Smart Scheduling: A Pilot Project of Workforce Scheduling in Radiation Oncology

ABSTRACT

Introduction: In the case of the radiation oncology department, the large number of visits faced by patients illuminates the critical need for optimal time management. Focused on three central themes: daily waiting times, diagnosis-to-treatment waiting times, and appropriate staffing for the present workload, the research highlights the impact of inefficient time management on patient satisfaction and overall operational efficiency. The time and energy invested in a schedule are high and frequently many scheduling conflicts occur even after the schedule is made. The ability to schedule different employees in the most optimal manner would increase the productivity of the radiation oncology department.

Methods: The scheduling software was constructed using Python language and importations of libraries from the Tkinker software for the Graphical User Interface. The software is a constraint-based algorithm that allocates staff to different sites based on each radiation therapy clinic’s staffing requirements.

Results and Discussion: This work developed a basic software that creates a randomized schedule of employees. While this would benefit the team by curating a schedule that has no functional mistakes, the algorithm provides a foundation for the data collection that will facilitate the future incorporation of artificial intelligence (AI). This would allow for deeper learning overtime of the software to develop a schedule that is optimal for the success of the individual and, thus, the entire team. This pilot project aimed to generate interest regarding the introduction of AI to current scheduling software in the context of the radiation oncology department.
INTRODUCTION

The large number of visits faced by patients with a cancer diagnosis underscores the critical need for optimal time management in the field of radiation oncology. Effective time management not only impacts the success of clinical treatments but also plays a pivotal role in shaping patient satisfaction and determining staffing requirements for managing workloads. Patients undergoing radiation therapy (RT) are commonly required to visit a RT clinic several times, spanning from days to weeks, in order to complete their treatment; daily waiting times are a vital factor to consider for patient satisfaction in this cohort as small reductions in waiting times have a cumulative impact on overall treatment experience [1–3]. In addition to daily waiting times, additional factors including the number of visits before treatment begins contribute to patient satisfaction [4, 5]. Further investigation of patient satisfaction reveals close links with staffing requirements, emphasizing the importance of aligning human resources with the present workload [6, 7].

Unsurprisingly, several studies have demonstrated higher patient satisfaction rates with shorter daily wait times among cancer patients [1]. According to Famiglietti et al., [8] daily waiting times had the second highest influence on overall patient satisfaction for patients in radiation oncology. Daily waiting times exert greater strain on these patients due to the cumulative factors of commuting to an RT center, total time spent in the clinic, and time spent in treatment. Gupta et al highlights work done by Lim et al, analyzing the time spent in healthcare of patients with pancreatic adenocarcinoma and concluded a median total of 36 days for a curative surgical treatment. Patients prescribed with adjuvant RT had an addition 30 days, almost doubling the total time spent receiving treatment for their cancer diagnosis [9, 10]. Longer waiting times have been significantly correlated with increased patient-reported pain scores [1]. Another study reported an escalation of emotional distress and anxiety levels during time spent in radiation therapy clinic waiting rooms [2].

The waiting time between initial cancer diagnosis to treatment time is another crucial factor to acknowledge while addressing patient satisfaction and clinical success. A survey conducted by Paul Ba Hons et al. [4 (p321)] reported that while 52% of patients experienced concern during each phase of their diagnosis and treatment journey, a higher rate of patients reported concern from diagnosis to initial radiotherapy (31%) compared to other waiting times. The initial anxiety caused by a cancer diagnosis is only augmented by the wait for definitive treatment, further decreasing patient satisfaction and resulting in a poorer quality of life [11, 5]. Studies have observed a strong association between delay in cancer treatment and increased mortality [12–14].

While numerous factors are associated with a patient’s timely access to care such as geographical distance from care, health insurance, and accessibility of treatment machines [15], factors internal to the radiation oncology team should be addressed. Radiation therapy treatment is a product of several steps, including diagnosis, consultation, simulation, and the treatment planning process. These steps are prepared and delivered by a collaborative group each with varying roles and responsibilities and temporal constraints. Medical physicists are specialists in research and training in radiation treatment techniques and carry a wide range of responsibilities within the department, ranging from treatment planning to patient chart checks to quality assurance [17]. The radiation oncology team members utilize treatment planning software to cultivate a treatment plan unique to each patient; the process can take up to a week or more [16]. Delays in treatment planning can lead to rescheduling of appointments and increased waiting time. By addressing practical components of the department, such as optimizing RT center workflow, the patients’ time and care are prioritized and unnecessary burdens can be minimized.

