Complexity Results for the Basic Residency Scheduling Problem

Jiayi Guo, David R. Morrison, Sheldon H. Jacobson, Janet A. Jokela

Abstract

Upon graduation from medical school, medical students join residency programs to complete their clinical training and fulfill specialty board certification requirements. During residency, they are assigned several years of clinical rotations, where they work under the supervision of physician faculty in a variety of different settings, to ensure that they gain the requisite training prior to beginning independent practice. These rotations typically last a short period of time, and the problem of determining a schedule for all the residents in a program can be quite tedious. In this paper, a basic Residency Scheduling Problem (RSP) that produces a one-year schedule is defined, and a proof of NP-completeness is presented. Furthermore, a specific model of the residency scheduling program for the internal medicine residency program at the University of Illinois College of Medicine at Urbana-Champaign is studied. Finally, a method for determining alternate optima is presented, as well as a method for handling schedule perturbation in the case that residents are unable to complete all or part of their assignment.

1 Introduction and Background

To become eligible for specialty board certification, medical school graduates are required to participate in a postgraduate residency training program. The residency program assigns residents to rotation periods in local hospitals and clinics so that the residents gain the necessary educational experience and training before they begin to practice on their own. Each rotation period typically lasts between four and six weeks (Ozkarahan, 1994), at which point the residents transition to a different type of rotation.

There are several different requirements that must be met as residents progress through a residency program. First, each resident has a set of educational requirements that must be satisfied to ensure that the resident has sufficient experience in a range of specific disciplines (Franz and Miller, 1993). Secondly, as residents work in a variety of clinical environments and provide valuable contributions to many hospitals' physician work forces, some key hospital rotations must be adequately staffed every rotation period. Thus, a set of teaching service requirements is imposed that ensures a certain number of residents to cover these rotations during every period (Franz and Miller, 1993). Finally, residents may be susceptible to stress, fatigue, and medical errors after long hours of intense clinical work (Topaloglu and Ozkarahan, 2011). Thus, the Accreditation Council for Graduate Medical Education (ACGME), a private non-profit organization that accredits residency programs, imposes a set of rules and regulations for resident duty hours (ACGME).

In many cases these sets of requirements come in conflict with each other. For example, if the teaching service demands and educational requirements are not appropriately balanced, the residency program must negotiate with the hospital on behalf of its residents to come to a resolution that meets the programs' educational needs. Additionally, most residency programs have budgetary constraints, which limit the number of residents that can be supported.

It is therefore important to have a method for developing a schedule (i.e., an assignment of residents to rotations) that satisfies all constraints, does not over-burden the residents with teaching service demands at the expense of educational requirements, meets the duty hour standards imposed by the ACGME, and is easy to modify in the event that a resident becomes sick or is unable to work. Typically, a schedule is made empirically via trial and error (Day et al., 2006). Unfortunately, this approach is quite time-consuming and costly, and is susceptible to error, as the scheduling problem is non-trivial to solve.

The significant contributions of this paper are threefold: first, a formal definition of a basic residency scheduling problem (RSP) is presented; this model includes only the most basic constraints that are common to all residency programs. The computational complexity of this model is then studied, and computational results are reported for a greedy algorithm and for CPLEX on a randomly-generated database of test problems. Secondly, an integer programming model which can simultaneously determine a rotation schedule and day-to-day working schedule for all residents is provided for a one-year period of time; computational results are shown for a real problem demonstrating that this IP is an effective strategy for solving RSP in practice. Finally, methods are presented for determining alternate optima that may be useful when disruptions to the schedule occur, which is currently a significant issue for hospital schedulers.

This paper is organized as follows: in Section 2, related research on this problem is discussed. Section 3 provides the formal problem statement and model assumptions. Section 4 shows that the optimization version of the RSP is NP-hard in general, and presents several polynomial-time special cases of the problem, as well as a polynomial-time greedy algorithm for RSP. In Section 5, computational results from the model for a specific residency program are presented, as well as for a database of randomly-generated RSP instances. Finally, Section 6 offers some concluding comments and directions for future research.

2 Related Work

There are some related papers in the literature; however, those papers only capture some of the constraints present in the model presented here. Day et al. (2006) develop an integer programming model for a weekly work hour schedule, only taking ACGME rules and a few hospital specific requirements into account. Topaloglu (2006) proposes a multi-objective programming model to balance the weekday and weekend shift assignments based on different coverage requirements, seniority levels, and resident preferences. Furthermore, Topaloglu (2009) formulates a mixed-integer programming model based on teaching service demands, work rules, and resident preferences. None of these papers address the educational requirements present in RSP.

In a slightly different setting, Wang et al. (2007) develop a genetic algorithm for the physician scheduling problem to generate a balanced schedule. In their model, constraints are divided into hard constraints and soft constraints. They use a genetic algorithm to generate a schedule that satisfies all the hard constraints and as many soft constraints as possible. Greedy algorithms have also been developed for various scheduling problems (for example, in the nurse scheduling literature (Gutjahr and Rauner, 2007)).

Most exact algorithms for related scheduling problems use a Dantzig-Wolfe decomposition of the integer program, together with a branch-and-price algorithm to find and verify the optimal solution. Brunner and Edenharter (2011) provides an example of this for the physician scheduling problem; this problem is able to handle physicians with different experience levels in the model. However, this problem deals only with a monthly shift assignment for a single type of rotation. A similar paper uses a branch-and-price algorithm to schedule nurses in a multi-objective setting (Maenhout and Vanhoucke, 2009). Finally, Beliën and Demeulemeester (2004) and Beliën and Demeulemeester (2006) provide a branch-and-price algorithm for the medical trainee scheduling problem, which is similar to the residency scheduling problem but does not consider additional day-to-day work requirements for the trainees.

Additionally, Franz and Miller (1993) regard RSP as a large-scale multi-period staff assignment problem with the objective of maximizing residents' schedule preference. They develop a rounding heuristic method to find an approximate schedule solution quickly. Although this paper includes teaching service demands and educational requirements, it does not consider the assignment of shifts given the ACGME duty hour rules, nor does it discuss how to find alternate optima, an important practical question in solving the RSP. Furthermore, Miller and Franz (1996) also solve the assignment problem of employees by a rounding heuristic method. This problem is similar to the basic RSP model described in this paper, though some of the specific constraints differ.

Cohn et al. (2009) provide a practical study of implementing a residency scheduling solver at the Boston University School of Medicine; in this document, they describe how to use MIP solvers to construct a variety of good schedules that can be used by hospital schedulers for their residency program. However, the schedules that they produce are on a day-to-day level, as the program described does not need to have residents assigned to particular rotations for a longer period of time. They also do not discuss any theoretical results regarding their work.

Finally, Osogami and Imai (2000) discuss the complexity of the nurse scheduling problem and showed the problem to be NP-complete. Moreover, a polynomial-time algorithm for a special case of the problem can solve the RSP problem when educational requirements and teaching service demands are disjoint. However, the general formulation is different from the formulation presented here.

