POM-Mar-18-OA-0219

A Network-Based Formulation for Scheduling Clinical Rotations

Andre A. Cire
Department of Management, University of Toronto Scarborough and Rotman School of Management,
Toronto, Ontario, M1C 1A4, andre.cire@rotman.utoronto.ca

Adam Diamant
Schulich School of Business, York University, Toronto, Ontario, M3J 1P3,
adiamant@schulich.yorku.ca

Tallys Yunes
Miami Business School, Coral Gables, Florida, 33124-8237,
tallys@miami.edu

Alejandro Carrasco
School of Medicine, American University of the Caribbean, Coral Gables, Florida, 33134,
acarrasco@aucmed.edu

We investigate the scheduling practices of a medical school that must assign a cohort of students to a series of clinical rotations, while respecting both operational and quality-of-service requirements. Students become available to start clerkship progressively throughout the year and can complete rotations at hospitals in different geographic regions. Each hospital may offer a subset of the clinical rotations, with different start dates, capacities, and cost rates. We propose a novel network-flow model based on decision diagrams, a graphical structure that compresses the state space of a dynamic program, to model feasible schedules. We demonstrate that our network model has several interesting structural features, is computationally superior as compared to a classical mixed-integer linear program, and can be used to generate useful insights that can aid in managerial decision-making. Using a data set collected from the American University of the Caribbean, we perform a counterfactual analysis which shows that had our scheduling approach been implemented, a cost reduction of approximately 19% on average could have been achieved. To understand how assignment decisions can affect future costs, we develop a discrete-event simulation of the licensing exam and clerkship scheduling process. We then compare our exact scheduling approach with current practice and achieve an average cost reduction of 25%. We also show that this cost reduction is robust with respect to estimation and forecast uncertainty, specifically, the licensing exam failure rate and the future cohort size.

Key words: Rotation Scheduling; Medical Training; Network-Based Formulation; Optimization; Simulation

History: Received: April 2018; Accepted: October 2018 by Sergei Savin, after 1 revision.

1. Introduction

The assignment of personnel to shifts is one of the most fundamental problems in scheduling (Ernst et al., 2004). Of particular interest is the field of health care, where scarce resources must be appropriately utilized and budgets carefully balanced. We consider a context where medical students are assigned to a series of rotations of different lengths at hospitals in various geographic locations. This in-hospital component of medical training (i.e., clerkship) provides students with an opportunity to develop clinical skills and ensure they learn how to deliver patient-centered, evidence-based care. It also exposes students to a range of specialization options, such as general surgery, internal medicine, psychiatry, pediatrics, and obstetrics & gynecology. Finally, this type of experiential education is mandatory; only after the successful completion of these rotations are students eligible to graduate and participate in the well-known hospital residency matching programs. In 2016, for example, 19,254 students graduated from 147 medical schools, with the size of a cohort ranging from 50 to 400 students for each school (Association of American Medical Colleges, 2016).

Because clerkship is a required component of medical education, the scheduling of students to core rotations is the responsibility of the medical school. Nonetheless, several factors make this task complex. First, a student can begin their clerkship only after passing a licensing exam, which takes place continuously throughout the year, is scheduled months in advance, and is administered on specialized computers at test centers worldwide. Thus, there is a constant (dynamic) flow of students who become eligible to begin in-hospital training. Second, although the pass rate among first-time test takers is high, some students do fail; in 2015, 4% of first-time test takers had to retake their exam at a later date (U.S. Medical Licensing Examination, 2015). For these students, the start of clerkship must be postponed and the sequence of rotations may have to be rescheduled. Third, multiple hospitals may be affiliated with a single medical school's clerkship program, each offering different rotations, on varying starting dates, with distinct capacities in terms of how many students they can accept on each date. Finally, every medical student in a cohort must be assigned to all core rotations. Since the duration of a clerkship program is typically one to two years, multiple cohorts interact and the length of a medical school's planning horizon ranges from 12 to 36 months.

The assignment of students to clinical rotations has a direct impact on the students' careers, their educational experience, and the school's financial status. We address the complexities of clerkship scheduling by formulating the *rotation assignment and scheduling problem* (RASP). The RASP seeks a minimum-cost policy that assigns a cohort of medical students, who become eligible to start their training over a period of several months, to a series of required clinical rotations, while observing operational and quality-of-service (QoS) requirements. We first present a mixed-integer linear programming (MILP) formulation of the RASP which, although capable of finding optimal solutions, is prohibitively large for practical-sized problem instances. As a result, we propose an alternative network-based formulation (NBF) that is derived from a *decision diagram*, a graphical structure that encodes the set of all feasible clinical rotations, i.e., those that satisfy a medical school's operational and QoS requirements. We prove that both formulations are equivalent and show several structural properties of the NBF that establish its theoretical attractiveness and, more importantly, its usefulness for obtaining managerial insights. We then use real data to generate a collection of problem instances to demonstrate that the NBF exhibits significantly superior computational performance over a range of parameter values when compared to the MILP model.

Although our formulation and solution approach is quite general, the study is motivated by our experience with the American University of the Caribbean (AUC) School of Medicine (AUC, 2018). Each year, AUC schedules between 300 and 350 students to five core rotations at one or more of 25 affiliated hospitals. Rotations are completed one at a time, in any order, and last either 6 or 12 weeks. Hospitals are located in several US states as well as a number of countries around the world (e.g., the UK and Australia). Each hospital may offer a subset of the five rotations, with different start dates, capacities, and cost rates. AUC commits to a certain level of service, ensuring, for example, that students complete rotations in a small number of geographic regions and that they are not idle for excessively long periods between rotations. The school's scheduling decisions are constrained, however, by national licensing requirements (Federation of State Medical Boards and National Board of Medical Examiners, 2018) that limit when a student can begin clerkship, operational restrictions on where and when a student can complete a particular rotation, and QoS constraints that ensure AUC provides its students with a positive learning experience.

Of particular concern is the current practice's demand on human resources. Several staff members are responsible for interacting with students and ensuring that the school's scheduling restrictions are obeyed. Further, assignment decisions are made on an ad-hoc basis using informal rules and somewhat ambiguous institutional guidelines. Previous attempts at an analytical scheduling approach that captured the full complexity of the problem have either proved too computationally intensive, did not allow for subsequent managerial analysis, or did not take into account rescheduling decisions. As a result, it is important that any proposed method address these issues.

We apply the NBF technique to AUC's rotation assignment and scheduling problem by incorporating school-specific constraints. Using six years' worth of student and administrative data collected from AUC, we present two experiments to estimate the cost savings of implementing our scheduling approach in practice. First, we perform a counterfactual analysis by comparing AUC's historical scheduling decisions to those derived from our scheduling policy. We demonstrate that had our scheduling approach been implemented, the estimated savings would have been approximately 19% per year on average. This is due to the better use of capacity at hospitals where AUC has contracts that guarantee a fixed number of rotation slots per cohort. It also indicates that AUC negotiated too many contracts given the number of students they needed to place. We then develop a discrete-event simulation of the licensing exam and clerkship scheduling process to understand how AUC's strategic decisions can affect future performance. We compare our network-based solution with the school's current scheduling practice (at best, a greedy approach), demonstrating that they can expect an average cost reduction of 25% going forward. We show that this result is robust to mis-specifications in the failure rate, i.e., the difference between the NBF and the greedy approach increases as the failure rate decreases. We also demonstrate that our approach dominates the school's current scheduling practice for a wide range of congestion levels (i.e., the number of students versus the number of affiliated hospitals and their corresponding capacities). In some cases, a cost reduction of more than 50% is achieved. Only when the size of the cohort is exceedingly small does the greedy approach offer similar performance as compared to the NBF technique. Finally, we discuss our model's applicability as a managerial decision-making tool by interviewing staff about how it can be utilized going forward.

2. Literature Review

Scheduling in the broad context of health care is an important and active area of research. With the appropriate management of resources, the cost of delivering health services can be reduced while simultaneously providing patients with greater and more equitable access to care. Systems of such type have been developed for many areas of the health services space, in particular for scheduling care providers; see, e.g., Hall (2012), Nickel et al. (2012), Guo et al. (2014), Cappanera and Scutellà (2014), Bard et al. (2014), Smet et al. (2016). Our focus is on the scheduling of medical students to educational opportunities that enable them to learn how best to become care providers in the future.

The nurse scheduling/rostering problem has an extensive literature (e.g., Warner, 1976, Cheang et al., 2003, Burke et al., 2004, Van den Bergh et al., 2013). The objective is to create work schedules over a planning horizon that ensure the appropriate number of nurses, each with a requisite skill level, are present for each shift. The flexibility of the schedule is limited by organizational constraints, such as the number of working hours before a break must be taken, financial restrictions that limit the amount of overtime that is allowed, and temporal constraints such as the number of consecutive work days. A feature that is often important in nurse scheduling are the preferences of those being scheduled (e.g., seniority, holidays, day/night shifts). From conversations with AUC staff, students do indicate which hospitals or geographic regions they prefer. Nonetheless, these are used as guidelines by the human schedulers, having lower priority than more general QoS requirements.

Several papers address the scheduling of medical residents to shifts, a stage that comes after students graduate from medical school. Cohn et al. (2009) develop a mathematical programming tool that constructs annual on-call schedules for residents specializing in psychiatry at the Boston University School of Medicine. Agarwal (2016) formulate and solve a multi-objective optimization problem for generating resident schedules for Rochester General Hospital. Bard et al. (2016) develop several optimization-based heuristics to construct annual block schedules for family medicine residents at the University of Texas Health Science Center. Smalley and Keskinocak (2016) propose two integer programming models to assign surgical residents to shifts for the Surgeon Residency Program at Emory University School of Medicine. Finally, Lemay et al. (2017) introduce a pair of algorithms that solve a sequence of optimization problems that, in addition to ensuring appropriate patient coverage and adequate training opportunities, incorporates resident preferences for downtime. A feature that concerns resident scheduling, but not the RASP, is their working conditions (Ludmerer, 2009) and the quality of patient care provided. In addition, as addressed in Smalley and Keskinocak (2016), there is a difference between *rotation* and *shift* scheduling for residents; the former being an inter-day problem, and the latter an intra-day problem. For the RASP, intra-day decisions do not play a role.

The RASP is also closely related to classical machine scheduling problems, with students, hospitals, start dates, and capacity constraints playing the role of jobs, machines, release dates, and resource constraints, respectively (Baptiste et al., 2001, Pinedo, 2008). Specifically, the rotations can be perceived as tasks of a particular job subject to additional deadline constraints and assignable to any subset of machines that can perform them. Problems of this sort are notoriously difficult to solve efficiently. For example, even if the order of the rotations were fixed and students were already assigned to hospitals, the RASP would reduce to an instance of the job-shop scheduling problem, which is NP-Hard in general (Garey et al., 1976) and still challenging for realistically-sized problems.

Our work is closely related to scheduling models that take advantage of an underlying network structure. Eppen and Martin (1987) present an alternative formulation of the multi-item capacitated lot-sizing problem by reformulating it as a shortest-path problem. Beasley and Cao (1998) analyze the crew scheduling problem and use a dynamic programming approach to represent the complex MILP model as a network. Raffensperger (1999) applies this technique to the cutting stock, tank scheduling, and traveling salesperson problems. Côté et al. (2011) introduce a grammar-based approach to solving a large-scale MILP model where the cost of assigning an identical set of workers to multiple activities during a single day of operation is minimized. Restrepo et al. (2012) analyze a similar setting, but present a column generation approach where the solution of the subproblem (i.e., a feasible shift schedule) is obtained by formulating and solving a shortest-path problem with resource constraints.

In this stream of research, an optimization model is reformulated as a network flow problem on an extended variable space that yields better relaxations and solves more quickly. We add to this research by representing the problem as a decision diagram, which exploits redundancy to reduce the size of the network and is the basis for our network-based MILP model. Moreover, we demonstrate that our network can be used to derive additional managerial insights and as a decision support tool. The idea of a network-based formulation also appears in Zhang et al. (2016). For a comprehensive treatment of extended formulations, see Conforti et al. (2010) and Lancia and Serafini (2018); for decision diagrams and optimization, we refer the reader to Bergman et al. (2016).

The remainder of the paper is organized as follows. We formally describe the RASP in Section 3 and then introduce a classical MILP model in Section 3.1. In Section 3.3, we introduce multivalued decision diagrams (MDDs) and, in Section 3.5, we use the RASP's MDD to create a network-based formulation (NBF) for the problem. In Section 4, we apply our solution approach to AUC by incorporating their specific QoS constraints into our models. Our numerical analysis with both actual and randomly generated RASP instances appears in Section 5. Finally, we provide managerial insights and concluding remarks in Sections 6 and 7. All proofs are located in Appendix B.

3. The Rotation Assignment and Scheduling Problem (RASP)

Consider a set of medical students $\mathcal{I} = \{1, \ldots, I\}$ that are to be scheduled to a set of clinical rotations $\mathcal{J} = \{1, \ldots, J\}$ at a set of hospitals $\mathcal{K} = \{1, \ldots, K\}$. Let t index the weeks in the planning horizon, which ranges from $t = 1$, the earliest time any student can start in-hospital training, up to $t = T$, the latest end time of any student's final rotation. Each student $i \in \mathcal{I}$ must complete all rotations $j \in \mathcal{J}$ exactly once before week T. Let $\mathcal{I}(t) \subseteq \mathcal{I}$ be the set of students who are eligible to start clerkship on week t because they have successfully passed their licensing exam prior to week t. Further, for each $i \in \mathcal{I}$, we define t_i to be the earliest week student i can begin clerkship, i.e., $t_i = \min\{t \,|\, i \in \mathcal{I}(t)\}$.

We consider a deterministic version of the problem where $\mathcal{I}(t)$ is known with certainty for all t. This is a sensible assumption for two reasons. First, the licensing exam is booked 3-6 months in advance and test scores are typically released eight weeks after the exam is taken. Medical schools are generally aware of these dates and typically schedule students by assuming these dates are firm. Second, most students pass their exam; among the 20,213 students who took the US licensing exam in 2015, 96% passed on their first attempt (U.S. Medical Licensing Examination, 2015). Thus, although students do fail, this occurrence happens infrequently. Nevertheless, in Section 4.2, we address the issue regarding the re-scheduling of students who do not pass their exam.

