A multi-neighborhood, lexicographic local search algorithm for the Integrated Healthcare Timetabling Competition 2024

Ahmad Othman
ahmaoth@gmail.com

Marco Chiarandini
marco@imada.sdu.dk

February 2025

1 Solution Approach

Our entry to the Integrated Healthcare Timetabling Competition of 2024 (IHTC-2024) [1] is a heuristic, local search algorithm that uses multiple neighborhood structures and manages hard and soft constraints lexicographically.

Terminology and description of the Integrated Healthcare Timetabling Problem (IHTP) are presented in [1]. Theory and practice of local search algorithms can be found in [3, 2]. We represent candidate solutions as a 2-tuple, with the assigned values for each patient and the nurse assigned to each specific room, day and shift pair. Candidate solutions always assign all mandatory patients, and a nurse to each room, day and shift pair. Moreover, they implicitly satisfy hard constraints H2 (Compatible rooms), H5 (Mandatory versus optional patients) and H6 (Admission days) by restricting the domains of the values. However, they may violate some of the other constraints. The evaluation of a solution gives a 2-tuple of two integers, being the hard and weighted sum of soft constraint violations respectively, with two solution evaluations being compared lexicographically based on their evaluation tuples. We, then, consider eight neighborhood structures defined by the corresponding local moves:

- Unassign an optional patient.
- Assign a patient to an admission day, room, and operating theater (OT).
- Assign a room, day and shift to a nurse.
- Swap values of two patients.
- Swap a single room of two nurses on the same shift.
- Swap all rooms between two nurses on the same shift.
- Kick: Set the values of a patient to the values of another patient, and randomize the latter patient.
- Kick out: Kick a patient on an optional patient and unassign them.

Solution evaluations are always calculated incrementally.

1.1 Preprocessing

We reduce the search space by discarding initially assignments that would make solutions clearly infeasible. We discard: i) potential admission days for a patient if their surgeon is unavailable on those days (for H3), ii) all the allowed admission days and OT for a patient if the OT is unavailable on that day (for H4), iii) admission days for a patient if scheduling them on that day would violate the room gender mix hard constraint with an existing occupant (for H1), and iv) admission days for a patient if scheduling them there would violate the room capacity constraint with other occupants (for H7).

1.2 Search Algorithms

Our final algorithm consists of four different configurations of two main algorithms: Simulated Annealing (SA) and Iterated Local Search (ILS). These four algorithms run in parallel and the best solution among them is selected.

Our SA algorithm uses exponential acceptance rate with different temperatures for the hard and soft constraints. If the proposed move increases the hard constraint violations, we accept it according to the exponential acceptance criterion on the current hard constraint temperature. If the move does not change the hard constraint violations, we repeat the same test for the soft constraint violations using the current soft constraint temperature. The temperature is cooled linearly according to the remaining time, starting at max temperature (600 sec) until 0.[1] A move is selected for evaluation by first randomly sampling the neighborhood structures with appropriate probabilities and then uniformly selecting a move within the chosen structure.

Our ILS algorithm iteratively performs a first-improvement local search that explores the full multi-neighborhood exhaustively. The local search cycles through the neighborhoods and continues from the same position after accepting a move. It terminates after completing a full cycle without accepting any moves, indicating a local optimum. The ILS then perturbs the solution using various kicks and restarts the search, keeping a record of the best seen solution.

1.3 Algorithm Selection

We selected offline the four algorithm configurations to be run in parallel on the four threads made available by the rules of the competition by solving optimally a sequence of set covering problems with cardinality constraints.

[1]Since the current temperatures are a function of the remaining time, using the same random seed might not always yield exactly the same result.

1.4 Results on the Public Dataset

The program is written in C++ 20 and compiled with GNU g++. All runs were performed on an Intel Core i5-8500 processor with 6 physical cores at 3.00 GHz. To make the computational budget (i.e., the time limit) comparable to the 600 sec. that will be imposed on the official PC of the competition, an AMD Ryzen Threadripper PRO 3975WX (3.50 GHz) we derive a transformation factor of 3.50/3.00, which we used to calculate a time limit of 700 sec for our machine.

The results are displayed in Table 1.

Instance	Objective	Instance	Objective	Instance	Objective
i01	3849	i11	26203	i21	24703
i02	1264	i12	12430	i22	47861
i03	10490	i13	17865	i23	38870
i04	1911	i14	10065	i24	35815
i05	12881	i15	12486	i25	11859
i06	10728	i16	10983	i26	65091
i07	5026	i17	40535	i27	51828
i08	6363	i18	39094	i28	79571
i09	6682	i19	44587	i29	12475
i10	20885	i20	30819	i30	37943

Table 1: Results of the algorithm on the 30 public dataset instances. Each instance has been run with 700 seconds. Results have been cross-validated with the validator provided by the competition.

References

[1] Sara Ceschia, Roberto Maria Rosati, Andrea Schaerf, Pieter Smet, Greet Vanden Berghe, and Eugenia Zanazzo. The integrated healthcare timetabling competition 2024—problem description and rules—. In *Proceedings of the 14th International Conference on the Practice and Theory of Automated Timetabling*, pages 52–65, 2024.

[2] Holger Hoos and Thomas Stützle. *Stochastic Local Search: Foundations and Applications.* 2004.

[3] Wil Michiels, Emile Aarts, and Jan Korst. *Theoretical Aspects of Local Search.* Monographs in Theoretical Computer Science. Springer, 2007.