EURO2025 LEEDS

A multi-neighborhood, lexicographic local search algorithm for the IHTC 2024

Ahmad Mahir Othman Marco Chiarandini

Department of Mathematics and Computer Science

University of Southern Denmark

June 23, 2025

Outline

Problem Formulation

- The problem integrates the following decisions:
  - Patient admission scheduling
  - Patient-to-room assignment
  - Nurse-to-room assignment
  - Operating theaters assignment

Day	Day 1			Day 2			...	Day 7n		
	early	late	night	early	late	night		early	late	night
Room 1	$\{p_1\}$			$\{p_1, p_3\}$			...			
	$\{n_1\}$	$\{n_2\}$	$\{n_3\}$	$\{n_4\}$	$\{n_2\}$	$\{n_3\}$				
Room 2	$\{p_2\}$			$\{p_2\}$						
	$\{n_1, n_5\}$	$\{n_4\}$	$\{n_3\}$	$\{n_1\}$	$\{n_2\}$	$\{n_3\}$	...			
OT 1	$\{p_1, p_2\}$			$\{\}$			...			
OT 2	$\{\}$			$\{p_3\}$			...			

- Solutions must **satisfy** 9 hard constraints (H1-H9) and **minimize a weighted sum** of 8 soft constraint violations (S1-S8)

- Competition imposes a **time limit of 10 minutes** and **maximum of 4 threads**

Outline

Our Attempts

- **Integrated MILP model** solved with Gurobi
  - 4 sets of binary decision variables
  - 11 sets of auxiliary variables
  - 26 sets of constraints
- **Hybrid methods** combining MILP and heuristics:
  - solve subproblems optimally
  - combine subproblem solutions heuristically
- Pure **local-search based metaheuristic algorithm**, the one submitted

Integrated MILP Model Results with Gurobi

Name	Continuous	Binary	Constraints	Setup	Solve	LB	Objective	Gap
i01	4496	5156	33932	3 s	600 s	3509	4148	15.4 %
i02	6675	7856	52346	5 s	600 s	784	1579	50.3 %
i03	6603	8053	57838	5 s	600 s	10011	10615	6.0 %
i04	16344	20089	199225	18 s	600 s	916	8651	89.4 %
i05	52888	58128	908939	79 s	602 s	12436	15223	18.3 %
i06	51695	80683	504856	67 s	601 s	10422	10853	4.0 %
i07	30914	39512	559676	55 s	601 s	2179	7765	71.9 %
i08	206957	323121	3523121	476 s	605 s	-	-	-
i09	37763	45194	731402	66 s	601 s	2630	45212	94.2 %
i10	88579	146727	1219228	152 s	602 s	14315	38640	62.9 %
i11	50219	63443	1123126	113 s	602 s	25463	34961	27.4 %
i12	64638	106568	816554	99 s	601 s	-	-	-
i13	46011	59145	848878	92 s	601 s	7605	56033	86.4 %
i14	123819	195679	2224097	244 s	603 s	5952	35315	83.1 %
i15	80626	96216	2026319	194 s	603 s	-209647	49312	399.5 %
i16	114330	220757	2001705	232 s	602 s	-	-	-
i17		out of memory						
i18	316210	483579	7385718	981 s	609 s	-	-	-
i19		out of memory						
i20	122902	233671	1971083	283 s	603 s	-	-	-
i21		out of memory						
...		out of memory						

Hybrid MILP Methods

- Solved three assignment subproblems:
  - patient day assignment
  - nurse assignment
  - operating theater assignment
- as perturbation to Iterated Local Search
- best solution so far used as warm start

Outline

1. The Integrated Healthcare Timetabling Model
2. Our Solution Attempts
3. Submitted Metaheuristic Approach
4. Analysis

Main Characteristics

- Pre-processing of hard constraints
- Multi-neighborhood structure
- Efficient evaluation of moves
- Handle hard and soft constraints lexicographically

$$\min_{s \in \mathcal{S}} \quad \left(\sum_{i=1}^{9} v_{\mathtt{Hi}}(s), \sum_{i=1}^{8} w_{\mathtt{Si}} \cdot v_{\mathtt{Si}}(s) \right)$$

