Proceedings of the 2008 Winter Simulation Conference
S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, J. W. Fowler eds.

MAXIMIZING THE UTILIZATION OF OPERATING ROOMS WITH STOCHASTIC TIMES USING SIMULATION

Jean-Paul M. Arnaout
Sevag Kulbashian

Department of Industrial Engineering
Lebanese American University
Byblos, LEBANON

ABSTRACT

This paper addresses a surgery rooms scheduling problem. The problem is modeled as a parallel machine scheduling problem with sequence dependent setup times and an objective of minimizing the makespan. This is a NP-hard problem and in this paper, a solution heuristic is developed and compared to existing ones using simulation. The results and analysis obtained from the computational experiments proved the superiority of the proposed algorithm *LEPST* over the other algorithms presented.

1 INTRODUCTION AND BACKGROUND

Now more than ever, the healthcare industry needs to respond to the swell in challenges and pressures in its environment. The constant increase in number of patients and surgeries require new unconventional methods that aim at increasing the efficiency and particularly improving the Operating Room (OR) utilization. The latter is one of the largest revenue and cost centers in the healthcare industry (Tsoy et al. 2004); i.e. it has a great potential of reducing expenses, especially that it affects many parts in the hospital such as resource management, financial management, purchasing, and many other areas along with the patients' waiting times.

The current norm in hospitals is that patients wait long hours before they are checked in and prepared for surgery. Unsuccessful scheduling of ORs is simultaneously decreasing the rooms' utilization and increasing the patients' delays, an act that is very costly. Operations managers' main target is to attain more accurate scheduling techniques in order to improve the ORs utilization.

Following the above, this paper addresses the problem of operating rooms' scheduling (non emergency cases) inside a local hospital with an objective of maximizing the utilization of the rooms using a dispatching rule. A dispatching rule is a priority rule, based on which the operations are organized in the different rooms. Currently, most hospitals including the one in this study assign block times to surgeons' groups based on their requests. 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.

In our local hospital, the operations that are non emergency cases are scheduled and planned on a daily basis. There is a special room for the emergency cases, which does not affect the OR scheduling. The hospital has four operating rooms, and eleven different major types of operations are executed in those rooms. Each operation type has a certain stochastic duration, and also a preparation phase before the operation. The duration of the preparation is affected by the sequence of the operations in a particular OR. This scheduling problem when translated to machine scheduling becomes the scheduling of jobs with sequence dependent setup times on identical parallel machines with the objective of minimizing the maximum completion time. This is at least a NP-hard problem as the simplified problem of two identical machines with no setup times is NP-hard in the ordinary sense (Bruno, Downey, and Frederickson 1981). Discrete event simulation will be used to model and test three heuristics for the problem addressed in this paper.

The rest of this paper is organized as follows. In section 2, the related research is summarized. In section 3, the problem statement and objective function are presented. Section 4 contains description of the heuristics developed and used. The simulation model verification and validation are presented in section 5; the computational results and output analysis are described in section 6. Finally, we conclude our results in section 7.

2 RELATED RESEARCH

The maximization of the OR utilization has been the aim of several researchers. Few of them used simulation as a tool to model and solve healthcare scheduling problems.

978-1-4244-2708-6/08/$25.00 ©2008 IEEE

Shmitz and Kwak (1972) developed a Monte Carlo simulation model for OR and recovery room usage. The authors used their model to determine the improvements if bed capacity is increased, and the surgical procedures' time and space requirements for both the operating and recovery rooms. Dexter et al. (1999) used computer simulation for scheduling operating rooms' Block Time in order to improve utilization; no preparation time was considered in their paper. Dexter (2000) analyzed the effects of scheduling on operating rooms' labor costs, and the effect of having a high OR utilization on the salaries of the personnel. Moreover, the author addressed some operational goals. Tsoy et al. (2004) introduced a genetic algorithm approach for the surgery operating rooms scheduling problem. The authors assumed the operations times to be deterministic. Jebali, Alouane, and Ladet (2006) introduced a two-step approach for operating room scheduling, where operations are assigned to ORs in the first step, and then these operations are sequenced in the second step. The authors considered all times to be deterministic. For more readings on techniques to construct the operating room schedule, the reader may refer to (Guinet and Chaabane 2003; Kharraja, Chaabane, and Marcon 2002).

