RAD-SMART: Rule-Guaranteed Scheduling with Live Re-planning to Reduce Waiting in a Single-Machine Radiotherapy Department

Akshay Dinesan, Abhinand T M, P S Raghotham Rao, Shirley Lewis Salins, Umesh Velu

Health-a-thon 2026, Cancer track · Manipal · 24 October 2026

Abstract

Background. Radiotherapy patients attend every weekday for one to seven weeks, so a single linear accelerator treats 70–90 patients a day. Appointment times are usually given by hand, without knowing how long each session takes. In our department, one patient in ten waits more than 109 minutes from arrival to treatment, and 22% are treated more than an hour after their appointment.

Methods. RAD-SMART is an assistive scheduling system with four steps. It predicts each session's machine-minutes, plans the day with a mixed-integer program that cannot break the department's hard rules, tests the plan in a digital twin, and tells patients in their own language, by a translated message and a voice note, through a page on their phone. Each course gets one standing time slot for its whole duration. When the machine breaks down, the rest of the day is re-planned under the department's own downtime rules. A person approves every change. We evaluated it on two datasets, kept apart: the department's 55 recorded days (October–December 2024), re-planned in a twin calibrated on those records, and synthetic patients, including 12 weeks of courses and two machine breakdowns.

Results. On the department's own days, the 90th-percentile wait fell from 108.6 to 70.4 minutes (−35%), and the share treated more than an hour late from 22.2% to 5.7%. The busiest days gained most (130 to 70 minutes). Over 12 synthetic weeks with standing slots, the 90th-percentile wait fell from 81.6 to 49.3 minutes (−40%). After a 150-minute breakdown, live re-planning cut the median wait from 104 to 39 minutes, with a new plan in 14 seconds. An independent checker found no broken hard rule. All 112 translated patient messages, in four Indian languages, kept every appointment time. The costs were a later evening, a short delay for one new patient in five and fewer requested times kept.

Conclusion. Rule-guaranteed optimisation with live re-planning can remove most long waits in a single-machine department without new machines. A 90-day pilot will test this on real days.

Keywords: radiotherapy; scheduling; patient waiting time; mixed-integer programming; digital twin; machine downtime; India

1. Introduction

Radiotherapy is not a normal outpatient clinic. A curative course lasts five to seven weeks and a palliative course one to two, with a session every weekday. One machine therefore carries 70–90 patients a day. It must also absorb new patients, urgent palliative starts and long, complex procedures. A routine session takes 7–15 minutes of machine time, while a new start or a complex case takes 20–60.

In India the pressure is high: 69% of radiotherapy facilities have only one machine [1]. When times are given by hand, too many patients are asked to come at once. They wait, usually with a family member, and they wait every day of the course. Audits at Indian centres report mean daily waits of around 37 minutes even with digital check-in [2].

Operations research has a long record in radiotherapy, but mostly for choosing the *day* a patient starts treatment, in large, multi-machine, fully digital centres [3–6]. Models that predict session length work well enough to

schedule by the minute [7]. Newer approaches, such as reinforcement learning [8] and large language models used directly as schedulers [9], cannot give the guarantee a department needs for rules such as "no new patient after 17:00".

We asked a simpler question: in a one-machine Indian department, can a scheduler that guarantees the department's rules, gives each course a fixed slot and re-plans live after a breakdown cut the long waits patients remember? This paper describes RAD-SMART and reports its results on the department's own records and on synthetic patients.

2. Setting and data

The department. One Elekta Versa HD runs from 08:30 until about 01:00, in three shifts, with an operational ceiling of 90 patients a day. Most patients speak Kannada or Tulu. A second machine arrives in 2027.

The records. The department shared anonymised arrival and treatment records for 55 working days, 15 October to 31 December 2024: 3,137 patient-days for 263 patients. After removing missing and impossible entries, 2,685 waits remain. We used them only as aggregates. They show:

- a median wait of 34 minutes and a 90th percentile of 109 minutes;
- 22% of patients treated more than an hour after their appointment, and only 29% within 15 minutes of it;
- waits that build through the day, from a median of 19 minutes for patients arriving before 10:00 to 66 minutes for those arriving after 19:00;
- a rising load, from 54 patients a weekday in October to 67 in December, with worse waits on busy days (a 90th percentile of 130 minutes on days with 70 or more patients).

The rules. The department confirmed its rules in writing on 21 September and 24 October 2026 (Table 1). They are stored as configuration that the department can change.

