# The Integrated Healthcare Timetabling Competition 2024
# — Problem description and rules —

Sara Ceschia[1], Roberto Maria Rosati[1], Andrea Schaerf[1], Pieter Smet[2], Greet Vanden Berghe[2], and Eugenia Zanazzo[1]

[1] Polytechnic Department of Engineering and Architecture, University of Udine, Italy
`{sara.ceschia,robertomaria.rosati,andrea.schaerf,eugenia.zanazzo}@uniud.it`
[2] Department of Computer Science, KU Leuven, Gent, Belgium
`{greet.vanden.berghe,pieter.smet}@kuleuven.be`

**Abstract.** Research in timetabling often focuses on clear-cut academic problems. Real-world healthcare optimization problems, however, encompass additional challenges due to various decision and optimization problems being intertwined. Moreover, general timetabling methodologies are not necessarily suitable for addressing such integrated problems. In the interest of stimulating research on the specifics of integrated scheduling problems in healthcare, this paper introduces the Integrated Healthcare Timetabling Competition 2024 We begin by describing the problem formulation, which integrates three critical problems in healthcare: surgical case planning, patient admission scheduling and nurse-to-room assignment. Next, we discuss the data sets and file formats, along with the solution checker (validator) that we provide for the participants. Finally, we state the rules of the competition and explain how participants will be ranked. All up-to-date information concerning the competition is available at the competition's website `https://ihtc2024.github.io`.

**Keywords:** Healthcare · Integrated optimization problem · Competition.

## 1 Introduction

Integrated healthcare scheduling deals with the coordination of resources related to various services within a single healthcare system. It aims to streamline and optimize the flow of patients across different departments and facilities of the hospital. The benefits of integrated healthcare optimization are manifold: enhanced patient experience, improved operational efficiency, and optimized resource utilization across the entire hospital system.

Contributions to integrated healthcare applications have been surveyed by Rachuba et al. [6]. They identify three levels of increasing integration, ranging from solving a single problem while incorporating the constraints coming from the other problems (level 1), to sequentially solving two or more problems using the output of one problem as input for the next one (level 2), and finally to simultaneously solving two or more problems at once in a single stage (level 3).

A recent proposal for level 3 integration by Brandt et al. [1] addresses the simultaneous resolution of two operational problems that are critical in hospitals: the Patient-to-Room Assignment (PRA) and the Nurse-to-Patient Assignment (NPA) problems.

The Integrated Healthcare Timetabling Competition 2024 (IHTC) revises the problem introduced by Brandt et al. and generalizes it by incorporating a third important optimization problem in hospitals, namely Surgical Case Planning (SCP)[7]. The resulting integrated problem, which we call the *Integrated Healthcare Timetabling Problem* (IHTP), brings together three $\mathcal{NP}$-hard problems and requires the following decisions: (*i*) the admission date for each patient (or admission postponement to the next scheduling period), (*ii*) the room for each admitted patient for the duration of their stay, (*iii*) the nurse for each room during each shift of the scheduling period, and (*iv*) the operating theater (OT) for each admitted patient.

The IHTP is subject to many hard and soft constraints. Some of these constraints relate to a specific subproblem, while others arise from their interactions. The IHTP is a special case of real-world timetabling at hospitals, which are often subject to additional constraints. Furthermore, we consider the *static, deterministic* variant of the IHTC, in which all information for a fixed scheduling period is known at the time of solving.

The remainder of this paper is organized as follows. Section 2 provides the problem definition. Section 3 introduces the datasets and the validator made available to participants for evaluating their solutions. Finally, Section 4 describes the rules of the competition. Appendix A is also included as supplementary material, which describes the file formats. All up-to-date information concerning the competition is available at the competition's website `https://ihtc2024.github.io`.

## 2 Problem definition

After first introducing the basic concepts of the IHTP, we will define the hard and soft constraints of the problem and explain how they must be evaluated throughout the entire scheduling horizon.