3 The Residency Scheduling Problem

This section presents a generic version of the RSP when restricted to a one-year schedule (that is, a schedule that ignores multi-year requirements) in both decision and optimization form. A specific instance of the problem that arises from the internal medical residency program at the University of Illinois College of Medicine at Urbana-Champaign is also provided.

3.1 The General RSP

To formally define the residency scheduling problem, the following definitions and assumptions are made. Let $\mathcal{R}$ be a set of residents, $\mathcal{P}$ be a set of rotation periods, and $\mathcal{T}$ be a set of specific rotations that are present in every rotation period. In addition, a function $H : \mathcal{P} \times \mathcal{T} \to \mathbb{N}_0$ is given, specifying the necessary teaching service demands for each rotation in each rotation period. Also, a function $E : \mathcal{R} \times \mathcal{T} \to \mathbb{N}_0$ specifies the educational requirements for each resident. It is assumed that if two residents have identical educational requirements, they are interchangeable with respect to the model.

Definition 1 (Residency Scheduling Problem)**.** *Given a set $\mathcal{R}$ of residents, a set $\mathcal{P}$ of rotation periods, and a set $\mathcal{T}$ of rotations, as well as functions H and E specifying the teaching service and educational requirements, respectively, does there exist a feasible assignment of residents to rotations satisfying all constraints? In other words, a function $f : \mathcal{R} \times \mathcal{T} \times \mathcal{P} \to \{0,1\}$ is sought satisfying*

a) $\sum_{r\in\mathcal{R}} f(r,t,p) \geq H(p,t)\ \forall\ p \in \mathcal{P}, t \in \mathcal{T}$,

b) $\sum_{p\in\mathcal{P}} f(r,t,p) \geq E(r,t)\ \forall\ r \in \mathcal{R}, t \in \mathcal{T}$,

c) $\sum_{t\in\mathcal{T}} f(r,t,p) = 1\ \forall\ r \in \mathcal{R}, p \in \mathcal{P}$.

Constraint (a) stipulates that every rotation in every period is staffed with enough people to satisfy the given teaching service demands. Constraint (b) ensures that residents meet their educational requirements for the year, and constraint (c) ensures that each resident is assigned to exactly one rotation in each rotation period.

Note that this model produces a fixed single-year schedule. In other words, if residents have requirements that must be satisfied over the course of multiple years, this model cannot handle such constraints, though they should be straightforward to incorporate to the integer programming model if necessary. Also, note that the teaching service demands and educational requirements are equivalent mathematically, since by interchanging the roles of the residents and the rotation periods, the educational requirements can be viewed as teaching service demands, and vice versa.

An optimization version of RSP can also be formulated which seeks to minimize the number of residents necessary to satisfy all of the teaching service demands. This variant must satisfy all of the same constraints as in Definition 1 except for (c). For this version of the problem, it is assumed that an upper bound on the number of residents available for each year is provided, based on the residency program budget and ACGME specific requirements.

Table 1: Teaching service demands for the residency program at UIUC (see Appendix A for definitions).

Rotation Type	PGY1/Prelim	PGY2/3
CAM	6	3
PAM	3	3
NF	2	2
CCC	1	2
VAN, PG, CC	1	0
VAG, VAT, CID, AMB	0	1

In practice, it is not sufficient to simply assign residents to rotations; the schedule must also determine the working hours from week-to-week for each resident, subject to the ACGME rules. This model does not take these rules into consideration; however, these rules can generally be modeled as additional side constraints on the problem that need to be satisfied, as shown in the following section.

3.2 The Residency Program at the University of Illinois College of Medicine at Urbana-Champaign

The University of Illinois College of Medicine at Urbana-Champaign (UIUC-COM) has the following constraints for their residency program. Each rotation period lasts for four weeks; thus, there are thirteen periods per year. Furthermore, within each rotation period, there are 36 different rotations that residents may take. However, only 13 of these rotations are present in constraints when restricted to a one-year schedule, such as is considered by this model; all the remaining requirements are either elective rotations or required across three years.

There are four different types of residents in the UIUC-COM program, based on seniority. The four types are classified as Post-Graduate Year (PGY) 1, 2, and 3, and preliminary PGY1 residents. Within a group, the teaching service demands and educational requirements do not differ from resident to resident, but there are unique requirements across the different groups. The teaching service demands and education requirements for this program are listed in Tables 1 and 2.

Furthermore, some of the rotations present at UIUC-COM are **shift rotations**, which means that the residents on these rotations may be assigned to work additional hours during the week; this is called **taking a shift**. There are specific requirements for the shift rotations limiting how

Table 2: The (one-year) baseline educational requirements for the residency program at UIUC (see Appendix A for definitions; note that these may change over time, and additional requirements are present that span multiple years, and thus are not considered in this model).

Rotation Type	PGY1	PGY2	PGY3	Prelim
CAM or PAM	6	4	3	6
CCC	1	1	1	1
ICR-VAC	1	0	0	0

frequently residents can take a shift within a given time period. These constraints also must be considered when constructing a schedule.

Additional notation for the UIUC-COM residency program and a list of the full names of the 13 rotations appearing in constraints is provided in Appendix A, together with the abbreviations used in the remainder of the paper. In addition to the teaching service demands and educational requirements presented in Tables 1 and 2, there are a number of added constraints that must be included in the model which cannot be exactly represented via the general RSP model. These additional constraints are presented in Appendix B, and the full IP model for the program at UIUC-COM is provided in Section 5.

4 The Complexity of the RSP

This section presents the decision version of the minimum residency scheduling problem. In order to consider the most basic model possible, additional constraints such as those described in Section 3.2 are ignored. It is shown that the RSP is NP-complete; in particular, the combination of the educational requirements and teaching service demands often come into conflict, making the problem challenging in general (note that Gibert and Hofstra (1988) show that an assignment problem with even a single side constraint is NP-hard; however, the structure of their side constraints are different than those present in RSP, so a separate proof of NP-completeness is needed).

4.1 RSP is NP-hard

A reduction for the RSP is given in two parts; first, a reduction from the Minimum 3-Set Cover problem to the Minimum 3-Multiset 3-Cover problem is shown. Then, a reduction from the Minimum 3-Multiset 3-Cover problem to the RSP is shown.

Definition 2 (Minimum 3-Set Cover)**.** *Given a collection $\mathcal{C}$ of subsets of a finite set S such that $|C| = 3$ for each $C \in \mathcal{C}$, together with a positive integer $K \leq |\mathcal{C}|$, the* **Minimum 3-Set Cover** *(3SC) problem seeks a cover $\mathcal{C}^* \subseteq \mathcal{C}$ of S such that $|\mathcal{C}^*| \leq K$.*

It is well-known that 3SC is NP-hard (Garey and Johnson, 1979). The Minimum 3-Multiset 3-Cover problem is an extension of 3SC that seeks to cover each element of S three times.