The most common objectives studied in parallel machine scheduling are minimization of completion time, tardiness, and makespan. The identical parallel machine problem with the objective of minimizing the makespan has been proven to be a NP problem (Sethi 1977; Garey and Johnson 1979). Due to this complexity, it became a common and acceptable practice to find suitable heuristics instead of optimal solutions for such problems. Following the conventional scheduling notation, the identical parallel machine problem can be denoted by $P_m//C_{max}$, where *m* refers to the number of machines, and C_{max} to the maximum completion time (makespan). In this problem (without setup times) with jobs having a deterministic processing time, the LPT (largest processing time) heuristic gives near optimal results (Graham 1969; Pinedo 1995). As for the same problem but with jobs having a stochastic processing time, the LEPT (largest expected processing time) rule minimizes the expected makespan in the class of nonpreemptive jobs (Pinedo 1995). Lee and Pinedo (1997) addressed the same problem with setup times where all times are deterministic and presented a three phase heuristic for minimizing the sum of the weighted tardiness. Monma and Potts (1989) addressed the problem of two identical parallel machines with batch setup times and claimed that pseudo polynomial algorithms exist for minimizing the makespan when the number of batches is fixed. Min and Cheng (1999) introduced a genetic algorithm for the same objective but without setup times and where the processing times are deterministic.

Stochastic machine scheduling problems have been considered, among others, by Glazebrook (1979), Weiss and Pinedo (1980), Bruno, Downey, and Frederickson (1981), Weber, Varaiya, Walrand (1986), Weiss (1992), Mohring, Shulz, and Uetz (1999), Javidi, Song, and Teneketzis (2001), and Arnaout, Rabadi, and Mun (2006). However, and up to our knowledge, no previous research has addressed the generation of schedules in identical parallel machines with stochastic processing and sequence dependent setup times, and this is where the contribution of this paper lies.

3 PROBLEM STATEMENT

The main objective of this research is to find a dispatching rule to schedule the operations in the ORs in a way that maximizes the utilization of the rooms.

3.1 The OR Scheduling Problem

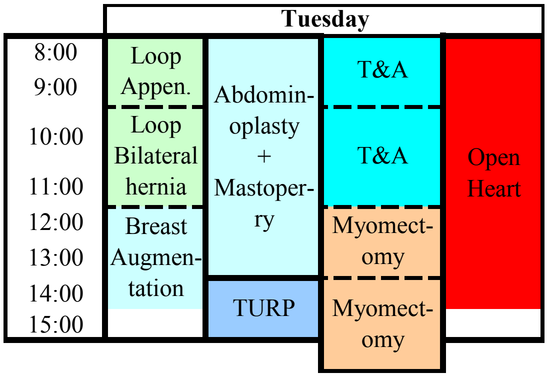
Scheduling is present in many areas of the hospital, one of which is the OR. The assignment of daily operations affects several parts of the hospital, e.g. resource management, financial management, and patients waiting times. As the salaries of the staff accounts for most OR costs, it is of great importance that the number of staff remains at its critical needed value, because in the OR scheduling case, the best way to maximize the personnel utilization is to maximize the OR utilization (Dexter et al. 1999).

The OR scheduling problem is translated to machine scheduling as follows:

- The operating rooms are regarded as machines.
- The operations are regarded as jobs.
- Operations can be performed in any room, as on any machine.
- Operations can be performed at any time which is translated as job independence.
- Operations need a preparation of equipment and personnel before they start, this translates to that jobs have a sequence dependent setup time.
- The objective is to organize operations in a way to complete all the operations as soon as possible, by minimizing the maximum completion time at the end of the working day. This can be translated to the minimization of the maximum completion time C_{max}.