For each rotation $j \in \mathcal{J}$, let $\Delta_j \in \mathbb{Z}_+$ be its duration in weeks and let $\mathcal{K}(j) \subseteq \mathcal{K}$ be the set of hospitals that offer rotation j. Conversely, let $\mathcal{J}(k)$ be the set of rotations offered by a hospital $k \in \mathcal{K}$. In addition to the set of rotations that are offered, a hospital is defined by several other attributes:

Rotation Capacity: Let C_{jkt} be the maximum number of students who can start rotation $j \in \mathcal{J}(k)$, at hospital k, on week t. If students cannot begin a rotation at hospital k, on week t, $C_{jkt} = 0$.

Fixed Cost: In exchange for a fixed amount c^F_k, hospital k may sign a multi-year contract with the medical school guaranteeing a pre-determined number of rotation slots per year. The set of hospitals that impose this type of fee structure is denoted by $\mathcal{K}_F$; we refer to these institutions as fixed-cost hospitals. Since fixed-cost hospitals have different characteristics (e.g., rotations offered, capacities, management procedures), the terms of each contract may also differ. As a result, even if $c^F_k = c^F_{k'}$ for hospitals $k, k' \in \mathcal{K}_F$, the number of guaranteed slots may not be equal.

Further, each hospital negotiates a different contract duration (in years) with the medical school. To this end, let $U_k(t)$ be a function that maps the week number t to the appropriate contract *year* for some hospital $k \in \mathcal{K}_F$; let u_k be the maximum duration of any contract that can be signed with hospital $k \in \mathcal{K}_F$. In our setting, the duration of each contract can be no less than one year while multi-year, consecutive and non-consecutive contracts can also be signed. Finally, note that fixed-cost hospitals may still charge a per student fee of c_{jk} for rotation $j \in \mathcal{J}(k)$.

Variable Cost: The set of hospitals that do not negotiate fixed contracts, but instead, charge an amount c_{jk} for an assigned student to complete rotation $j \in \mathcal{J}(k)$ is denoted by $\mathcal{K}_V$ where $\mathcal{K}_V(j) = \mathcal{K}_V \cap \mathcal{K}(j)$; we refer to these institutions as variable-cost hospitals. To facilitate a more compact formulation, we define $U_k(t)$ for a hospital $k \in \mathcal{K}_V$. It is an identity function, i.e., $U_k(t) = t$, that maps the week number t to the appropriate *week* for some hospital $k \in \mathcal{K}_V$. The interpretation here is that for variable-cost hospitals, only weekly contracts for a single slot per rotation can be signed with the medical school. Thus, $u_k = T$ represents the maximum weekly contract that can be signed with hospital $k \in \mathcal{K}_V$ over the planning horizon.

In practice, if a fixed-cost hospital charges a rotation cost c_{jk}, it is much smaller than the fee charged by variable-cost hospitals. However, partitioning the set of hospitals into these cost classes allows the medical school to track when their accounts must be settled. For fixed-cost hospitals, payment for in-hospital training is remitted prior to the start of the planning horizon. For variable-cost hospitals, payment can be delayed until just before the start of the student's rotation.

Each student $i \in \mathcal{I}$ must be assigned a schedule $\mathcal{S}_i$ which corresponds to a sequence of J stages, one for each core rotation. In stage $s = 1, \ldots, J$, the student starts rotation j_s, at hospital k_s, on week t_s. That is, a schedule for a student i is the sequence

$$\mathcal{S}_i = ((j_1, k_1, t_1), (j_2, k_2, t_2), \ldots, (j_J, k_J, t_J)). \tag{1}$$

A schedule is feasible if it satisfies two *core* requirements, as follows. These requirements are scheduling restrictions that are typically observed by all medical schools in North America.

R.1 A schedule must contain all rotations in $\mathcal{J}$ exactly once, in any order.

R.2 Rotations must be completed one at a time (no overlapping).

A feasible solution to the RASP is an assignment of I feasible schedules to students (i.e., one schedule for every student in a cohort) satisfying the following licensing and capacity constraints.

R.3 Student i cannot start clinical rotations before week t_i (i.e., before passing the licensing exam).

R.4 The number of students that start rotation j, at hospital k, on week t, must not exceed C_{jkt}.

The objective of the RASP is to find a collection of feasible schedules for a given cohort of students $\mathcal{I}$ that satisfies the licensing/capacity restrictions and minimizes the sum of fixed and variable costs.

3.1. A MILP Model

We first present a time-indexed MILP formulation for the RASP to determine $\mathcal{S}_i$ for every $i \in \mathcal{I}$. This type of model is attractive because different objectives and side constraints can be easily incorporated. It also provides strong dual bounds (Van den Akker et al., 2000) compared to event-based formulations, and is standard when formulating resource constraints (Artigues, 2017).

Let x_{ijkt} be a binary variable that equals one if student i is assigned to start rotation j, at hospital k, on week t, and zero otherwise. Let z_{ku} be a binary variable that equals one if there is at least one student assigned to a hospital $k \in \mathcal{K}_F$ during contract year u or a hospital $k \in \mathcal{K}_V$ during week u and zero otherwise. Then, the MILP formulation of the RASP is given by

$$Z_{RASP} = \min \left\{ \sum_{k \in K_F} \sum_{u=1}^{u_k} c_k^F z_{ku} + \sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{k \in \mathcal{K}(j)} \sum_{t=t_i}^{T} c_{jk} x_{ijkt} \right\}, \tag{MP}$$

subject to:

$$\sum_{k \in \mathcal{K}(j)} \sum_{t=t_i}^{T} x_{ijkt} = 1, \qquad \forall\, i, j, \tag{2}$$

$$\sum_{j=1}^{J} \sum_{k \in \mathcal{K}(j)} \sum_{s=t-\Delta_j+1}^{t} x_{ijks} \leq 1, \qquad \forall\, i, t \geq t_i, \tag{3}$$

$$\sum_{i \in \mathcal{I}(t)} x_{ijkt} \leq C_{jkt} z_{ku}, \qquad \forall\, j, k \in \mathcal{K}(j), t, u \in U_k(t), \tag{4}$$

$$x_{ijkt} \in \{0,1\}, \qquad \forall\, i, j, k \in \mathcal{K}(j), t \geq t_i, \tag{5}$$

$$z_{ku} \in \{0,1\}, \qquad \forall\, k, u \tag{6}$$

The objective minimizes the fixed and variable costs associated with assigning students to hospitals and clinical rotations. Constraint (2) enforces (**R.1**): each student is assigned to exactly one of each type of rotation and all rotations are completed. Constraint (3) enforces (**R.2**): no more than one rotation can be completed at any given time. Requirement (**R.3**) is enforced by not creating x_{ijkt} variables with $t < t_i$. Constraint (4) enforces (**R.4**) for fixed- and variable-cost hospitals: the maximum capacity of rotation j, at hospital k, during week t. If week t is a valid start date for rotation j at hospital k, $C_{jkt} > 0$; otherwise $C_{jkt} = 0$. It also ensures that $z_{ku} = 1$ whenever a student is assigned to hospital $k \in \mathcal{K}_F$ during contract year u or, for hospital $k \in \mathcal{K}_V$, week u.

3.2. Formulation Size: A Case For and Against Column Generation

Problem (MP) has $O(I \times J \times K \times T)$ variables and $O(I \times T)$ constraints. In the case of AUC, this amounts to over 4,000,000 decision variables and approximately 50,000 constraints. Further, many medical schools have QoS requirements, i.e., additional scheduling restrictions that are unique to the school (see Section 4 for AUC's QoS constraints). When faced with such a large model, a typical approach is to formulate an alternative MILP model that creates one variable for each feasible schedule, as defined by (1). These schedules are sometimes referred to as *patterns*. In such a model, the complexity of what constitutes a feasible pattern is embedded in a pattern-generation step that is exogenous to the MILP. As a consequence, the model itself becomes relatively straightforward: pick one pattern for each student so that costs are minimized and a small number of simple side constraints are satisfied. The issue with these formulations is the large number of variables (all distinct feasible schedules/patterns), which leads to the popular approach known as *branch-and-price*, i.e., column generation embedded inside a branch-and-bound tree search (Barnhart et al., 1998).

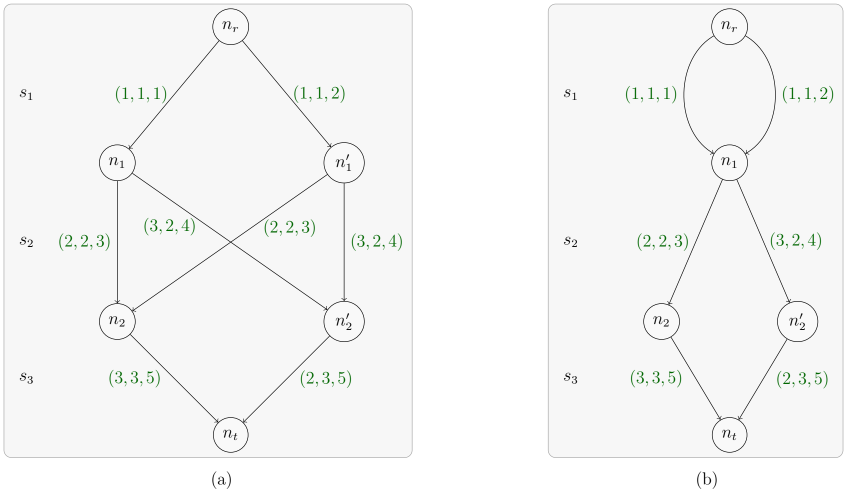
Branch-and-price algorithms are notoriously difficult to implement and fine tune for good performance. There are typically several issues that must be addressed, such as choosing an appropriate branching scheme (Vanderbeck, 2011), dual stabilization techniques to improve convergence (Clautiaux et al., 2011), primal heuristics (Joncour et al., 2010), and proper management of the pool of generated columns (Barnhart et al., 1998). As a result, we present an alternative approach that is, in many ways, equivalent to column generation but possesses several advantages. This includes a significantly more compact (but still exact) representation of the set of feasible patterns, dual bounds that are just as strong, the ability to perform sensitivity analysis (e.g., see Section 6), and a much simpler implementation that takes advantage of existing solvers. As shown in the following sections, the idea is to develop network-based reformulation of the problem, extracted from a dynamic programming model, that is equivalent to (MP) in that it captures all feasible student schedules.

3.3. A Multivalued Decision Diagram for the RASP

We present a reformulation of the RASP based on decoupling the assignment and scheduling elements of the problem. This is achieved by encoding all feasible schedules into a *multivalued decision diagram* (MDD) (Bergman et al., 2016). This network-style representation carries significantly more structure than the MILP model. In particular, in certain settings, the optimal solution can be obtained by solving a minimum-cost flow problem on the MDD. Further, this MDD can help derive additional managerial insights associated with rescheduling students and dealing with undesired assignments.

Formally, we create a directed acyclic graph $D = (N, A)$ that compactly encodes the set of feasible schedules $\mathcal{S}$ that can be assigned to a student. The set of nodes N is partitioned into $J+1$ layers $L_0, \ldots, L_J$ where L_0 and L_J are singletons containing a root node n_r and a terminal node n_t, respectively. Arcs in A only connect nodes in adjacent layers of D. An arc $a = (n, n')$, with $n \in L_s$, $n' \in L_{s+1}$, is associated with the triplet $\theta(a) = (j_s, k_s, t_s)$ which represents the s-th stage in the student's schedule. Thus, a path of length J specified by the sequence of arcs $(a_1, a_2, \ldots, a_J)$ starting at n_r and terminating at n_t (also known as an n_r, n_t-path) encodes a feasible schedule $\mathcal{S} = (\theta(a_1), \theta(a_2), \ldots, \theta(a_J))$. In this graphical representation, there exists a one-to-one mapping between the set of feasible schedules and all n_r, n_t-paths in D. Using the network alone, the RASP (without **R.3** and **R.4**) now reduces to finding a set of paths of length J in D with the lowest total cost such that all medical students are assigned a schedule and all core requirements are met.

Example 1. Consider an instance of the RASP with three hospitals ($K = 3$) and three rotations ($J = 3$), each lasting one week. The first hospital offers rotation 1 starting on weeks $t = \{1, 2\}$ (i.e., C_{111} and $C_{112} > 0$). The second hospital offers rotations 2 and 3 starting on weeks $t = 3$ and $t = 4$, respectively. The third hospital also offers rotations 2 and 3, but they both start on week $t = 5$.



(a)

(b)

Figure 1 Two distinct MDDs encoding the same set of feasible rotation schedules of Example 1.

Two MDDs encoding the set of feasible schedules for Example 1 are presented in Figure 1. The path (n_r, n_1, n'_2, n_t) in Figure 1(a) encodes a schedule where a student starts rotation 1 at the first hospital on week $t = 1$, proceeds to rotation 3 at the second hospital on week $t = 4$, and starts rotation 2 at the third hospital on week $t = 5$. Figure 1(b) shows the exact same set of schedules in a more compact fashion. Our methodology constructs the smallest of such graphs, known as a *reduced MDD*, for any arbitrary instance of the RASP. This is achieved by exploiting the fact that feasible schedules (i.e,. paths through the network) often share equivalent subsequences, leading to redundancies that can be eliminated. For example, in Figure 1(b), the two paths/schedules starting with $(1, 1, 1)$ and $(1, 1, 2)$ have the same subsequences in the following layers and, hence, do not need to be represented by two distinct nodes in their layer as they were in Figure 1(a).

The MDD is constructed by developing a dynamic program (DP). Nodes encode the states of the system (i.e., partial feasible schedules) while arcs represent feasible actions (i.e., possible rotation assignments) that can be taken at each state. Hence, the DP enumerates all possible states, which are implicitly encoded as paths from n_r to n_t in the MDD. In our case, paths are constructed using a recursive model that considers both the set of clinical rotations left to complete at each stage and the weeks in which the remaining rotations can begin. Thus, the MDD is a compressed version of the underlying state-transition graph of the DP, where isomorphic subtrees are merged and costs are attributed to arcs as opposed to state transitions (Hooker, 2013).