Definition 3 (Minimum 3-Multiset 3-Cover)**.** *Given a collection $\mathcal{M}$ of multiset subsets of a finite set U such that $|M| = 3$ for each $M \in \mathcal{M}$, together with a positive integer $J \leq |\mathcal{M}|$, the* **Minimum 3-Multiset 3-Cover** *(3M3C) problem seeks a 3-cover of U, that is, a collection $\mathcal{M}^* \subseteq \mathcal{M}$ such that $|\mathcal{M}^*| \leq K$ and every element in U is covered at least three times by sets in $\mathcal{M}^*$.*

Lemma 1. *3M3C is NP-complete.*

Proof. By checking that every element in U is covered at least three times, it can be seen that 3M3C is verifiable in polynomial time—that is, that 3M3C is in NP. Thus, it remains to show that 3M3C is NP-hard. This is shown by a reduction from 3SC.

Given an instance of 3SC, construct T as a copy of the input set S. In other words, $T \cap S = \varnothing$, and $|T| = |S|$. List elements in S to be $S = \{s_1, s_2, ..., s_n\}$, and list elements in T to be $T = \{t_1, t_2, ..., t_n\}$. Construct a collection of subsets of $S \cup T$ to be $\hat{\mathcal{C}}$ such that $\hat{\mathcal{C}} = \{\{s_i, t_i, t_i\} \mid i = 1, 2, ..., n\}$. Construct another collection of subsets of $S \cup T$ to be $\bar{C}$ with $\bar{C} = \{\{s_j, t_{j-1}, t_{j-1}\} \mid j = 2, 3, ..., n\} \cup \{\{s_1, t_n, t_n\}\}$.

Then, it suffices to show that solving 3SC is equivalent to solving 3M3C with $U = S \cup T$, $\mathcal{M} = \mathcal{C} \cup \hat{\mathcal{C}} \cup \bar{\mathcal{C}}$, and $J = K + 2|S|$. To see this, note that since every element of T must be covered at least three times, then a solution $\mathcal{M}^*$ to 3M3C must satisfy $\mathcal{M}^* \supseteq \hat{\mathcal{C}}$ and $\mathcal{M}^* \supseteq \bar{\mathcal{C}}$. Thus every element of S is covered exactly twice, meaning that the remaining sets in $\mathcal{M}^*$ form a cover of $\mathcal{C}$. Since $|T| = |S|$, this implies that $\mathcal{C}^* = \mathcal{M}^* - \hat{\mathcal{C}} - \bar{\mathcal{C}}$ has size less than or equal to K, as desired. Similarly, any cover $\mathcal{C}^*$ of S can be immediately transformed into a cover of U by simply taking $\mathcal{M}^* = \mathcal{C}^* \cup \hat{\mathcal{C}} \cup \bar{\mathcal{C}}$. ∎

To establish that the basic RSP is NP-complete, the decision version of the minimization variant presented in Section 3 is considered. This problem takes as an additional input a positive integer

N and seeks a feasible assignment of residents using N or fewer residents.

Theorem 1. *The minimization version of RSP is NP-complete.*

Proof. Given an assignment of residents to rotations, one can verify in polynomial time that each resident is assigned to exactly one rotation in every period, and that each teaching service demand and educational requirement is satisfied. Thus, RSP is in NP, and it remains to show that it is NP-hard. To do this, a reduction from 3M3C is shown.

Given an instance of 3M3C, let $\mathcal{T} = U$ be the set of rotations, and let $\mathcal{P} = \{1, 2, 3\}$ be the set of three rotation periods. Finally, for each $M \in \mathcal{M}$, construct a resident r, and define

$$H(p,t) = 1 \ \forall \ p \in \mathcal{P}, t \in \mathcal{T}$$

$$E(r,t) = \begin{cases} 1 & \text{if } t \in M \\ 0 & o.w. \end{cases} \quad \forall \ r \in \mathcal{R}$$

Finally, let $N = J$. Then, if there is a feasible schedule using N or fewer residents, then there exists a 3-multiset 3-cover of size J or less for 3M3C, and vice versa.

First, note that if there exists a feasible schedule, then each rotation t is covered at least once per rotation period, since the teaching service demands $H(p,t)$ must be satisfied in every period. Thus, each element in U is covered at least three times by some sets in $\mathcal{M}$ corresponding to the residents assigned to rotation t in the three rotation periods. Therefore, since every element in U is covered by three sets in M, these sets are a valid 3-cover.

Then, it remains to show that, given a valid 3-cover, a feasible schedule of size less than N can be constructed. Let $m = |U|$. Since $\mathcal{M}^*$ contains at least $3 \cdot |\mathcal{M}^*|$ (not necessarily distinct) elements, and all elements in U have been covered at least three times, $|\mathcal{M}^*| \geq m$. There are two cases to consider:

$|\mathcal{M}^*| = m$ In this case, every element of U is covered exactly three times by $\mathcal{M}^*$. Let $A_{|U|\times|U|}$ be the assignment matrix of sets to elements that they cover of U; then, note that every row and column sum is 3, so A can be decomposed into three permutation matrices A^1, A^2, and A^3 (West, 2001). Each permutation matrix can be taken as an assignment of $|U|$ residents to rotations in a particular period. Every rotation is covered once in each period (satisfying the teaching service demands), and each resident has satisfied his educational requirements; furthermore so this is a valid assignment.

$|\mathcal{M}^*| > m$ This problem is reduced to the first case by adding additional dummy elements as follows. Since every element in U is covered three times, mark three distinct occurrences of $u_1, u_2..., u_m$ in sets in $\mathcal{M}^*$ to keep. Then, any unmarked elements in $\mathcal{M}^*$ can be replaced with dummy elements. Since there are exactly $3m$ marked elements, and a total of $3 \cdot |\mathcal{M}^*|$ elements in the sets of $\mathcal{M}^*$, the total number of unmarked elements is $3 \cdot (|\mathcal{M}^*| - m)$. Therefore, construct $(|\mathcal{M}^*| - m)$ dummy elements to add to the set $\mathcal{U}$, and arbitrarily replace each triple of unmarked elements in $\mathcal{M}^*$ with one dummy element. Then, $|\mathcal{U}| = \mathcal{M}^*$, and the problem reduces to the previous case.

■

4.2 Polynomial Special Cases

There are two special cases for the RSP that are polynomially-solvable. The first is when there are no educational requirements (or conversely, no teaching service demands); another special case is when there is only one rotation period.

