstein, C. Rabetoy, and B. Miller. The nursing shortage and the quality of care. *The New England Journal of Medicine*, 347(14): 1118–1119, 2002.

[30] J. Van den Bergh, J. Beliën, P. De Bruecker, E. Demeulemeester, and L. De Boeck. Personnel scheduling: A literature review. *European Journal of Operational Research*, 226(3):367–385, 2013.

[31] C. Van Huele and M. Vanhoucke. Decomposition-based heuristics for the integrated physician rostering and surgery scheduling problem. *Health Systems*, 4(3), 2014.

[32] M. Widl and N. Musliu. The break scheduling problem: Complexity results and practical algorithms. *Memetic Computing*, 6(2):97–112, 2014.

[33] T. Wong, M. Xu, and K. Chin. A two-stage heuristic approach for nurse scheduling problem: A case study in an emergency department. *Computers & Operations Research*, 51:99–110, 2014.

[34] I. H. Wright, C. Kooperberg, B. A. Bonar, and G. Bashein. Statistical Modeling to Predict Elective Surgery Time: Comparison with a Computer Scheduling System and Surgeon-provided Estimates. *The Journal of the American Society of Anesthesiologists*, 85(6):1235–1245, 1996.

[35] W. Xiang, J. Yin, and G. Lim. A short-term operating room surgery scheduling problem integratingmultiple nurses roster constraints. *Artificial Intelligence in Medicine*, 63(2):91–106, 2015.

Appendix A. Calculating the Reduced Cost $\overline{RC}_c$

In general, the reduced cost of Z_c^ρ can be written as $c_c^\rho - \pi y_c^\rho$, where c_c^ρ is the corresponding objective function coefficient, π is a vector of dual variables and y_c^ρ is the column vector whose components are y_{ickh}^ρ for all i, k and h. The relevant values of π are given in Table 8. The reduced cost of Z_c^ρ is given in Equation (48).

Table 8: Master problem dual variables

Constraint	Dual variable	Index range
(26)	$\pi_{ih}^1 \geq 0$	$\forall i \in \mathcal{I}, h \in \mathcal{H}$
(27)	$\pi_i^2 \geq 0$	$\forall i \in \mathcal{I}$
(28)	$\pi_{ckh}^3 \geq 0$	$\forall c \in \mathcal{C}, k \in \mathcal{K}, h \in \mathcal{H}$
(30)	π_{ih}^4 and $\pi_{ih}^5 \geq 0$	$\forall i \in \mathcal{I}, h \in \mathcal{H}$
(32)	$\pi_{ij}^6 \geq 0$	$\forall i \in \mathcal{I}, j \in \mathcal{J}$
(34)	$\pi_{ih}^7 \geq 0$	$\forall i \in \mathcal{I}, h \in \mathcal{H}$
(36)	$\pi_{ic}^8 \geq 0$	$\forall i \in \mathcal{I}, c \in \mathcal{C}$
(37)	$\pi_{ic}^9 \geq 0$	$\forall i \in \mathcal{I}, c \in \mathcal{C}$
(38)	$\pi_c^{10} \geq 0$	$\forall c \in \mathcal{C}$

$$\begin{aligned}\overline{RC}_c^\rho =& c_c^\rho - \pi y_c^\rho = 0 - \left[\sum_{i,h} \pi_{ih}^1 \cdot \sum_k y_{ickh}^\rho\right] - \left[\sum_i \pi_i^2 \cdot \sum_{k,h} y_{ickh}^\rho\right] - \left[\sum_h \pi_{ckh}^3 \cdot \sum_{i,k} y_{ickh}^\rho\right] \\ &- \left[\sum_{i,h} \pi_{ih}^4 \cdot \sum_k (y_{ick(h+1)}^\rho - y_{ickh}^\rho)\right] + \left[\sum_{i,h} \pi_{ih}^5 \cdot \sum_k (y_{ickh}^\rho - y_{ick(h+1)}^\rho)\right] - \left[\sum_{i,j \in \mathcal{J}} \pi_{ij}^6 \cdot P_{cj}^3 \cdot \sum_{k,h} y_{ickh}^\rho\right] \\ &+ \left[\sum_{i,h} \pi_{ih}^7 \cdot \sum_s P_{is}^1 \cdot P_{sh}^9 \cdot \sum_{c \in \mathcal{C},k} y_{ickh}^\rho\right] - \left[\sum_i \pi_{ic}^8 \cdot nc_{ic}^\rho\right] - \left[\sum_i \pi_{ic}^9 \cdot cd_{ic}^\rho\right] - \pi_c^{10}\end{aligned} \tag{48}$$

