International Journal of Engineering Innovation & Research
Volume 2, Issue 2, ISSN: 2277 – 5668

A Simulation Model for the Real-Life Job-Shop Scheduling Problem

Prof. Dr. Gamal M. Nawara
Zagazig University, Faculty of
Engineering Industrial Engineering Department

Eng. Wael S. Hassanein
Zagazig University, Faculty of
Engineering, Industrial Engineering Department
Email: wsalsaid@gmail.com

***Abstract* - Job-shop scheduling problem (JSSP) is very important and has a lot of applications in industrial and service sectors. With good job-shop scheduling, facilities can save time and money. Real-life JSSP is a very big challenge to researchers because of its high complexity. Standard JSSP is a typical NP-hard problem in the strong sense without considering any additional factors. In the real-life JSSP, we need to consider one or more of some factors like transportation times, parallel processing, setup times, and so many factors according to the problem nature. In this paper, we presented a simulation model by Arena simulation software to solve an example from the OR-Library with some modifications considering some factors from the real-life JSSP to show the capability of the proposed approach as shown in section 4. The objective function is to minimize the makespan.**

***Keywords* - Job-Shop Scheduling Problem, Makespan, Simulation, Arena, Parallel Processing.**

I. Introduction

For the last 50+ years and till now the JSSP is representing a very big challenge to researchers from different disciplines, mainly Operations Research, Management Science, Computer Science, and Manufacture Science. JSSP has numerous applications in manufacturing and service sectors. In manufacturing, JSSP is the process of assigning jobs to machines satisfying the precedence and resource constraints over time such that certain objective(s) is optimized. The JSSP is one of the strongest NP-hard problems [1]. Because of this, a lot of researchers applied their different methodologies on its problems to approve their methods. Practitioners did not get benefit of most of these researches as these researchers were used to solve the benchmark problems with some assumptions as shown in section 3. These assumptions took the problem very far away from the real-life problem. In the real-life JSSP, we have to consider one or more of some factors like; transportation times, parallel processing, setup times, machine breakdowns, and so many factors according to the application. In other words, we have to relax one or more of the assumptions assumed by the researchers. In this paper, we relaxed 3 assumptions by our simulation model as shown in example in section 4.

II. Literature Review

For the last 5 decades, a lot of researchers from different disciplines were trying to solve the JSSP with their different methodologies. Some of them were just trying to prove their methods with this very complex problem. Most of these researchers were used to solve the JSS benchmarks and compare their solutions to the optimum or best known solutions for them. These benchmarks don't represent the real-life JSSP because they neglect a lot of factors highly needed in the real life JSSP. Even some recent researches just tried to solve the benchmarks without considering any factors from the real life problem, for example researchers in [2] developed a hybrid Differential Evolution—Tabu Search algorithm for the solution of the JSSP. Researchers in [3] developed a hybrid Genetic algorithm for the JSSP. Researchers in [4] developed a hybrid algorithm which combines global equilibrium search, path relinking and tabu search to solve the JSSP. Researchers in [5] developed an effective team process algorithm (TPA) for the JSSP. All the previous researchers and so many others just solved the benchmarks with their assumptions as shown in section 3 and the objective was to minimize the makespan.

Some recent researchers tried to consider one or more of the factors related to the real-life problem or in other words tried to relax one or more of the assumptions. In the following paragraph we will show some examples of these recent researches for the last 5 years.

For example in [6] the authors allowed the preemption. In [7] the authors relaxed two assumptions, they included transportation time and machine breakdowns. In [8] the authors solved the problem with random processing times under the objective of minimizing the expected total tardiness. In [9] the authors considered stochastic processing times with random machine breakdowns. In [10] the authors solved the problem with random processing times under the objective of minimizing the expected sum of storage expenses and tardiness penalties. In [11] the authors considered transportation delay having three objectives to optimize: makespan, total material handling costs, and closeness rating score. In [12] the authors developed anartificial bee colony algorithm based on problem data properties for the standard JSSP with the objective of minimizing total weighted tardiness. In [13] the authors developed a genetic algorithm for solving total weighted tardiness and sum of earliness and tardiness penalties of JSSP. In [14] the authors developed a mixed-integer linear programming model along with an electromagnetism-like algorithm for JSSP with sequence-dependent set-up times. In [15] the authors developed a simulation model for the Dynamic job-shop scheduling with sequence-dependent setup times. Finally in [16] the authors used petri nets to model the JSSP considering batches, setup times, transportation and parallel processing under the objective of minimizing the makespan.