In the case of this paper, we have chosen the scheduling of the operating rooms inside a local hospital in Lebanon. In table 1, a sample schedule of daily operations is given. Figure 1 presents the OR problem as a machine scheduling representation, where each block is an operation (job) and the dotted line divides between the preparation (setup) and the processing times. Note that the scheduling of the operations is done on a daily basis.

Table 1: An example of four ORs scheduling



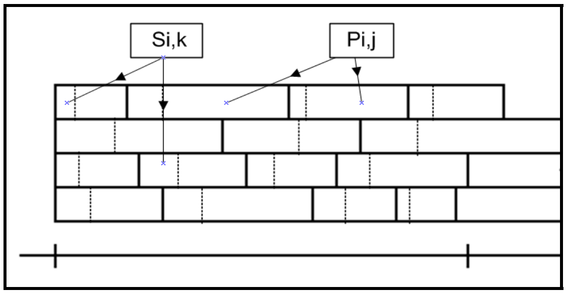


Figure 1: Representation of a 4 machine scheduling model with sequence dependent setup times (Fredi et al. 2001).

Some of the constraints/assumptions for this problem are listed as follows. First, the operations are considered independent from one another; the operating rooms are identical and parallel meaning any operation can be scheduled on any OR at any time; the scheduling of operations is a daily process, the quantity of operations scheduled is based on the daily capacity, and all the operations are scheduled before the day begins.

The problem statement in machine scheduling terms is the following:

$$P_m/S_{ik}/\ C_{max}$$

Where,
P_m: m parallel identical machines.
S_{ik}: The jobs are constrained with setup times.
C_{max}: The objective is to minimize C_{max} (maximum completion time).

3.2 Problem Constraints

There are m identical parallel machines, and n independent jobs to be scheduled on the machines. Each machine is able to process any type of jobs and a job has the same processing time on any machine (Identical). The jobs are simultaneously available at the beginning of the scheduling horizon (at time zero). Further more, each job can be processed on any of the machines but needs to be processed by one machine only, and each machine is capable of processing only one job at a time. Job preemption is not allowed and there is no processing precedence on any of the machines. The machine setup times are dependent on jobs' sequence where setup times depend on both the job just completed and the next job to be processed. Setup times are assumed to be machine independent such that regardless to which machine jobs *k* and *i* are assigned, s_{ki} would be the setup time required if job *i* is scheduled after job *k*.

Our main objective is to have the maximum utilization of the operating rooms. The makespan, defined as max (C_1, C_2…..,C_n), is equivalent to the completion time of the last job to leave the system. A minimum makespan usually implies a high utilization of the machines (Pinedo 1995). Hence, by minimizing C_{max}, we maximize the utilization of the operating rooms.

4 HEURISTIC ALGORITHMS

In the following sections, different heuristics are presented and compared in order to determine the most appropriate one for our problem. The jobs' processing and setup times are stochastic and drawn from different uniform distributions. Whenever a job is called by any algorithm to be sorted with the other jobs or sent to a machine, it will be assigned a processing time and setup time following some uniform distribution; this is discussed more in section 6.

4.1 LPT and SPT

Longest Processing Time First (LPT) rule gives the priority to the job that has the longest processing time to be scheduled first. Under LPT, the shortest job is the last job to start its processing and also the last job to finish its processing.

Shortest Processing Time First (SPT) gives the priority to the job that has the smallest processing time to be scheduled first. Under SPT, the longest job is the last job to start its processing and also the last job to finish its processing.

During the schedule's execution for the multiple machine case, LPT and SPT send the jobs to the machine that has the minimum completion time.

4.2 LEPT and SEPT

LEPT and SEPT are very similar to LPT and SPT, but they differ in that they consider the average processing times of jobs. In other words, the advantages over the LPT and SPT are the ability to schedule jobs in stochastic environments (Pinedo 1995).

In order to use the LEPT or SEPT, we must calculate the averages of the expressions of processing times. In particular, whenever we want to schedule a job on the machines, LEPT will pick from the unscheduled jobs the one with the maximum EP_i, while SEPT will pick the job with

the minimum EP_i; the chosen job using either rule will be sent to the machine with the smallest completion time.
Note that EP_i is the expected processing time of job *i* on any machine for *i* = 1,…, n.
The limitation of LEPT and SEPT for the OR problem is that they do not account for setup (preparation) times.