Lemma 2. *If $E(r, t) = 0$ for all $r \in \mathcal{R}$ and $t \in \mathcal{T}$, then the RSP is polynomially-solvable.*

Proof. If there are only teaching service demands, then all residents are interchangeable, since there are no educational requirements. Thus, for a particular period p_0, the number of required residents is $\sum_{t\in\mathcal{T}} H(p_0, t)$. In all, the minimal number of residents for the program is $\max_{p\in\mathcal{P}} \sum_{t\in\mathcal{T}} H(p, t)$, which is the maximum number of required residents for any period. If the total number of residents is less than required number of residents, then the problem is infeasible. To actually determine the assignment, a greedy algorithm can be used. ■

The proof is identical for the following corollary:

Corollary 1. *If $H(p, t) = 0$ for all $p \in \mathcal{P}$ and $t \in \mathcal{T}$, then the RSP is polynomially-solvable.*

Next, the following result is shown:

Lemma 3. *If $|\mathcal{P}| = 1$, then the RSP is solvable in polynomial time.*

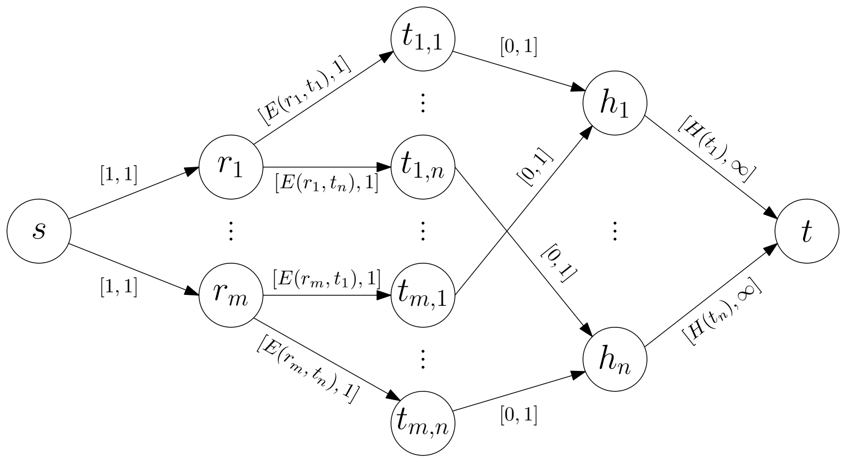

Figure 1: The network construction for the case when the number of rotation periods is exactly one.

Proof. Note first of all that in this condition, E must be a binary function, since no resident is allowed to take more than one rotation. Under this assumption, the following network flow model can be constructed, shown in Figure 1.

To build the network, construct a source node s and a sink node t, as well as $|\mathcal{R}|$ nodes $r_1, r_2..., r_m$ for each resident in the program. For each resident $r_i \in \mathcal{R}$, construct nodes $t_{i,1}, t_{i,2}..., t_{i,n}$, one for each rotation type in $\mathcal{T}$, where $|T| = n$. Finally, construct nodes $h_1, h_2..., h_n$ to represent the teaching service demands for each rotation type.

Add an arc from s to each r_i with upper and lower bounds equal to 1 (since each resident must be assigned to at exactly one rotation), and add arcs from r_i to $t_{i,j}$ for each resident node r_i and each rotation type node $t_{i,j}$; set the upper bound on these arcs to 1, and the lower bound to $E(r_i, t_j)$ (which is either 0 or 1, by assumption). Finally, for each rotation type node $t_{i,j}$, add an arc to h_j with unbounded capacity, and an arc from h_j to t with unbounded capacity and lower bound equal to $H(t_j)$. Since all capacities are integral, and there are $O(mn)$ nodes and $O(mn)$ arcs in the network, a feasible integer flow can be found in polynomial time using any standard network algorithm (Papadimitriou and Steiglitz, 1998). It is clear that a feasible flow corresponds to a feasible schedule, and vice versa. ∎

4.3 A Greedy Heuristic

This section presents the `ScheduleResidents` algorithm, a greedy assignment algorithm that demonstrates good results when solving the feasibility version of RSP. Note that greedy algorithms are not new algorithms for this class of problems (see, for example, Bellanti et al. (2004), which embeds a greedy algorithm inside a local-search heuristic for the nurse scheduling problem).

From a high level, this greedy heuristic first attempts to satisfy all of the teaching service demands present in the problem. Once these demands have been satisfied, it then fills in any remaining educational requirements for the residents (though note that the algorithm could equally attempt to satisfy educational constraints first, and then cover the teaching service demands).

The `ScheduleResidents` algorithm first performs a pre-check for infeasibility; if any resident is assigned more educational requirements than there are periods, or if any rotation has more teaching service demands than there are residents, then the problem is infeasible (note that the converse is not true). If the problem instance passes this check, the algorithm then attempts to assign residents to rotations in the following manner:

1. If there exists a rotation in some period whose teaching service demands have not been satisfied and for which some resident has unsatisfied educational requirements, then choose an unassigned resident with non-zero educational requirement to cover that rotation. Repeat until no more residents can be assigned.

2. If no resident exists with unsatisfied educational requirements for a particular rotation that has unsatisfied teaching service demands in some period, choose an unassigned resident to assign to this rotation.

3. If residents exist with unsatisfied educational requirements for some rotation, choose a period to assign them to that rotation.

Since the sum of the teaching service demands in every period is at most the number of residents (due to our pre-check), at the end of step 2, it is guaranteed that all teaching service requirements will be satisfied. However, it may be the case that it is not possible to satisfy all the remaining educational requirements in step 3. Furthermore, it is possible that a different assignment may exist that is feasible. Therefore, `ScheduleResidents` does not guarantee feasibility.

It is well-known that for many greedy algorithms, the manner of tie-breaking can have a dramatic impact on the performance of the algorithm. Therefore, `ScheduleResidents` employs a number of tie-breaking mechanics to attempt to ensure feasibility of the problem. These are described below:

1. In step 1, for a rotation t in a rotation period p, a resident is chosen with largest remaining educational requirement for rotation t. In the case of a tie, a resident is chosen with the largest number of remaining educational requirements across all rotations. Finally, ties in this last case are broken randomly.

2. In step 2, a resident is chosen to cover a rotation as the resident with the fewest remaining educational requirements.

3. In step 3, residents are assigned arbitrarily to rotations to satisfy their educational requirements.

Note that this greedy heuristic is polynomial in the number of residents, rotation periods, and rotations; when it returns a feasible solution, this solution is guaranteed to be correct, since all residents have satisfied their educational requirements, and all periods have satisfied teaching service demands.

4.4 The Feasibility Problem

The complexity of the feasibility version of RSP remains unknown; that is, given sets $\mathcal{R}, \mathcal{P}, \mathcal{T}$, and teaching service demands $H(p,t)$ and educational requirements $E(r,t)$, is there an assignment of *all* residents to rotations that satisfies all constraints?

The main difficulty in resolving this issue appears to be the individuality of the residents with respect to their educational requirements. The problem that arises when reducing from a known NP-complete problem like set cover (as in the minimization version of RSP) comes in choosing the correct subsets to transform into residents. Other reductions have been considered, namely from 3SAT and Vertex Cover, but there does not seem to be a natural way to encode the constraints of these problems as teaching service or educational requirement functions.

Nevertheless, the difficulty of simultaneously resolving both teaching service demands and educational requirements, as well as results on similar problems in the literature, suggests that the following conjecture is true:

Conjecture 1. *The feasibility version of the RSP is NP-complete.*

In the following section, computational results from the residency program at UIUC-COM are presented, as well as a discussion of how alternate optima can be obtained.