We now present a dynamic programming formulation that models a feasible student schedule over a finite planning horizon. Such a model provides the theoretical basis for calculating the size of D which, for practical instances of our problem, is sufficiently small.

Proposition 1. *A schedule $\mathcal{S}$, as defined in (1), is feasible if and only if there exists a sequence of rotations-to-go ($\mathcal{G}$) and earliest start time (τ) pairs $(\mathcal{G}_0,\tau_0),(\mathcal{G}_1,\tau_1),(\mathcal{G}_2,\tau_2),\ldots,(\mathcal{G}_J,\tau_J)$, where $\mathcal{G}_s \subseteq \mathcal{J}$ and $\tau_s \in \mathbb{Z}_+$ for $s=0,\ldots,J$, such that $(\mathcal{G}_0,\tau_0)=(\mathcal{J},0)$ and*

$$\mathcal{G}_s = \mathcal{G}_{s-1} \setminus \{j_s\}, \tag{7}$$

$$\tau_s = t_s + \Delta_{j_s}, \tag{8}$$

$$(j_s,k_s,t_s) \in \mathcal{A}(\mathcal{G}_{s-1},\tau_{s-1}), \tag{9}$$

for $s=1,\ldots,J$, where $\mathcal{A}(\mathcal{G}_s,\tau_s)=\{(j,k,t): j\in\mathcal{G}_s, k\in\mathcal{K}(j), t\geq\tau_s\}$.

A consequence of Proposition 1 is that every feasible schedule can be obtained by enumerating all $(\mathcal{G}_s,\tau_s)$ pairs. Assuming a planning horizon of T weeks, the total number of pairs is bounded by $O(2^J\times T)$, which is small in many practical instances (e.g., if $J=5$ and $T=104$, $2^J\times T=3328$). Even in large-scale implementations, the number of such pairs can be exponentially smaller than the set of feasible schedules. In fact, in Section 4, we show that the number of $(\mathcal{G}_s,\tau_s)$ pairs remains tractable even when AUC's QoS constraints are incorporated into the network.

3.4. MDD Construction

The MDD is built in two phases. In the first phase, we construct a directed acyclic graph $D=(N,A)$ encoding the set of transitions of the recursive model from Proposition 1. Specifically, the set of nodes N is partitioned into layers $L_0,L_1,\ldots,L_J$. There exists a one-to-one mapping between a node $n\in L_s$ and a pair $(\mathcal{G}_s,\tau_s)$ that is reachable at stage s by some feasible schedule. The pair associated with a node is henceforth denoted by $\xi(n)$, e.g. $\xi(n_r)=(\mathcal{G}_0,\tau_0)$. An arc $a=(n,n')$ connects nodes in adjacent layers, $n\in L_s$, $n'\in L_{s+1}$, if and only if there exists a tuple $\theta(a)=(j_s,k_s,t_s)\in\mathcal{A}(\xi(n))$ in some feasible schedule which transitions $\xi(n)$ to $\xi(n')$ according to (7)–(8). This procedure is summarized in Algorithm 1. It considers each pair $\xi(n)$ (i.e., a node in the network) exactly once. Because a node can have at most $|\mathcal{A}(\xi(n))|$ outgoing arcs, the algorithm to generate the network has a (conservative) worst-case time complexity of $O(2^J\times T)\times O(J\times K\times T)=O(J\times 2^J\times K\times T^2)$.

Example 2. Figure 1(a) depicts the MDD obtained after applying Algorithm 1 to the instance described in Example 1. In particular, $\xi(n_r)=(\{1,2,3\},0)$, $\xi(n_1)=(\{2,3\},2)$, $\xi(n_1')=(\{2,3\},3)$, $\xi(n_2)=(\{3\},4)$, $\xi(n_2')=(\{2\},5)$, and $\xi(n_t)=(\emptyset,6)$.

Algorithm 1 MDD Compilation Algorithm

```
1: Create node $n$ associated with pair $\xi(n) = (\mathcal{G}_0, \tau_0)$.
2: Let $L_0 = \{n_r\}$, $A = \emptyset$.
3: For each $s = 1, \ldots, J-1$:
    (a) Let $L_{s+1} = \emptyset$.
    (b) For each node $n \in L_s$ and $(j_s, k_s, t_s) \in \mathcal{A}(\xi(n))$:
        (i) Let $n'$ be a node such that $\xi(n') = (\mathcal{G}_{s+1}, \tau_{s+1})$ in $L_{s+1}$ according to (7)–(8).
            If no such node exists, add it to $L_{s+1}$.
        (ii) Create arc $a = (n, n')$ associated with $\theta(a) = (j_s, k_s, t_s)$ and add it to $A$.
4: Return $D = (N, A)$.
```

The second phase consists of merging *equivalent* nodes of D to remove redundancy. Two nodes $n, n' \in L_s$ are considered equivalent if, for any arc-specified path $(a_1, \ldots, a_{J-s})$ starting at node n and ending at a node $\tilde{n} \in L_J$, there exists another path $(a'_1, \ldots, a'_{J-s})$ starting at n' and ending at $\tilde{n}$ such that $\theta(a_i) = \theta(a'_i)$ for all $i = 1, \ldots, J-s$. That is, any partial schedule that reaches a state $\xi(n) = (\mathcal{G}_s, \tau_s)$ can be completed in the same way as any partial schedule that reaches $\xi(n') = (\mathcal{G}'_s, \tau'_s)$. The task of identifying and merging equivalent nodes is called *reduction* and is described in Algorithm 2. The reduction phase has a quadratic worst-case time complexity in the number of nodes of D. More sophisticated methods can be implemented with a linear time complexity (Wegener, 2000). We note that the reduction operation is a critical step in our context. In our numerical experiments, MDDs were approximately 50% smaller on average after the second phase as compared to the first.

Algorithm 2 MDD Reduction Algorithm

```
1: Merge all nodes of $L_J$ into a single terminal node $n_t$.
2: For $s = J-1, J-2, \ldots, 1$:
    (a) For each pair of nodes $n, n' \in L_s$, $n \neq n'$:
            If for each arc $a = (n, \tilde{n})$ there exists an arc $a' = (n', \tilde{n})$ with $\theta(a) = \theta(a')$ for some
            $\tilde{n} \in L_{s+1}$, redirect incoming arcs of $n'$ to $n$ and remove $n'$ from $L_s$.
```

Example 3. Figure 1(b) depicts the result of applying Algorithm 2 to the MDD in Figure 1(a). Starting in the second-to-last layer, nodes n_2 and n'_2 cannot be merged because $\theta((n_2, n_t)) \neq \theta((n'_2, n_t))$. Going up one layer, for every arc leaving node n_1 there exists an arc leaving node n'_1 with the same endpoint (either n_2 or n'_2) and the same value of $\theta(\cdot)$. Therefore, n_1 and n'_1 can be merged.

3.5. A Network-Based Formulation

The decision diagram D encodes the set of all feasible schedules, i.e., those that satisfy core requirements **R.1** and **R.2**. Thus, determining feasible schedules that minimize the sum of all variable costs can be obtained by solving a minimum-cost flow problem. All feasible solutions to the RASP are feasible schedules in D, although not all feasible schedules are feasible solutions to the RASP, i.e., they may not satisfy **R.3** and **R.4**. Further, fixed-cost hospitals must also be included in the formulation. Thus, we present a network-based MILP formulation (NBF) of the RASP which uses the MDD's underlying graphical structure as input. The network encodes all feasible schedules while the MILP includes fixed-cost hospitals and enforces **R.3** and **R.4** to find the minimum cost solution.

Lemma 1. *Given a decision diagram $D = (N, A)$, let $\delta^+(n)$ and $\delta^-(n)$ denote the set of outgoing and incoming arcs at a node $n \in N$, respectively. For any $m \in \mathbb{Z}_+$, there exists a one-to-one correspondence between m-sized subsets of feasible schedules and integral points of the network-flow polyhedron*

$$\texttt{flow}(D,m) = \left\{ \mathbf{y} \in \mathbb{R}_+^{|A|} : \sum_{a \in \delta^+(n_r)} y_a = m \quad \text{and} \quad \sum_{a \in \delta^-(n)} y_a - \sum_{a \in \delta^+(n)} y_a = 0, \; \forall\, n \in N \setminus \{n_r, n_t\} \right\},$$

where y_a is a non-negative decision variable that corresponds to the amount of flow on arc a.

By Lemma 1, this reformulation of the RASP consists of finding an m-unit flow over D satisfying **R.3** and **R.4**. For each arc $a \in A$, with a slight abuse of notation, let j_a, k_a, and t_a be the rotation, hospital, and start week associated with $\theta(a)$, respectively. In addition, let $\mathcal{C}(j,k,t) \subseteq A$ be the set of arcs a with $j_a = j$, $k_a = k$, and $t_a = t$, and let $c_{j_a k_a}$ be the rotation cost per unit of flow on arc a. If y_a is the amount of flow on arc a, then the reformulation of the RASP is given by

$$Z_{RASP} = \min \left\{ \sum_{k \in K_F} \sum_{u=1}^{u_k} c_k^F \, z_{ku} + \sum_{a \in A} c_{j_a k_a} y_a \right\}, \tag{NBF}$$

subject to:

$$\mathbf{y} \in \texttt{flow}(D, I), \tag{10}$$

$$\sum_{a \in \mathcal{C}(j,k,t)} y_a \le C_{jkt} z_{ku}, \qquad \forall\, j, k \in \mathcal{K}(j), t, u \in U_k(t), \tag{11}$$

$$\sum_{\{a \in \delta^+(n_r) \,:\, t_a \ge t\}} y_a = \sum_{s \ge t} |\mathcal{I}(s) \setminus \mathcal{I}(s-1)|, \qquad \forall\, t, \tag{12}$$

$$y_a \in \{0, \ldots, I\}, \qquad \forall\, a, \tag{13}$$

$$z_{ku} \in \{0, 1\}, \qquad \forall\, k, u. \tag{14}$$

where we assume $\mathcal{I}(0) = \emptyset$. The objective of (NBF) is equivalent to that of (MP). Constraint (10) ensures $\mathbf{y}$ is a subset of I feasible student schedules (by Lemma 1). Constraint (11) plays the role of

(4), as $\mathcal{C}(j,k,t)$ aggregates flow units (students) with the same rotation j, hospital k, and start week t. Constraint (12) ensures that students do not start rotations before they become available.

An benefit to using an MDD for the reformulation is that the underlying network structure can be leveraged to derive efficient algorithms for special cases of NBF that are relevant in practice. For instance, AUC does not often make joint contracting and scheduling decisions. Instead, they first negotiate fixed contracts with hospitals and then assign students to rotations several months later. In this special case, $c_k^F = 0$ for all $k \in \mathcal{K}_F$ as rotation slots for fixed-cost hospitals have already been guaranteed. Thus, it is advantageous to assign as many students to these hospitals as possible. AUC, however, may wish to impose additional scheduling restrictions when defining a feasible assignment of students to rotations. As long as these extra requirements can be incorporated directly in the decision diagram (which may increase the size of the graph), the next proposition demonstrates that the problem can still be solved efficiently in the size of the network.

Proposition 2. *Let $c_k^F = 0$ for all $k \in \mathcal{K}_F$ and let $D = (N, A)$ be constructed so that all feasible paths in the network satisfy any additional requirements imposed by the medical school. If the hospital capacities are sufficiently large, NBF can be solved in strongly polynomial time in the size of D.*

Moreover, in some medical schools, students are required to complete clerkship in a specific order (Wimmers et al., 2007). In this case, optimal schedules can also be derived efficiently in the size of D if no additional requirements are imposed by the school, as we prove in Proposition 3.

Proposition 3. *Let $c_k^F = 0$ for all $k \in \mathcal{K}_F$ and suppose that no additional requirements are imposed by the medical school. If students must complete their rotations in a pre-determined sequence, the optimal solution to NBF can be obtained in polynomial time in the size of D.*

We note that the network structure of D implicitly handles the complexity of the sequencing decisions in this reformulation. However, the general version of the RASP, which includes hospital capacities and possibly more general scheduling requirements (e.g., AUC's QoS constraints), still remains theoretically hard. We formalize this statement below.

Proposition 4. *The decision version of NBF is strongly NP-Hard even when $c_k^F = 0$ for all $k \in \mathcal{K}_F$ and D is polynomial in the size of the problem input.*

4. Incorporating Quality-of Service Requirements: AUC Case Study

In this section, we demonstrate how to incorporate QoS requirements into our model. These requirements are unique to the individual medical school. They represent additional constraints that must be added to the RASP to ensure rotation schedules satisfy internal procedures but are also well-received by prospective students. For AUC, these requirements are:

R.5 *Student Preferences:* Students may request to be placed at hospitals in specific areas of the country. Let $r_k \in \{1,\dots,R\}$ identify the geographic region where hospital k is located. For example, at AUC, some of the affiliated hospitals are located in New York City, London, England, and Miami, Florida. The medical school tries to fulfill these requests whenever possible.

R.6 *Relocation:* It is costly for students to resettle in different geographic regions. Therefore, each student must complete all of the required rotations with no more than $\bar{r}$ region relocations and no back-and-forth movement among them. In addition, as a measure of QoS, AUC ensures that at least $\alpha\%$ of their students complete all of their rotations within a single region.

R.7 *Idleness:* For each student, there can be no more than ε weeks of separation between the end of one rotation and the beginning of the next rotation.

R.8 *All-or-None:* Any student assigned to a fixed-cost hospital must complete all the rotations offered by that hospital (in an effort to maximize the use of the fixed contract).