4.3 Longest Expected Processing with Setup Time (LEPST)

The LEPST is the dispatching rule that is introduced for OR scheduling in this study; it is an extension to LEPT where the setup times are considered. The main challenge of the problem is the consideration of the sequence dependent setup time. It is shown that the $P_m/S_{j,k}/C_{max}$ problem is strongly NP-hard (Pinedo 1995).

Let *S* be a set containing the unscheduled jobs.

1. Find job *i* and machine *j* where (1) is at its maximum:

$$[C_{kj} + (EP_i * \alpha) + Max(S_{j,k,i})] \quad (1)$$

   where $i \in S$ (index of unscheduled jobs), *j* is the machine index, and *k* is the previous job on that specific machine *Mj*. C_{kj} refers to the completion time of the last job on machine *j*, and the control parameter α value was determined from Figure 2.
2. After finding both *i* and *j*, assign job *i* to machine *Mj*, and remove job *i* from list *S*.
3. If *S* = Ø, STOP; else go to *Step* 1.

In *LEPST*, by giving the priority to jobs that have the maximum setup time, there will be less jobs having high setup times assigned at the end of the schedule; hence affecting less the C_{max}.

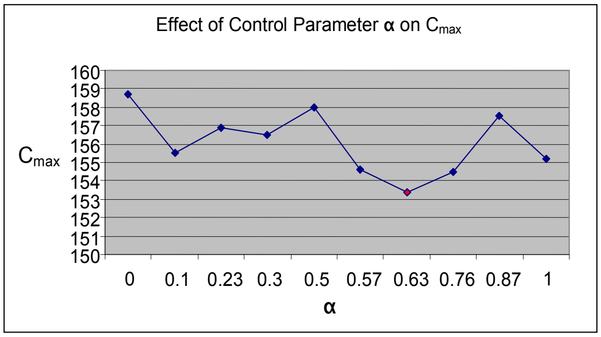


Figure 2: Control Parameter α

The chart in Figure 2 describes how the maximum completion time of jobs fluctuates when α is changed while applying *LEPST* to the problem at hand. Note that the values of α are between 0 and 1, where 1 indicates that the processing and setup times will have equal weights when selecting a job, and 0 indicates that only setup time will decide on the job selection. We can conclude from Figure 2 that the algorithm is giving the best solution when α = 0.63, and this will be the value to be used in the proposed heuristic *LEPST*.

In order to test the importance of α, we compared *LEPST* to itself when α was set equal to 1; i.e. given both processing and setup times the same weight. One hundred replications were run for both models and a paired t-test was conducted on the output. Output Analyzer from Rockwell Systems was used for the analysis and the result indicated that on a 95% Confidence Interval, α = 0.63 will lead to better Cmax that α = 1 and the difference is statistically significant. The output analyzer report is shown in Figure 3 below.