5 Computational Results and Alternate Optima

As the basic RSP has been shown to be NP-complete, and the feasibility version is conjectured to be NP-complete, it is necessary to resort to an exponential-time exact algorithms for this problem. In this section, an integer-programming (IP) model is presented. This integer programming model is actually an extension of the basic RSP described in the previous section. It includes all of the educational constraints and teaching service constraints, and contains a number of additional side constraints that encompass ACGME regulations. Furthermore, it has additional variables that allow a day-to-day work schedule to be constructed simultaneously to the rotation schedule.

To create an integer program for the RSP, conditions (1)-(3) from Definition 1 can be taken as constraints for the program, and $f(r, t, p)$ can be treated as binary variables signifying whether resident r takes rotation t in rotation period p. If it is only necessary to determine a feasible schedule, then the objective function is irrelevant. In the optimization version, the objective function is given by

$$\min \sum_{r \in \mathcal{R}} Y_r \tag{1}$$

where Y_r is a binary variable for each resident that is 1 if and only if resident r has been assigned to at least one rotation.

Then, condition (2) needs to be replaced by the following two constraints in the IP formulation:

$$\sum_{p \in \mathcal{P}} f(r, t, p) \geq Y_r \cdot E(r, t) \ \forall \ r \in \mathcal{R}, t \in \mathcal{T} \tag{2}$$

$$\sum_{p \in \mathcal{P}} f(r, t, p) \leq Y_r \cdot |\mathcal{P}| \ \forall \ r \in \mathcal{R}, t \in \mathcal{T} \tag{3}$$

The first constraint above ensures that if resident r is assigned to a rotation, then all of his educational requirements will be satisfied. The second ensures that if he is not assigned to any rotation, then $f(r,\cdot,\cdot) = 0$. The objective function for the IP can then be taken to be (1).

Furthermore, to handle the additional ACGME requirements, binary variables $f(r,t,p)$ signify that resident r takes rotation t in rotation period p. In addition, binary decision variables $Z(r,s,l,p)$ indicate whether resident r takes shift s on day l in rotation period p. These variables are referred to as **shift variables**. The following list of constraints is the translation of all single-year conditions from (1a)-(3g) in Appendix B into constraints for the integer program formulation. The notation used in this model is described in Appendix A.

$$\sum_{p\in\{10,11,12,13\}} f(r,\mathrm{ER},p) \geq 1 \ \forall\ r \in \mathcal{R}2 \tag{4}$$

$$f(r,\mathrm{NF},p) + f(r,\mathrm{NF},(p+1)) \leq 1 \ \forall\ r \in \mathcal{R}, p \in \{1..12\} \tag{5}$$

$$Z(r,s,d,p) \leq f(r,s,p) \ \forall\ r \in \mathcal{R}, p \in \mathcal{P}, s \in \mathcal{S}, d \in DP \tag{6}$$

$$\sum_{r\in\mathcal{R}1\cup\mathcal{R}p} Z(r,s,d,p) \geq 1 \ \forall\ p \in \{\text{CAM, PAM}\}, \tag{7}$$

$$d \in \{1..5, 7..12, 14..19, 21..26, 28\} \tag{8}$$

$$\sum_{r\in\mathcal{R}1\cup\mathcal{R}p} Z(r,s,d,\mathrm{NF}) \geq 2 \tag{9}$$

$$\forall\ d \in \{1..5, 7..12, 14..19, 21..26, 28\} \tag{10}$$

$$\sum_{r\in\mathcal{R}2\cup\mathcal{R}3} Z(r,s,d,p) \geq 1 \ \forall\ p \in \{\text{CAM, PAM}\}, \tag{11}$$

$$d \in \{1..5, 7..12, 14..19, 21..26, 28\} \tag{12}$$

$$\sum_{r\in\mathcal{R}2\cup\mathcal{R}3} Z(r,s,d,\mathrm{NF}) \geq 2 \ \forall\ d \in \{1..5, 7..12, 14..19, 21..26, 28\} \tag{13}$$

Additionally, the full integer program for UIUC-COM includes all constraints specified in Tables 1 and 2.

While it may be very difficult to solve an integer program to optimality, the RSP is generally small enough that it can be solved quickly in practice. The following section demonstrates that a feasible solution for the residency program at UIUC-COM can be found in reasonable computation time.

Table 3: A running time comparison for the lower-classman problem

# prelims	# PGY1	CPU time (s)
1	17	0.05
5	14	0.08
20	20	0.24
50	50	0.61

5.1 Computational Performance for UIUC-COM

From the constraint set provided in Section 3.2, the assignment of PGY1 and prelim residents has no effect on the assignment of the PGY2/3 residents. Thus, the problem can be decomposed into two smaller integer programming problems, referred to as the **lower-classman** and **upper-classman** problems, respectively. If n_i ($i \in \{1, 2, 3\}$) is the number of i^{th}-year residents and n_p is the number of prelim residents, the total number of variables for the first integer program is $1352(n_p + n_1)$ and the total number of constraints is $1131 \cdot n_p + 1132 \cdot n_1 + 1027$. Similarly, the total number of variables for the upper-classman problem is $1365(n_2 + n_3)$ and the total number of constraints is $1144 \cdot n_2 + 1145 \cdot n_3 + 1027$.

However, note that the size of the integer program can be reduced substantially by disregarding the assignment to shift rotations. There are many different equally-good assignments of residents to shifts, and a valid assignment can be computed in a post-processing step. Ignoring the variables created by the shift rotations, the problem size reduces to $104(n_p + n_1)$ variables and $39 \cdot n_p + 40 \cdot n_1 + 91$ constraints for the lower-classman problem, and similarly for the upper-classman problem the problem size reduces to $117(n_2 + n_3)$ variables and $52 \cdot n_2 + 53 \cdot n_3 + 91$ constraints. The post-processing assignment of residents to shift rotations simply takes all of the available residents for the rotation, and rotates through which of them are assigned to the shift for that rotation.

To determine the computational results, the RSP was formulated as an integer program and solved using CPLEX 12.5 on an Intel Core i7-930 2.8GHz quad-core processor, with 12 GB of available memory. At UIUC in recent years, the upper bound on the number of PGY1, 2, and 3 is 14 residents each, with 5 prelim residents. The running time results for this case, and a few other hypothetical examples, are provided in Tables 3 and 4.

Table 4: A running time comparison for the upper-classman problem

# PGY2	# PGY3	CPU time (s)
8	9	0.05
14	14	0.18
25	25	0.11
50	50	0.24

5.2 Computational Performance for the General RSP

To further gain insight into the complexity of the feasibility version of RSP, a database of randomly-generated test problems was created, and the performance of CPLEX and the greedy heuristic in Section 4.3 was analyzed for these problems. There are four different classes of problems that were generated:

- Type 1 - Random: In this mode, the educational requirements and teaching service demands are generated uniformly at random. It is ensured that the number of educational requirements and teaching service demands does not exceed the maximum for each resident and period.