### 2.1 Basic concepts

We begin by introducing the time horizon and physical resources involved in the IHTP:

**Scheduling period:** The scheduling period is defined as a number $D$ of consecutive days. $D$ is always a multiple of seven, and can vary from 14 (two weeks) to 28 (four weeks).

**Shifts:** A shift denotes a nurse's working period during a day. We assume three non-overlapping shifts per day: *early*, *late*, and *night*. The entire scheduling period thus consists of $3D$ shifts. Each shift is denoted by an integer ranging from 0 to $3D-1$. The early, late and night shifts on the first day are numbered 0, 1, and 2, respectively. For the second day, the shifts are numbered 3, 4, and 5. This pattern continues until the end of the scheduling period.

**Operating theaters:** All OTs are identical in that they are suitable for accommodating any type of surgery. Each OT has a daily maximum capacity, expressed in minutes. Some OTs might be unavailable on specific days, indicated by a maximum capacity of 0 minutes on those days.

**Rooms:** Rooms host the patients during their recovery. These rooms are characterized by their capacity, expressed in terms of the number of beds. Room equipment is not explicitly taken into account. However, as will be outlined in what follows, some rooms might be declared unsuitable for some patients.

Next, we describe the human resources that are involved in the IHTP:

**Nurses:** Each nurse has a skill level. Levels are strictly ordered (hierarchical) and represented by an integer that ranges from 0 (lowest) to $L-1$ (highest), where $L$ is the number of skill levels. Furthermore, each nurse has a predetermined roster, which is defined as a set of shifts that the nurse has been assigned to, along with the maximum workload the nurse can accommodate in each shift. This roster is fixed and cannot be changed.

**Surgeons:** Each surgeon has a maximum operating time on each day, which is 0 when the surgeon is unavailable on that day. If a surgeon is available, we assume their surgical team is also available. In other words, the surgeon and their team form an atomic indivisible resource (called *surgeon* for simplicity).

Note that the maximum nurse workload is shift-dependent as nurses can carry out auxiliary activities during some specific working shifts, thereby reducing their availability. Also note that we assume an *open scheduling policy* [3], which means that all surgeons can operate in all OTs.

The patient is the central entity of the problem. The following information is provided for each patient:

- **mandatory/optional**: mandatory patients must be admitted during the scheduling period, while the admission of optional patients can be postponed until a future scheduling period.
- **release date**: earliest possible admission date for the patient.
- **due date**: latest possible admission date, provided only for mandatory patients.
- **age group**: the age group of the patient (e.g., infant, youth, adult, elder). The list of age groups is fully ordered.
- **gender**: the gender of the patient.
- **length of stay**: duration of the hospitalization in days.

- **incompatible rooms**: set of rooms that must not be allocated to the patient because, for example, they do not have the specific equipment or the necessary isolation.
- **surgery duration**: the expected duration of the patient's surgery, which is assumed to always take place on the day of admission.
- **surgeon**: the surgeon who carries out the patient's surgery.
- **workload**: the workload profile generated by the patient, which is described by a vector, starts at the early shift of the admission day and ends at the night shift of the discharge day. The length of the vector equals 3 times the patient's length of stay.
- **minimum skill level**: the minimum nurse skill level required by the patient for each shift they are staying in the hospital; described by a vector similar to the patient workload vector.

Note that both the workload and the minimum skill level required for a patient can vary based on the shift and how long the patient has been in the hospital, as these factors are related to the patient's treatments and stage of recovery. Both values are usually lower during night shifts and higher during the initial days of the stay.

### 2.2 Solution

The solution of an IHTP instance consists of the following decisions:

i. the admission date for patients, or, in the case of optional patients, potentially their postponement to the next scheduling period;
ii. the allocation of a room for each admitted patient;
iii. the assignment of a single nurse to each occupied room, for each shift within the scheduling period;
iv. the assignment of patient surgeries to OTs, for each day of the scheduling period.