III. Problem Description

In our paper [17] we presented the basic model for solving the standard JSSP with these assumptions:

1. Processing times are deterministic.
2. All jobs are ready to be processed at time zero.
3. Only one job can be processed on each machine at a given period of time.
4. Each job visits each machine once at most.
5. The machines are continuously available.
6. There's only one machine of each type of machines.
7. No preemption of operations is allowed.
8. The transportation times between different machines are neglected.
9. The setup times for different jobs are neglected or added to the processing time.

As we said in our paper [17] these assumptions are taking the problem very far away from the real-life problem. In this paper we will show how to relax 3 of these assumptions to simulate the real-life problem as shown in the example in section 4. The 3 assumptions which we will relax are 4, 6, and 8.

IV. Proposed Model

Our example is the famous benchmark problem Ft06 with some modifications as shown in table 1, we will relax assumptions 4, 6, and 8 from previous section as follows:

1. Any job can visit any machine more than once, it is shown here in job 2 and job 5, these two jobs will visit machine 6 two times.
2. We can include any number of identical machines of any type of machine, here there are two identical machines of type 6, as it's the bottle neck machine.
3. We can include any transportation time from any location to any location, here we will include a 3 minutes transportation time from the release point to any machine, between any two machines, and from any machine to the pickup point.

It is required to give the best schedule with the minimum make span.

Table.1. A 6x6 JSS example

J	Operation sequence (Processing time)						
1	3(1)	1(3)	2(6)	4(7)	6(3)	5(6)	-
2	2(8)	3(5)	5(10)	6(10)	1(10)	4(4)	6(10)
3	3(5)	4(4)	6(8)	1(9)	2(1)	5(7)	-
4	2(5)	1(5)	3(5)	4(3)	5(8)	6(9)	-
5	3(9)	2(3)	5(5)	6(4)	1(3)	4(1)	6(10)
6	2(3)	4(3)	6(9)	1(10)	5(4)	3(1)	-

In our basic model [17] we divided the model into 3 parts as follows:

- *Part one :* Will be responsible for creating jobs, defining all variables and attributes, releasing jobs, and starting sequencing.
- *Part two:* will be responsible for the machines, the manufacturing process, and the exportation of the information to the Excel file. This part will be done for each machine. The same process will be repeated for all machines.
- *Part three:* In this part jobs will exit the manufacturing system, and the system will dispose the parts.

Our model will consist of the same three parts with some modifications to fit our assumptions here. We will discuss these modifications to relax the 3 predefined assumptions as follows:

- *Assumption number 1:* for any job to visit any machine more than once, we can add any steps to the process plan of this part. For example instead of 7 steps for job 2 we added another step as this job has to visit machine 6 twice. In Arena from *advanced transfer* go to *sequence*, go to part 2 process plan, then add the new step with the processing time of 10 minutes.
- *Assumption number 2:* here machine 6 is the bottleneck machine so we can duplicate it to improve the efficiency of this system. To do so we will go to cell 6 in the Arena model, go to resources, and add a set of machines with *cyclical* as a selection rule. From *basic process* select *set* then add these two machines in cell 6. In *ReadWrite*module we will add an assignment as an attribute with the name of machine index, this attribute will tell us which job was processed on which machine.
- *Assumption number 3:* to add a transportation time of 3 minutes we added for all routes a route time of 3 minutes.

V. Verification and Validation

We already verified and validated our basic model in [17], and the problem Ft06 was solved by the basic model. After these modifications to the problem we will just review all the modules with their connections and their logic then all the entered data has to be reviewed. Second thing we will run the program for one replication and go to the excel file to check the solution. In this check we will review the processing times and the feasibility of the solution by following one or two of the jobs with its different movements from one machine to another with its processing and transportation times.

VI. Results and Discussion

Now the model is ready, with random as a priority rule, we will adjust it to 100 replications and get the results. Ofcourse, we can run it for more replications, the one hundred replications are an example. We used Arena software version 14 to build the model. After running the model for 100 replications, we got the minimum makesp an of 87 minutes with the machine schedules as in the six tables from 2 to 7 for the 6 cells. The run took less than two minutes on a Dell® Vostro PC (Intel® Core(TM) i5–2400 CPU @ 3.10GHZ with 4 GB RAM). Note that we

International Journal of Engineering Innovation & Research
Volume 2, Issue 2, ISSN: 2277 – 5668

have to add 3 minutes to the finish time for each jobto transfer the job from the last machine to the pickup point. Here in table 7 job 2 and 5 visited machine 6 two times this is the last 2 rows, the last column to the right shows which machine was used to process the jobs shown in the first column.