- Type 2 - Balanced Random: In this mode, the educational requirements for a particular resident and rotation are generated uniformly in the range $[0, \lceil |\mathcal{P}|/|\mathcal{T}| \rceil]$, and the teaching service demands are generated uniformly in the range $[0, \lceil |\mathcal{R}|/|\mathcal{T}| \rceil]$ (again, ensuring that the number of requirements is not too large). These problems have a more balanced structure to them, since the demands are spread out more evenly.

- Type 3 - Correlated Requirements: In this mode, the total number of teaching service demands for a particular rotation (across all periods) is set equal to the total number of educational requirements for that rotation (across all residents). This correlates the values for educational requirements and teaching service demands.

- Type 4 - Inversely Correlated Requirements: This mode is the opposite of Mode 3; it sets the total number of teaching service demands for a rotation (across all periods) equal to $\lceil |\mathcal{P}||\mathcal{R}|/|\mathcal{T}| \rceil - \sum_{r \in \mathcal{R}} E(r,t)$; this has the effect of negatively correlating the educational and teaching service demands for a rotation.

Table 5: A running time comparison for the randomly-generated database of RSP instances; times are averages of CPU seconds over 20 instances

Type	$\|\mathcal{R}\|$	$\|\mathcal{P}\|$	$\|\mathcal{T}\|$	CPLEX	GREEDY
1	50	20	50	0.06	0.01
2	50	20	50	3.15	0.01
3	50	20	50	2.69	0.01
4	50	20	50	1.25	0.01
1	100	40	100	0.50	0.01
2	100	40	100	210	0.06
3	100	40	100	193	0.05
4	100	40	100	176	0.04
1	200	60	200	3.11	0.07
2	200	60	200	3600	0.52
3	200	60	200	3600	0.52
4	200	60	200	3600	0.40

The problems in this database do not incorporate any additional constraints due to ACGME rules or otherwise. The feasibility version of these problems was solved using both CPLEX 12.5, as well as with the greedy algorithm from Section 4.3. The most challenging instances of the problem were found when the number of residents and the number of rotations were equal, and the number of periods was not too high. Therefore, 20 problems of each type were generated with the following parameters for $(|\mathcal{R}|, |\mathcal{P}|, |\mathcal{T}|)$: $(50, 20, 50), (100, 40, 100)$, and $(200, 60, 200)$. All experiments were given a time limit of 1 hour. The results from these experiments are provided in Table 5

A few observations can be made about these results. First, it was noted that no problems of Type 1 generated by our generator were actually feasible, despite passing the greedy algorithm's pre-check routine; further analysis shows that this is unsurprising, since it tends to generate extremely unbalanced distributions, which tend to be infeasible. CPLEX was able to discover this fact quite quickly in most cases. Secondly, the greedy algorithm is able to find feasible solutions for all generated instances except for the Type 1 instances, and in all cases it is able to find a feasible solution more quickly than CPLEX, usually by several orders of magnitude. This is primarily because the integer programs formulated for the RSP grows as the product of $|\mathcal{R}|$, $|\mathcal{P}|$, and $|\mathcal{T}|$, which means that CPLEX has to do substantially more work to solve the problems.

One might then ask the question, "Why not use the greedy algorithm for all version of the problem?" There are three reasons for this:

1. The greedy algorithm does not currently have any mechanism in place to handle the day-to-day scheduling of residents to rotations, whereas the integer program described above can also handle this problem.

2. For problems of relatively small size, it is desirable to use the integer programming model to produce a solution, since most integer programming software packages are easily able to produce many alternate optima. This is desirable for hospital schedulers so that additional qualitative metrics might be applied to determine the best schedule for the year.

3. For the greedy algorithm, it is not clear how to incorporate various objective functions into the solution, for instance to minimize the number of residents used, or to handle schedule perturbations.

5.3 Finding Alternate Optima

The minimal number of residents needed to maintain feasibility is the most important goal of the RSP, but another equally important question is how to determine alternate optima. It may be the case that there are additional (qualitative) concerns that the scheduler has when designing a schedule for the residents that cannot be incorporated into the integer programming model. In this case, it is desirable to be able to produce several different optimal solutions and allow the scheduler to choose from among them. Furthermore, determining the alternate optima allows sensitivity analysis to be performed to determine how susceptible the optimal solution is to schedule disruptions.

Since there are (potentially) an exponential number of optimal solutions, it is necessary to distinguish between two different types of alternate optima. The first type is the different combination of residents based on their seniority level. For example, in the lower-classman problem, it may be the case that x PGY1 and y preliminary residents produces an optimal solution; then, if y PGY1 residents and x preliminary yields a feasible schedule, it is also optimal. The second type of alternate optima is that there are different assignments within a fixed combination of residents. However, since the model assumes that each resident in the same seniority group is interchangeable, this type of alternative optima is not considered.

To find the first type of alternate optima, the objective function in Equation (1) can be modified so that each Y_r is multiplied by some weight value. Then, each assignment of weights to residents causes the resident to be more or less likely to be considered when compared to his or her peers. For example, assigning each preliminary resident weight one and each PGY1 resident weight zero will result in a schedule maximizing the number of PGY1 residents used, and will only use preliminary residents in the event that the PGY1 residents cannot satisfy all of the teaching service demands.

To fully characterize all of the first type of alternate optima, it is necessary to solve $\min(n_p, n_1)$ different integer programs, using distinct weights in $\{1, 2, ..., n_p + n_1\}$ for each resident. As before, assign weights $1, 2, ..., n_1$ to the PGY1 students; the integer program will prioritize them over the prelim students. Since all residents within a particular seniority level are indistinguishable, the only thing that matters for determining a schedule is the ranking of the weight of residents between different groups. Therefore, the weights can then be rotated cyclically, so that the PGY1 students have weight $2, 3, ..., n_1+1$, and the prelim residents are given weights $n_1+1, n_1+2, n_1+3, ..., n_1+n_p$. By rotating the weights $\min(n_p, n_1)$ times, all relative rankings will be considering, and all alternate optima will be found.

The alternate optima results for the residency problem at UIUC-COM are shown in Figure 2. The first graph in Figure 2 indicates that the minimum number of required PGY1/prelim residents needed to satisfy all requirements is 16; any combination of $n \geq 6$ prelims and $16 - n$ PGY1 residents can be assigned to a feasible schedule. If fewer than 6 prelim residents are available, the minimum number of required residents increases to 17. Similarly, the second graph in Figure 2 shows that as the number of PGY2 residents increases, the required number of PGY3 residents decreases, though not at the same rate. In particular, it can be seen that one fewer PGY3 resident is needed for every two additional PGY2 residents. Despite needing to solve a number of integer programs to determine these solutions, it only took 3.83 total CPU seconds to determine all of the first type of alternate optima.

