A GENETIC ALGORITHM APPROACH FOR SURGERY OPERATING ROOMS SCHEDULING PROBLEM

Galina Tsoy, Old Dominion University
Jean-Paul Arnaout, Old Dominion University
Timm Smith, Old Dominion University
Ghaith Rabadi, Ph.D., Old Dominion University

Abstract
This paper addresses a surgery rooms scheduling problem. The problem is modeled as a parallel machine scheduling problem with the objective of minimizing the makespan (as a primary objective) and minimizing the total completion time (as a secondary objective). Many methods have been used to address this problem in the traditional manufacturing environment. In this paper, we use Genetic Algorithms (GAs) to solve the problem in the surgery room-scheduling context.

Introduction and Background

In order to effectively address the mounting pressures of an ever-changing healthcare environment, it is imperative that the healthcare industry responds to market challenges that require high efficiency and improved performance at the lowest possible cost. In this paper, we will consider strategies for searching out process improvements in surgery departments. The Operating Room (OR) is one of the largest revenue and cost center in the healthcare industry, and therefore, ORs represent a significant opportunity to improve margins.

Currently in United States hospitals, patients wait long hours before they are checked in and prepared for surgery. The first patient's admission delay turns into the next patient's delay in the preparation of operating rooms, consequently throwing off the whole surgical schedule. Ineffective scheduling that is frequently subject to change is very costly. The current situation does nothing to improve hospitals' margins or patients' satisfaction; therefore, a task of essential importance for the operation manager is the creation of more accurate schedules that minimize patient delay times and/or maximize surgery room utilization.

To create a more accurate schedule and compare with the old scheduling system, data was obtained from Sentara Leigh Hospital in Virginia Beach including surgery date; scheduled time; patient type; estimated procedure time; and actual procedure time for every day for 6-month period. Only the estimated procedure time was used for constructing the new scheduling model. Currently, most hospitals including Sentara Leigh Hospital assign block times to surgeons' groups based on their requests and on the previous month's performance of utilization of their block times. Then, surgeons contact the administrative office for a specific starting time for surgery and request a reservation within allocated block time. The reservation within a certain time block is done on "first come first served" basis. The total number of rooms in the Ambulatory Surgery Center (ASC) is six. Procedures are assigned to the rooms based on their duration and their category.

In most ORs, the management's main goals are to improve the utilization of the ORs and reduce the patients' waiting time. As a result, two objective functions were considered in this paper: a primary objective function to minimize the makespan (schedule length) and a secondary objective to minimize the total completion time. With the makespan as a primary objective function, the problem becomes NP-hard and Genetic Algorithms (GAs) will be used to solve it. GAs were effective at solving this problem type when compared to other heuristics such as the Longest Processing Time first (LPT) rule and Simulated Annealing algorithm (Lui and Cheng, 1999). Therefore, GAs will be used with this parallel machine scheduling problem in the context of surgery room scheduling.

Once the GA finds its best solution, the schedule will be optimized with respect to the secondary objective function of minimizing the total completion time.

Problem Statement

This paper addresses two variants of the parallel machine scheduling problem: The first is when all procedures are eligible to be performed in all rooms, and the second is when certain procedures can be performed in certain rooms only. The latter condition will be referred to as the "machine eligibility restriction", where not all jobs (i.e., procedures) are eligible to be processed on all machines (i.e., ORs). The objective of the study is to introduce a new scheduling method that maximizes ORs throughput or utilization and minimizes patient waiting times to improve process performance in the ORs compared to the existing scheduling method. The primary objective is to minimize the OR makespan. The *makespan* is defined as the time necessary to finish all procedures. This is the same as minimizing the maximum completion time of the last procedure to be completed in all operating rooms. A minimum makespan usually implies a high utilization of the OR(s). The secondary objective is to minimize the patient waiting times to have a surgery. This can be accomplished by minimizing the total completion time of all procedures. Note that the total completion time for a job is the time it spends in the system, including processing and waiting for other procedures to finish. This is *not* the same as the makespan.

Problem Model

For this study, two models were developed. The first model is used with the scheduling problem without machine eligibility restrictions, while the second model is used with machine eligibility restrictions. In both cases the makespan for identical parallel machines is minimized as a primary objective function. The problems addressed can be modeled as identical parallel machine scheduling problem with minimizing the makespan because an OR represents a resource (or a machine) and since procedure times are not dependent on the room to which they are assigned, then the ORs can be treated as identical parallel machines. Note that the reason the machines are considered identical is because they have the same processing speeds to process a certain task (Eiselt and Sandblom, 2004).