5 MODEL VERIFICATION

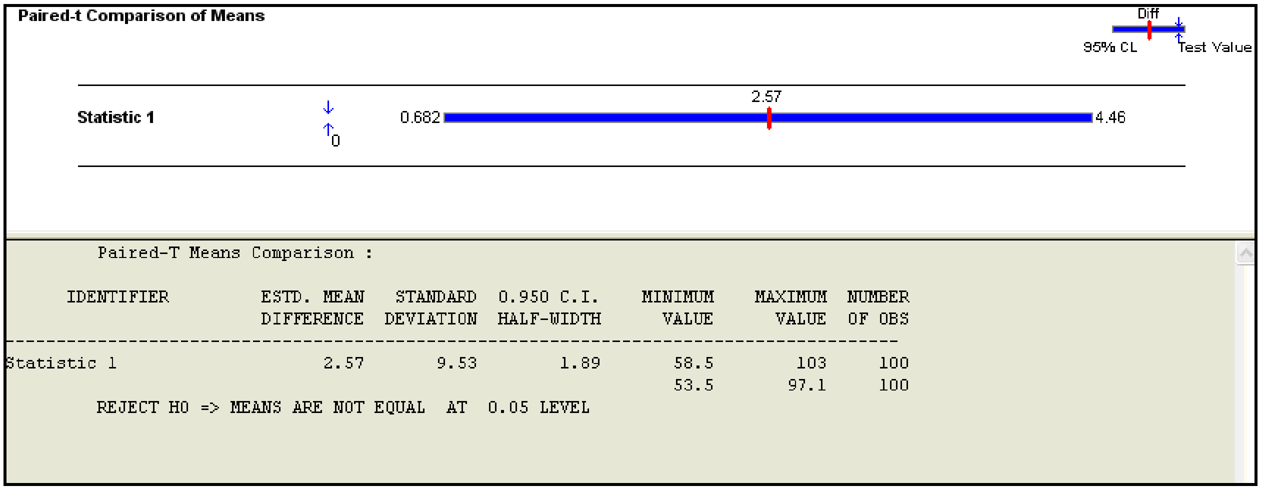
Verification is the process of ensuring that the simulation model behaves in the way it was intended according to the modeling assumptions made (Kelton et al. 2004).

Different methods were applied in verifying the behavior of our models:

- We used first deterministic data instead of stochastic data for both the processing and setup times; this allowed us to predict the system's behavior.
- We let only a single entity enters the system, and then followed this entity through all the decisions nodes to ensure that the model's logic is correct.
- We monitored the model's animation, which made it easier to detect any errors in our logic.
- Finally, we put several variable animations, which enabled us to determine which batch number is first scheduled, and which batch is separated.

6 COMPUTATIONAL TESTS

The above heuristics have been modeled and compared using the simulation software Arena. The popularity of simulation has been increasing over the past decade mainly due to its ability to deal with very complicated models of correspondingly complicated systems (Kelton, Sadowski, and Sturrock 2004). The reason stochastic data was used for the processing and setup times is to ensure a more real representation of the healthcare environment, where most of the time an operation will not finish on a specific time, but on a range between two times. Simulation is considered to be one of the best approaches to deal with such source of randomness. The jobs' (operations) processing times and machines' setup (preparation) times



```
        Paired-T Means Comparison :

     IDENTIFIER         ESTD. MEAN   STANDARD  0.950 C.I.    MINIMUM    MAXIMUM  NUMBER
                        DIFFERENCE  DEVIATION  HALF-WIDTH     VALUE      VALUE   OF OBS
-----------------------------------------------------------------------------------
Statistic 1                   2.57       9.53        1.89      58.5        103     100
                                                               53.5       97.1     100
        REJECT H0 => MEANS ARE NOT EQUAL  AT  0.05 LEVEL
```

Figure 3: Comparing Cmax for α=0.63 and α=1

are stochastic following uniform distributions; these times were drawn from the hospital real data and a sample of operations times is as follows:

Operation type I → U[1.2,1.5]
Operation type II → U[2.2,2.3]
Operation type III → U[6.5,6.9], etc…

Recall that these values will not be known until the operation is actually being performed in the OR (machine). The reason uniform distributions were used is due to their high variances, ensuring that the presented heuristics are being tested under unfavorable conditions (Weng, Lu, and Ren 2001).

Our simulation model logic is described in Figure 4. Each of the tested heuristics was run for 75 replications (determined through experiments to deliver decent confidence intervals), and paired t-tests were conducted between all the rules (two rules at a time), and a difference between two rules is proven to be statistically significant whenever the p-value was less than 0.05. These tests proved (for 95% statistical significance) the following: LEPST performed the best, LEPT second, and SEPT last.

In particular, the results indicated that the utilization of the ORs inside the local hospital in this study can be increased by using the LEPST algorithm with an α of 0.63. Over a considerable period of time, the rule resulted in a reduction of about 20 minutes per day. The logic behind α = 0.63 is that there is approximately 60% weight to the processing time when assigning an operation. This can be understood as the ratio of processing/setup time of the hospital is relatively high.