The combination of the *Relocation* and *Idleness* restrictions ensure that final schedule does not place an undue burden on the capability of a student to complete their clerkship to the best of their ability. For example, AUC ensures that students are idle for no more than eight weeks between rotations ($\varepsilon = 8$), that students can be assigned to at most two geographic regions ($\bar{r} = 1$), and that 75% of all students must be assigned to exactly one region ($\alpha = 75$). Note that there are more hospitals than regions, as some hospitals are assigned to the same geographic region. For example, AUC is affiliated with several hospitals in New York City which are all assigned to the same region. A hospital in Miami, Florida, however, would be considered an entirely different geographic region.

We note that (**R.5**)-(**R.8**) represent the standard QoS constraints that AUC must obey when scheduling students to rotations. Additional idiosyncratic restrictions that represent each student's personal preferences may be added to the model. In this case, the solution becomes highly specialized and provides one schedule for every student in the cohort. However, these restrictions vary from year to year and, thus, we focus on QoS requirements that remain constant for all cohorts. The added benefit is that a feasible schedule can now be assigned to any student i provided $t_i \geq t$. More specifically, the output of the model represents flow on the network D. Any path from n_r to n_t with positive flow is a sequence of rotations that a student could follow. The value of the flow equals the number of students who will have that exact same schedule. Therefore, when students sit with their advisors to pick a schedule, they can be presented with several options (i.e., different paths in D) that satisfy their starting date t_i and are not yet assigned to other students. Each such path may be (subjectively) more or less desirable according to each student's personal preferences.

4.1. MILP Representation of AUC's QoS Requirements

We first add AUC's QoS requirements to the time-indexed MILP formulation for the RASP introduced in Section 3.1, i.e., problem (MP). Let v_{it} be a binary variable that equals one if student i

completes their last rotation at time t and zero otherwise. Let w_{ir} be a binary variable that equals one if student i is assigned to a hospital in geographic region r and zero otherwise. Finally, let $\tilde{w}_i$ be a binary variable that equals zero if student i completes their entire training in one geographic region, and equals one if i's rotations take place in two different regions (recall that $\bar{r}=1$ for AUC).

$$\sum_{t=t_i}^{T} v_{it} = 1, \qquad \forall\, i, \tag{15}$$

$$\sum_{k\in\mathcal{K}(j)} x_{ijkt} - \sum_{s=t+\Delta_j}^{t+\Delta_j+\varepsilon} \sum_{j'\neq j} \sum_{k\in\mathcal{K}(j')} x_{ij'ks} \le v_{it}, \qquad \forall\, i,j,t\ge t_i, \tag{16}$$

$$v_{it} + \sum_{s=t+1}^{T} \sum_{k\in\mathcal{K}(j)} x_{ijks} \le 1, \qquad \forall\, i,j,t\ge t_i, \tag{17}$$

$$\sum_{t=t_i}^{T} x_{ijkt} \le w_{ir_k}, \qquad \forall\, i,j,k\in\mathcal{K}(j), \tag{18}$$

$$\sum_{r=1}^{R} w_{ir} \le 1 + \tilde{w}_i, \qquad \forall\, i, \tag{19}$$

$$\sum_{i=1}^{I} \tilde{w}_i \le 0.25\, I, \tag{20}$$

$$x_{ij_1k_1t_1} + x_{ij_2k_2t_2} + \sum_{s=t_2+1}^{T} x_{ij_3k_3s} \le 2, \qquad \begin{array}{r} \forall\, \{k_1,k_2,k_3\}\in\mathcal{K}\,|\, r_{k_1}\neq r_{k_2}, r_{k_2}\neq r_{k_3}, \\ i, t_i \le t_1 < t_2,\ \text{distinct } j_1,j_2,j_3, \\ j_1\in\mathcal{J}(k_1), j_2\in\mathcal{J}(k_2), j_3\in\mathcal{J}(k_3), \end{array} \tag{21}$$

$$\sum_{t=t_i}^{T} x_{ijkt} = \sum_{t=t_i}^{T} x_{ij'kt}, \qquad \forall\, i,k\in\mathcal{K}_F, j \text{ and } j'\in\mathcal{J}(k), j<j', \tag{22}$$

$$w_{ir}, \tilde{w}_i, v_{it} \in \{0,1\}, \qquad \forall\, i,r,t, \tag{23}$$

Constraints (15)–(17) enforce (**R.7**). That is, (15) ensures that each student completes all core rotations before week T, (16) ensures that two consecutive rotations (when $v_{it}=0$) cannot be separated by more than ε weeks, and (17) prevents rotation assignments after the last rotation is completed (when $v_{it}=1$). Although this last constraint is redundant, it strengthens the model. Constraints (18)–(21) enforce (**R.6**). Constraint (18) sets w_{ir_k} to one when student i is assigned to hospital k, (19) sets $\tilde{w}_i$ to one if two w_{ir} variables with distinct r values are set to one for student i (meaning i visited two different regions), and (20) ensures that at most 25% of AUC's students complete rotations in two different geographic regions, i.e., have $\tilde{w}_i = 1$. Note that (21) is necessary to ensure that, if a student completes their rotations in two different regions, they cannot oscillate between them, i.e. once they change to the second region, they cannot change back. Constraint (22) enforces (**R.8**): if a student is assigned to a fixed-cost hospital, they complete all rotations offered at that institution. Finally, notice that (**R.5**) is not explicitly enforced in the MILP model. Instead, once a solution is obtained, students meet with an advisor and are presented with a few schedules from which to choose. By incorporating (15)–(23) in (MP), we obtain the AUC-specific RASP formulation.

Constraints (15)–(23) highlight the challenge of using a MILP approach for this type of problem. Requirements that are easily stated can be difficult to formalize mathematically. In our case, an additional $O(I\times R+I\times T)$ variables and $O(I\times J\times T)$ constraints are added to MP. For practical-sized instances, this amounts to tens of thousands of additional variables, and hundreds of thousands of additional constraints, resulting in increased computational complexity.

4.2. A Network-Based Formulation of AUC's QoS Requirements

We now discuss how to encode the QoS requirements (**R.6**), (**R.7**), and (**R.8**) in the RASP's MDD, and consequently, in the NBF. Although the overall construction and reduction procedures, as presented in Algorithms 1 and 2, remain the same, the state information and transition functions of the DP must be augmented. That is, for every stage s, we replace the original state information pair $(\mathcal{G}_s,\tau_s)$ with $(\mathcal{G}_s,\tau_s,\eta_s,\zeta_s,\mathcal{F}_s)$, where the three new pieces of information are defined as follows:

- $\eta_s\in\mathbb{Z}_+$: the number of distinct regions to which the student has been assigned to up to stage s;
- $\zeta_s\in\{0,1,\dots,R\}$: the region to which the student is assigned in stage s;
- $\mathcal{F}_s\in[\{0\}\cup\mathcal{K}_F]^J$: an ordered tuple of length J where the number at position j, $\mathcal{F}_s(j)$, indicates whether or not rotation j was completed at a fixed-cost hospital. If $\mathcal{F}_s(j)=0$, either rotation j has not been completed by stage s or it was completed at a hospital in $\mathcal{K}\setminus\mathcal{K}_F$. Otherwise, $\mathcal{F}_s(j)$ identifies the hospital from $\mathcal{K}_F$ where rotation j was completed.

Given the augmented state space, feasible transitions in the MDD must abide by the QoS constraints.

(**R.7**): For any s, if a rotation finishes at time τ_s, the next rotation must start no later than $\tau_s+\varepsilon$.

(**R.6**): Recall that $\bar{r}$ is the maximum number of region changes a student can experience. Let $\mathbb{I}(\ell)$ be an indicator function that evaluates to 1 if condition ℓ is true and 0 otherwise. For any schedule $\mathcal{S}$ as in (1), we require that the number of region changes be no more than $\bar{r}$, i.e.,

$$\sum_{s=2}^{J}\mathbb{I}(r_{k_{s-1}}\neq r_{k_s})\le\bar{r}. \tag{24}$$

(**R.8**): Given a schedule $\mathcal{S}$, define $\mathcal{J}(\mathcal{S},k)=\{j_s\mid k_s=k\text{ for }(j_s,k_s,t_s)\in\mathcal{S},\ s=1,\dots,J\}$ to be the set of rotations performed at a hospital k. For any fixed-cost hospital $k\in\mathcal{K}_F$, a feasible $\mathcal{S}$ must be such that either all or none of the rotations in $\mathcal{J}(k)$ are performed at k, i.e.,

$$\Big(\mathcal{J}(k)=\mathcal{J}(\mathcal{S},k)\Big)\text{ or }\Big(\mathcal{J}(\mathcal{S},k)=\emptyset\Big). \tag{25}$$

We now present an updated version of Proposition 1 that recursively defines a feasible schedule using $(\mathcal{G}_s,\tau_s,\eta_s,\zeta_s,\mathcal{F}_s)$. To simplify our exposition when it comes to enforcing (**R.8**), let $\mathcal{H}(\mathcal{G}_s,\mathcal{F}_s,j)$ be the set of hospitals where a student can perform rotation j, given that rotations in $\mathcal{G}_s$ are left to be completed, and fixed-cost hospitals in $\mathcal{F}_s$ have been used. Such a set is defined as follows:

(a) If all entries in $\mathcal{F}_s$ are zero, no fixed-contract hospitals have been assigned to the partial schedule up to stage s. Thus, we have $\mathcal{H}(\mathcal{G}_s, \mathcal{F}_s, j) = \{k : j \in \mathcal{J}(k) \text{ and } (k \in \mathcal{K} \setminus \mathcal{K}_F \text{ or } k \in \mathcal{K}_F \wedge \mathcal{J}(k) \subseteq \mathcal{G}_s)\}$. That is, rotation j can either be completed at a variable-cost hospital or at a fixed-cost hospital provided, due to the *all-or-none* requirement, that all rotations offered by that institution are in the set $\mathcal{G}_s$, i.e., are left to be completed.

(b) Otherwise, we must verify whether a fixed-cost hospital that offers rotation j has been assigned to the partial schedule in a previous stage. This leads to two conditions:

(i) If there exists some rotation $j' \neq j$ such that $\mathcal{F}_s(j') = k'$ for some fixed-cost hospital $k' \in \mathcal{K}_F$ and hospital k' offers rotation j, then $\mathcal{H}(\mathcal{G}_s, \mathcal{F}_s, j) = \{k'\}$. In this case, some rotations were already completed at fixed-cost hospital k', and because $j \in \mathcal{J}(k')$, due to the *all-or-none* requirement, rotation j must also be completed at hospital k'.

(ii) Otherwise, $\mathcal{H}(\mathcal{G}_s, \mathcal{F}_s, j)$ is defined as in (a).

Proposition 5. *A schedule $\mathcal{S}$, as defined in (1), is feasible if and only if there exists a sequence of tuples $(\mathcal{G}_0, \tau_0, \eta_0, \zeta_0, \mathcal{F}_0), \ldots, (\mathcal{G}_J, \tau_J, \eta_J, \zeta_J, \mathcal{F}_J)$ with $\mathcal{G}_s \subseteq \mathcal{J}$, $\tau_s \in \mathbb{Z}_+$, $\eta_s \in \mathbb{Z}_+$, $\zeta_s \in \{0, 1, \ldots, R\}$, $\mathcal{F}_s \in [\{0\} \cup \mathcal{K}_F]^J$ for $s = 0, \ldots, J$, such that $(\mathcal{G}_0, \tau_0, \eta_0, \zeta_s, \mathcal{F}_0) = (\mathcal{J}, 0, 0, 0, (0)^J)$ and, for $s = 1, \ldots, J$,*

$$\mathcal{G}_s = \mathcal{G}_{s-1} \setminus \{j_s\},$$

$$\tau_s = t_s + \Delta_{j_s},$$

$$\eta_s = \eta_{s-1} + \mathbb{I}(r_{k_{s-1}} \neq r_{k_s}), \tag{26}$$

$$\zeta_s = r_{k_s}, \tag{27}$$

$$\mathcal{F}_s = \mathcal{F}_{s-1} + \mathbb{I}(k_s \in \mathcal{K}_F) k_s e_{j_s}, \tag{28}$$

$$(j_s, k_s, t_s) \in \tilde{\mathcal{A}}(\mathcal{G}_{s-1}, \tau_{s-1}, \eta_{s-1}, \zeta_{s-1}, \mathcal{F}_{s-1}), \tag{29}$$

where e_j is a vector of length J with value 1 at component j and zeros otherwise, and

$$\tilde{\mathcal{A}}(\mathcal{G}_s, \tau_s, \eta_s, \zeta_s, \mathcal{F}_s) = \Big\{(j, k, t) : j \in \mathcal{G}_s,\ k \in \mathcal{H}(\mathcal{G}_s, \mathcal{F}_s, j),\ \tau_s \le t \le \tau_s + \varepsilon,\ \eta_s + \mathbb{I}(\zeta_s \neq r_k) \le \bar{r} + 1\Big\}.$$

Proposition 5 demonstrates that by using the augmented tuple $(\mathcal{G}, \tau, \eta, \zeta, \mathcal{F})$ and the valid transition function $\tilde{\mathcal{A}}(\cdot)$ in the MDD compilation procedure, requirements (**R.6**), (**R.7**), and (**R.8**) can be incorporated directly into the graphical structure D. However, it has more general implications. Specifically, it suggests that modelers of large-scale problems have additional flexibility with regards to how complex requirements may be represented in their solution approach. Whereas some restrictions may be better expressed using a math programming representation, others can be incorporated into a decision diagram via dynamic programming. Thus, the NBF approach allows the modeler to examine how best to express problem requirements, i.e., by experimenting between the MDD and the network-based MILP representation and studying the effect on computational performance. Another important result is that, even with the additional AUC constraints, the MDD grows sub-exponentially in all parameters except for J, as shown in the following proposition.

Proposition 6. *The size of the MDD increases exponentially only in the number of rotations J.*

Proposition 6 implies that our approach can accommodate large numbers of students, time periods, and hospitals. Specifically, if the number of rotations is fixed, the size of the network that underlies the MDD is pseudo-polynomial with respect to the input, as it depends on the length of the planning horizon, but is polynomial in all the remaining parameters. For this problem setting, the result is especially advantageous as the number of core rotations is typically small ($J = 5$). They are standardized by the Council for Graduate Medical Education (2017)—an independent organization that sets the standards for the delivery of medical education in the US—and are unlikely to change often. Other problem parameters, however, may fluctuate significantly from year to year.