6 Conclusions and Future Work

This paper solves a medical resident scheduling problem using an integer programming model that considers the teaching service demands, ACGME educational requirements, and duty hour rules. Firstly, it shows the minimization version of the RSP is NP-complete. Then, it presents an integer

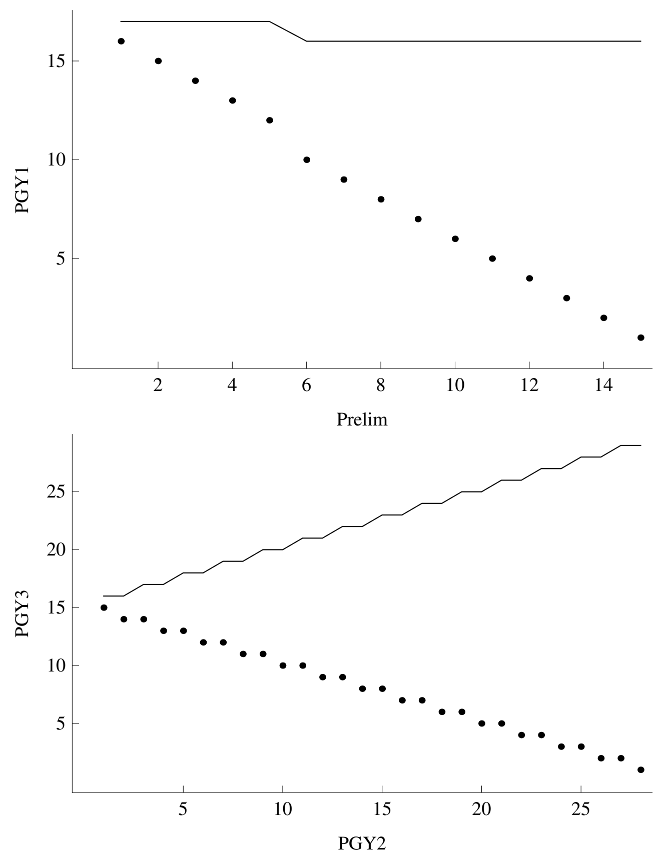

Figure 2: For a fixed number of Prelim/PGY2 residents, the minimum number of required PGY1/PGY3 students. The solid line shows the total number of minimum required residents.

program that solves the RSP exactly, and reports computational results demonstrating that the solution time is quick for problems of practical interest. Moreover, an analysis of the alternate optima shows that schedulers can produce several alternative feasible schedules that all use the minimal number of residents. Finally, it is shown how to create a new schedule in the event that one of the residents becomes sick or unable to complete his scheduled rotations.

Further work can be done on this problem; first, it is unknown whether the feasibility version of RSP is NP-complete. This is an important question, since in many cases hospitals work with a given set of residents; they do not get the option of only scheduling the minimum number. Therefore, an important next step for RSP is to resolve the status of Conjecture 1.

Secondly, it would be beneficial to be able to develop a multi-year schedule instead of the single-year schedule presented here. This is a particularly challenging problem, because residents enter and leave the residency program every year, which has the potential to offer significant disruptions to a long-term schedule, or even introduce infeasibilities into the schedule! Thus, the important research questions to be answered in this setting involve determining if a periodic, steady-state multi-year schedule can be developed, and if so, what the period is; it will also be important to understand how such a steady-state solution responds to schedule disruptions. A recent paper by Turner et al. (2013) gives a two-stage stochastic optimization procedure to address this issue at the operational level, but does not consider the assignment of residents to rotations at the block level.

Additionally, as the residency scheduling problem grows in size and difficulty, it is quite likely that the IP model presented in this paper will not be adequate to solve the problem efficiently. Most scheduling problems of a similar nature use a Dantzig-Wolfe decomposition to produce a formulation that yields tighter bounds, and it is likely that such an approach will be necessary here, as well. Thus, in order to address the more complex instances of this problem, it would be good to develop a branch-and-price algorithm based on such a decomposition.

Finally, as this problem is solved by hand in most hospitals, a user-friendly software package should be developed so that other residency programs besides the one at the University of Illinois College of Medicine at Urbana-Champaign can make use of these results. This will be a useful labor-saving contribution for residency programs across the country.

A List of Rotation Types and Additional Notation for UIUC-COM

Carle Foundation Hospital (CFH)

Provena Covenant Medical Center (PCMC)

Veteran's Administration Hospital (VA)

- CFH adult medicine (CAM)
- PCMC adult medicine (PAM)
- CFH/PCMC night float (NF)
- CFH critical care (CCC)
- VA Nephrology (VAN) *
- PCMC Gastroenterology (PG) *
- Carle Cardiology (CC) *
- VA Geriatrics (VAG) *
- VA Triage (VAT) *
- Carle Infectious Disease (CID) *
- Ambulatory (AMB) *
- Elective rotation for PGY2/3 residents (ER)
- Introduction to clinical research/vacation (ICR-VAC)

In the above list, starred problems indicate "backup rotations", that is, rotations that provide additional coverage in case of illness or other schedule disruptions. The following list provides the notation used in the model for the UIUC-COM residency program:

- $\mathcal{R}X, X \in \{p, 1, 2, 3\}$: the set of prelim, PGY1, PGY2, and PGY3 residents, respectively

Table 6: A list of working hours for all rotation types at UIUC-COM.

Rotation Type	Regular Hours	Shift Hours
CAM/PAM	7am-5pm	7am-7pm
NF	7pm-7am	7pm-7am
CCC	7am-5pm	7am-11am (next day)
Others	7am-5pm	N/A

- $n_X, X \in \{p, 1, 2, 3\}$: the number of prelim, PGY1, PGY2, and PGY3 residents, respectively
- $\mathcal{P} = \{1, ..., 13\}$: the set of 4-week rotation periods, ordered chronologically
- $\mathcal{T}$: the set of specific rotation types, listed in Appendix A
- $\mathcal{S}$: the set of shift rotations, that is, CAM, PAM, NF, and CCC. The hours for the shift rotations are provided in Table 6.
- $DP = \{1, ..., 28\}$: the days of a rotation period
- $D = \{1, ..., 7\}$: the days of the week

B Additional Requirements for UIUC-COM

The following list all of the additional side constraints present in the UIUC-COM residency scheduling problem (ACGME).

1. Additional Educational Requirements
   (a) PGY2 residents prefer to have elective in rotation periods 10-13. This model treats the preference as a hard constraint.
   (b) A resident cannot take NF for two consecutive rotation periods in each year; a resident cannot take more than two NF rotations in a year.
   (c) Every resident must take at least three and no more than six critical care rotations over three years. †
   (d) Residents must be assigned to emergency medicine rotations for at least four weeks of direct experience in blocks of not less than two weeks over the course of three years. †

2. Additional Teaching Service Demands

   (a) If a resident takes a shift, then he/she must take the rotation for that shift.

   (b) There must be at least one PGY1/prelim and one PGY2/3 covering the CAM/PAM/NF $\times$ 2 rotations 24 hours, 6 days a week. Note that the last day in a week is covered by resident hospitalists from the hospitalist rotations.