7 CONCLUSIONS

In this paper, we have introduced an effective heuristic algorithm, *LEPST*, for maximizing the utilization of operating rooms in a local hospital in Lebanon. This problem is equivalent to the minimization of makespan on identical parallel machines with sequence dependent setup times. *LEPST* was compared to two other algorithms, *LEPT* and *SEPT*. All three algorithms were modeled and tested through simulation, and our conclusions were drawn using a large number of replications and several statistical tests. Computational experiments showed that *LEPST* significantly outperformed the other algorithms.

While the conducted tests show very promising results for OR scheduling, it is worth noting that the rules were compared under static schedules; i.e. operations were available at time 0 of each day. An extension to this work would be to test *LEPST* (or an expansion of it) under dynamic schedules where operations are arriving over time; such study would be also appropriate for emergency room scheduling.

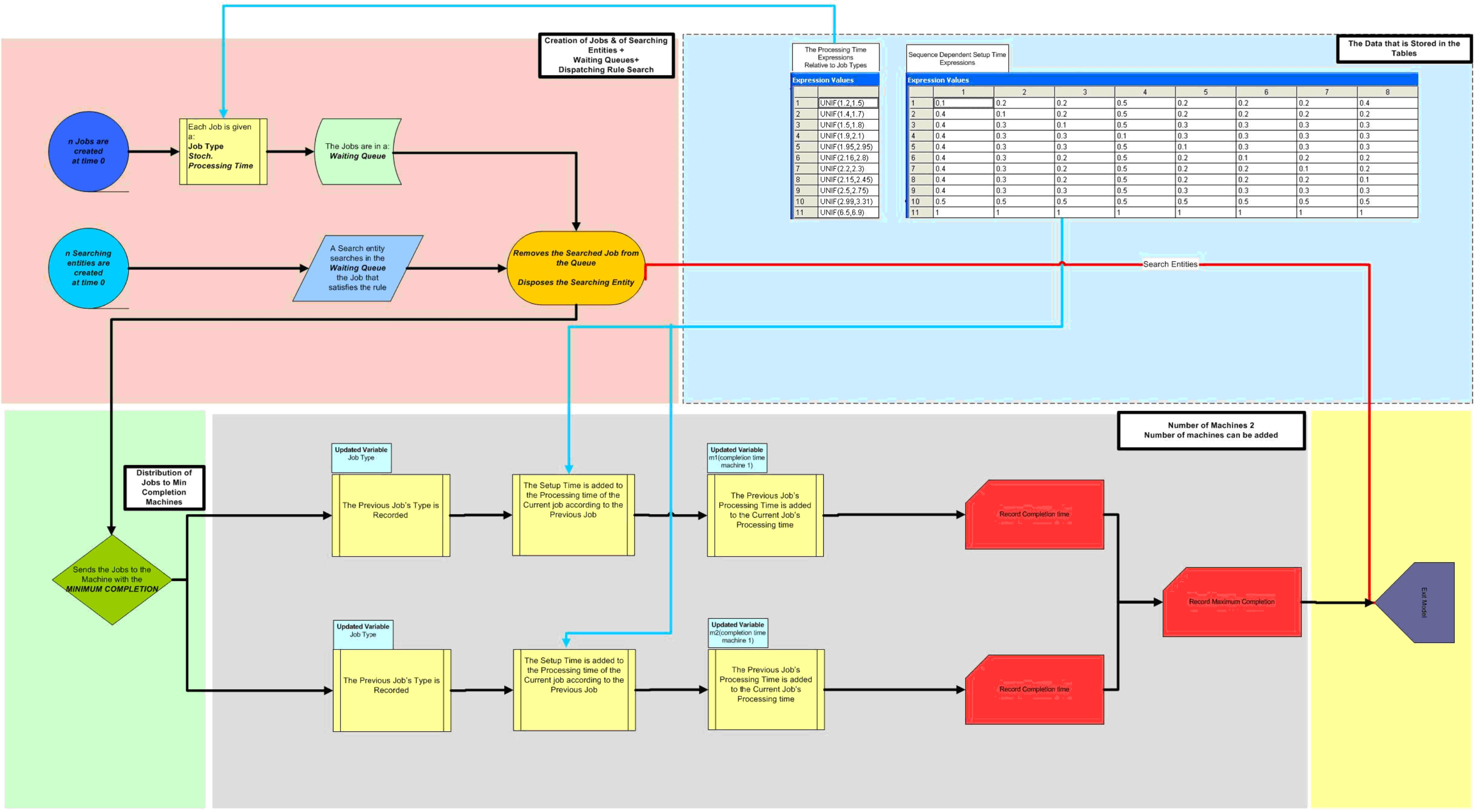
REFERENCES

Arnaout, J-P., G. Rabadi, and J.H. Mun. 2006. A Dynamic Heuristic for the Stochastic Unrelated Parallel Machine Scheduling Problem. *International Journal of Operations Research* 3: 136-143.

Bruno, J.L., P.J Downey, and G.N Frederickson. 1981. Sequencing tasks with exponential service times to minimize the expected flow time or makespan. *Journal of the ACM* 28: 100-113.

Dexter, F., A. Macario, R Traub, M. Hopwood, and D. Lubarsky. 1999. An operating room scheduling strategy to maximize the use of operating room block time: computer simulation of patient scheduling and survey of patients' preferences for surgical waiting time. *Anesthesia & Analgesia* 89: 7-20.

Dexter, F. 2000. A strategy to decide whether to move the last case of the day in an operating room to another empty operating room to decrease overtime labor costs. *Anesthesia & Analgesia* 91: 925-928.