The efficiency of the solution procedure also has implications with regards to students who fail their licensing exam and whose sequence of rotations must be rescheduled. Such students require schedules that begin after the date of their next exam. The next proposition demonstrates that the MDD network can efficiently identify an optimal set of alternative schedules with the lowest cost.

Proposition 7. *Under the assumptions of Proposition 2, the b least-cost schedules can be obtained in polynomial time in the size of D for any integer $b > 0$.*

This result implies that, although we assume a deterministic formulation for the problem, our solution methodology can efficiently account for the stochastic features of the real system.

5. Computational Experiments: An AUC Case Study

In this section, we present a comprehensive numerical study of the RASP. We first provide a performance analysis to demonstrate that the solution times, number of instances solved, and problem sizes that can be solved to optimality using the NBF approach (Sections 3.3 and 3.5) are better than the MILP model (Section 3.1). We then perform a counterfactual study, parametrized by six years' worth of student and administrative data, that compares AUC's historical scheduling decisions to what would have happened had the NBF framework been implemented. To understand how the optimal assignment of students to rotations can affect future costs, we develop a discrete-event simulation (DES) of the licensing exam and clerkship scheduling process. We use the DES to compare the long-run costs of our solution technique to a greedy heuristic. Finally, we perform a sensitivity analysis to determine how future costs may be affected by estimation uncertainty (i.e., the licensing exam failure rate) and forecast uncertainty (i.e., the future size of an AUC cohort).

For the counterfactual analysis and the DES, we compare the NBF framework to a greedy benchmark. The greedy algorithm first orders students in decreasing order of t_i. Then, one student a time, it sequentially schedules them to a series of rotations with the minimum cost. Notice that the algorithm first schedules the student with the latest start week and progressively work backwards until it gets to the current week; this reduces the chance of generating an infeasible schedule. Interviews with staff, along with a retrospective comparison of schedules (see Section 5.2), indicate that the greedy algorithm is the most coherent way to operationalize AUC's current scheduling practices. This is because scheduling decisions at AUC are made in an ad-hoc manner where staff do not attempt to solely minimize cost. Instead, staff use their experience to assign each student to a schedule (as opposed to assigning a schedule to a student) and periodically consider idiosyncratic student preferences when making assignment decision. Nevertheless, whenever cost considerations are taken into account, placements are assigned greedily. Additionally, we did not design new heuristic approaches to use as benchmarks since they would return suboptimal solutions and the NBF approach is exact.

The MILP and NBF were solved using Gurobi 7.5.2 (Gurobi Optimization, 2016). The discrete-event simulations were developed in Java. Simulation experiments were performed on an Intel® Core™ i7-3770 with a 3.40GHz processor and 16 GB of RAM. For each simulation, we employed the independent-replications method with a burn-in of 50 years and trial length of 100 years.

5.1. Performance Analysis

We first compare solution times and the number of instances solved between the MILP representation of AUC's RASP and the NBF approach over a wide range of problem parameters. For a planning horizon of $T = 52$ weeks, 3,240 test instances were generated by varying the cohort size $(10, 25, 50, 100, 200, 300)$, the number of hospitals $(5, 10, 20, 30)$, the proportion of hospitals located in unique regions $(20\%, 40\%, 60\%)$, the percentage of fixed- versus variable-cost hospitals $(25\%, 50\%, 75\%)$, the number of rotations $(3, 5)$, the rotation lengths in weeks (3,5,7), and the maximum idle time between rotations in weeks (3,5,7). Detailed results that carefully outline the parameters selected for each experiment are presented in Appendix A. Figure 2 summarizes the data from all of the experiments and gives an overview of its implications. Specifically, the NBF approach solves 2,763 (85%) of the test instances to optimality in under 30 minutes, while only 109 (4%) are solved by the MILP model. Note that all optimal MILP solutions match with the optimal NBF solutions. Further, of the problems solved by the NBF, 90% can be completed in under 5 minutes. These results suggest that for any choice of input parameters, the NBF approach significantly outperforms the MILP model by efficiently solving more problem instances in less time.

To further illustrate the differences in performance between the time-indexed MILP model and the NBF approach, we consider a collection of experiments with a planning horizon of $T = 52$ weeks.

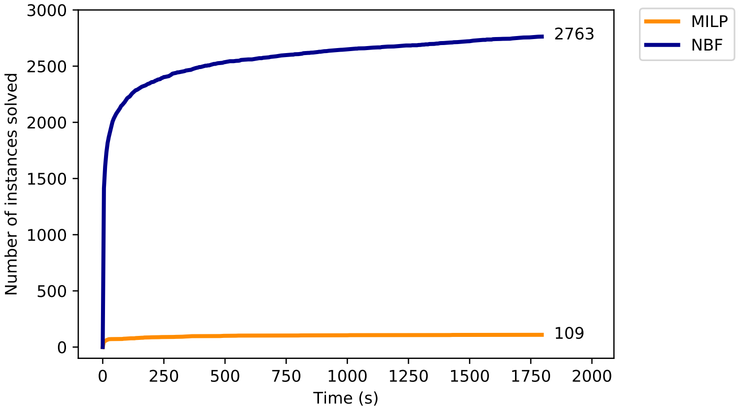
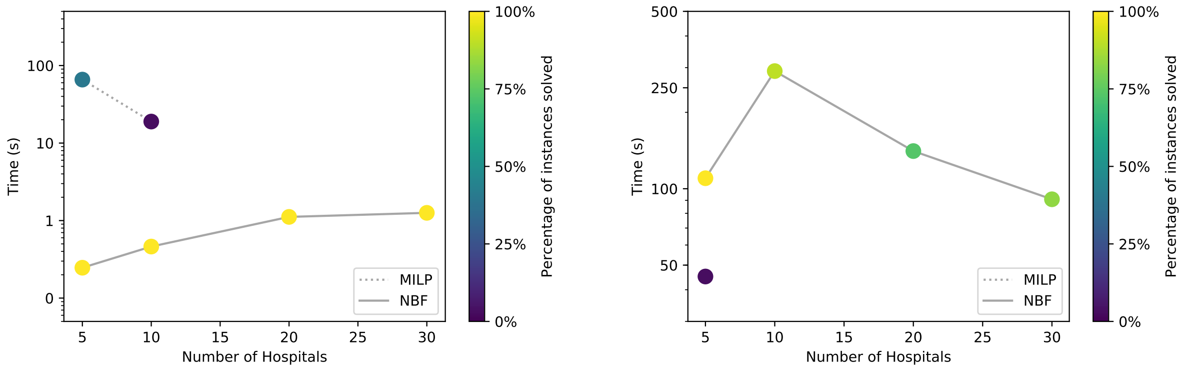


Figure 2 Number of problems solved to optimality in under 30 minutes (out of 3,240). Test instances were generated by setting the planning horizon to $T = 52$ weeks and then varying the cohort size $(10, 25, 50, 100, 200, 300)$, the number of hospitals $(5, 10, 20, 30)$, the proportion of hospitals located in unique regions $(20\%, 40\%, 60\%)$, the percentage of fixed- versus variable-cost hospitals $(25\%, 50\%, 75\%)$, the number of rotations $(3, 5)$, the rotation lengths in weeks (3,5,7), and the maximum idle time between rotations in weeks (3,5,7).

We hold fixed the proportion of hospitals located in unique regions, the percentage of fixed- versus variable-cost hospitals, and the maximum idle time between rotations at 60%, 50%, and 5 weeks, respectively. In the first set of experiments (Figure 3) we vary the number of hospitals $(5, 10, 20, 30)$ while holding the number of rotations fixed ($J = 3$ in the left plot and $J = 5$ in the right). The shade of each data point represents the proportion of problems solved from thirty randomly generated instances; five randomly sampled rotation lengths $(3, 5, 7)$ for each of the six different cohort sizes $(10, 25, 50, 100, 200, 300)$. For $J = 3$, the NBF solves all problem instances to optimality within 30 minutes regardless of the number of available hospitals. The MILP model, on the other hand, does not solve a single instance when there are more than 10 hospitals. Further, when there are fewer than 10 hospitals, the solution speed of the NBF is orders of magnitude smaller than the MILP. For $J = 5$, the NBF once again exhibits much better performance. Whereas the MILP model can only solve a single instance with five hospitals, the NBF approach solves all problems instances. Further, when there are thirty hospitals, upwards of 60% of all randomly generated instances can be solved to optimality in under 2 minutes. Since the number of hospitals affiliated with AUC is 25, these results suggest that the NBF is more appropriate for realistically-sized instances.

In the second set of experiments (Figure 4), we vary the cohort size $(10, 25, 50, 100, 200, 300)$ while holding the number of rotations fixed ($J = 3$ in the left plot and $J = 5$ in the right). The shade of each data point now represents the number of problems solved from 20 randomly generated instances; five randomly sampled rotation lengths $(3, 5, 7)$ for each of the four values associated with the number of available hospitals $(5, 10, 20, 30)$. We observe similar performance as in Figure 3 in that for $J = 3$,



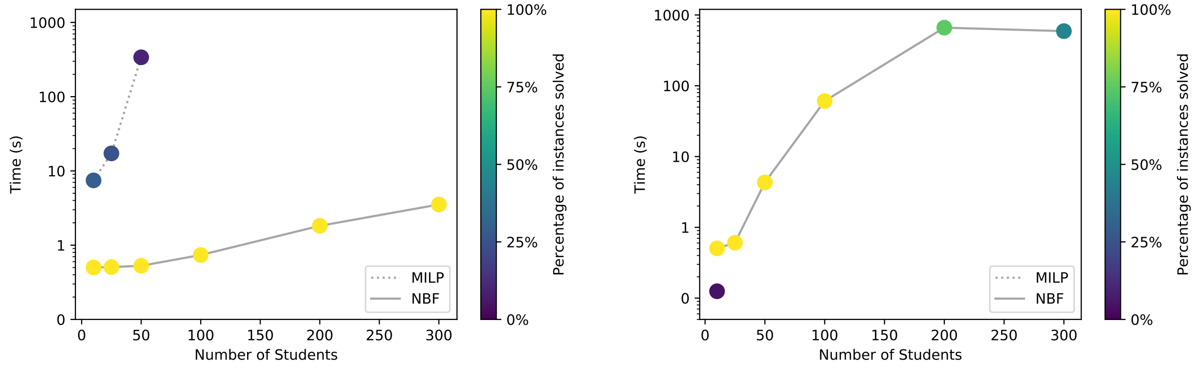
(a) J=3 (b) J=5

Figure 3 **Time (seconds) and the percentage of instances solved as a function of the number of hospitals. The proportion of hospitals located in unique regions, the percentage of fixed- versus variable-cost hospitals, and the maximum idle time between rotations are held fixed at 60%, 50%, and 5 weeks, respectively. The shade of each data point represents the number of solved instances from five randomly sampled rotation lengths $(3,5,7)$ for each cohort size $(10,25,50,100,200,300)$.**

the NBF solves all problem instances to optimality within 30 minutes. The MILP model does not solve all instances even in the case of fewer than 50 students per cohort. For $J = 5$, the MILP model can only solve one problem instance with 10 students. In contrast, the NBF solves to optimality all instances with up to 100 students, and upwards of 50% of the instances when there are 200 or more students in a cohort. This further demonstrates that the NBF is a more robust alternative for AUC, as the cohort size over the last several years has consistently exceeded 300 students.

In the final set of experiments, we evaluate the impact of side constraints on MDD size. In figure 5, we plot the number of nodes of the non-reduced MDD (x-axis) versus the number of nodes of the reduced MDD (y-axis) for all $3,240$ test instances. On average, the reduced MDD is typically half the size of the non-reduced MDD (6,715 vs. 3,691 nodes). The relative averages are also consistent as a function of the number of rotations. For example, for $J = 3, 4$, and 5, there are approximately 1.82 times the number of nodes in the non-reduced MDD as compared to the reduced MDD. Finally, while the inclusion of QoS constraints leads to an MDD with a theoretical bound that is significantly larger than the original MDD, we find that the actual difference varies considerably. In fact, the addition of QoS constraints often results in substantial reductions in the number of nodes in D.

The performance analysis shows the effectiveness of the NBF approach at solving realistically-sized problem instances. As the number of hospitals and the size of a cohort increase, so do the number of QoS requirements that ensure students have an excellent clerkship experience. Whereas any MILP model would naturally get larger to include more students (i.e., decision variables) and extra constraints for the additional hospitals and QoS requirements, the graphical structure associated



(a) J=3 (b) J=5

Figure 4 Time (seconds) and the percentage of instances solved as a function of the cohort size. The proportion of hospitals located in unique regions, the percentage of fixed- versus variable-cost hospitals, and the maximum idle time between rotations are held fixed at 60%, 50%, and 5 weeks, respectively. The shade of each data point represents the number of solved instances from five randomly sampled rotation lengths $(3,5,7)$ for for each number of hospitals $(5,10,20,30)$.

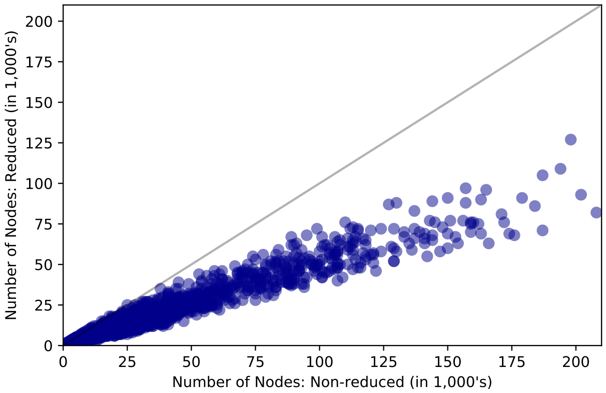


Figure 5 The total number of nodes between reduced and non-reduced MDDs for all instances.

with the NBF approach may, in some cases, shrink. This implies that as long as new constraints do not add to the MILP reformulation (i.e., Section 3.5), AUC may see computational speedups for future cohorts as the scheduling requirements get more complex.