A growing demand for radiotherapy requires a proportional increase in human resource utilization [6]. To conduct safe treatments, Ahn et al. [7] demonstrated “a need for systematic arrangements concerning staffing requirements and manpower, as well as for management of treatment machines.” A lack of a proper employee base is a negative factor in both risk management and flow of patients’ treatments. Third et al. [18] evaluated the increasing demand for RT between 2012 and 2020, reporting a 45% increase in patients compared with the growth of human resource utilization of a 29% increase in staffing levels (medical dosimetrists, physicians, and medical physicists). These results demonstrate the importance of proper staffing in accordance with growing workloads to decrease burnout and compassion fatigue. In addition to regular maintenance of the growing radiotherapy equipment, high-dose radiation treatment plans such as stereotactic radiosurgery and stereotactic body RT, often
require a physician and a physicist to be on site during treatment. Without the appropriate staff present, treatment must be rescheduled leading to decreased patient satisfaction and increased burden. Few studies have explored different methods to optimize staffing requirements for the present workload in radiation oncology [19–21]. Therefore, the purpose of this project is to address the present issues of employee scheduling by developing basic software to allocate different human resources (dosimetrists, physicists, and physicians) to different sites at Stony Brook University Hospital. Stony Brook University Hospital has over 200+ locations throughout Long Island, with two main locations for RT treatment. Oncologists and physicists travel between these two sites depending on patient load and treatment type.

Optimization algorithms are already an important subset of many artificial intelligence (AI) algorithms, facilitating the production of optimal solutions within a set of constraints. When optimization algorithms are combined with other AI algorithms like reinforcement learning (RL), their ability to determine predictive suggestions based on identified patterns and interaction with feedback is enhanced. RL learns optimal decision-making strategies through interaction and feedback with an environment [21–24]. RL is currently used to solve problems in optimization, scheduling, and resource sharing [24, 25]. Ipek et al. proposed an RL-based approach to solve the limitations of conventional approaches for the efficiency and performance of scheduling decisions of controllers which ultimately yielded higher performance levels [25]. Furthermore, Che et al. proposed a deep reinforcement learning (DRL) algorithm to address the multi-resource scheduling problem for the traffic of cellular networks which resulted in greater convergence speeds compared to conventional RL [24]. Various integrations of RL to address the downfall of conventional methods for efficiency demonstrate the versatility and potential of RL in optimizing complex systems.

The application of RL within these industrial sectors provides a strong basis for the transition into clinical and administrative applications in radiation oncology. Factors like increased workloads, treatment specific staff requirements, site-specific considerations, staff constraints and preferences and many more perpetuate challenges within the department. The predictive capability would contribute to a more adaptive scheduling methodology by considering the multifactorial challenges of patient daily waiting times, diagnosis-to-treatment planning, and employee satisfaction.

**METHODS**

The purpose of the developed platform is to create a graphical user interface (GUI) application to generate employee schedules based on the individual constraints. Individual constraints include unavailability on certain days throughout the work week that may vary from week to week. For example, a physicist may be unable to work Thursday and an oncologist cannot come in on Friday. This variability can cause delays in treatment if not properly reported and human resources are not properly allocated for patient treatments occurring that week. In order to ease implementation and consideration of HIPAA compliance, names of employees and patients will not be used in the beta version and schedules will be formed using initials and broad descriptions of treatment.

Work schedules are created utilizing the Tkinter library in Python and visualized in Visual Studio Code [see Supplementary File 1]. Stony Brook University Hospital IT team will check and monitor the code running on a 64-bit Windows 10 Enterprise computer with Xeon Silver 4110 CPU processor and 32 GB memory. After importing the necessary libraries: tkinter, itertools, and random, dictionaries were created to store each employee and their constraints labeled as ‘employee_constraints.’ There are a series of functions that all contribute to the final product. The function “add_employee_and_constraint()” takes the employee’s name and constraint from the entry fields on the GUI and adds them to the dictionary. After the employee is added to the dictionary, two buttons, random day off and remove constraint, will appear in the left frame for the specific employee. This provides the user with flexibility to adjust constraints. The display field is then updated with the current employees and their constraints with the “update_employee_list()” function. The script also allows the user to specify the number of locations and the number of people required at each location to increase the usability for different radiation oncology departments that have more than one location performing radiotherapy.

The “generate_schedule()” function will take the user inputs about the employees, constraints, and the number of people needed at each location and the function “generate_employee_schedule()” provides the algorithm to generate the employee schedules. If this function is pressed again, it will continue to shuffle the employees while considering constraints. A “clear_employee_list()” is
also added to the application to clear the dictionary while updating the displayed dictionary for user visibility. Finally, the code includes GUI components such as entry fields, labels and buttons. Variable buttons like “random day off for:” and “remove constraint for:” are added dynamically as additional employee inputs are added. All of these functions act simultaneously in order to present the following output: a GUI application to generate employee schedules based on each individual’s specific constraints and the constraints of each location. The flow chart depicts the sequential order of input from the user and how it is taken into the various functions, previously explained, which ultimately leads to the custom randomized schedule, as shown in Figure 1.

The ability of the script to generate randomized schedules based on a simple constraint function provides the foundation for further advancements. Through the implementation and run of the beta version, data collection will include but are not limited to workload (measured in number of cases), employee self-reported satisfaction, patient satisfaction (through survey), work efficiency (number of scheduling conflicts), and treatment planning efficiency (number of postpones). This data will allow the integration of optimization formulas, as previously discussed, into the scheduling algorithm to elevate its efficiency.