- Simulated annealing + Iterated local search
- 4 threads running in parallel with different parameters

Solution Representation

Map of patients to (day, room, OT) + Map of (room, day, shift) to nurse:

```
dict[Patient, tuple[Day, Room, OT]]
{'p1': ('day1', 'room1', 'OT1'), 'p2': ('day1', 'room1', 'OT1'), ...}

dict[tuple[Day, Shift, Room], Nurse]
{('day1', 'early', 'room1'): 'n1', ('day1', 'early', 'room2'): 'n1', ...}
```

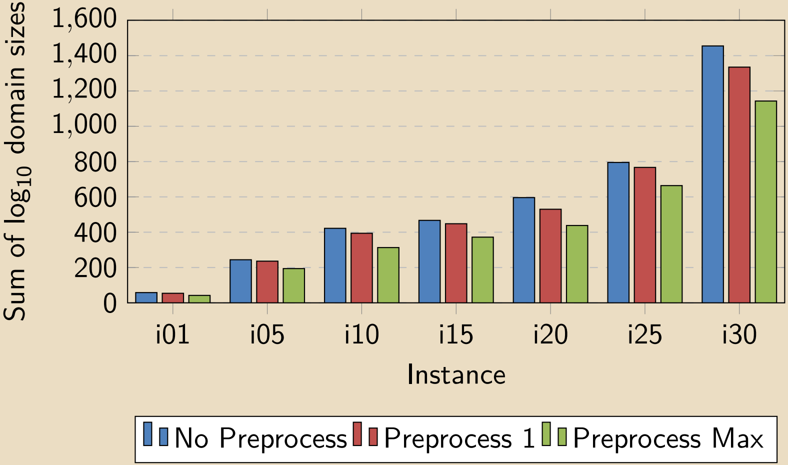
Both dictionaries are exhaustively initialized

Preprocessing of hard constraints

- Hard constraints:
  - H1 No gender mix
  - H2 ~~Compatible rooms~~
  - H3 Surgeon overtime
  - H4 OT overtime
  - H5 ~~Mandatory versus optional patients~~
  - H6 ~~Admission day~~
  - H7 Room capacity
  - H8 ~~Nurse presence~~
  - H9 ~~Uncovered room~~
- Entailed by solution representation H5, H8, H9
- Entailed after variable domain pruning the unary hard constraints H2, H6
- Variable domain pruning by considering pre-assignments, aka occupants for H1, H7
- Variable domain pruning for null capacity of surgeons and OTs in H3, H4

Impact of Preprocessing on Search Space

Search space restricted to patients expressed as cartesian product of their domain size



Multi-neighborhoods

8 neighborhood structures:

- 3 **atomic moves**:
  - MoveSetPatient, MoveRemovePatient, MoveSetNurse
- 5 **chain moves** made of multiple atomic moves:
  - MoveSwapPatients, MoveKickPatient, MoveKickPatientOut
  - MoveSwapNurseRoomsAll, MoveSwapNurseRoomsSingle

```
dict[Patient, tuple[Day, Room, OT]]
{'p1': ('day1', 'room1', 'OT1'), 'p2': ('day1', 'room1', 'OT1'), ...}

dict[tuple[Day, Shift, Room], Nurse]
{('day1', 'early', 'room1'): 'n1', ('day1', 'early', 'room2'): 'n1', ...}
```

Solution Evaluation and Move Evaluation

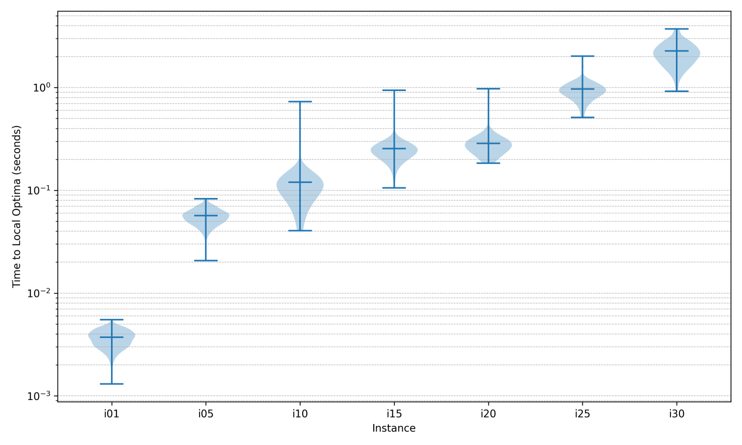
- Solution evaluation $([v_{\mathtt{Hi}}(s)]_{i=1}^{9}, [v_{\mathtt{Si}}(s)]_{i=1}^{8})$
- Preference Model: Lexicographic ordering