5.2. Counterfactual Study: What could have been?

In this section, we compare AUC's historical scheduling decisions to what would have happened had our framework been implemented. We collected data on 1,962 medical students who studied at AUC between August 2012 and January 2018. Each entry in the data set included a unique (masked) student identifier, the rotation, the name of the hospital where the rotation was completed, and the start and end dates of the rotation. We also obtained a set of administrative documents outlining the contract details for each of the 25 affiliated hospitals. For each hospital, we examined the contract to determine the type of rotations offered, the fixed (c_k^F) and variable costs (c_{jk}) associated with training, the number of students that started a rotation on a given date (C_{jkt}), and the hospital's address (r_k). Hospitals within a 50 mile radius of each other were classified as being in the same geographic region. In accordance with the Council for Graduate Medical Education (2017), each student completed $J = 5$ clinical rotations: internal medicine, general surgery, pediatrics, obstetrics & gynecology, and psychiatry. The first two rotations were twelve weeks in length, while the latter three were six weeks. We note that, for privacy reasons, neither data set indicated whether a student failed their licensing exam nor how many times the exam was written.

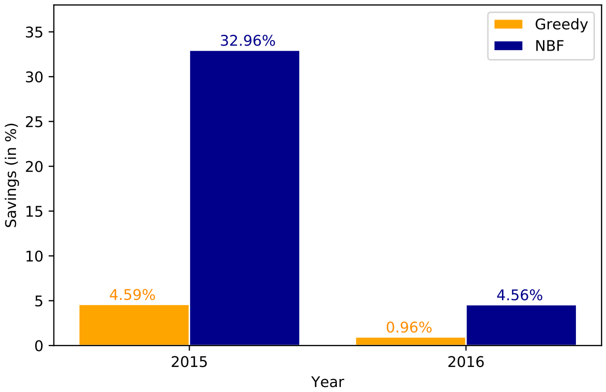


Figure 6 Comparison against historical placement costs (current practice) as a function of time.

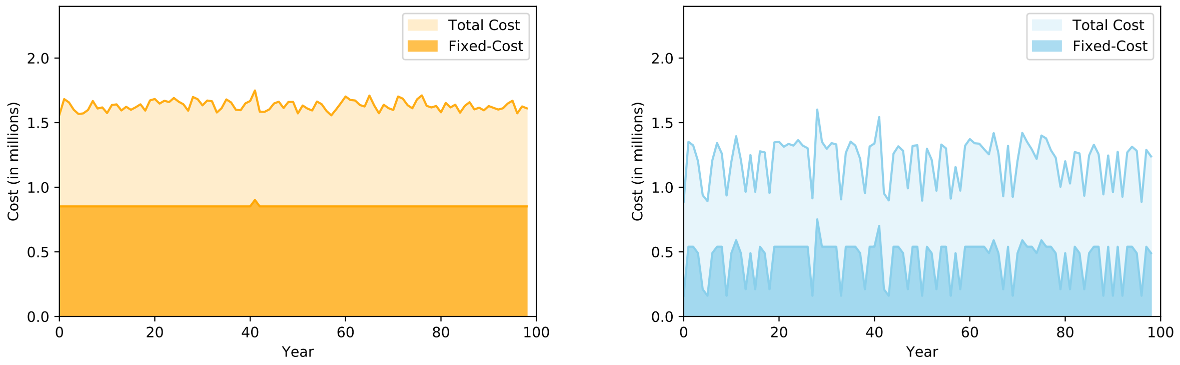
We perform a counterfactual study by comparing the yearly placement costs of three scheduling approaches using past capacity and student availability data. The first approach (current practice) represents what actually transpired. The second (Greedy) approach schedules students to rotations according to the greedy algorithm. The third approach uses the NBF technique. For each scheduling model starting with the September 2014 cohort, students are assigned to rotations after passing their licensing exam. For the Greedy and NBF approaches, the placement costs of years 2015 and 2016 are obtained and compared to current practice (see Figure 6). Two important insights emerge. First, substantial savings, approximately 19% per year on average, could have been realized using the NBF technique. Although some savings are due to the better use of capacity, the majority of the cost reduction is associated with limiting the number of fixed-cost hospitals under contract. For example, in 2015, NBF needed to negotiate far fewer contracts with fixed-contract hospitals as compared to current practice. Second, we observe that the greedy approach bounds AUC's placement cost from below. Although, as can be expected, the greedy approach does not fully capture the idiosyncrasies of current scheduling practices, the result suggests that it is an appropriate benchmark.

5.3. Simulation Results: Evaluation of Future Costs

Using results from a DES, we now provide some insight as to the cost savings AUC can expect in the future. We use a discrete uniform distribution with support [300, 350] to model the number of students in a cohort. Although we do not have enough data to find the true cohort size distribution, discussions with staff indicate that this distribution and range is reasonable. We note that the mean cohort size estimated from the data set is 327. Given the realization for the number of students in a cohort for a given year, the empirical distribution models the week each student receives their exam results. Since a student may pass or fail the exam, the probability they begin clerkship is geometrically distributed with a success probability of 96% (one minus the failure rate). This is the average pass rate of first-time test takers for 2016. To ensure that AUC's QoS constraints are satisfied (see Section 4), a student that begins clerkship cannot be idle for more than eight weeks ($\varepsilon = 8$). Further, at most 25% of students in a cohort can be assigned to two geographic regions ($\bar{r} = 1$) while all other students must complete in-hospital training in a single geographic region.

Since clerkship typically comprises the last two years of a four-year degree in medicine, we assume a rolling planning horizon of $T = 104$ weeks (i.e., two years with 52 weeks per year). If the week represents the beginning of a new school year, the size and exam weeks of the cohort are realized. Then, the scheduling policy (either greedy or the NBF) assigns students a schedule. On all other weeks, we realize which students can begin in-hospital training (by following their previously scheduled sequence of rotations) and which students have failed their exam. For those that have failed, we re-assign them to a new schedule either using the greedy approach or by following Proposition 7. We also continuously track and update the capacity levels of each rotation as the current cohort affects the number of rotation slots available for next year's cohort. We note that in the DES, no time limit was imposed on the NBF although all instances were solved to optimality within 2 hours.

We find that, on average, AUC can expect around a 25% reduction in cost using the NBF approach. The translates into a real-dollar cost reduction of several hundred thousand dollars, a managerially significant quantity that validates the pursuit of a methodology that finds optimal assignment policies. Figure 7 presents a time-series plot of costs per year. In the left figure, costs associated with the greedy approach are presented, and in the right figure, the costs accrued by the NBF are given. Notice that the greedy approach is more consistent than the NBF as the year-over-year costs are smaller. However, the NBF always assigns students to schedules with a smaller placement cost, and in some cases, finds assignments that are half as costly. This is because the NBF consistently opens fewer fixed-cost hospitals as compared to the greedy heuristic. This intuition is confirmed by examining the proportion of the total cost that can be attributed to fixed-cost hospitals in Figure 7. We also observe that the greedy approach tends to concentrate the use of fixed-cost hospitals early in the planning horizon, whereas the NBF technique tends to use these hospitals more evenly over time.



(a) Greedy Heuristic (b) NBF

Figure 7 Placement cost per year (in millions) as a function of time (year) for a single trial.

5.4. Simulation Results: Estimation and Forecast Uncertainty

Our results thus far assume that the licensing exam failure rate at AUC is equal to the national average and that the average number of students in a cohort is fixed as compared to the number of affiliated hospitals. However, the actual failure may by higher (or lower) and the ratio between the number of students to the number of available rotation slots may not be held fixed. Thus, we now investigate how sensitive our results are to mis-specifications in the underlying stochastic quantities. This gives AUC further insight into the value of a more systematic approach to student scheduling and also validates the choice of a deterministic model (as opposed to a stochastic program, for instance).

In Figure 8, we examine how the failure rate affects costs. As expected, we find that as the failure rate increases, the placement cost AUC incurs also increases. This is independent of the scheduling approach and highlights the fact that as uncertainty plays a larger role in the scheduling environment, including stochasticity in any model becomes compulsory. Nonetheless, since the failure rate at AUC has been approximately equal to the national average over the last several years (AUC Medical School, 2018a), our approach is reasonable. We also find that the NBF reduces placement costs by approximately 20%-25% as compared to current practice, and is robust with respect to a wide range of failure rate values (see Figure 8). Thus, the NBF approach yields a substantial reduction in placement costs and, as AUC acts to increase the licensing exam pass rate (AUC Medical School, 2018b), the resulting cost savings will approach the true minimum.

In Figure 9, we study how the size of the cohort, given that the number of available rotation slots are held fixed, affects placement costs. For both approaches, we find that costs increase as a function of the cohort size, as expected. However, we also find that the NBF is clearly superior when the supply of rotation slots does not substantially exceed demand. More specifically, for small cohorts, any scheduling approach is sufficient; there are few students and many feasible schedules to choose from. In the middle range where the cohort size is between 150 and 250 (more realistic) and the

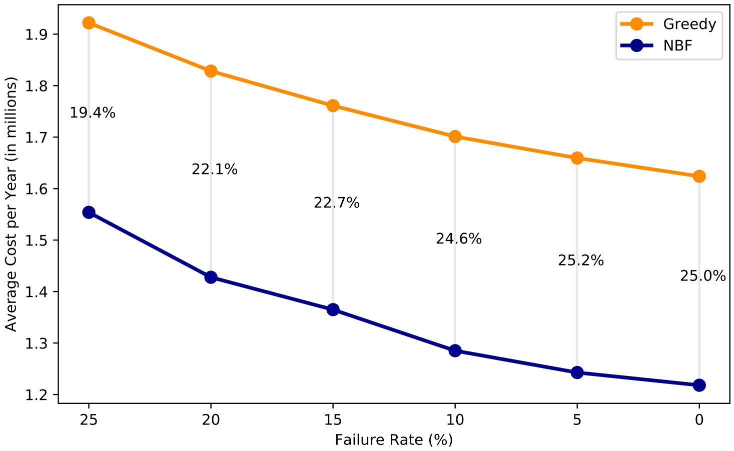


Figure 8 Average placement cost per year (in millions) as a function of the per student failure rate. Note that the probability a student can begin clerkship is geometrically distributed with success probability equal to one minus the failure rate. The average values over 100 trials are presented.

supply of rotation slots is still ample, the difference between the NBF and the greedy approach is the largest. In this region, there are many feasible schedules although choosing the optimal one is computationally difficult. This is why the NBF yields such large cost improvements as compared to the greedy approach. AUC typically operates in the upper range, where demand for rotation slots is large and supply is limited. In this region, the relative cost savings are reduced (as compared to the middle range) as the ratio of demand to supply approaches one. Nevertheless, the absolute cost savings in this region are the most substantial due to the large number of students that must be placed. Further, because AUC typically negotiates fixed contracts with hospitals before assigning students to rotations, supply will never far outpace demand. Since it is likely that AUC will continue to operate in this region for the foreseeable future, results such as these are encouraging.

6. Managerial Insights

In addition to its computational benefits, the underlying graphical structure in our NBF approach has useful managerial implications. For example, by creating a pattern store (a collection of feasible paths in D), administrators can ensure that specific schedules are either not assigned, or are more likely to be assigned to students in future cohorts. These individual patterns can be inserted or removed from the network using traditional graphical separation techniques. We refer to Cire and Hooker (2014) and Bergman et al. (2016) for MDD methodologies to include and remove paths, alongside their associated computational complexities.

This flexibility allows the medical school to utilize feedback solicited from students who have finished clerkship. Negative experiences could be directly incorporated into the assignment process,

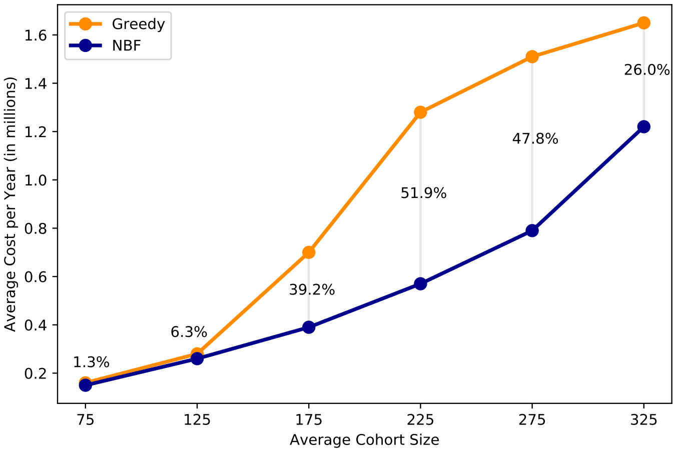


Figure 9 **Average placement cost per year (in millions) as a function of the mean number of students in a cohort. Note that the probability a student can begin clerkship on a given week is sampled from the empirical distribution. The average values over 100 trials are presented.**

i.e., particularly problematic schedules would be eliminated by removing the corresponding paths in the MDD. Systematically representing these idiosyncratic restrictions as constraints in a MILP model would be more cumbersome to implement and difficult to maintain. Moreover, while, in theory, it is possible to answer some of these questions with an MILP model, we did not observe its performance to be robust to larger problem sizes. Specifically, while the number of hospitals and students of a real-life problem instance are around 25 and 200-300, respectively, the current state-of-the-art MILP model cannot solve instances with more than 10 hospitals and 50 students.

Our approach also allows a medical school to conduct analysis for capacity planning, demand management, and contract negotiations. For example, the maximum number of students that can be assigned to any hospital can be obtained by replacing the objective function of the NBF to maximize flow (instead of minimize cost). Note that for the classical time-indexed MILP, this would require adding extra students until the model eventually became infeasible. Further, by adjusting the cost of the arcs in the decision diagram, a medical school can determine a threshold beyond which it would be too costly to assign students to any affiliated hospital. These managerial questions are of interest to any medical school in the planning stages, particularly when determining how many students should receive an offer of admission. Although additional MILP constructs could be incorporated to address these concerns, these insights are already obtainable as a direct consequence of the NBF.