Creation of Jobs & of Searching Entities + Waiting Queues+ Dispatching Rule Search
n Jobs are created at time 0
Each Job is given a: Job Type Stoch. Processing Time
The Jobs are in a: Waiting Queue
n Searching entities are created at time 0
A Search entity searches in the Waiting Queue the Job that satisfies the rule
Removes the Searched Job from the Queue
Disposes the Searching Entity
The Data that is Stored in the Tables
Search Entities
Distribution of Jobs to Min Completion Machines
Sends the Jobs to the Machine with the MINIMUM COMPLETION
Number of Machines 2
Number of machines can be added
Updated Variable Job Type
The Previous Job's Type is Recorded
The Setup Time is added to the Processing time of the Current job according to the Previous Job
Updated Variable m1(completion time machine 1)
Updated Variable m2(completion time machine 1)
The Previous Job's Processing Time is added to the Current Job's Processing time
Record Completion time
Record Maximum Completion
Exit Model

The Processing Time Expressions Relative to Job Types

Expression Values

1	UNIF(1.2,1.5)
2	UNIF(1.4,1.7)
3	UNIF(1.5,1.8)
4	UNIF(1.9,2.1)
5	UNIF(1.95,2.95)
6	UNIF(2.16,2.8)
7	UNIF(2.2,2.3)
8	UNIF(2.15,2.45)
9	UNIF(2.5,2.75)
10	UNIF(2.99,3.31)
11	UNIF(6.5,6.9)

Sequence Dependent Setup Time Expressions

Expression Values

	1	2	3	4	5	6	7	8
1	0.1	0.2	0.2	0.5	0.2	0.2	0.2	0.4
2	0.4	0.1	0.2	0.5	0.2	0.2	0.2	0.2
3	0.4	0.3	0.1	0.5	0.3	0.3	0.3	0.3
4	0.4	0.3	0.3	0.1	0.3	0.3	0.3	0.3
5	0.4	0.3	0.3	0.5	0.1	0.3	0.3	0.3
6	0.4	0.3	0.2	0.5	0.2	0.1	0.2	0.2
7	0.4	0.3	0.2	0.5	0.2	0.2	0.1	0.2
8	0.4	0.3	0.2	0.5	0.2	0.2	0.2	0.1
9	0.4	0.3	0.3	0.5	0.3	0.3	0.3	0.3
10	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
11	1	1	1	1	1	1	1	1

Figure 4: Simulation model flow and Logic

Garey, MR, and DS Johnson. 1979. Computer and intractability: a guide to the theory of NP-completeness. W H Freeman: San Francisco.