Table 1. The department's hard rules, which a plan may never break.

Rule	Detail
New starts	Finish by 17:00
Complex cases (SBRT, SRT, TBI, CSI)	Between 10:00 and 17:00, when senior staff are present
Blood irradiation	The machine is reserved 13:30–14:00
Hospice (MHRC) patients	Treated 16:00–17:00, in time for the 17:00 bus
Public transport	Patients who depend on it finish by 21:00
Shared accessories	One breast board and one breath-hold (ABC) unit, shared with the CT simulator
Urgent starts	Treated by 18:00
Standing slot	Each course keeps one time slot until it ends

Soft priorities are traded off: preferred times for paying patients, earlier times for older patients, and late-evening slots for inpatients, hostel residents and people living nearby.

3. Methods

3.1 Overview

RAD-SMART works in four steps: **predict** each session's minutes, **plan** the day, **test** the plan in a digital twin, and **tell** staff and patients. It decides only *when* and *on which machine* a session takes place, never whether or how a patient is treated. Every output is a proposal: the doctor approves each course and its slot, and a technologist or

doctor approves the day and any breakdown response. A language model is used only to explain decisions and draft messages. It never schedules, and it never sees names or phone numbers.

3.2 Predicting session minutes

Gradient-boosted regressors predict the expected, median and cautious (80th-percentile) minutes of each session from tags already on the treatment card: technique, first fraction or not, imaging, mobility, accessories and machine. Because the department does not yet record session durations, RAD-SMART starts from the department's own estimate table and learns from measured times once the pilot records them.

3.3 Planning the day

The day is a grid of 2-minute steps. A mixed-integer program places each session on an eligible machine at a start time. Every rule in Table 1 is a hard constraint. The objective penalises unplaced sessions most heavily, and also overtime, late finishes for patients who are not flexible, and distance from each patient's preferred time. To solve a 70–90-patient day quickly, it works coarse-to-fine: it first levels the load in 15-minute buckets, then places new starts, complex cases and hospice patients exactly, then fits everyone else around them. We used the open-source HiGHS solver [10].

A separate function, independent of the solver, re-checks every plan against the hard rules before anyone sees it.

3.4 Standing slots

The department keeps each patient's time for the whole course. When the doctor approves a course, RAD-SMART proposes two things:

1. a **new-start slot** on the first day, finishing before 17:00; and
2. one **standing slot** for every later session: a time that is free on every treatment date of the course, keeps every hard rule, and leaves 150 minutes before 17:00 free for future new patients.

Among the free times it picks the one with the lowest soft cost. A patient's preferred part of the day (morning 08:30–12:00, afternoon to 17:00, evening to 21:00), when stated, is a soft goal that replaces "older patients earlier". If a patient asks for another time, RAD-SMART proposes the nearest time that keeps every rule. As courses end and slots free up, a *better-time* pass offers patients outside their part of the day a standing slot inside it, free on every remaining date and keeping the new-start reserve; offers are ranked by preference weight × fractions left × minutes outside, each offered slot is held so no two offers collide, and staff agree any change with the patient. On days built from standing slots, the checker adds a ninth rule: every patient is in their slot.

3.5 Breakdown re-planning

A breakdown lasts from 15 minutes to 48 hours. The technologist enters the expected downtime, and RAD-SMART applies the department's rules:

- **Up to 60 minutes:** nobody is sent home, and everyone is treated the same day with up to 60 minutes of overtime.
- **Over 60 minutes:** RAD-SMART sends back only as many patients as are needed to free the lost machine time, choosing first those who have not left home. A gap-sensitive course is never sent back without the oncologist. The missed session is added at the end of the course.
- **Always:** new starts wait for the machine and still finish by 17:00, or move to the next working day. Patients already waiting are treated first. Patients at least 45 minutes from their time are messaged not to travel until they get a new time.

The re-plan freezes what has started and re-solves the rest of the day, penalising extra waiting for patients already in the department more heavily than moving patients who are still at home. Nothing reaches a patient until a technologist or doctor approves.

3.6 Digital twin and evaluation

The twin is a discrete-event simulation of the treatment day. It includes lognormal session times, 2% no-shows, zero to three urgent starts a day, imaging holds, and patients arriving early or late as the records show for each hour of the day. "Today's booking" gives each patient an appointment drawn from the department's recorded spread of appointment times, treated first-come-first-served. "RAD-SMART" uses the optimised plan, with patients asked to report 20 minutes before their slot (45 minutes for pelvic patients, who must drink water first). Both policies face the same patients and the same random events (common random numbers), so the comparison is fair.