$$\min_{s \in \mathcal{S}} \quad \left(\sum_{i=1}^{9} v_{\mathtt{Hi}}(s), \sum_{i=1}^{8} w_{\mathtt{Si}} \cdot v_{\mathtt{Si}}(s) \right)$$

- Moves assessed efficiently considering only **incremental evaluations** using auxiliary data structures
- Moves are evaluated **lazily**
- Application of moves must update the auxiliary data structures

Time to Local Optima

- Time to local optima using **first-improvement local search** on the union neighborhood

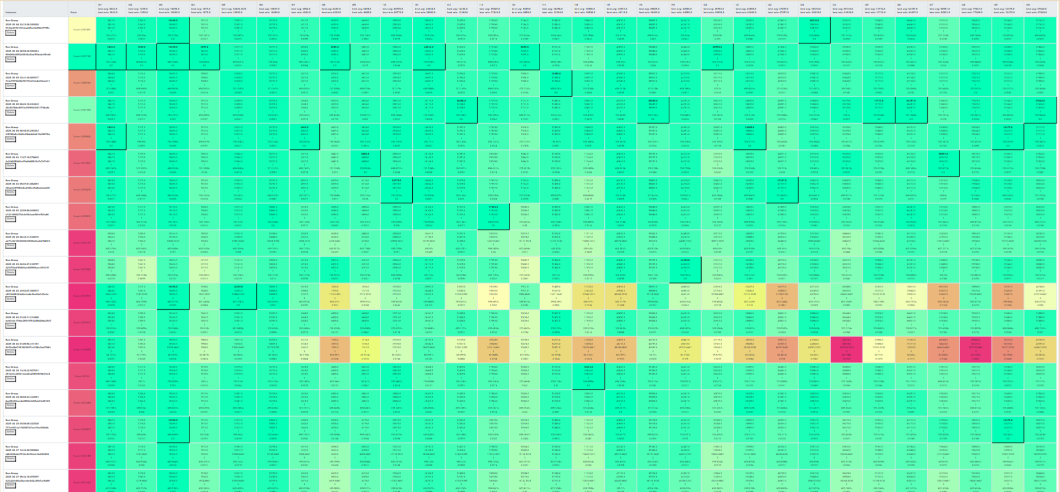


Heuristic Algorithm

- Start with a **random initial solution**, then run simulated annealing for T seconds followed by iterated local search for $600 - T$ seconds
- Simulated annealing:
  - Select a neighborhood from weighted probabilities, and then select move uniformly at random from neighborhood
  - Maintain two temperatures for hard and soft constraints
  - Temperature decreases linearly according to remaining time
- Iterated local search:
  - First-improvement on the union neighborhood
  - 21 kinds of perturbations complementary to the defined neighborhoods
- Implemented in Python and in C++ following the ROAR-NET API specification
- T, neighborhood weights and other algorithm parameters were tuned with irace

Analysis

- irace yielded several sets of non-significantly different configurations
- We collected results on each public instance by the winning configurations and computed **gap from instance best known**



Exploiting the 4 Threads

Task: select 4 configurations

Input: Results on each public instance by the winning configurations and computed gap from instance best known

repeat

set a threshold in non-decreasing order;

convert gap matrix to a $\{0, 1\}$ matrix according to threshold;