Glazebrook, K.D. 1979. Scheduling tasks with exponential service times on parallel machines. *Journal of Applied Probability* 16: 685-689.

Graham, R.L. 1969. Boundaries on multiprocessing timing anomalies. *SIAM Journal of Applied Mathematics* 17: 416–429.

Guinet, A., and S. Chaabane. 2003. Operating theatre planning. *International Journal of Production Economics* 85: 69–81.

Javidi, T., N-O. Song, and D. Teneketzis. 2001. Expected makespan minimization on identical machines in two interconnected queues. *Probability in the Engineering and Informational Sciences* 15: 409-443.

Jebali, A., A. Alouane, and P. Ladet. 2006. Operating rooms scheduling. *International Journal of Production Economics* 99: 52-62.

Kelton, D., R. Sadowski, and D. Sturrock. 2004. *Simulation with Arena.* 3rd ed. McGraw-Hill Companies, New York.

Kharraja, S, S. Chaabane, and E. Marcon. 2002. Evaluation de performances pour deux stratégies de programmation opératoire de bloc. In: *Actes de la 2eme Conf Int Francophone d'Automatique*, Nantes, France.

Lee, Y., and M. Pinedo. 1997. Scheduling jobs on parallel machines with sequence-dependent setup times. *European Journal of Operational Research* 100: 464-474.

Min, L., and W. Cheng. 1999. A genetic algorithm for minimizing the makespan in the case of scheduling identical parallel machines. *Artificial Intelligence in Engineering* 13: 399-403.

Mohring, R., A. Shulz, and M. Uetz. 1999. Approximation in stochastic scheduling: the power of LP-based priority policies. *Journal of the ACM (JACM)* 46: 924-942.

Monma, CL., and CN. Potts. 1989. On the complexity of scheduling with batch setups. *Operations Research* 37: 798-804.

Pinedo, M. 1995. *Scheduling: Theory, Algorithms, and Systems*. Prentice Hall: New Jersey.

Sethi, R. 1977. On the complexity of mean flow time scheduling. *Mathematics of Operations Research* 2:320-330.

Shmitz, H., and N. Kwak. 1972. Monte Carlo Simulation of Operating-Room and Recovery-Room Usage. *Operations Research* 20: 1171-1180.

Tsoy, G., J-P. Arnaout, T. Smith, and G. Rabadi. 2004. A Genetic Algorithm Approach for Surgery Operating Rooms Scheduling Problem. In: *Proceedings of the 25th National Conference of the American Society for Engineering Management*, Alexandria, Virginia.

Weber, R.R., P. Varaiya, and J. Walrand. 1986. Scheduling jobs with stochastically ordered processing times on parallel machines to minimize expected flowtime. *Journal of Applied Probability* 23: 841-847.

Weiss, G., and M. Pinedo. 1980. Scheduling tasks with exponential service times on non-identical processors to minimize various cost functions. *Journal of Applied Probability* 17: 187-202.

Weng, M., J. Lu, and H. Ren. 2001. Unrelated parallel machine scheduling with setup consideration and a total weighted completion time objective. *International Journal of Production Economics* 70: 215-226.

AUTHOR BIOGRAPHIES

JEAN-PAUL M. ARNAOUT is an Assistant Professor at the Industrial Engineering Department at the Lebanese American University, Lebanon. He received his PhD and M.S. from the Department of Engineering Management and Systems engineering at Old Dominion University, Norfolk, Virginia in 2006 and 2003 respectively. He received his bachelor's degree in Mechanical Engineering from the University of Balamand, Lebanon. His Research interests include Optimization Techniques, Modeling and Simulation, and Scheduling and Rescheduling. He can be reached at `jparnaout@lau.edu.lb`

SEVAG KULBASHIAN is a BE student at the Industrial and Mechanical Engineering Department at the Lebanese American University. His Research interests include Optimization Techniques and Simulation and Modeling.