The feasible region of each subproblem consists of the remaining hard and soft constraints from the NAM that were not included in the LMP. So, The subproblem for case c is

$$\text{Minimize} \quad \overline{RC}_c \tag{49}$$

$$\text{subject to} \quad y_{ickh} \leq \sum_{s \in S} P_{is}^1 \cdot (P_{sh}^8 + P_{sh}^9), \qquad \forall i \in \mathcal{I}, k \in \mathcal{K}, h \in \mathcal{H} \tag{50}$$

$$y_{ickh} \leq P_{ch}^6 \cdot \left(\sum_{q \in \mathcal{Q}, p \in \mathcal{P}} P_{ikqp}^2 \cdot P_{cqph}^4\right), \qquad \forall i \in \mathcal{I}, k \in \mathcal{K}, h \in \mathcal{H} \tag{51}$$

$$\sum_{i \in \mathcal{I}} y_{ickh} \geq P_{ch}^6, \qquad \forall k \in \mathcal{K}, h \in \mathcal{H} \tag{52}$$

$$\sum_{h \in \mathcal{H}} y_{ickh} \leq P_c^7 \cdot \sum_{q \in \mathcal{Q}, p \in \mathcal{P}, h \in \mathcal{H}} \left(P_{cqph}^4 \cdot P_{ikqp}^2\right) \cdot x_{ick}, \qquad \forall i \in \mathcal{I}, k \in \mathcal{K} \tag{53}$$

$$\sum_{k\in\mathcal{K}} x_{ick} \leq 1, \qquad \forall i \in \mathcal{I} \tag{54}$$

$$\sum_{k\in\mathcal{K},h\in\mathcal{H}} y_{ickh} + P_c^7 \cdot (cd_{ic} + (1 - nc_{ic})) \geq P_c^7, \quad \forall i \in \mathcal{I} \tag{55}$$

$$\sum_{k\in\mathcal{K},h\in\mathcal{H}} y_{ickh} \leq P_c^7 \cdot \sum_{k\in\mathcal{K},q\in\mathcal{Q},p\in\mathcal{P},h\in\mathcal{H}} \left(P_{cqph}^4 \cdot P_{ikqp}^2\right) \cdot nc_{ic}, \qquad \forall i \in \mathcal{I} \tag{56}$$

$$x_{ick}, y_{ickh}, nc_{ic} \in \{0,1\}, \qquad \forall i \in \mathcal{I}, c \in \mathcal{C}, k \in \mathcal{K}, h \in \mathcal{H} \tag{57}$$

Appendix B. Proof of Proposition 1

A feasible solution of the NAM can be constructed from an optimal solution of the ISM.

Proof. We take two steps to prove this claim. The first step is to show that any feasible solution of the ISM satisfies the hard constraints in the NAM. We then show that $(\bar{x}, \bar{y})$ can be constructed based on $\hat{x}$.

Let us examine the constraints of both models. In terms of the feasibility condition, constraint (46) is equivalent to constraint (1), constraint (42) is equivalent to constraints (2) and (4), constraint (45) is equivalent to constraints (3) (both constraints limit the workload of each nurse during the day), constraint (43) is equivalent to constraint (5), and constraint (44) is equivalent to constraints (6) and (7). Thus, we can see that constraints of the ISM are equivalent to a special case of the NAM in which each scheduled case must have one nurse per role and deviations are not allowed.

Suppose that the feasible region of the ISM is not empty. Let $\hat{x} = \{\hat{x}_{ick}\}_{\forall i\in\mathcal{I},c\in\mathcal{C},k\in\mathcal{K}}$ be a feasible solution of the ISM and $(\bar{x}, \bar{y})$ be a feasible solution of the NAM, where $\bar{y} = \{\bar{y}_{ickh}\}_{\forall i\in\mathcal{I},c\in\mathcal{C},k\in\mathcal{K},h\in\mathcal{H}}$. In particular, it is easy to see that $\bar{x} = \hat{x}$ according to variable definitions. Because all surgical cases have already been scheduled prior to the nursing staff assignment, $\hat{x}_{ick}$ dictates which time intervals (h) will be assigned to nurse i for case c. Hence, $\bar{y}_{ickh} = 1$ if $\hat{x}_{ick} = 1$ and $ST_c \leq h \leq ET_c$ for any $i \in \mathcal{I}, c \in \mathcal{C}, k \in \mathcal{K}, h \in \mathcal{H}$; 0 otherwise. Because an optimal solution is also a feasible solution, a feasible solution of the NAM can be constructed from an optimal solution of the ISM. □

Appendix C. Obtained Results for Data sets 3, 4 and 5

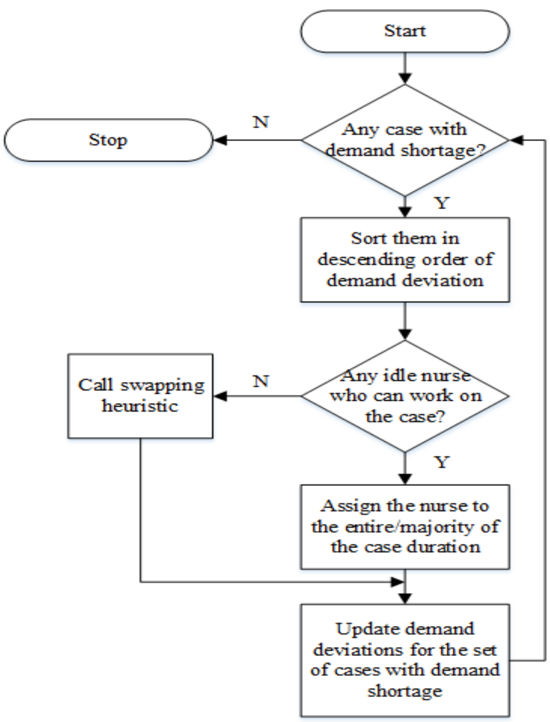
Table 9: Numerical results for data sets 3, 4 and 5