We considered the operating rooms in the ASC as six identical parallel machines. In this sense, determining the ORs schedule will be a matter of determining the best combination of procedures that would minimize the makespan across all six ORs. Let *m* represent the number of operating rooms, *n* the number of jobs or procedures, and p(*j*) is the processing time of job *j* (or procedure *j*). Define x(*i*,*j*) as binary variable that represents the assignment of job *j* to machine *i*, where If x(ij) = 1, then job *j* is assigned to machine *i*. Otherwise, if x(ij) = 0, then job *j* is not assigned to machine *i*. Note that, x(*i*,*j*)p(*j*) represents the processing time of job *j* on machine *i*.

The primary objective function will then be as defined by equation (1):

Minimize Cmax =

$$[\min\{\max[\sum_{j=1}^{n} x(1,j)p(j),\ldots,\sum_{j=1}^{n} x(m,j)p(j)]\} \quad (1)$$

subject to:

$$\sum_{i=1}^{m} x(i,j) = 1, \forall 1,\ldots,n \quad (2)$$

$$x(i,j) \in \{0,1\} \quad (3)$$

The objective function in (1) is the overall makespan of the schedule. The set of constraints in (2) ensure that a procedure cannot be assigned to more than one room. The constraints in (3) state that the decision variables *x*(*i*,*j*) must be binary.

The problem at hand is NP-hard even for two machines (Garey and Johnson, 1979) where the search space grows exponentially and is equal to m^n. We are assuming in this paper that other resources such as surgeons and nurses are available when needed.

The second model considers machine eligibility restrictions where certain procedures can be assigned only to certain rooms among the six rooms. For example, procedure (a) may require to be performed only in room 1; while procedure (b) must be performed in room 4 for instance. Some procedures, however, may be assigned to any room. The same objective of minimizing the makespan is used here also.

Solution Methodology.

Genetic Algorithms (GAs). GAs are used to minimize the makespan as a primary objective function. GAs are probabilistic search techniques adapted from the natural selection process to solve optimization problems. GA’s are based on the concept of *the survival to the fittest*.

The basic approach to implementing a GA consists of the following eight steps.

Step 1: Initialization – An initial population of feasible solutions (or individuals) is created, usually randomly.

Step 2: Evaluation – Once the initial population is generated, each solution is evaluated for fitness based on the objective function. As a result, each individual is ranked based on it fitness.

Step 3: Selection – Good solutions are given a higher chance of passing their ‘genes’ to the following generations based on the value of their fitness. Individuals of high fitness will have higher chance of survival.

Step 4: Crossover – This is the main reproduction operator that takes two individuals and mix them to produce new children (i.e., solutions) that inherit genes (or properties) from the parent individuals.

Step 5: Mutation – This is an operator that works with typically a low probability to reintroduce lost genes into the population.

Step 6: Evolution – New individuals are evaluated and inserted back into the population. After ranking all individuals, the ones with high fitness remain and the ones with low fitness are removed. This method is known as the rank-based method.

Step 7: Stopping – Steps 3 through 6 are repeated until no improvement is achieved, and this is when the GA stops and reports its best solution.

The Shortest Processing Time First (SPT) Heuristic. The SPT rule was applied to the solution obtained by the GA to further minimize the total completion time. One can think of minimizing the total completion time, as minimizing the average waiting time of the patients in the system. The rationale behind the SPT rule is that this rule is known to find optimal solutions for minimizing the total completion time on identical parallel machines assuming that this objective is the only objective function (Pinedo, 2002). Since the total completion time is considered here as a secondary objective, the SPT would help finding the minimum total completion time for the solution obtained by the GA.

Using the SPT rule also helps limit the variability in the duration of the procedures. Mathematically, short procedures have smaller standard deviations. The procedures with smaller deviations have a higher probability to finish on time or, if not on time, with less waiting time for the subsequent procedures. Therefore, the smaller standard deviation for the duration of a procedure, the less, on average, the next patient will wait (Lebowitz, 2003).

Numerical Examples. Two examples are presented to show how the models with and without machine eligibility restrictions were applied to the OR scheduling problem. Both examples were adapted from real data provided by Sentara Leigh Hospital in Virginia Beach, Virginia.

Example 1. The data used in this example was from the busiest days for ASC OR group where there was 28 procedures scheduled for one day to be processed in all 6 rooms. The GA was implemented using a spreadsheet-based software plug-in to MS-Excel called *Evolver* from *Palisade Incorporation*. GAs were applied to find a solution that minimizes the makespan. The assignment of procedures to rooms obtained by the GA is shown in Exhibit 1.