Table.2: Schedulefor cell 1

Cell 1			
Job Seq.	**Processing Time**	**Start Time**	**Finish Time**
1	3	7	10
4	5	22	27
3	9	30	39
6	10	39	49
5	3	51	54
2	10	54	64

Table.3: Schedule for cell 2

Cell 2			
Job Seq.	**Processing Time**	**Start Time**	**Finish Time**
2	8	3	11
6	3	11	14
4	5	14	19
1	6	19	25
5	3	25	28
3	1	42	43

Table.4: Schedule for cell 3

Cell 3			
Job Seq.	**Processing Time**	**Start Time**	**Finish Time**
1	1	3	4
3	5	4	9
5	9	9	18
2	5	18	23
4	5	30	35
6	1	66	67

Table 5: Schedule for cell 4

Cell 4			
job seq.	**processing time**	**Start time**	**Finish time**
3	4	12	16
6	3	17	20
1	7	28	35
4	3	38	41
5	1	57	58
2	4	67	71

Table 6: Schedule for cell 5

Cell 5			
Job Seq.	**Processing Time**	**Start Time**	**Finish Time**
2	10	26	36
5	5	36	41
4	8	44	52
3	7	52	59
6	4	59	63
1	6	63	69

Table 7: Schedule for cell 6

Cell 6				
Job Seq.	**Processing Time**	**Start Time**	**Finish Time**	**Machine Index**
3	8	19	27	1
6	9	23	32	2
1	3	38	41	1
5	4	44	48	1
2	10	39	49	2
4	9	55	64	2
5	10	61	71	1
2	10	74	84	2

Best solution for our example is represented by Gantt Chart in Fig. 1, where we have one machine from each type, and two machines from type 6 (6/1 and 6/2). We can also get extra information about the system performance from Arena reports which will be very beneficial like; machine utilization, queue length, waiting time, transfer time, and others.

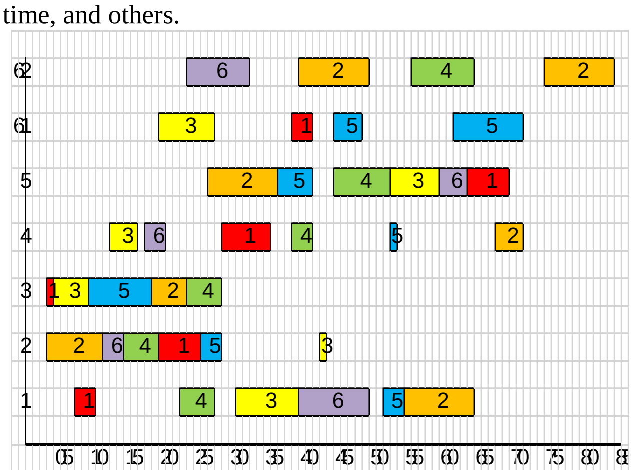

Fig.1. Gant Chart for the best schedule

VII. CONCLUSION

In this paper, we introduced a simulation model for the real-life JSSP using Arena simulation software, output analyzer, Microsoft Excel. An example considering three factors from the real-life JSSP was given. These factors were as follows; considering transportation times, allowing any job to visit any machine more than once, and using multiple resources. The model was capable for handling such problem and gave good solution in reasonable time.

VIII. Future Work

The JSSP is very famous and it has a lot of applications. There are a lot of factors and objectives to consider when needed according to the application. In the following papers we will consider extra factors like machines breakdowns, stochastic processing times, jobs will not be available at time zero, and other factors. Regarding the objective function we will consider other objectives like minimizing lateness or tardiness, we will also consider multi objective functions.

References

[1] J.K. Lenstra, A.H.G.R. Kan, P. Brucker, Complexity of machine scheduling problems, Annals of Discrete Mathematics 7 (1977) 343–362.

[2] Antonin Ponsich, Carlos A. CoelloCoello, A hybrid Differential Evolution—Tabu Search algorithm for the solution of Job-Shop Scheduling Problems, Applied Soft Computing, Volume 13, Issue 1, January 2013, Pages 462-474, ISSN 1568-4946, 10.1016/j.asoc.2012.07.034.

[3] Ren Qing-dao-er-ji, Yuping Wang, A new hybrid genetic algorithm for job shop scheduling problem, Computers & Operations Research, Volume 39, Issue 10, October 2012, Pages 2291-2299, ISSN 0305-0548, 10.1016/j.cor.2011.12.005.