Validation. Replaying today's booking over all 55 recorded days, the twin gives a 90th-percentile wait of 106.1 minutes against 108.6 recorded, and 24.2% treated more than an hour late against 22.2%. It under-predicts long waits on quieter days and over-predicts them on the busiest. Today's figures are therefore always quoted from the records.

Two datasets, kept apart.

- **Original dataset:** the 55 recorded days, in four groups by size. A day of each size is re-planned by RAD-SMART in the twin, and the result is weighted by the number of days. "Without" is what was recorded.
- **Simulated dataset:** synthetic patients under the department's rules. (a) 12 weeks of courses: new patients arrive every working day and are booked both ways from the same start date; after seven weeks, when courses fully overlap, 20 working days are played 15 times each. (b) One busy 87-patient day, simulated 300 times. (c) The same day with a 45-minute breakdown at 11:00 and a 150-minute breakdown at 10:00.

Measures. The primary measure is the 90th-percentile wait from arrival to entering the treatment room: the wait that one patient in ten exceeds. Secondary measures are the median wait, the share treated within 15 minutes of the appointment, the share treated more than an hour late, and costs such as overtime and when the last patient finishes.

3.7 Implementation

The engine is about 3,700 lines of Python, using scikit-learn, NumPy and SciPy with HiGHS. It also runs inside CARE, the open-source hospital information system of the Open Healthcare Network, as a backend plugin (Django) and a frontend plugin (React), with no change to CARE's own code. The whole application is packaged as one Docker image built on CARE's production image, and runs live on Microsoft Azure (Container Apps with a managed PostgreSQL database), where anyone can try it with synthetic patients. The doctor starts from a **RADSMART** button on the patient's visit page, enters the course and approves the proposed slots. The technologist runs the day from a day board that records arrival, room entry and exit, and machine status, and declares breakdowns there.

Patients are told through a page on their own phone, opened from one private link sent on WhatsApp, with no app and no login. It shows the next treatment times, the department's messages with a voice note, and a QR code that reception scans to check the patient in; four buttons let the patient say they are on the way, late, cannot come, or want another time, and staff decide what to do. After every approval, RAD-SMART drafts a message for each patient affected, from fixed templates in English. Sarvam Translate puts it into the patient's language (Kannada, Malayalam, Hindi or Konkani; Sarvam has no Tulu model, so Tulu speakers get Kannada, which they read) and Sarvam Bulbul reads it aloud; a technologist reviews it and presses send in WhatsApp. A translation is used only if it keeps exactly the clock times of the English text; otherwise the English is sent. On request, Sarvam-105B explains a proposed slot in plain words from the engine's own reasons, and an answer that mentions any time not in the plan is discarded. Only the message text reaches these services, never a name, hospital number or phone number.

4. Results

4.1 Original dataset: the department's own days

RAD-SMART reduced the long wait on every size of day (Table 2, Figure 1). The busier the day, the larger the gain: with today's booking the long wait grows with the size of the day, while with RAD-SMART it stays near 70 minutes. The median wait moved from 34 to 30 minutes, and the share treated within 15 minutes of the appointment rose from 28.9% to 49.5%.

Table 2. Original dataset: recorded (without) and re-planned by RAD-SMART in the twin (with).

Day size (days)	90th-percentile wait, without → with	More than 1 hour late, without → with
Up to 44 patients (7)	79 → **62 min** (−21%)	14.5% → **4.7%**
45–59 patients (20)	98 → **68 min** (−31%)	21.8% → **6.1%**
60–69 patients (18)	110 → **70 min** (−36%)	20.8% → **5.8%**
70 or more patients (10)	130 → **70 min** (−46%)	27.4% → **5.5%**
All 55 days	108.6 → **70.4 min** (−35%)	22.2% → **5.7%**