From Exhibit 1, we can see that the decision variables are the 0's and 1's that are located in the 'room' columns. The GA was applied to assign the 0's and 1's in a way that will minimize the schedule's makespan (Cmax) while keeping the constraints satisfied. It is important to note that the reason for ordering the procedures from the smallest processing time to the largest according to the SPT rule is to ensure that the secondary objective function of minimizing the total completion time will also be reduced. The far right column, 'Procedure to room constraint', makes sure that each procedure is assigned to one and only one room. The Cmax for each room, Cmax(m), is the sum product of the decision variables for a room and the processing times of each procedure assigned to that room. For example, for Room 1, Cmax(1) is the sum product of 1*p(4) + 1*p(10) + 1*p(18) + 1*p(6) + 1*p(25) + 1*p(15) = 30 + 30 + 30 + 45 + 45 + 47 = 227 minutes. The overall Cmax for all rooms is max{Cmax(1), Cmax(2), …, Cmax(6)} = 240 minutes.

The results show that the GA tried to balance the procedure workload across the rooms by minimizing the sum of the processing times of each room; in this case, 4 of the 6 rooms had a Cmax of 240 minutes. After the GA converges to its best solution, the secondary objective function of minimizing the total completion time is minimized by applying the SPT rule. For example, the SPT job sequence for Room 2 is 5, 3,7,19, and 21 with a total completion time of 30+75+120+180+240 = 645 minutes.

This example was taken for one of the busiest days for the ASC, and all procedure were scheduled in 240 minutes (or 4 hours), while typically the hospital currently take a whole day of 7.5 hours to perform all procedures. We carefully emphasize here that this is not a general conclusion to make because this was merely an example of one day. Also there are many practical aspects that we did not consider in our model such as lunch breaks, delays, and the stochastic nature of surgery procedures. This example, however, points out that there could be space for improvement if the procedures were not scheduled based on First Come First Served basis within a certain block.

Example 2. This example shows the implementation of the identical parallel machine model with machine eligibility restrictions to minimize the makespan. The same data set from Example 1 was used for this example with the addition of the machine eligibility constraint as shown in Exhibit 2. In this example the machine eligibility constraint for each job is defined in the 'Machine Required' column. The machine required for each job corresponds to the operating room number necessary to perform the procedure. Jobs that can be performed in any room are assigned '0' in the 'Machine Required' column and are randomly assigned to a room.

When the GA starts, the jobs that have a specific room requirement are assigned to that room. As for the jobs that do not have any specific machine requirement (with 0 assignment in the 'Machine Required' column), they will be assigned randomly to any of the 6 ORs.

For this example, the makespan obtained by GA was 270 minutes. The SPT rule can be also applied in this model to minimize the secondary objective function of the total completion time.

Conclusion

The Operating Room scheduling problem is addressed in this paper. The problem with and without eligibility restrictions is modeled as an identical parallel machine scheduling problem. The primary objective function is minimizing the makespan and the secondary objective function is minimizing the total completion time of patients. Genetic Algorithms have proven to be successful in solving large-scale combinatorial optimization problems and they have been applied to the problem at hand to minimize the makespan. The Shortest Processing Time first rule was used to minimize the secondary objective function of the total completion time. Two examples were presented to show how the problem can be modeled as an identical parallel machine scheduling problem.

The GA showed to reach a good solution for both examples. The GA, however, does not does guarantee finding optimal solutions. More computational experiments will be necessary to make conclusions on the performance of the solution methodology compared to the existing scheduling methodology.

Exhibit 1. Identical Parallel Machine For Minimizing The Makespan.

Job	Proc Time	Room 1	Room 2	Room 3	Room 4	Room 5	Room 6	Procedure to room constraint
4	30	1	0	0	0	0	0	1
5	30	0	1	0	0	0	0	1
9	30	0	0	1	0	0	0	1
10	30	1	0	0	0	0	0	1
12	30	0	0	0	1	0	0	1
14	30	0	0	0	0	0	1	1
17	30	0	0	0	0	1	0	1
18	30	1	0	0	0	0	0	1
2	45	0	0	1	0	0	0	1
3	45	0	1	0	0	0	0	1
6	45	1	0	0	0	0	0	1
7	45	0	1	0	0	0	0	1
11	45	0	0	1	0	0	0	1
25	45	1	0	0	0	0	0	1
26	45	0	0	0	0	1	0	1
27	45	0	0	0	0	0	1	1
15	47	1	0	0	0	0	0	1
8	60	0	0	0	1	0	0	1
16	60	0	0	1	0	0	0	1
19	60	0	1	0	0	0	0	1
20	60	0	0	0	0	0	1	1
21	60	0	1	0	0	0	0	1
22	60	0	0	1	0	0	0	1
24	60	0	0	0	1	0	0	1
1	75	0	0	0	0	1	0	1
13	90	0	0	0	0	0	1	1
23	90	0	0	0	0	1	0	1
28	90	0	0	0	1	0	0	1
	C max (m)	227	240	240	240	240	225	
	Cmax =	240						

Exhibit 2. Identical Parallel Machine For Minimizing The Makespan With Machine Eligibility