Solution method	Staff shortage ($\mathcal{DE}$)	Overtime assignments ($\mathcal{DF}$)	Idle times ($\mathcal{DS}$)	Time (sec)	CWI
NAM	0	5	10	42400	2.39
SPM	12	6	14	42367	8.96
MGPM	0	6	13	18654	2.52
CGS	12	6	14	11.31	9.07
ISM	12	6	14	0.23	9.07
SWAP	12	6	14	0.63	9.24

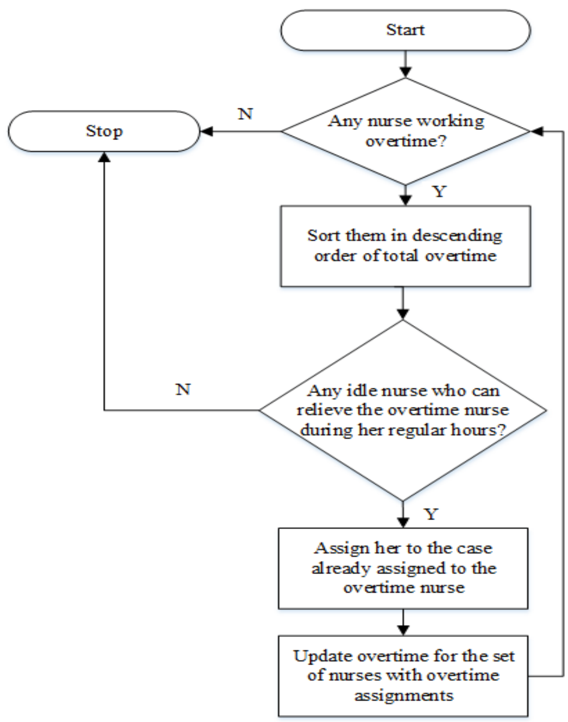
Solution method	Staff shortage ($\mathcal{DE}$)	Overtime assignments ($\mathcal{DF}$)	Idle times ($\mathcal{DS}$)	Time (sec)	CWI
NAM	12	4	18	37610	10.83
SPM	12	6	13	37606	8.88
MGPM	0	5	13	18502	2.57
CGS	12	6	17	28.14	9.30
ISM	12	6	17	0.25	9.31
SWAP	12	6	9	0.65	8.73

Solution method	Staff shortage ($\mathcal{DE}$)	Overtime assignments ($\mathcal{DF}$)	Idle times ($\mathcal{DS}$)	Time (sec)	CWI
NAM	21	4	16	54080	16.01
SPM	22	4	20	54059	14.45
MGPM	4	0	15	34171	4.01
CGS	22	7	18	35.65	14.87
ISM	22	7	18	0.68	14.87
SWAP	22	5	15	2.33	14.41

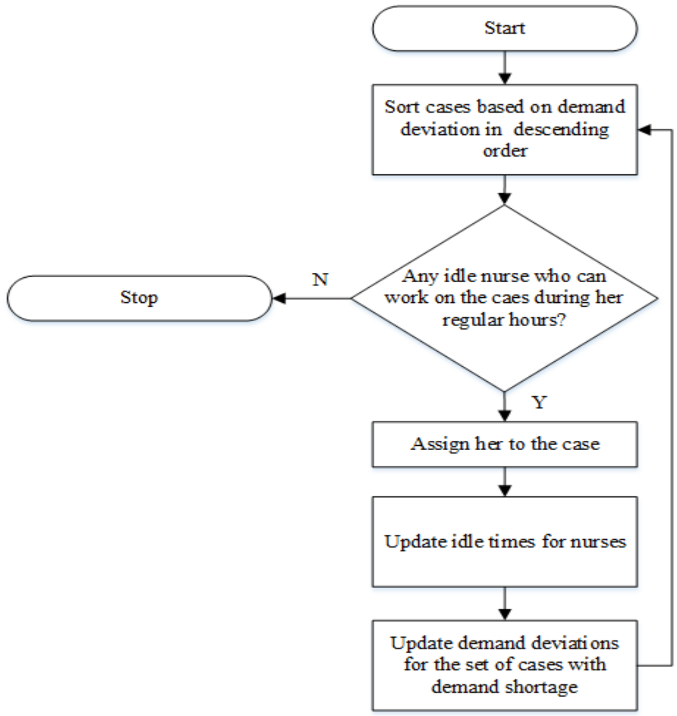
7 Appendix D. SWAP Improvement Phase Flowcharts



(a) Flowchart of staff shortage reduction



(b) Flowchart of staff shortage reduction



(c) Flowchart of idle time reduction