   (c) It is preferable to have one PGY2/3 providing coverage for CCC 24 hours, 7 days per week. This model treats the preference as a hard constraint.

   (d) PGY2/3 residents must be scheduled on a CCC shift no more frequently than every-third-night (when averaged over a four-week period).

3. ACGME Rules

   (a) Residents must not be scheduled for more than 80 hours per week, averaged over a 4-week period.

   (b) Residents must have at least one full (24 hours) day out of 7, free of patient care duties, averaged over 4 weeks.

   (c) Residents must have a minimum rest period of 10 hours between duty periods.

   (d) PGY1/Prelim residents cannot work more than 16 hours straight.

   (e) In house calls must occur no more frequently than every third night, averaged over a four-week period.

   (f) The residency program must provide opportunities for experience in geriatric medicine, neurology, psychiatry, allergy/immunology, dermatology, medical ophthalmology, office gynecology, otorhinolaryngology, non-operative orthopedics, palliative medicine, sleep medicine, and rehabilitation medicine in three years. †

   (g) The residency program must provide opportunities for experience in all internal medicine subspecialties; that is, cardiology, critical care, endocrinology, hematology, gastroenterology, infectious disease, nephrology, oncology, pulmonary disease, and rheumatology. †

Requirements marked with † are requirements that must be satisfied over the course of the three-year residency program, and thus are not considered by this model.

Note that the ACGME duty hour rules can be eliminated, since they can always be satisfied, given the additional teaching service demands (2b)-(2d). To see this, note that if a resident takes a rotation which is not a shift rotation, then the resident only works from 7am-5pm, Mon.-Fri. (see Table 6), which is a 50-hour work week. Moreover, he has two weekend days to rest. In this case, all conditions (3a)-(3e) are satisfied.

Additionally, if a resident is assigned to CAM, PAM or NF, in the worst case, he works from 7am-7pm 6 days in a week (by condition (2b)), which means he will work 72 hours in that week, but will have a full day of rest in between. Thus again, all ACGME rules are satisfied.

In the final case, if a PGY2/3 resident is assigned to the CCC shift rotation, then there must be at least three such residents assigned, by condition (2c) and (2d). Then, by rotating through each of the three residents, each resident gets one day of shift work from 7am to 11am (next day), followed a regular work day from 7am to 5pm on the third day, followed by another shift work day, and then take a rest on the sixth day. Therefore, the average working hour is 77 hours a week, which clearly satisfies conditions (3a)-(3e). Therefore, the ACGME rules are redundant for this model, and are not considered further.

C Acknowledgments

The authors would like to thank Chaitanya Are, a chief resident at UIUC-COM, and Tracey Johnson, program coordinator, for their valuable input regarding the residency program at the University of Illinois. The authors would additionally like to thank three anonymous referees whose suggestions resulted in a significantly improved version of this paper. The computational results reported were obtained at the Simulation and Optimization Laboratory at the University of Illinois, Urbana-Champaign. This research has been supported in part by the Air Force Office of Scientific Research (FA9550-10-1-0387), the Department of Defense (DoD) through the National Defense Science & Engineering Graduate Fellowship (NDSEG) Program, and the National Science Foundation through the Graduate Research Fellowship Program. Additionally, the third author was supported in part by (while serving at) the National Science Foundation. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Government.

References

ACGME. *Program Requirements for Graduate Medical Education in Internal Medicine*. Accreditation Council for Graduate Medical Education (ACGME), 2009. URL `http://www.acgme.org/acwebsite/rrc_140/140_prindex.asp`. accessed Mar. 2012.

J. Beliën and E. Demeulemeester. Heuristic branch-and-price for building long term trainee schedules. https://lirias.kuleuven.be/handle/123456789/85433, 2004.

J. Beliën and E. Demeulemeester. Scheduling trainees at a hospital department using a branch-and-price approach. *European Journal of Operational Research*, 175(1):258–278, Nov 2006.

F. Bellanti, G. Carello, and R. Tadei. A greedy-based neighborhood search approach to a nurse rostering problem. *European Journal of Operational Research*, 153(1):28–40, Feb 2004.

J. Brunner and G. Edenharter. Long term staff scheduling of physicians with different experience levels in hospitals using column generation. *Health Care Management Science*, 14(2):189–202, Apr. 2011.

A. Cohn, S. Root, C. Kymissis, J. Esses, and N. Westmoreland. Scheduling medical residents at boston university school of medicine. *Interfaces*, 39(3):186–195, May 2009.

T. Day, J. Napoli, and P. Kuo. Scheduling the resident 80-hour work week: An operations research algorithm. *Current Surgery*, 63(2):136–141, Mar.-Apr. 2006.

L. Franz and J. Miller. Scheduling medical residents to rotations: Solving the large-scale multiperiod staff assignment problem. *Operations Research*, 41(2):269–279, Mar.-Apr. 1993.

M. Garey and D. Johnson. *Computers and Intractability A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, New York, 1979.

K. Gibert and R. Hofstra. Multidimensional assignment problems. *Decision Sciences*, 19(2):306–321, Jun. 1988.

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

B. Maenhout and M. Vanhoucke. Branching strategies in a branch-and-price approach for a multiple objective nurse scheduling problem. *Journal of Scheduling*, 13(1):77–93, Mar. 2009.

J. Miller and L. Franz. A binary-rounding heuristic for multi-period variable-task-duration assignment problems. *Computers and Operations Research*, 23(8):819–828, Aug. 1996.

T. Osogami and H. Imai. Classificaion of various neighborhood operations for the nurse scheduling problem. *IBM TRL Research Report*, RT0373, Aug. 2000.

I. Ozkarahan. A scheduling model for hospital residents. *Journal of Medical Systems*, 18(5):251–265, 1994.

S. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity.* Dover Publications, Inc. Mineola, New York, 1998.

S. Topaloglu. A multi-objective programming model for scheduling emergency medicine residents. *Computers and Industrial Engineering*, 51(3):375–388, Nov. 2006.

S. Topaloglu. A shift scheduling model for employees with different seniority levels and an application in healthcare. *European Journal of Operational Research*, 198(3):943–957, Nov. 2009.

S. Topaloglu and I. Ozkarahan. A constraint programming-based solution approach for medical resident scheduling problems. *Computers and Operations Research*, 38(1):246–255, Jan. 2011.

J. Turner, K. Kim, S. Mehrotra, D. DaRosa, M. Daskin, and H. Rodriguez. Using optimization models to demonstrate the need for structural changes in training programs for surgical medical residents. *Health Care Management Science*, 16(3):217–227, Sep 2013.

C-W. Wang, L-M. Sun, M. Jin, C. Fu, L. Liu, C-H. Chan, and C. Kao. A genetic algorithm for resident physician scheduling problem. In *8th Annual Conference on Genetic and Evolutionary Computation*, 2007.

D. West. *Introduction to Graph Theory.* Prentice-Hall, Inc. Upper Saddle River, NJ, 2001.