We assume that patients are always admitted and discharged after the night shift and before the early shift. Note that a patient stays in only one room during the entire length of their stay, meaning a patient cannot be transferred from one room to another.

We also assume that all patients undergo surgery, and that this takes place on the day of admission. In addition, as each patient's surgeon is predetermined, the day of admission automatically determines the total surgery time of each surgeon on each day. By contrast, the OT must be selected. This assignment does not include the precise operating time, only the date. The IHTP does not consider the order of surgeries in an OT.

Finally, note that it is necessary to assign a nurse to a room on a given shift only if that room contains patients on the day to which the shift belongs. Nevertheless, assigning nurses to empty rooms would be feasible and does not incur additional costs.

### 2.3 Constraints

We divide the constraints into four sets: (i) those related to the PAS problem, (ii) those related to the NRA problem, (iii) those related related to the SCP problem, and finally (iv) those related to the integration of the three problems. In addition, constraints are categorized as either hard (starting with **H**) or soft (starting with **S**). The former must always be satisfied, while the latter contribute to the objective function. Violations of soft constraint $\mathbf{S}_i$ are multiplied by weight $W_i$. Note that the soft constraint weights are instance-specific and thus given in each input file.

#### Constraints on Patient Admission Scheduling

**H1** No gender mix: Patients of different genders may not share a room on any day.
**H2** Compatible rooms: Patients can only be assigned to one of their compatible rooms.
**S1** Age groups: For each day of the scheduling period and for each room, the maximum difference between age groups of patients sharing the room should be minimized.

#### Constraints on Nurse-to-Room Assignment

While the IHTP does not explicitly require the assignment of nurses to patients, the combination of patient-to-room assignments and nurse-to-room assignments determines which nurses are responsible for which patients. The following constraints (**S2**, **S3**, and **S4**) depend on the resulting nurse-patient assignment.

**S2** Minimum skill level: The minimum skill level a nurse must have to provide the required care for a patient during each shift of their stay should be met. If the skill level of the nurse assigned to a patient's room in a shift does not reach the minimum level required by that patient, a penalty is incurred equal to the difference between the two skill levels. Note that a nurse with a skill level greater than the minimum required can be assigned to the room at no additional cost.
**S3** Continuity of care: To ensure continuity of care, the total number of distinct nurses providing care to a patient during their entire stay should be minimized. The given rosters assume maximum one shift per day for each nurse, hence the number of different nurses who take care of a patient is at least 3.
**S4** Maximum workload: For each shift, the total workload induced by patients staying in rooms assigned to a nurse should not exceed the maximum workload of that nurse in that shift.

**Constraints on Surgical Case Planning**

**H3** Surgeon overtime: The maximum daily surgery time of a surgeon must not be exceeded.
**H4** OT overtime: The duration of all surgeries allocated to an OT on a day must not exceed its maximum capacity.
**S5** Open OTs: The number of OTs opened on each day should be minimized. Note that if an OT has no patients assigned for a particular day, it should not open on that day.
**S6** Surgeon transfer: The number of different OTs a surgeon is assigned to per working day should be minimized.

**Global constraints**

**H5** Mandatory versus optional patients: All mandatory patients must be admitted within the scheduling period, whereas optional patients may be postponed to future scheduling periods.
**H6** Admission day: A patient can be admitted on any day from their release date to their due date. Given that optional patients do not have a due date, they can be admitted on any day after their release date.
**S7** Admission delay: The number of days between a patient's release date and their actual date of admission should be minimized.
**S8** Unscheduled patients: The number of optional patients who are not admitted in the current scheduling period should be minimized.

### 2.4 Boundary data

We assume that some patients are already present in the hospital on the first day of the scheduling period. We use the term *occupants* to refer to these special patients. While these occupants contribute to the occupancy of the rooms and to all related constraints, their admission date and room assignment are fixed. Occupants do not contribute to the OTs' occupancy because their surgery occurred during the preceding scheduling period.