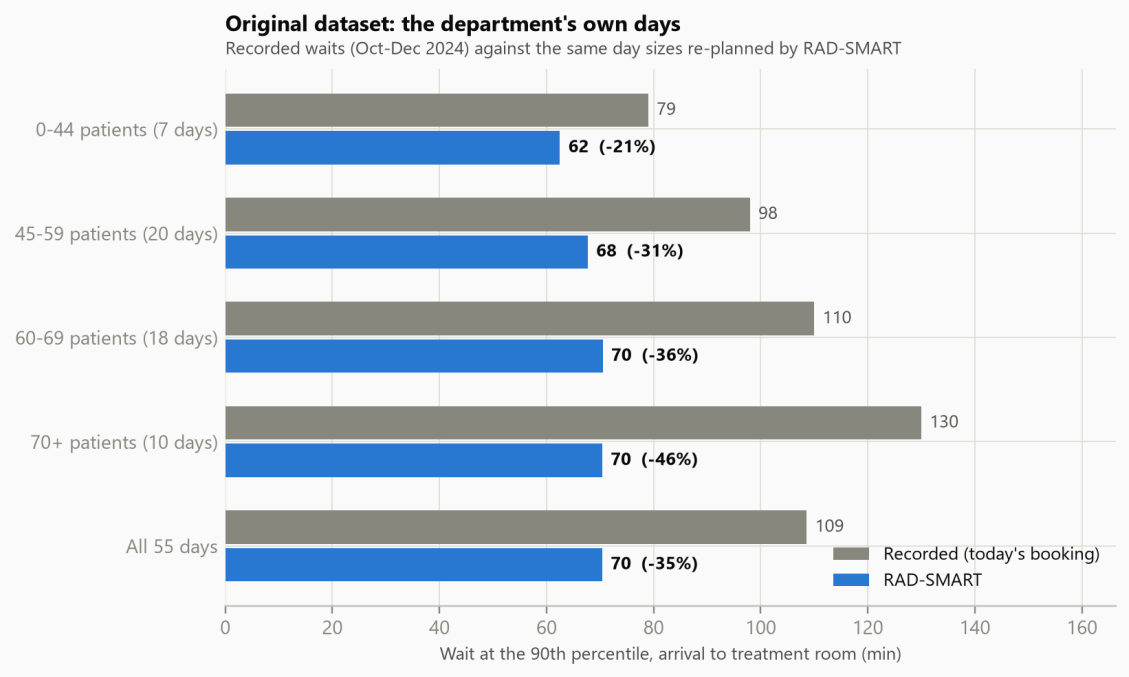


Figure 1. Original dataset: 90th-percentile wait by size of day. Grey: recorded. Blue: the same day sizes re-planned by RAD-SMART.

4.2 Simulated dataset: synthetic patients

Table 3 and Figure 2 give the results. The 12-week test is the closest to practice, because each course keeps its slot from start to finish. No hard rule was broken on any of the 20 evaluation days.

Table 3. Simulated dataset: today's booking (without) against RAD-SMART (with), same patients and events.

Scenario	Median wait	90th-percentile wait	More than 1 hour late
12 weeks with standing slots (about 63 patients a day)	40 → **20 min**	82 → **49 min** (−40%)	19% → **4%**
One busy day, 87 patients	133 → **36 min**	210 → **76 min** (−64%)	66% → **6%**
45-minute breakdown at 11:00	177 → **37 min**	257 → **81 min** (−69%)	74% → **8%**
150-minute breakdown at 10:00	280 → **39 min**	358 → **110 min** (−69%)	83% → **14%**