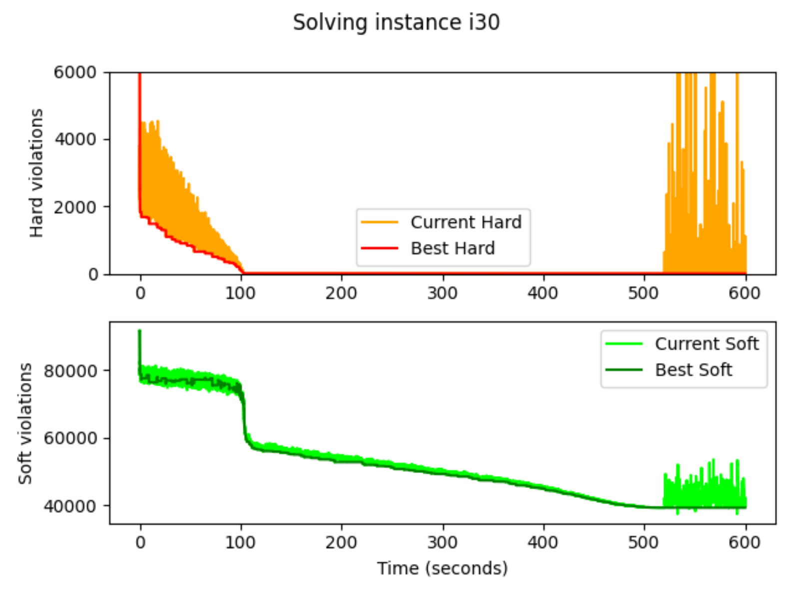
solve a set covering problem;

until 4 configurations cover all instances;

Outline

1. The Integrated Healthcare Timetabling Model
2. Our Solution Attempts
3. Submitted Metaheuristic Approach
4. Analysis

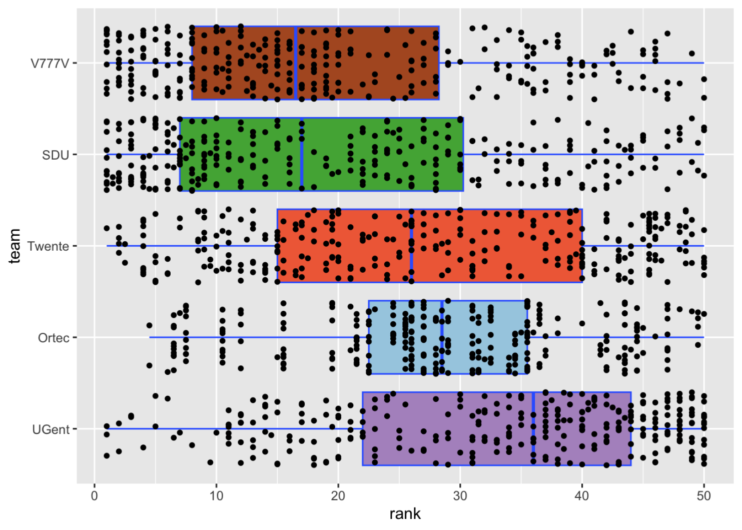
Algorithm Profiling



Competition Results

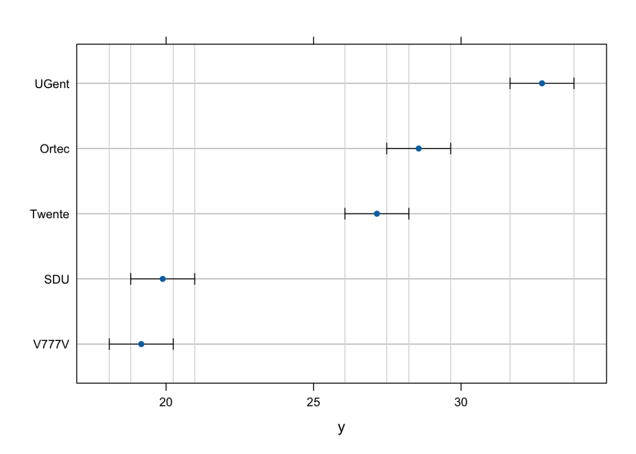
	team	median_rank	mean_rank
1	V777V	16.50	19.16
2	SDU	17.00	19.89
3	Twente	26.00	27.15
4	Ortec	28.50	28.56
5	UGent	36.00	32.74

Competition Results



Competition Results

Rank-based Friedman post-hoc test [Conover, 1999], the same used in irace



If intervals overlap then difference not statistically significant

Conclusions

- A classical metaheuristic approach performed well given the proposed time limit
- No construction heuristic was developed
- A structured approach suggested by the ROAR-NET API was helpful and flexible

Thank you!