For patients admitted during the current scheduling period and who stay after the end of it, no penalties are incurred after the end of the horizon.

## 3 Datasets and validator

Problem instances are supplied as JSON files following the structure outlined in Appendix A. Each instance is contained within a single file. We provide a *public* dataset composed of 30 instances, named `i01`, . . . `i30`, with a scheduling period ranging from two to four weeks and a number of patients ranging from approximately 50 to 500. In addition, we provide five instances, `test01`, . . . , `test05`, for testing and debugging purposes. We also provide a solution for each test instance. We will employ a different *hidden* dataset to evaluate the participants' submissions. This dataset will be shared with the participants at the end of the competition. Both the *public* and *hidden* datasets are generated using the same instance generator, which utilizes realistic patterns and distributions.

Generated solutions must be saved as JSON files adhering to the format described in Appendix A. The validator, which certifies the feasibility and quality of a given solution, is provided as a C++ source code and should be compiled using, for example, the GNU compiler g++. The validator receives the instance and solution files as command line parameters, as demonstrated in the following example.

```
> ./IHTP_Validator.exe input_file.json sol_file.json
```

The command line output of the validator appears as follows:

```
VIOLATIONS:
RoomGenderMix....................0
PatientRoomCompatibility.........0
SurgeonOvertime..................0
OperatingTheaterOvertime.........0
MandatoryUnscheduledPatients.....0
AdmissionDay.....................0
RoomCapacity.....................0
NursePresence....................0
UncoveredRoom....................0
Total violations = 0

COSTS (weight X cost):
RoomAgeMix............................5 (  5 X   1)
RoomSkillLevel.......................21 (  1 X  21)
ContinuityOfCare.....................43 (  1 X  43)
ExcessiveNurseWorkload................0 (  1 X   0)
OpenOperatingRoom...................100 ( 50 X   2)
SurgeonTransfer.......................0 (  5 X   0)
PatientDelay.........................50 ( 10 X   5)
ElectiveUnscheduledPatients...........0 (300 X   0)
Total cost = 219
```

If `verbose` is added as a third parameter, the details of each single cost element are also printed:

```
Room r0 is age-mixed 1/2 in day 1
Nurse n5 is underqualified for occupant a1 in room r0 in shift 3 (day1@early)
Nurse n6 is underqualified for patient p5 in room r0 in shift 4 (day1@late)
...
6 distinct nurses for occupant a0
4 distinct nurses for occupant a1
...
Operating theater t0 is open on day 1
Operating theater t0 is open on day 4
```

```
Patient p0 has been delayed for 1 days
Patient p1 has been delayed for 2 days
...
```

## 4 Competition rules

This competition seeks to encourage research into automated timetabling and scheduling methods for solving an integrated healthcare problem, with prizes offered for the most successful methods. As with any set of rules for any competition it is possible to work within the letter of rules but outside their spirit. We, as organizers, ask all participants to respect both the letter and spirit of these rules. Failing to do so will result in disqualification.

**Rule 1:** We reserve the right to update the rules at any time if they believe it is necessary for the sake of ensuring the correct operation of the competition. Any change of rules will be notified in the repository.

**Rule 2:** The competition has deadlines concerning when all submissions must be uploaded. These deadlines are strict and no extensions will be given under any circumstances.

**Rule 3:** Participants may use any programming language. The use of third-party software is allowed under the following restrictions:
- either it is open source (`https://opensource.org/osd`) or it provides a free, unlimited academic license;
- its behavior is (reasonably) documented;
- it runs under a commonly-used operating system (Unix/Linux, Windows, or MAC OS).

**Rule 4:** The solution method should take as input a file in the format described, and produce as output a solution file in the correct format. The algorithm must stop after 10 minutes wall time. Parallel computing is allowed, using up to 4 threads.