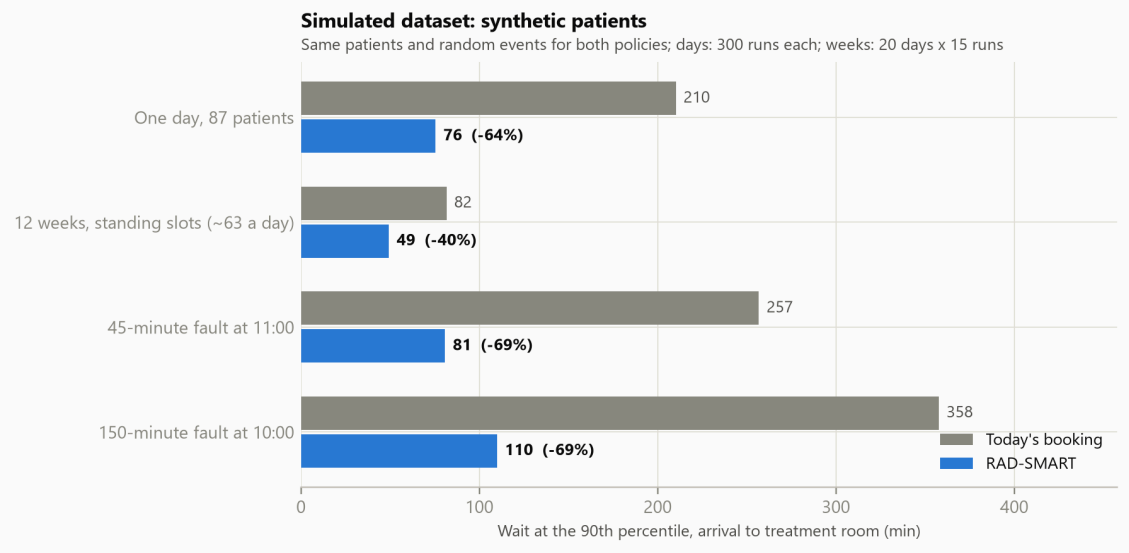


Figure 2. Simulated dataset: 90th-percentile wait under today's booking (grey) and RAD-SMART (blue).

The 87-patient day is ten patients more than the busiest recorded day. The twin's projection of today's booking at that load (210 minutes) is pessimistic; against the busiest recorded days (130 minutes), RAD-SMART's 76 minutes is still 42% lower.

4.3 Breakdowns

The fair test of live re-planning compares the same RAD-SMART plan with and without the re-plan (Table 4).

Table 4. Breakdowns: the RAD-SMART plan kept unchanged against live re-planning (300 simulated days each).

	45 minutes at 11:00	150 minutes at 10:00
Median wait, kept → re-planned	52.6 → **37.3 min**	104.4 → **39.4 min**
90th-percentile wait, kept → re-planned	102.2 → **80.8 min**	210.1 → **110.0 min**
Time to re-plan	10.1 s	13.9 s
Patients sent back	0	1 (had not left home)
New starts moved to the next day	1	2
Overtime on the day	8 min	50 min

Under the up-to-60-minute rule nobody was sent home. Under the over-60-minute rule, one patient who had not yet travelled was sent back, and everyone else was treated the same day.

4.4 Prediction and computation

On 2,000 held-out synthetic sessions, gradient boosting predicted session length with a mean absolute error of 2.15 minutes, against 2.62 for a table of averages. The 87-patient day (about 4,800 variables and 1,662 constraints) was solved to a proven optimum in 7.1 seconds with predicted minutes and 30.9 seconds with the table of averages, on a laptop. With averages alone, the 90th-percentile wait was 82 minutes rather than 76, so most of the benefit is available before any session times are recorded.

4.5 Patient messages

We tested the language step on synthetic text, with the app's own settings. Seven message templates (a new course, a pelvic course, a new standing time and four breakdown messages), each with four sets of times, were translated into all four languages. All 112 translations kept every time, so none would have fallen back to English; the median was 1.3–2.0 seconds. Translating back into English and comparing with the source (chrF, 0–100) gave medians of 76–80 by language, with the lowest scores on the pelvic message, which native speakers will check

first. For 20 synthetic plans, Sarvam-105B gave an explanation the time check accepted every time, with no invented time; 16 stated both arrival times. Back-translation is only a rough check of meaning: native speakers at the department will rate every template before the pilot.

4.6 Costs

Over the 12 weeks, standing slots had clear costs:

- **A later evening.** The last patient finished at about 00:07 instead of 21:55, still inside the department's hours, because patients are spread out and time before 17:00 is kept for new starts.
- **Some new starts wait.** 21.9% of new patients started later than with today's booking, by 0.42 working days on average.
- **Fewer requested times kept.** Patients who asked for a particular time were treated within 15 minutes of it 3.9% of the time, against 31.4%. Today they are given that time but then wait in the queue.
- **Overtime** of 8 minutes a day on average.

5. Discussion

Principal findings. On both datasets RAD-SMART removed most of the long waits: a 35% lower 90th-percentile wait on the department's own days and 40% lower over 12 synthetic weeks, with lateness of over an hour falling to about 5%. The typical wait changed little, because patients are still asked to arrive early to prepare. What changes is the long, unpredictable wait, and the build-up of delay through the day. After a breakdown, a re-plan in seconds kept the day close to normal for nearly everyone.

Why it works. Today's booking gives many patients the same time and lets the queue absorb the difference. RAD-SMART books by predicted minutes, so the machine's day is full but not over-full, and it keeps the rules as constraints rather than hoping they hold. This matches results from Dutch and Belgian centres, where optimisation made appointment times far more regular [3, 5], but at the scale of minutes within one day and on a single machine.

