A Metaheuristic Approach to Solve an Outpatient Surgery Scheduling Problem

Lydia Bouchlaghem, Fatima Ghedjati, Alexandre Philippot

Université de Reims Champagne-Ardenne
Chemin Des Rouliers, 51100 Reims
Lydia.bouchlaghem@univ-reims.fr

1 Introduction

Operating room scheduling is an ongoing research topic with direct practical applicability. Managing an operating room (OR) department is a complex task that requires a broad view of the hospital system including staff management, resource allocation, and patient appointment scheduling. In practice, many difficulties of such managerial tasks stem from the OR department being a shared facility, used by multiple medical specialties, where outpatient and inpatient surgeries coexist simultaneously in an uncertain environment.

Outpatient surgeries allow the patient to enter and leave the hospital on the day of the intervention, whereas inpatient surgeries require an overnight stay in the hospital (post-surgery recovery and monitoring, complex multi-stage surgeries, risky pre-existing conditions for the patient…). With the improved technology and advances in anesthesia and pain control, many less invasive surgical procedures are now being performed on an outpatient, or ambulatory, basis. Typically, as emphasized by *Denton et al.* [1], outpatient surgery encompasses three stages: preoperative, intraoperative, and postoperative. The preoperative stage begins with the patient's arrival at the outpatient facility and ends when the patient is transferred to the OR bed, whereas the intraoperative stage is defined as the time between when the patient reaches the OR bed and the time when he is admitted to the recovery area. The postoperative stage defines the time between the patient's arrival at the recovery area and the time that the surgeon finishes the follow-up care with the patient. Each of these stages is critical to a successful surgical procedure.

In this paper we consider the problem of scheduling an outpatient surgery unit in a hospital similar in structure to the Reims CHU. Our main goal is to provide a general framework, and a first approach solution, to this scheduling problem under which further discussion on uncertainty management, multi-objective scheduling and predictive techniques can be efficiently led. As such, we target here the minimization of patients' length of stay considering the three stages (preoperative, intraoperative, and postoperative) and under a block planning strategy. We choose to model our problem as a three stage Hybrid Flow Shop (HFS) scheduling problem proven to be NP-hard. Our solution approach consists of using Genetic Algorithms (GA's) to efficiently solve the proposed problem.

2 Problem description and solution approach

In this paper, we focus on the scheduling of a specified number of patients in an ambulatory surgery center. Our goal is to minimize the patient's length of stay. The number of patients and patient's types are predetermined in the higher-level planning according to available resources. The problem considers several patient types with deterministic service times at each stage and punctual arrivals. The stages work according to a "first come, first served" rule except for the first stage, which admits patients according to the schedule. Each type of patient is served by a specific specialty, which intensifies the importance of resource compatibility in this problem.

Under the block planning strategy, OR resources are pre-affected to a particular specialty (a group of surgeons of the same specialty) for a specified timespan at a specified time. The number of available surgeon types per time block is provided by the surgeon schedules, which are determined based on the surgeons' availability time window. Each operating room is therefore assigned a specialty and a specific surgeon group for a predefined block of time; this is known as "master surgery scheduling," which is developed by hospital management. The problem considers resource availability such as ORs, preparation beds, and recovery beds.

Our problem can be seen as a three-stage HFS problem, where a set of *n* patients are assigned through three consecutive stages, starting from stage *1*. Stages *1* and *3* are composed of a set of beds (placed respectively in the preparation and recovery areas), while stage *2* is composed of non-identical operating rooms organized by medical specialties. Each patient j is assigned only one bed from the beds in stages *1* and *3*, and a single operating room in stage *2*. We consider the following assumptions: Each patient must pass through the three stages, and each stage is characterized by a duration (preparation, operation, and recovery durations). Pre-emption is not permitted. The intervention of patient *j* cannot start unless the intervention of patient *j-1* is finished. For generality purposes, we also suppose that the beds in the preparation and recovery areas are available in sufficient number (no blocking constraint). Therefore, the objective is to minimize the makespan.:

$$C_{max} = max_{i=1..n} C_i \quad (1)$$

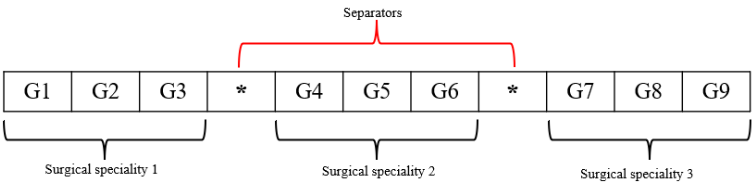
For this criterion, *Gupta et al.* [2] showed that two-stage HFS problems are NP-hard in cases with at least two machines on one of the two stages. Thus, the problem we consider in this paper is also NP-hard. Heuristic and/or meta-heuristic approaches allow us to find acceptable solutions in a reasonable time, in these cases. Because multiple authors used them successfully for this category of problems, we chose a GA. We can cite in this regard: Lin et al [3] or Timucin et al [4].

The basic idea behind GAs is to simulate the process of natural selection by creating a population of potential solutions to a problem, and then iteratively applying genetic operators to the population to generate new, more fit solutions. The genetic operators are based on the mechanics of biological evolution. A fitness function is used to evaluate the quality of each solution in the population, and the best solutions are selected to create the next generation. This process is repeated until a stopping criterion is met.

Based on the work provided by *Cheng et al.* [5], our proposed approach consists of representing the solution as follows: A **gene** (G) corresponds to a data structure containing data relative to the patient's pathway: his assigned bed, start and end times in stage 1; his assigned operating room of the relevant surgical specialty, start and end times in stage 2; his assigned bed, start and end times in stage 3. A **chromosome** (S) is a vector composed of genes and symbols of separation (denoted * to help distinguish genes belonging to the same surgical specialty). For instance, with 9 patients (j = {1, ..., 9}) and 3 operating rooms (3 surgical specialities), we have: S = {G3, G4, G5 * G8, G7, G9 * G2, G6, G1}.

Our approach uses the three classical genetic operators namely: selection, crossover, and mutation.

Rank selection: consists of choosing from the initial population the N individuals that minimize the objective function (1). **Two-point crossover**: allows to switch the set of genes which lies between two randomly chosen points, these points need to lie within separators (*) to avoid mixing the surgical specialities. **Mutation**: Consists of swapping two randomly selected genes belonging to the same surgical specialty in a randomly chosen chromosome. After each mutation we repair the chromosome, by recalculating the start and end times and adjusting the chromosome. In each generation, we evaluate the solutions, and save the best solution. The process terminates when the predetermined stopping criteria is satisfied.



3 Experimentations

In this section, we present results based on a series of numerical experiments. First, we describe how we generated our test cases. Then, we present the approach used as a comparison for the considered GA: a heuristic approach proposed by Sadaani et al. [6]. Finally, we compare the results of the approaches on the generated test cases.

All experiments were conducted on an HP EliteBook laptop running 64-bit Windows 10 with an Intel(R) Core (TM) i7-1185G7 CPU and 32GB of RAM. The performance of the proposed GA is evaluated using nine test surgery cases that are different in the number of patients, surgery durations, and their required OR resources. The durations were generated based on a normal random distribution. For the sake of diversifying the test cases, the surgeries are divided into five categories: short (S), medium (M), long (L), extra-long (E) and special (SP). This classification is based solely on the length of the surgery and does not take into account the surgical specialty. Table **A.1** shows in detail the parameters considered for the generation of the durations. Table **A.2** provides the detail information about the nine test cases.

Table 1 Approaches comparison.

		GA			NEHDLBM		
Number of patients	**Number of Operating rooms**	**MIN**	**MAX**	**AVG**	**MIN**	**MAX**	**AVG**
10	**6**	499	332,4	277	472	376,7	499
15	**6**	913	502,3	455	1037	584,9	913
20	**6**	898	724,1	661	1106	855,3	898
30	**10**	1524	928,7	689	1816	1059	1524
40	**10**	1209	975,1	814	1495	1063	1209
50	**10**	1500	1176	992	1611	1237,8	1500
60	**13**	1400	1114,4	917	1469	1212,3	1400
80	**13**	1701	1322,3	1122	1808	1414,1	1701
100	**13**	1989	1733,7	1630	2254	1826,2	1989

To compare our results, we chose to implement the heuristic proposed by *Sadaani et al.* [6], which consists of combining Nawaz-Enscrore-et-Ham heuristics with the Palmer Index and Last Busy Machine heuristic, referred to in this article by *NEHDLBM*. The authors applied this heuristic to an operating room scheduling problem, considering the three operative stages, and described the problem as a three-stage hybrid flow shop problem with the goal of minimizing the makespan.