**Rule 5:** The solution method may be either deterministic or stochastic. In both cases, participants must be prepared to show that the results are repeatable within the given computational time. In particular, participants using a stochastic algorithm should do their utmost best to code their program in such a way that the run producing each submitted solution can be replicated by reusing the same random seed.

**Rule 6:** Participants must submit (i) solutions for all instances from the public dataset and (ii) a clear and concise description of their algorithm before the first competition deadline. A set of 5 finalists will be determined by ranking the participants on each public instance.

If the first 5 finalists all use licensed software, the number of finalists will be increased to 6 by adding the best-ranked solver using only open-source software.

An infeasible or missing solution will equate to the last position in the ranking for that particular instance. The mean average of the ranks across all instances will produce the participant ordering, of which the first 5 are then selected as finalists. Section 4.2 provides additional details on how the ordering will be established.

**Rule 7:** We will rerun the finalists' solution methods on the hidden dataset using the same time limit specified in Rule 4. The official PC will be a AMD Ryzen Threadripper PRO 3975WX, 3.50 GHz, running Ubuntu Linux 22.4. A different operating environment might be used in exceptional cases if necessary. It is the responsibility of competition participants to ensure that all files and information needed to run their code is provided to us.

**Rule 8:** The final ranking of the finalists will be based on the ranks obtained for each instance for a set of trials on hidden instances. Section 4.2 provides an explanation of the procedures to be used.

### 4.1 Dates

The competition will be announced at different conferences during the summer of 2024, including PATAT 2024 and ORAHS 2024. The competition will then officially begin on September 1, 2024. On this date, we will release the public dataset, the specifications, and the validator. The deadline for submission of participants' best solutions and a description of their solution method is March 1, 2025. Notifications of the finalists will be sent out on April 1, 2025. The winners will be announced at the EURO 2025 conference in Leeds, UK (June 22-25, 2025).

### 4.2 Adjudication procedure

We follow the same adjudication procedure used in the First and Second International Nurse Rostering Competitions (INRC-I, INRC-II) [4,2], which was originally imported from the Second International Timetabling Competition (ITC-2007) [5]. The procedure is repeated here for the sake of completeness.

Let $m$ be the total number of problem instances and $k$ the number of participants who produce a solution for all $m$ instances. Let $X_{ij}$ be the result supplied (and verified) by participant $i$ for instance $j$. Each $X_{ij}$ is the value of the objective function $s$, for participant $i$ on instance $j$. In case participant $i$ is unable to provide a feasible solution for instance $j$, $X_{ij}$ is assigned a conventional value $M$ larger than the result supplied by any other participant for that instance.

The matrix $X$ of results is transformed into a matrix of ranks $R$ by assigning to each $R_{ij}$ a value from 1 to $k$. That is, for instance $j$ the supplied $X_{1j}$, $X_{2j}$, $\ldots$,$X_{kj}$ are compared with each other and rank 1 is assigned to the smallest value observed, rank 2 to the second smallest, and so on to rank $k$, which is assigned to the largest value for instance $j$. Ranks are assigned for all the instances. We use average ranks in case of ties. If a solution method produces an infeasible solution, it will be assigned the highest rank for the corresponding instance. The rule of average ranks for tie-breaking is not applied in case of infeasibility: solution methods that generate infeasible solutions or fail to generate solutions at all are assigned rank $k$ for the corresponding instance, in which k is the total number of participating solution methods.

Consider the example with $m = 6$ instances and $k = 7$ participants in Table 1. Table 2 shows the ranks.