It is this versatility that AUC finds compelling about the network-based model. Interviews with scheduling staff at AUC have indicated that our approach can be used to answer other managerially relevant questions, as well as to analyze certain what-if scenarios. Consider the following examples:

1. *What is the maximum number of slots per rotation that AUC can effectively use from a hospital?* By maximizing the flow over the arcs associated with that hospital in the decision diagram, and setting the capacity of the remaining arcs and hospitals to infinity, the number of slots can be obtained. Such a solution would also account for the underlying complex combinatorial structure of the problem imposed by the operational requirements. For example, suppose that a hospital only offers a single rotation j, but there exists some rotation $j' \neq j$ that is only available at a fixed-cost hospital which also offers rotation j. Because of the *all-or-none* requirement, the number of slots requested from the first hospital should effectively be zero.

2. *What if a certain hospital were to dramatically increase (or decrease) the number of slots available to a cohort?* This can be answered by adjusting the arc and hospital capacities accordingly. For a bound on the marginal cost associated with increasing or decreasing capacities, it is also possible to use the reduced cost of the arc variables in the NBF. Such bounds will be more accurate than the time-indexed MILP formulation, since the network flow model represents an ideal formulation of the set of student schedules (i.e., without operational requirements, the extreme points of the network flow model are integral and represent a feasible student schedule).

3. *How different are schedules that move students through clerkship the quickest versus those that keep them within a small geographic region?* If particular arcs corresponding to schedules completed at the end of the planning horizon are removed from the decision diagram, it forces students to complete clerkship earlier. The decision diagram can then be modified to only allow transitions between regions that are geographically close.

4. *How many fixed-cost hospitals should AUC secure contracts with?*

   In general, guaranteeing rotation slots far in advance requires significant investment as expensive contracts with fixed-cost hospitals must be signed. By taking better advantage of the capacity available at variable-cost hospitals, which offer a smaller cost per placement, AUC can reduce their overall cost. Further, if too much capacity is concentrated in a small number of fixed-cost hospitals, a large proportion of students must have their schedules restricted to the dates offered by those hospitals, limiting the universe of feasible schedules that can be created.

   AUC values their current relationships with fixed-cost hospitals because they can be trusted to provide a steady number of rotation slots every year (thereby reducing uncertainty). In contrast, capacity at variable-cost hospitals is more volatile. Therefore, it is not clear whether AUC has any interest in reducing the number of contracts they secure. Nevertheless, our model can be used by AUC, and any medical school for that matter, to better negotiate contracts by means of what-if scenarios. The model can be run several times, with different inputs for hospital capacities, costs, etc., and the resulting outputs would form the basis of a contract negotiation strategy in which the school argues for specific capacity levels, price adjustments, and so on.

Many other questions can be similarly answered by running and re-running our model after fixing certain variables (or arcs of the decision diagram) to specific values. In fact, Alejandro Carrasco, Senior Director of Business Operations, remarked that "being able to quickly answer these types of questions would significantly improve our operations, allowing us to deliver better quality schedules for our students, and with much less manual effort."

7. Concluding Remarks

We investigate the scheduling practices of a medical school that must assign a cohort of medical students, who become eligible over a period of several months, to a series of clinical rotations at hospitals in different geographic regions. The student schedules must obey certain core restrictions, licensing requirements, and quality-of-service constraints, while keeping the school's placement costs to a minimum. To this end, we propose a network-based formulation (NBF) and show that it is significantly superior to a mixed-integer linear programming (MILP) model. Using a data set collected from the American University of the Caribbean (AUC) we perform a counterfactual analysis and develop a discrete-event simulation of the licensing exam and clerkship scheduling process. We find that AUC could have saved approximately 19% per year on average in student placement costs had our scheduling approach been implemented. Using a DES, we then compare our exact scheduling approach with current practice and demonstrate that in the future, AUC can expect an average cost reduction of 25%. We also show that this cost reduction is robust to mis-specifications in the licensing exam failure rate and the size of a cohort when the number of affiliated hospitals is held constant.

Our NBF is created by using decision diagrams to reformulate a particularly large optimization problem as a network model with side constraints. We use dynamic programming, and its ability to reduce a complex problem to a series of simpler subproblems, together with MILP modeling, which more easily represents complicated relationships between decision variables and best communicates various problem restrictions. By carefully choosing which constraints to incorporate into the decision diagram versus which to represent in the MILP model, one can obtain significant computational speed-up and exact solutions to large-scale problems that may be difficult to implement and fine tune using other methods such as branch-and-price. Future work will aim to provide more guidance as to what classes of constraints are better represented in the state-transition graph versus those that are better suited for a mathematical programming representation.

Acknowledgments

The first two authors were supported by a Discovery Grant provided by the Natural Sciences and Engineering Research Council of Canada (NSERC). The authors are grateful to the department editor, the associate editor, and the three anonymous referees, whose comments and suggestions have considerably improved the study. We would also like to sincerely thank the staff from the School of Medicine at the American University of the Caribbean for the support and time dedicated to this project.

References

Agarwal, A. 2016. Balancing medical resident education and workload while ensuring quality patient care. Ph.D. thesis, Rochester Institute of Technology.

Ahuja, Ravindra K, Thomas L Magnanti, James B Orlin. 1993. *Network flows: theory, algorithms, and applications*. Prentice Hall.

Artigues, Christian. 2017. On the strength of time-indexed formulations for the resource-constrained project scheduling problem. *Operations Research Letters* **45**(2) 154 – 159. doi:https://doi.org/10.1016/j.orl.2017.02.001. URL http://www.sciencedirect.com/science/article/pii/S016763771730072X.

Association of American Medical Colleges. 2016. Total enrolment by US medical school. URL https://www.aamc.org/download/321526/data/factstableb1-2.pdf,https://www.aamc.org/download/321532/data/factstableb2-2.pdf. Accessed October 25, 2017.

AUC. 2018. American university of the carribean medical school. URL https://www.aucmed.edu. Accessed March 29, 2018.

AUC Medical School. 2018a. Facts & figures about AUC's medical school. URL https://www.aucmed.edu/about/facts-and-figures.html. Accessed February 18, 2018.

AUC Medical School. 2018b. Facts & figures about AUC's medical school. URL https://www.aucmed.edu/academics/usmle-preparation.html. Accessed February 18, 2018.

Baptiste, Philippe, Claude Le Pape, Wim Nuijten. 2001. *Constraint-Based Scheduling*. Kluwer Academic Publishers.

Bard, Jonathan F, Yufen Shao, Xiangtong Qi, Ahmad I Jarrah. 2014. The traveling therapist scheduling problem. *IIE Transactions* **46**(7) 683–706.

Bard, Jonathan F., Zhichao Shu, Douglas J. Morrice, Luci K. Leykum, Ramin Poursani. 2016. Annual block scheduling for family medicine residency programs with continuity clinic considerations. *IIE Transactions* **48**(9) 797–811.

Barnhart, Cynthia, Ellis L. Johnson, George L. Nemhauser, Martin W. P. Savelsbergh, Pamela H. Vance. 1998. Branch-and-price: Column generation for solving huge integer programs. *Operations Research* **46**(3) 316–329.

Beasley, J. E., B. Cao. 1998. A dynamic programming based algorithm for the crew scheduling problem. *Computers and Operations Research* **25**(7-8) 567–582.

Bergman, D., A. A. Cire, W.-J. van Hoeve, J. N. Hooker. 2016. *Decision Diagrams for Optimization*. Springer.

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

Cappanera, Paola, Maria Grazia Scutellà. 2014. Joint assignment, scheduling, and routing models to home care optimization: A pattern-based approach. *Transportation Science* **49**(4) 830–852.

Cheang, B., H. Li, A. Lim, B. Rodrigues. 2003. Nurse rostering problems—a bibliographic survey. *EUR J OPER RES* **151**(3) 447–460.

Cire, A. A., J.N. Hooker. 2014. The separation problem on binary decision diagrams. *Proceedings of the International Symposium on Artificial Intelligence and Mathematics (ISAIM)*.

Clautiaux, François, Cláudio Alves, José Valério de Carvalho, Jürgen Rietz. 2011. New stabilization procedures for the cutting stock problem. *Journal on Computing* **23**(4) 530–545.

Cohn, A. M., S. Root, C. Kymissis, J. Esses, N. Westmoreland. 2009. Scheduling medical residents at Boston University school of medicine. *Interfaces* **39**(3) 186–195.

Conforti, Michele, Gérard Cornuéjols, Giacomo Zambelli. 2010. Extended formulations in combinatorial optimization. *4OR* **8**(1) 1–48.

Côté, Marie-Claude, Bernard Gendron, Louis-Martin Rousseau. 2011. Grammar-based integer programming models for multiactivity shift scheduling. *Management Science* **57**(1) 151–163.

Council for Graduate Medical Education. 2017. Program requirements. URL http://www.acgme.org/What-We-Do/Accreditation/Common-Program-Requirements. Accessed June 7, 2017.

Eppen, Gary D., R. Kipp Martin. 1987. Solving multi-item capacitated lot-sizing problems using variable redefinition. *Operations Research* **35**(6) 832–848.

Ernst, Andreas T, Houyuan Jiang, Mohan Krishnamoorthy, David Sier. 2004. Staff scheduling and rostering: A review of applications, methods and models. *EUR J OPER RES* **153**(1) 3–27.

Federation of State Medical Boards and National Board of Medical Examiners. 2018. The united states medical licensing examination. URL http://www.usmle.org. Accessed March 29, 2018.

Garey, Michael R., David S. Johnson. 1990. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA.

Garey, Michael R, David S Johnson, Ravi Sethi. 1976. The complexity of flowshop and jobshop scheduling. *Mathematics of operations research* **1**(2) 117–129.

Guo, J., D. R. Morrison, S. H. Jacobson, J. A. Jokela. 2014. Complexity results for the basic residency scheduling problem. *Journal of Scheduling* **17**(3) 211–223.

Gurobi Optimization, Inc. 2016. Gurobi optimizer reference manual. URL http://www.gurobi.com.

Hall, R., ed. 2012. *Handbook of Healthcare System Scheduling*, *International Series in Operations Research & Management Science*, vol. 168. Springer US.

Hooker, John N. 2013. Decision diagrams and dynamic programming. Carla Gomes, Meinolf Sellmann, eds., *Proceedings of CPAIOR 2013*, vol. 7874. Springer Berlin Heidelberg, 94–110.

Joncour, Cédric, Sophie Michel, Ruslan Sadykov, Dmitry Sverdlov, François Vanderbeck. 2010. Column Generation based Primal Heuristics. *International Symposium on Combinatorial Optimization (ISCO)*, *Electronic Notes in Discrete Mathematics*, vol. 36. 695–702.

Lancia, G., P. Serafini. 2018. *Compact Extended Linear Programming Models*. Springer.

Lemay, Brian, Amy Cohn, Marina Epelman, Stephen Gorga. 2017. New methods for resolving conflicting requests with examples from medical residency scheduling. *Production and Operations Management* **26**(9) 1778–1793.

Ludmerer, Kenneth M. 2009. Resident burnout: working hours or working conditions? *Journal of graduate medical education* **1**(2) 169–171.

Nickel, Stefan, Michael Schröder, Jörg Steeg. 2012. Mid-term and short-term planning support for home health care services. *EUR J OPER RES* **219**(3) 574–587.

Pinedo, Michael L. 2008. *Scheduling: Theory, Algorithms, and Systems*. 3rd ed. Springer.

Raffensperger, J. F. 1999. The marriage of dynamic programming and integer programming. *Proceedings of the 34th Annual Conference of the Operational Research Society of New Zealand (ORSNZ)*. 49–58.

Restrepo, María I, Leonardo Lozano, Andrés L Medaglia. 2012. Constrained network-based column generation for the multi-activity shift scheduling problem. *International Journal of Production Economics* **140**(1) 466–472.

Smalley, H. K., P. Keskinocak. 2016. Automated medical resident rotation and shift scheduling to ensure quality resident education and patient care. *Health Care Management Science* **19**(1) 66–88.

Smet, Pieter, Peter Brucker, Patrick De Causmaecker, Greet Vanden Berghe. 2016. Polynomially solvable personnel rostering problems. *EUR J OPER RES* **249**(1) 67–75.

U.S. Medical Licensing Examination. 2015. Performance data. URL http://www.usmle.org/performance-data/default.aspx. Accessed August 16, 2016.

Van den Akker, J.M., Cor A.J. Hurkens, Martin W.P. Savelsbergh. 2000. Time-indexed formulations for machine scheduling problems: Column generation. *Journal on Computing* **12**(2) 111–124.

Van den Bergh, Jorne, Jeroen Beliën, Philippe De Bruecker, Erik Demeulemeester, Liesje De Boeck. 2013. Personnel scheduling: A literature review. *EUR J OPER RES* **226**(3) 367–385.

Vanderbeck, François. 2011. Branching in branch-and-price: a generic scheme. *Mathematical Programming* **130**(2) 249–294.

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

Wegener, I. 2000. *Branching Programs and Binary Decision Diagrams: Theory and Applications*. SIAM monographs on discrete mathematics and applications, Society for Industrial and Applied Mathematics.

Wimmers, Paul F., Ted A.W. Splinter, Gregory R. Hancock, Henk G. Schmidt. 2007. Clinical competence: General ability or case-specific? *Advances in health sciences education* **12**(3) 299–314.

Zhang, H., L. Jackson, A. Tako, J. Liu, S. Dunnett. 2016. Network based formulations for roster scheduling problems. Edmund K. Burke, Luca Di Gaspero, Ender Özcan, Barry McCollum, Andrea Schaerf, eds., *Proceedings of the 11th International Conference on the Practice and Theory of Automated Timetabling (PATAT)*. 403–419.