Trade-offs are the department's to choose. Standing slots move patients into the evening, delay some new starts by less than half a day, and make requested times harder to give. These are settings: the new-start reserve, the urgent holds and the weights on preferred times can all be changed, and the department decides where to stand.

Limitations.

- **Replay, not a trial.** The original-dataset result comes from the twin, which reproduces today's long wait closely overall but not on every day size. It re-plans days of the recorded sizes; it does not re-run the exact patients, whose session lengths were never recorded.
- **Synthetic patients.** Session lengths and the patient mix are estimates. The real gain depends on how much of today's waiting is caused by scheduling rather than by transport, machine checks or paperwork.
- **Patient behaviour.** The benefit assumes that patients stop arriving very early once times are kept. If they keep today's habits, the 90th-percentile wait on the busy day is 110 minutes rather than 76, still below the busiest recorded days.
- **One machine.** Live sequencing was tested on one machine; two machines were tested only for the three-week capacity forecast.
- **Messages not yet rated by patients.** The translations were checked automatically and by back-translation, not yet by native speakers or patients.

Next step. A 90-day pilot on the Versa HD will run in three phases: two weeks of baseline measurement, three weeks in shadow mode, and eight weeks live. The primary target, a 90th-percentile wait at least 25% below the recorded 109 minutes, is set below the results above. If a hard rule is broken in practice, the department returns to manual scheduling.

6. Conclusion

A single radiotherapy machine does not need to mean long, unpredictable waits. Predicting session minutes, planning with guaranteed rules, giving each course a fixed slot and re-planning in seconds after a breakdown cut the long wait by about a third on the department's own days and by 40% over 12 synthetic weeks, at a cost the department can see and adjust. Because RAD-SMART runs inside an open-source hospital system and keeps a person in charge of every change, it can be piloted without new machines or a new hospital system.

Ethics and data. The prototype uses synthetic patients only. The department's records were anonymised before sharing and are reported only as aggregates; no row-level data is published. RAD-SMART is assistive and not diagnostic. The pilot will go to the Institutional Ethics Committee for review as a quality-improvement study before it begins.

Code availability. The engine, the CARE plugins and the scripts that produce every number and figure in this paper are at github.com/psraghothamrao8/RAD-SMART (`poc/run_poc.py`). The full research report gives every detail.

Author roles. Akshay Dinesan: team leader and clinical lead. Abhinand T M: technical lead. P S Raghotham Rao: lead engineer. Shirley Lewis Salins and Umesh Velu: clinical partners.

References

1. An analysis of radiotherapy machine requirements in India: impact of the pandemic and regional disparities. *J Med Phys.* 2024. PubMed 39526149.
2. Munshi A, Krishnakutty S, Sarkar B, Ganesh T, Mohanti BK. Daily waiting and treatment times at an advanced radiation oncology setup: a 4-year audit of consecutive patients from single institution. *J Cancer Res Ther.* 2021;17(2):523–529.
3. Vieira B, Demirtas D, van de Kamer JB, Hans EW, Jongste W, van Harten W. Radiotherapy treatment scheduling: implementing operations research into clinical practice. *PLoS ONE.* 2021;16(2):e0247428.
4. Vieira B, Hans EW, van Vliet-Vroegindeweij C, et al. Operations research for resource planning and -use in radiotherapy: a literature review. *BMC Med Inform Decis Mak.* 2016;16:149.
5. Frimodig S, Mercier C, De Kerf G. Automated radiation therapy patient scheduling: a case study at a Belgian hospital. arXiv:2303.12494, 2023.
6. Pham TS, Legrain A, De Causmaecker P, Rousseau LM. A prediction-based approach for online dynamic appointment scheduling: a case study in radiotherapy treatment. *INFORMS J Comput.* 2023;35(4):844–868.
7. Xie et al. Machine learning-based radiotherapy time prediction and treatment scheduling management. *J Appl Clin Med Phys.* 2023.
8. An online algorithm for integrated scheduling of pre-treatment and treatment appointments in radiotherapy using deep reinforcement learning. *Health Care Manag Sci.* 2026.
9. Large language models for AI-assisted radiotherapy scheduling: a feasibility study under realistic operational constraints. arXiv:2605.12896, 2026.
10. HiGHS: high-performance open-source linear optimisation software. highs.dev.