We run tests on the nine problem classes. Computational results are given in **Table 1**. Since our algorithms are by nature randomized, we ran them 10 times for each instance and reported the minimum (MIN), maximum (MAX) and average (AVG) makespan values. Table 2 shows that, on average, *GA* provides better results regarding makespan minimization in comparison *NEHDLBM.*

5 Conclusion and ongoing work

In this paper, we propose an approach for solving the outpatient surgery problem taking into account the three stages of the patient's journey, the surgery itself, the block scheduling strategy, and the resource constraints. In our framework, we described OR scheduling as a three-stage hybrid flow shop problem. Due to the computational complexity of this combinatorial optimization problem, we developed a *GA* algorithm with the objective of minimizing the maximum total length of stay of a patient (makespan). Nine test cases were built with different surgery sizes and available resources. An additional approach was considered as a comparison, namely *NEHDLBM*. The two approaches were compared, computational results showed that our approach is more effective regarding the makespan as the unique criterion.

The rest of our work will consist in integrating the improvements allowed by multi-objective approaches and machine learning models. we will seek to make the model more complete and useful by integrating the objectives and criteria put forward by the Reims hospital experts, for instance with regard to last minute cancellations.

References

[1] Denton, B., Rahman, A., Nelson, H., & Bailey, A. (2006). Simulation of a multiple operating room surgical suite. *Proceedings of the 2006 Winter Simulation Conference*. https://doi.org/10.1109/wsc.2006.323110

[2] Gupta, J. N. (1988). Two-stage, hybrid Flowshop scheduling problem. *The Journal of the Operational Research Society*, *39*(4), 359. https://doi.org/10.2307/2582115

[3] Timucin, T., & Birogul, S. (2018). Implementation of operating room scheduling with genetic algorithm and the importance of repair operator. *2018 2nd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*. https://doi.org/10.1109/ismsit.2018.8566697

[4] Lin, Y.-K., & Chou, Y.-Y. (2019). A hybrid genetic algorithm for operating room scheduling. *Health Care Management Science*, *23*(2), 249–263. https://doi.org/10.1007/s10729-019-09481-5

[5] Cheng, R., Gen, M., & Tozawa, T. (1995). Minmax earliness/tardiness scheduling in identical parallel machine system using genetic algorithms. *Computers & Industrial Engineering*, *29*(1-4), 513–517. https://doi.org/10.1016/0360-8352(95)00126-l

[6] Saadani, N. H., Guinet, A., & Chaabane, S. (2006). Ordonnancement des blocs operatoires. In *MOSIM: Conference francophone de MOdélisation et SIMulation* (Vol. 6).

Appendix A

Table A.1 The duration of pre/post-surgery stage and durations of the different surgery types

	Pre-surgery	**Surgery case**					**Post- surgery**		
		Small	Medium	Large	Extra-large	Special	Small	Medium	Large
Duration (minutes)	μ=8, σ = 2	μ=33, σ = 15	μ=86, σ = 15	μ=153, σ = 17	μ=213, σ = 17	μ=316, σ = 62	μ=59, σ = 16	μ=183, σ = 17	μ=210, σ = 41

(μ : mean, σ : standard deviation)

Table A.2 Test cases

Case	**Number of Patients**	**Number of Operating rooms**	**Number of Surgical specialties**	**Surgery Type (S, M, L, E, SP)**
1	10	6	3	(4,6,0, 0,0)
2	15	6	3	(4,9,2,0,0)
3	20	6	3	(4,12,2,1,1)
4	30	10	5	(6,19,2,2,1)
5	40	10	5	(8,22,4,5,1)
6	50	10	5	(8,26,8,5,3)
7	60	13	6	(10,28,14,5,3)
8	80	13	6	(19,37,16,5,3)
9	100	13	6	(21,43,20,9,7)

(S: Small, M: Medium, L: Large, E: Extra-long, SP: Special)