RESULTS

The output of the script is a GUI created using the Tkinter library, which allows users to interact with the employee scheduling application. There are several components and functions that appear when the script is run. On the left frame, there are input fields for specifications on the number of locations and, for each location, how many individuals are required at each location.
Below is an input field for employee names and constraints, which the user will utilize to specify specific constraints for the employee’s name that is entered. The button “Add Employee and Constraint” will add the employee and their constraints to the text box below for the user to visualize, as shown in Figure 2.

As employees and their constraints are added, “Random Day Off for ——” and “Remove Constraint for ——” buttons will appear dynamically for each employee, as shown in Figure 3a. This will allow the user to change up the constraints for each employee without having to use the “Clear List” button which resets the dictionary of each employee and their constraints. After the user has inputted all the values indicated in the left frame, a schedule (with all inputted constraints) will be generated in the right frame upon pressing the “Generate Schedule” button, as shown in Figure 3d. The output contains information on each day of the week, assigning employees to separate locations as needed. While this output follows all constraints, employee preferences can be taken into consideration at this point as the “Generate Schedule” button can be pressed several times if the output is not preferred initially. For instance, while each generated schedule meets all individual constraints, an employee may be available Friday but prefers a schedule that allows them to work Monday instead. This shuffling of schedules allows the manager to select the ideal schedule to increase staff satisfaction. The GUI is a functional and user-friendly tool for generating employee schedules with practical features such as dynamic constraint adjustments, visibility of employee constraints, and flexibility of schedule generation.

Figure 2 Graphical User Interface after code is run on computer. Left frame displays input fields for number of locations, number of people required (Personnel Needs) at each location (L 1, L 2), employee name, and constraint. One button to add employee and constraint and another to clear the list. There is a display field to preview employee(s) and their constraints. The right frame displays button to generate the schedule and a display field.

Figure 3 A–D. Displays the GUI of the software during operation. A. Values of the number of locations and personnel requirements at each site are inputted in the upper left of the frame. B. Staff initials and corresponding constraints are inputted and options for each staff’s constraints are visible below the display box. C. Final list of staff and constraints are demonstrated on the left frame. D. Right frame previews the display field with the resulting randomly generated schedule based on the variables and constraints shown on the left frame.
DISCUSSION AND CONCLUSION

This project delves into the importance of time management in the radiation oncology department, through the lens of patient satisfaction and operational efficiency. Exploration into the daily waiting times of patients, diagnosis-to-treatment waiting times, and proper staffing of employees highlight the need of strategic interventions such as the implementation of a scheduling script to address internal staffing requirements. Effortless integration into day-to-day procedures in any department is necessary and accomplished as the script is constructed utilizing the Tkinter library in Python, a tool on many, if not all, computers. Initially, the script will only need to be run on the manager’s computer terminal to provide the schedule for the whole department. The script provides a practical and user-friendly interface for generating schedules. The dynamic nature of the software allows for adjustments to individual constraints, another vital factor to consider during integration to be able to increase flexibility and satisfaction of the healthcare staff. While the current script serves as a valuable tool for generating schedules, it represents the initial step in a journey towards operational efficiency in the radiation oncology department. The practicality of the current script acts as a fundamental element for ongoing improvements as the urgency of proper time management, patient satisfaction, and departmental success comes to light.

While this current software focuses on constraint-based scheduling, it paves the way for future utilization of optimization formulas and AI. The first step in optimization is defining an objective function, representing the goal to be achieved, examples of such are: minimizing overtime, maximizing employee preferences, or balancing workload across locations. The establishment of constraints and decision variables will allow the optimization algorithm to find the best combination of decision variables to maximize (or minimize) the objective function. Decision variables are defined through the data collected using this constraint-based scheduling. By incorporating optimization formulas, the script would go beyond basic scheduling constraints to create a more efficient schedule that would adhere to and align with organizational goals and the wellbeing of employees.

Further incorporation of AI can elevate the optimization role of scheduling by analyzing previous data to identify patterns and ultimately predict future scheduling needs. AI models can utilize previous data to adapt to employee preferences which will increase job satisfaction.

ADDITIONAL FILE

The additional file for this article can be found as follows:

- Supplementary File 1. Smart Scheduling Code. DOI: https://doi.org/10.29024/jsim.202.s1

COMPETING INTERESTS

The authors have no competing interests to declare.

AUTHOR AFFILIATIONS

Jennifer Chen  
orcid.org/0009-0002-8604-1040  
Stony Brook University, Stony Brook, NY, US

Xin Qian  
Department of Radiation Oncology, Stony Brook University Hospital, Stony Brook, NY, US

REFERENCES


12. Hanna TP, King WD, Thibodeau S, et al. Mortality due to cancer treatment delay: Systematic review and meta-analysis. BMJ. Published online November 4, 2020; mko87. DOI: https://doi.org/10.1136/ bmj.mko87


TO CITE THIS ARTICLE:


Submitted: 16 March 2024
Accepted: 13 May 2024
Published: 06 August 2024

COPYRIGHT:

© 2024 The Author(s). This is an open-access article distributed under the terms of the Creative Commons Attribution 4.0 International License (CC-BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited. See http://creativecommons.org/licenses/by/4.0/.