**Table 1.** An example of submitted solution scores.

| Instance | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| Solution method 1 | 34 | 35 | 42 | 32 | 10 | 12 |
| Solution method 2 | 32 | 24 | 44 | 33 | 13 | 15 |
| Solution method 3 | 33 | 36 | 30 | 12 | 10 | 17 |
| Solution method 4 | 36 | 32 | 46 | 32 | 12 | 13 |
| Solution method 5 | 37 | 30 | 43 | 29 | 9 | 4 |
| Solution method 6 | 68 | 29 | 41 | 55 | 10 | 5 |
| Solution method 7 | 36 | 30 | 43 | 58 | 10 | 4 |

**Table 2.** Corresponding solution ranks for the example.

| Instance | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| Solution method 1 | 3 | 6 | 3 | 3.5 | 3.5 | 4 |
| Solution method 2 | 1 | 1 | 6 | 5 | 7 | 6 |
| Solution method 3 | 2 | 7 | 1 | 1 | 3.5 | 7 |
| Solution method 4 | 4.5 | 5 | 7 | 3.5 | 6 | 5 |
| Solution method 5 | 6 | 3.5 | 4.5 | 2 | 1 | 1.5 |
| Solution method 6 | 7 | 2 | 2 | 6 | 3.5 | 3 |
| Solution method 7 | 4.5 | 3.5 | 4.5 | 7 | 3.5 | 1.5 |

We define for each solution method the mean of the ranks. The finalists of the competition will be the 5 solution methods with the lowest mean ranks. In case of a tie for the last position, all tying methods will be included in the final (in this case the number of finalists will be more than 5). Table 3 shows the mean ranks for the example.

**Table 3.** Mean ranks.

| | |
|---|---|
| Solution method 1 | 3.83 |
| Solution method 2 | 4.33 |
| Solution method 3 | 3.58 |
| Solution method 4 | 5.17 |
| Solution method 5 | 3.08 |
| Solution method 6 | 3.92 |
| Solution method 7 | 4.08 |

In this case, the finalists would be solution methods 1, 3, 5, 6 and 7.

During the final phase of the competition, the evaluation process is repeated for the finalists with the following new elements:

1. The hidden dataset will be used.
2. We will run the solution methods of the finalists. We expect the finalists to offer support in the process of compiling and running their solution method.
3. For each problem instance, we will run 10 independent trials with random seeds. For each trial, we will compute the ranks and average them over all trials on all instances.

The winner is the participant with the lowest mean rank. In case of a tie, an additional trial will be run for all instances until a single winner is found.

### 4.3 Prizes

The top three teams will receive a cash prize (first prize €1100, second €700, third €400), and be offered one non-transferable free registration to EURO 2025, which will host a special track dedicated to the competition.

The best open-source finalist will receive a special prize of €200, which can be awarded in addition to the regular prize.

**Acknowledgements.** The group at University of Udine has been funded by the project "Models and algorithms for the optimization of integrated healthcare management" (no. 2020LNEZYC) supported by the Italian Ministry of University and Research (MUR) under the PRIN-2020 program. This research was also supported by the project "Human-centred decision support based on a new theory for personnel rostering", KU Leuven-IOF (C2E/23/012).

## References

1. Brandt, T., Klein, T.L., Reuter-Oppermann, M., Schäfer, F., Thielen, C., van de Vrugt, M., Viana, J.: Integrated patient-to-room and nurse-to-patient assignment in hospital wards. arXiv preprint 2309.10739 (2023)
2. Ceschia, S., Dang, N., De Causmaecker, P., Haspeslagh, S., Schaerf, A.: The second International Nurse Rostering Competition. Annals of Operations Research **274**(1), 171–186 (2019)
3. Guerriero, F., Guido, R.: Operational research in the management of the operating theatre: a survey. Health care management science **14**, 89–114 (2011)
4. Haspeslagh, S., De Causmaecker, P., Schaerf, A., Stølevik, M.: The first International Nurse Rostering Competition 2010. Annals of Operations Research **218**, 221–236 (2014)
5. McCollum, B., Schaerf, A., Paechter, B., McMullan, P., Lewis, R., Parkes, A.J., Di Gaspero, L., Qu, R., Burke, E.K.: Setting the research agenda in automated timetabling: The second International Timetabling Competition. INFORMS Journal on Computing **22**(1), 120–130 (2010)
6. Rachuba, S., Reuter-Oppermann, M., Thielen, C.: Integrated planning in hospitals: A review. arXiv preprint 2307.05258 (2023)