[4] Mohammad Mahdi Nasiri, FarhadKianfar, A GES/TS algorithm for the job shop scheduling, Computers & Industrial Engineering, Volume 62, Issue 4, May 2012, Pages 946-952, ISSN 0360-8352, 10.1016/j.cie.2011.12.018.

[5] Li, Y., & Chen, Y. An effective TPA-based algorithm for job-shop scheduling problem. Expert Systems with Applications, 38(3), 2913–2918, (2011). doi:10.1016/j.eswa.2010.08.006

[6] Abbas Ebadi, GhasemMoslehi, An optimal method for the preemptive job shop scheduling problem, Computers & Operations Research, Volume 40, Issue 5, May 2013, Pages 1314-1327, ISSN 0305-0548, 10.1016/j.cor.2012.12.004.

[7] Kaban, a. K., Othman, Z., &Rohmah, D. S. Comparison of dispatching rules in job-shop scheduling problem using simulation: a case study. International Journal of Simulation Modelling, 11(3), 129–140, (2012).doi:10.2507/IJSIMM11 (3)2.201

[8] Zhang, R., Song, S., & Wu, C. A hybrid differential evolution algorithm for job shop scheduling problems with expected total tardiness criterion. Applied Soft Computing Journal. (2012). doi:10.1016/j.asoc.2012.02.024

[9] Lei, D. Minimizing makespan for scheduling stochastic job shop with random breakdown. Applied Mathematics and Computation, 218(24), 11851–11858, (2012). doi:10.1016/j.amc. 2012.04.091

[10] Horng, S., Lin, S., & Yang, F. Expert Systems with Applications Evolutionary algorithm for stochastic job shop scheduling with random processing time q. Expert Systems with Applications, 39(3), 3603–3610, (2012). doi:10.1016/j.eswa.2011.09.050

[11] Shah, K., Ripon, N., Glette, K., Hovin, M., &Torresen, J. Job Shop Scheduling with Transportation Delays and Layout Planning in Manufacturing Systems : A Multi-objective Evolutionary Approach, 209–219, (2012).

[12] Zhang, R. An Artificial Bee Colony Algorithm Based on Problem Data Properties for Scheduling Job Shops. Procedia Engineering, 23, 131–136, (2011). doi:10.1016/j.proeng. 2011.11.2478

[13] Omar, M. K., Anuar, N. I. B., &Supaya, Y. Genetic algorithm for solving total weighted tardiness and sum of earliness and tardiness penalties of job-shop scheduling problem. 2011 Fourth International Conference on Modeling, Simulation and Applied Optimization, 1–6, (2011). doi:10.1109/ICMSAO.2011.5775549

[14] Roshanaei, V., Balagh, A. K. G., Esfahani, M. M. S., &Vahdani, B. A mixed-integer linear programming model along with an electromagnetism-like algorithm for scheduling job shop production system with sequence-dependent set-up times. The International Journal of Advanced Manufacturing Technology, 47(5-8), 783–793, (2009).doi:10.1007/s00170-009-2210-9

[15] Vinod, V., &Sridharan, R. Scheduling a dynamic job shop production system with sequence-dependent setups: An experimental study. Robotics and Computer-Integrated Manufacturing, 24(3), 435–449, (2008). doi:10.1016/j.rcim. 2007.05.001

[16] Zhang, H., Gu, M., & Song, X. Modeling and Analysis of Real-Life Job Shop Scheduling Problems by Petri nets. 41st Annual Simulation Symposium (anss-41 2008), (2007), 279–285, (2008). doi:10.1109/ANSS-41.2008.15

[17] Gamal M. Nawara and Wael S. Hassanein. Solving the Job-Shop Scheduling Problem by Arena Simulation Software. International Journal of Engineering Innovation & Research 2(2), 161-166, (2013).

Author's Profile

Prof. Dr. Gamal M. Nawara

He has a PhD in Industrial Engineering, Leipzig University, Germany, 1969. Senior member in IEE society. Ex-Vice, President for Graduate Studies and Research, Zagazig University Egypt. Ex-Dean, Faculty of Engineering, Zagazig University Egypt. Currently Secretary General Council of Private Universities, Egypt.He is interested inProduction and Maintenance Planning and Control.

Eng. Wael S. Hassanein

He got his B.Sc. and M.Sc. in industrial engineering, Zagazig University, Egypt, 1998 and 2005. Currently he is a PhD student, industrial engineering, Zagazig University, Egypt. He is interested in Operations research, production planning and simulation.