Cmax	270

Optimized Values			ROOM 1			ROOM 2			ROOM 3			ROOM 4			ROOM 5			ROOM 6		
JOB ID	Pj	Machine Required	Start	Comp	Loc.	Start	Comp	Loc.	Start	Comp	Loc.	Start	Comp	Loc.	Start	Comp	Loc.	Start	Comp	Loc.
9	30	0	0	0		0	0	here	0	0		0	0		0	0		0	0	
23	90	0	0	0		0	0		0	0		0	0	here	0	0		0	0	
18	30	0	0	0		0	0		0	0		0	0		0	0		0	0	here
5	30	5	0	0		0	0		0	0		0	0		0	30		0	0	
19	60	0	0	0		0	0		0	0	here	0	0		0	0		0	0	
8	60	0	0	0		0	0		0	0		0	0		0	0	here	0	0	
22	60	0	0	0		0	0		0	0		0	0		0	0		0	0	here
26	45	2	0	0		0	45		0	0		0	0		0	0		0	0	
17	30	0	0	0	here	0	0		0	0		0	0		0	0		0	0	
24	60	0	0	0		0	0		0	0	here	0	0		0	0		0	0	
21	60	0	0	0		0	0	here	0	0		0	0		0	0		0	0	
7	45	0	0	0		0	0	here	0	0		0	0		0	0		0	0	
3	45	3	0	0		0	0		0	45		0	0		0	0		0	0	
13	90	0	0	0		0	0		0	0		0	0		0	0	here	0	0	
10	30	0	0	0		0	0		0	0		0	0		0	0		0	0	here
6	45	6	0	0		0	0		0	0		0	0		0	0		0	45	
20	60	0	0	0	here	0	0		0	0		0	0		0	0		0	0	
27	45	3	0	0		0	0		45	90		0	0		0	0		0	0	
25	45	1	0	45		0	0		0	0		0	0		0	0		0	0	
4	30	4	0	0		0	0		0	0		0	30		0	0		0	0	
11	45	0	0	0		0	0		0	0	here	0	0		0	0		0	0	
16	60	6	0	0		0	0		0	0		0	0		0	0		45	105	
2	45	2	0	0		45	90		0	0		0	0		0	0		0	0	
28	90	4	0	0		0	0		0	0		30	120		0	0		0	0	
1	75	1	45	120		0	0		0	0		0	0		0	0		0	0	
14	30	4	0	0		0	0		0	0		120	150		0	0		0	0	
15	47	5	0	0		0	0		0	0		0	0		30	77		0	0	
12	30	0	0	0		0	0		0	0		0	0	here	0	0		0	0	
			Cj_1 =	120		Cj_2 =	90		Cj_3 =	90		Cj_4 =	150		Cj_5 =	77		Cj_6 =	105	
				90			135			165			120			150			120	
			Total	210		**Total**	225		**Total**	255		**Total**	270		**Total**	227		**Total**	225	

References

Eiselt H.A., C.L. Sandblom, *Decision Analysis, Location models, and scheduling problems*, Springer (2004).

Garey, M. R. and Johnson, D. S, *Computers and intractability: A guide to the theory of NP-completeness*, W. H. Freeman and Company. (1979)

Lebowitz, P.W., “Schedule the Short procedure First to improve OR efficiency,” *AORN Journal*, Vol.78, No.4 (Oct 2003), pp. 651-659.

Min, Lui, W. Cheng, “A Genetic Algorithm for minimizing the makespan in the case of scheduling identical parallel machines”, *Artificial Intelligence in Engineering*, Vol. 13 (March, 1999), pp. 399-403.

Pinedo, Michael, *Scheduling Theory, Algorithms, and Systems*, second edition, Prentice Hall (2002).

About the Author(s)

Galina Tsoy

Galina Tsoy is an MS student in Engineering Management at Old Dominion University. She has a bachelor degree in Civil Engineering Technology at Old Dominion University. She is currently finishing my thesis on queuing theory to estimate bed allocation in medical units and also on estimating surgical procedure times in the health care industry. She is a member of Society of Women Engineers and a member of Institute of Operations Research and Management Science.

Jean-Paul Arnaout

Mr. Arnaout is a PhD candidate in the department of Engineering Management and Systems Engineering at Old Dominion University. He received his M.E. in Engineering Management (2003) from Old Dominion University, and his B.S. in Mechanical Engineering from University of Balamand, LEB.

Current research interest is predictive-reactive scheduling of parallel machines under machine breakdown.

Timm J. Smith

Mr. Smith is a senior Mechanical Engineer for the Department of the Navy. He has over 22 years of experience in preventive and predictive shipboard maintenance. He also has several years of project management experience. Mr. Smith has a Masters of Engineering Management degree from Old Dominion University. He is a member of the Project Management Institute and the Naval Civilian Managers Association.