7. Zhu, S., Fan, W., Yang, S., Pei, J., Pardalos, P.M.: Operating room planning and surgical case scheduling: a review of literature. Journal of Combinatorial Optimization **37**, 757–805 (2019)

## Appendix A - File formats

Input and solution files are written in JSON. The input file contains a header part in addition to separate sections for nurses, rooms, operating theaters, surgeons, patients, and occupants. What follows is an example of the header part, containing the general data and the weights of the cost components.

```
{
  "days": 28,
  "skill_levels": 3,
  "shift_types": [
    "early",
    "late",
    "night"
  ],
  "age_groups": [
    "infant",
    "adult",
    "elderly"
  ],
  "weights": {
    "room_mixed_age": 5,
    "room_nurse_skill": 10,
    "continuity_of_care": 5,
    "nurse_eccessive_workload": 10,
    "open_operating_theater": 20,
    "surgeon_transfer": 1,
    "patient_delay": 5,
    "unscheduled_optional": 350
  }
 ...
}
```

What follows is a fragment of an example for the section about nurses. For each nurse, we have a unique identifier (id), the skill level and a list of working shifts with their respective maximum workloads.

```
"nurses": [
    {
      "id": "n00",
      "skill_level": 0,
      "working_shifts": [
        {
          "day": 0,
          "shift": "early",
          "max_load": 10
```

```
      },
      {
        "day": 1,
        "shift": "night",
        "max_load": 5
      },
      ...
  ]},
  ...
  ]
```

The structure of the sections concerning rooms, OTs, and surgeons is straightforward and shown in the following fragment.

```
"surgeons": [
   {
     "id": "s0",
     "max_surgery_time": [0, 360, 0, 600, 480, 0, 0, 600, ...]
   },
   ...
 ],
 "operating_theaters": [
   {
     "id": "t0",
     "availability": [0, 600, 720, 600, 600, 720, 720, ...]
   },
   ...
 ],
 "rooms": [
   {
     "id": "r0",
     "capacity": 2
   },
   {
     "id": "r1",
     "capacity": 3
   },
  ....
  ]
```

Finally, we introduce the structure of the patient data, divided into occupants (present at the beginning of the scheduling period) and regular patients.

```
"occupants": [
   {
     "id": "a0",
     "gender": "B",
```

```
      "age_group": "elderly",
      "length_of_stay": 2,
      "workload_produced": [2, 1, 1, 2, 3, 2],
      "skill_level_required": [1, 2, 0, 0, 0, 0],
      "room_id": "r21"
    },
    ...
    ]
 "patients": [
    {
      "id": "p28",
      "mandatory": true,
      "gender": "A",
      "age_group": "elderly",
      "length_of_stay": 3,
      "surgery_release_day": 3,
      "surgery_due_day": 17,
      "surgery_duration": 90,
      "surgeon_id": "s0",
      "incompatible_room_ids": ["r2"],
      "workload_produced": [1, 1, 1, 2, 1, 1, 1, 2, 1],
      "skill_level_required": [1, 2, 0, 2, 0, 0, 2, 1, 1]
    }
    ...
    }
```

The solution file format is divided into two sections: one concerning patients and one concerning nurses. The following fragment illustrates both sections.

```
{
  "patients": [
    {
      "id": "p00",
      "admission_day": 4,
      "room": "r3",
      "operating_theater": "t0"
    },
    ...
    ],
  "nurses": [
    {
      "id": "n00",
      "assignments": [
        {
          "day": 0,
          "shift": "early",
```

```
          "rooms": ["r1", "r4"]
        },
        {
          "day": 1,
          "shift": "night",
          "rooms": ["r4", "r5"]
        },
        ...
        ]
    },
    ...
    ]
}
```