METAHEURISTICS FOR OPTIMIZATION/DECISION PROBLEMS

PATIENT ALLOCATION IN HOSPITALS

ARTIFICIAL INTELLIGENCE

Group_A1_82

Rodrigo Resende - up202108750

Tomás Ferreira - up202002749

João Pedro Silva - up202208936

PROBLEM SPECIFICATION

Efficient patient allocation is crucial for enhancing hospital operations and patient outcomes.

The objective is to develop a system that assigns patients to appropriate hospital resources - such as beds, staff, and departments - while considering constraints like resource capacities and patient needs.

Related Work

Generating balanced workload allocations in hospitals: Discusses optimization models for patient flow in hospitals.

Data for 'Generating balanced workload allocations in hospital': Provides real-world data for testing and validation.

Meta-heuristic algorithms (e.g., genetic algorithms, tabu search) have been widely used in healthcare optimization problems.

Problem Formulation as an Optimization Problem

Solution Representation: `{patient_id: (ward, admission_day, length_of_stay)}`

Evaluation Function: `fitness = -(overtime + undertime + delay + 1e6 * overcapacity)`

Neighborhood/Mutation Functions: Swap patient assignments, reassign patients to different resources.

Crossover Function (for Genetic Algorithms): Combine two solutions to create a new one (e.g., random selection of assignments from parents).

Hard Constraints: No overloading of resources; High-priority patients must be allocated first.

Implementation Progress

Programming Language: Python (for flexibility and extensive libraries).

Development Environment: Visual Studio Code

Data Structures: Lists/arrays for patient and resource data; Dictionaries for mapping patients to resources.

Current Status: Parser for reading problem instances from .dat files; Full implementation of 4 optimization algorithms: Genetic, Hill-Climbing, Simulated Annealing and Tabu Search.

Implemented Algorithms

Genetic Algorithm (GA)

Parameters:
Population=50, Generations=100

Mutation: Adaptive (light/normal/strong)

Workflow:
Initialize with mixed strategies
Tournament selection → Crossover (ward-based)
Mutate → Elite retention

Strength: Global optimization for large instances

Hill-Climbing

Parameters:
Restarts=3, Neighbors=20/iter

Strategies: patient_swap, day_shift

Workflow:
Generate neighbors
Accept first improvement
Restart on plateau

Strength: Fastest (22.1s avg.)

Implemented Algorithms

Simulated Annealing (SA)

Parameters:
Temp: 1000→0.1, α=0.95
Reheat: 1.5x on stagnation

Workflow:
Mutate current solution
Accept via Metropolis criterion
Adapt cooling rate

Strength: Handles complex constraints (-1.1e5)

Tabu Search

Parameters:
Tabu list=50, Diversify every 50 iters
Neighborhood=30/iter

Workflow:
Generate candidate moves
Enforce tabu attributes (ward/day)
Aspiration for global best

Strength: Balanced performance (-9.8e4)

Experimental Results

Algorithm	Fitness	Runtime	Best Scenario
GA	0	1.7s	Large patient pools
Hill-Climbing	-10	0.21s	Quick prototypes
SA	0	5.57s	Complex constraints
Tabu	-1	4.44s	Balanced workloads

These are the results for data file s1m2.dat

Conclusions

Tabu Search achieved the best balance: Lowest cost (-9.8e4) and Moderate runtime (38.7s).

Hill Climbing was the fastest (22.1s) but the solution quality is limited.

Simulated Annealing excels in constraint-heavy scenarios.

Genetic Algorithm scales the best for large problems.

References

‘Generating balanced workload allocations in hospitals’ **- scientific article**

Dataset: Data for 'Generating balanced workload allocations in hospitals'

Metaheuristics used: Genetic Algorithm, Hill-Climbing, Simulated Annealing and Tabu Search

Python Libraries: rich, random and re

Code: https://github.com/Resende16/IA