Appendix A: Performance Analysis Results

Test instances were generated by setting the planning horizon to $T = 52$ weeks and then varying the number of hospitals $(5, 10, 20, 30)$ and the cohort size $(10, 25, 50, 100, 200, 300)$ for $J = 3$ and $J = 5$. Table 1 reports the average time in seconds (Time) and number of unsolved test instances (Instances), i.e., they failed to solve to optimality in 30 minutes or reached a memory limit, for the MILP model and NBF approach. Each entry represents 135 randomly generated problems; five randomly sampled rotation lengths $(3, 5, 7)$ for each proportion of hospitals located in unique regions $(20\%, 40\%, 60\%)$, percentage of fixed- versus variable-cost hospitals $(25\%, 50\%, 75\%)$, and maximum idle time between rotations in weeks (3,5,7).

Appendix B: Proofs of Lemmas and Propositions

Proof of Proposition 1:

Assume first that $\mathcal{S}$ is feasible and let $(\mathcal{G}_s, \tau_s)$ be defined by (7) and (8) for $s = 1, \dots, J$. Feasibility, specifically (**R.1**), implies that all j_s must be distinct yielding, for any $s \neq s'$, $j_s \neq j_{s'}$. Thus, $j_s \in \mathcal{G}_{s-1}$ by construction given that $\mathcal{G}_0 = J$. Since (**R.2**) and (8) imply $t_s \geq t_{s-1} + \Delta_{j_{s-1}} = \tau_{s-1}$ and, by feasibility, $k_s \in \mathcal{K}(j_s)$, we have that $(j_s, k_s, t_s) \in \mathcal{A}(\mathcal{G}_{s-1}, \tau_{s-1})$. Hence, condition (9) is satisfied.

Conversely, consider a sequence $(\mathcal{G}_s, \tau_s)$ satisfying (7)–(9) for some given schedule $\mathcal{S}$. We first show, by induction on the number of scheduling stages, that rotations $j_1, \dots, j_J$ are pairwise distinct. Therefore, all rotations are performed exactly once, satisfying (**R.1**). Note that this is trivially satisfied for $s = 1$. Suppose now such a condition is true for all elements $s' < s$ for some $2 \leq s' \leq J$. This implies that $\mathcal{G}_{s-1} = J \setminus \{j_1, \dots, j_{s-1}\}$ by condition (7) and the fact that $\mathcal{G}_0 = J$. Since $j_s \in \mathcal{G}_{s-1}$ by condition (9), we must have that $j_s \neq j_1, \dots, j_{s-1}$. Finally, (**R.2**) is satisfied because $t_s \geq \tau_{s-1} = t_{s-1} + \Delta_{j_{s-1}}$ by conditions (8) and (9), while the definition of $\mathcal{A}$ in (9) implies $k_s \in \mathcal{K}(j_s)$. Therefore, $\mathcal{S}$ is a feasible schedule. □

		$J = 3$ rotations				$J = 5$ rotations			
		MILP		**NBF**		**MILP**		**NBF**	
Hospitals	**Students**	Time(s)	Instances	Time(s)	Instances	Time(s)	Instances	Time(s)	Instances
5	10	2.47	0	0.0	0	2.89	75	0.01	0
	25	15.58	0	0.02	0	270.42	94	0.47	0
	50	469.09	56	0.07	0	795.07	130	7.12	0
	100	896.73	124	0.22	0	-	135	51.95	0
	200	-	135	0.42	0	-	135	101.93	1
	300	-	135	0.5	0	-	135	106.42	0
10	10	13.87	84	0.0	0	10.71	132	0.01	0
	25	-	135	0.01	0	-	135	0.4	0
	50	-	135	0.07	0	-	135	6.7	0
	100	-	135	0.38	0	-	135	152.35	0
	200	-	135	1.12	0	-	135	686.79	23
	300	-	135	1.38	0	-	135	749.21	61
20	10	-	135	0.01	0	-	135	0.01	0
	25	-	135	0.01	0	-	135	0.32	0
	50	-	135	0.07	0	-	135	5.41	0
	100	-	135	0.37	0	-	135	149.82	0
	200	-	135	2.6	0	-	135	838.67	86
	300	-	135	5.73	0	-	135	970.99	116
30	10	-	135	0.0	0	-	135	0.0	0
	25	-	135	0.01	0	-	135	0.26	0
	50	-	135	0.06	0	-	135	5.26	0
	100	-	135	0.41	0	-	135	72.49	0
	200	-	135	2.55	0	-	135	743.93	62
	300	-	135	6.51	0	-	135	1,382.98	128

Table 1 Performance comparison between MILP and NBF approaches.

Proof of Lemma 1:

By construction, a feasible schedule as defined by Proposition 1 maps to an n_r, n_t-path in D, and vice-versa. Thus, any subset of m feasible schedules can be alternatively formulated as sending m integral units of flow over D (Ahuja et al., 1993). As a result, we can formulate a network flow model represented by $\texttt{flow}(D, m)$, where y_a corresponds to the amount of flow on arc a. □

Proof of Proposition 2:

If there are no fixed-cost hospitals and the number of students requiring a specific rotation is smaller than the capacity of the hospitals that offer those rotations (i.e., **R.4** becomes inconsequential), the problem reduces to a minimum cost network flow over D. To account for **R.3**, we include an upper bound on the arc capacities emanating from the root node n_r. More specifically, if an arc a emanating from node n_r has triplet $\theta(a) = (j_1, k_1, t_1)$, then the capacity of this arc can be no greater than the sum of students who can start clerkship by t_1. This problem can be solved in strongly polynomial time by Orlin's algorithm, for example, with a worst-case complexity of $O(|A| \log |N| (|A| + |N| \log |N|))$. □

Proof of Proposition 3:

We show that solving NBF to optimality is equivalent to finding a minimum cost flow over an associated graph D', where $D \to D'$ is a polynomial-time transformation. First, the definition of the state space implies that, for any two nodes $n, n' \in L_s$, $\xi(n) \neq \xi(n')$. However, because there is a fixed sequence of rotations, $\mathcal{G}_s$ is identical for every $n \in L_s$. Thus, two nodes $n, n' \in L_s$ differ only due to the start week of rotation j_s at hospital k_s. Second, the assumption that rotations are completed in a fixed sequence implies that, given a layer s, $j_s \neq j'_s$ for any $s' \neq s$. As a result, for a given node $n \in L_s$, there are no other nodes in D with rotation j_s being performed at hospital k_s and beginning on week t_s. Hence, for a given $\tilde{n} \in L_{s+1}$, all arcs $(n, \tilde{n})$ for some $n \in L_s$ belong to the same capacity constraint, and no other arcs in D contribute.

With this observation, we now construct D'. Given D, for $s = 1, \ldots, J$, split each node $n \in L_s$ into two nodes: n_1 and n_2. All incoming arcs to node n become incoming arcs to n_1. All outgoing arcs from node n become outgoing arcs from n_2. Nodes n_1 and n_2 are then connected by a single arc with capacity $C_{j_s k_s t_s}$. Since each node of D is duplicated, the time complexity of this transformation is $O(|N|)$. We refer to the resulting augmented graph as D'. By solving a minimum cost flow problem over D' using the same technique as discussed in Proposition 2 to bound the arcs emanating from the root node, we find the minimum cost schedules while ensuring that all capacity constraints are enforced in polynomial time in the size of D. □

Proof of Proposition 4:

We reduce the decision version of the Maximum Independent Set Problem (ISP) to an instance of NBF where $c_k^F = 0$ for all $k \in \mathcal{K}_F$ and D is polynomial in the size of the problem input. Given an undirected simple connected graph $G = (V, E)$ with vertex set V and edge set $E \subseteq V \times V$, the ISP asks if there exists a vertex subset $V' \subseteq V$ with cardinality $M \leq |V|$ such that no two vertices ν, ν' in V' are adjacent, i.e., $\{\nu, \nu'\} \notin E$ for all $\nu, \nu' \in V'$, $\nu \neq \nu'$. The ISP is strongly NP-Hard (Garey and Johnson, 1990).

Consider a graph $G = (V, E)$ and a constant M provided as an input to the ISP. We construct an instance of NBF with two hospitals $\mathcal{K} = \{1, 2\}$, $J = |E|$ rotations, and $T = |E|$ weeks for the planning horizon. We establish that only $|V|$ schedules are feasible (representing, for example, arbitrary operational requirements established by AUC), each uniquely associated with a vertex $\nu \in V$. They are defined as follows.

- For all schedules, rotation $j \in \{1, \ldots, |E|\}$ can only be performed at week $t = j$.
- For the schedule associated with $\nu \in V$, rotation $j \in \{1, \ldots, |E|\}$ is performed at hospital 1 if and only if ν belongs to the j-th edge of E. Otherwise, it is performed at hospital 2.

We set all capacities of hospital 1 to one, and all capacities of hospital 2 to $|E| + 1$ (hospital costs can be fixed to zero). In such a case, G has an independent set with M vertices if and only if it is possible to schedule $I = M$ students that are available to start at week 0.

To verify this claim, notice that, because G is connected, every vertex $\nu \in V$ belongs to at least one edge in E. By construction, the schedule associated with ν performs at least one rotation in hospital 1, and hence, can only be allocated to a single student. Moreover, because of the unitary capacity of hospital 1, no two schedules can be allocated in a solution if their corresponding vertices share an edge. Because schedules can only be selected once and no two schedules can be selected if they belong to a common set (i.e., edge), there is an one-to-one correspondence between independent sets and solutions of the NBF instance. Finally, since there are $|V|$ schedules, D has at most $|V|$ nodes per layer (i.e., the extreme case where paths are disjointly represented), and the result follows. □

Proof of Proposition 5:

Because $(\eta_s, \zeta_s, \mathcal{F}_s)$ is defined independently from $(\mathcal{G}_s, \tau_s)$, we omit the proof that $(\mathcal{G}_s, \tau_s)$ is valid as it is identical to the proof of Proposition 1. We proceed by splitting the proof into two parts: the first recursively defines η_s and ζ_s, while the second focuses on $\mathcal{F}_s$.

Let $\{(\eta_s, \zeta_s)\}_{s=0}^{J}$ be a sequence satisfying conditions (26), (27), and (29) for some schedule $\mathcal{S}$. Suppose, for the purpose of contradiction, that (24) is violated by $\mathcal{S}$. Then, there must exist a minimum s' such that $\sum_{s=2}^{s'} \mathbb{I}(r_{k_{s-1}} \neq r_{k_s}) > \bar{r}$. Because $\zeta_s = r_{k_s}$ by (27), it follows from (26) that

$$\begin{aligned}
\eta_{s'-1} &= \eta_{s'-2} + \mathbb{I}\left(r_{k_{s'-2}} \neq r_{k_{s'-1}}\right) \\
&= \eta_{s'-3} + \mathbb{I}\left(r_{k_{s'-3}} \neq r_{k_{s'-2}}\right) + \mathbb{I}\left(r_{k_{s'-2}} \neq r_{k_{s'-1}}\right) \\
&= \dots \\
&= \sum_{s=2}^{s'-1} \mathbb{I}(r_{k_{s-1}} \neq r_{k_s}) = \bar{r},
\end{aligned}$$

where the last equality follows from the fact that s' is the first stage when (24) is violated. Since an indicator function increases the sum by at most one, the total region transitions up to $s'-1$ must be equal to $\bar{r}$. This implies that $r_{k_{s'}} \neq r_{k_{s'-1}}$, which would violate (29).

Conversely, let $\mathcal{S}$ be a schedule that observes (24) and create a sequence $\{\eta_s\}_{s=0}^{J}$ based on the transitions specified by (26). For any $s' = 2, \dots, J$, we have by construction that

$$\eta_{s'} = \eta_{s'-1} + \mathbb{I}\left(r_{k_{s'-1}} \neq r_{k_{s'}}\right) = \dots = \sum_{s=2}^{s'} \mathbb{I}(r_{k_{s-1}} \neq r_{k_s}) \leq \bar{r},$$

where the last inequality follows from (24). Suppose once again, for the sake of contradiction, that there exists some $s \leq J-1$ such that $r_{k_{s+1}} \neq r_{k_s}$. Then by (26), we would have $\eta_s + 1 > \bar{r}$.

Let $\{\mathcal{F}_s\}_{s=1}^{J}$ be a sequence as specified by (28) and (29) for some schedule $\mathcal{S}$. Suppose, for the purpose of contradiction, that (25) is violated by $\mathcal{S}$. Therefore, there exists $k \in \mathcal{K}_F$ for which some, but not all, rotations in $\mathcal{J}(k)$ are performed at k. Let s' be the first stage in which a rotation in $\mathcal{J}(k)$ is performed at k, and let s'' be the first stage after s' when the issue arises, that is, $\mathcal{F}_{s''}(j) \neq k$ for some $j \in \mathcal{J}(k)$. By the time we reach stage s'' in the construction of the schedule, $\mathcal{F}_{s''}$ is such that it falls into case (b)(i) in the definition of $\mathcal{H}(\mathcal{G}_{s''}, \mathcal{F}_{s''}, j)$. Hence, by definition, $\mathcal{H}(\mathcal{G}_{s''}, \mathcal{F}_{s''}, j) = \{k\}$, contradicting the assumption that $\mathcal{F}_{s''}(j) \neq k$.

The argument in the reverse direction is analogous. □

Proof of Proposition 6:

Since each node is associated with at least one recursive state by Algorithms 1 and 2, it suffices to show that there is an upper bound on the number of states in the system. A state that considers all AUC-specific constraints is represented by the tuple $(\mathcal{G}, \tau, \eta, \zeta, \mathcal{F})$ based on Proposition 5. For each element of the tuple, the total number of states is bounded as follows: $|\mathcal{G}| \leq 2^J$, $\tau \leq T+1$, $\eta \leq J-1$, $\zeta \leq R$, and $|\mathcal{F}| \leq (|\mathcal{K}|+1)^J + 1$. Thus, the number of nodes in the MDD is bounded by $O\left(J \times T \times R \times (2|\mathcal{K}+1|)^J\right)$. □

Proof of Proposition 7:

Similar to the proof of Proposition 2, the problem reduces to a minimum-cost network flow where we are required to send b units of flow from the MDD root node n_r to the terminal node n_t. □