SILESIAN UNIVERSITY OF TECHNOLOGY PUBLISHING HOUSE

SCIENTIFIC PAPERS OF SILESIAN UNIVERSITY OF TECHNOLOGY 2025
ORGANIZATION AND MANAGEMENT SERIES NO. 230

CLOUD COMPUTING FOR OPTIMIZATION JOB-SHOP SCHEDULING PROBLEM WITH CRANE IN FLEXSIM ENVIRONMENT

Piotr JANKE

Silesian University of Technology, Faculty of Organization and Management; piotr.janke@polsl.pl,
ORCID: 0000-0001-8065

Purpose: Production scheduling for job-shop problems is computationally advanced and classified as NP-hard problems. This paper develops a simulation model to optimize orders for job-shop scheduling problem with crane transport.
Design/methodology/approach: A simulation model was prepared in the FlexSim software and based on the Taillard job-shop test for 15 machines and 15 jobs. The crane was added to the model, and the layout of the production hall was prepared. The original time tables and the order on the machines were used. Created Makespan Cmax->min optimization job. This article presents the results of the search for the optimal and acceptable order sequence in production machines in this task. All calculations were performed on a computing cluster built for this purpose. The cluster consisted of 15 nodes and one master computer.
Findings: The results of the calculations were finally compared with the best solution found for this task with and without transport.
Originality/value: The approach to the problem presented in this publication circumvents the need for advanced mathematical apparatuses to describe modified classical situations in a practical situation in real production and beyond
Keywords: job-shop, FlexSim, Qptquest, cloud computing.
Category of the paper: research paper.

1. Introduction

In the production of highly customized goods and the operation of automated logistics systems, efficient scheduling is a daily challenge. Therefore, the nested problem is recognized as a standard model in planning and scheduling research. Although classical variants are well studied, the inclusion of conditions that are relevant from a practical point of view seems warranted (Lange, Werner, 2019). The workshop scheduling problem is one of the many classical planning and scheduling problems and is included in the NP class of problems. Problems of this type are computationally complex, especially for large tasks. In the classical approach, a mathematical apparatus with many assumptions is built to solve the task, in which it is difficult to take into account the practical realities that actually occur on the line. Especially, the problem arises when in a task of this type we want to take into account the logistics occurring in real production.

The main purpose of this article is to present a solution to the classic work-shop problem with modifications to make the problem realistic for the designed assembly hall. Transportation is taken into account in the article. Due to the computational complexity of the task, an experiment was designed using the concept of cloud computing.

The structure of the article is as follows: first, the basic types of mathematical problems related to the method and organization of production are explained. Next, current literature research related to the topic is presented. The next section presents the methodology of the approach to the problem, along with the layout of the production hall and the organization of the environment for calculations. The next step presents the results and a discussion. The whole is summarized.

2. Literature review

Production systems, such as flow-shop, job-shop and open-shop, differ significantly in how they organize and manage production processes. Each of these systems has its own unique characteristics that affect the production efficiency and flexibility. Flow-shop is a system in which all jobs must pass through machines in the same order. This arrangement simplifies scheduling but can lead to less flexibility in the event of disruptions. This is particularly beneficial in mass production, where repeatability and process standardization are key (Pinedo, 2012)**.** The job shop is a more flexible system in which jobs can pass through machines in a different order, depending on the specifications of each job. This arrangement allows for more flexibility and customization in production, but requires more complex scheduling. This is often used in unit or small batch production, where product variety is high (Sobaszek et al., n.d.). Open-shop is a system in which there is no fixed order in which tasks pass through machines. Tasks can be processed in any order on each machine, which allows for maximum flexibility, but also complicates the scheduling process. This type of system is used in situations where task priorities can change dynamically and flexibility is key (Pinedo, 2012). Differences between these systems are important for the efficiency and flexibility of production processes. Choosing the right system depends on the specifics of production and the requirements for flexibility and disturbance management.

In recent years, a lot of research has emerged on optimizing scheduling in various industrial contexts. Both flow-shop and nested problems are considered in the research (Alizadeh et al., 2020) presented a modified genetic algorithm to schedule outpatient visits, taking into account patient priorities and high demand for medical services. This research underscores the importance of effective medical resource management to improve service quality. Faraji Amiri, Behnamian (2020) have addressed the problem of scheduling in a green flow-shop with multiple targets under uncertainty, using a decomposition estimation algorithm. Their work focuses on sustainability and minimizing environmental impact in production processes An et al. (2020) have proposed a hybrid evolutionary algorithm to optimize production scheduling and cutting tool maintenance, taking into account total energy consumption. Their research shows the benefits of integrating different aspects of production management to achieve greater energy efficiency. Barzanji et al. (2020) discussed decomposition algorithms for the integrated process planning and scheduling problem. Their approach allows for more efficient management of complex production processes by breaking down the problem into smaller, more manageable parts. Bayu et al. (2020) presented gasoline mixing and distribution scheduling using a graphical genetic algorithm. Their research shows how advanced optimization techniques can be used in the chemical industry to improve process efficiency. (Boufellouh, Belkaid, 2020) have addressed dual-objective optimization algorithms for joint production and maintenance scheduling under global resource constraints. Their work underscores the importance of an integrated approach to production and maintenance management to achieve better operational performance. Brum et al. (2022) presented the automatic generation of iterated greedy algorithms for a flow-shop scheduling problem without permutations, minimizing the total completion time. Their research shows the potential of automation in optimization processes. Ghaleb et al. (2020) presented an integrated approach to production and maintenance scheduling for a single degrading machine with failures based on deterioration. Their research underscores the importance of considering machine condition in the production scheduling process, which can lead to increased operational efficiency and fewer breakdowns.

Kaczmar, Bányai (2022) have addressed the optimization of the flow shop scheduling problem, comparing the FlexSim simulation system with an evolutionary solution. Their work focuses on comparing different optimization methods and their impact on scheduling efficiency in a production environment. Lange, Werner (2019) have proposed a heuristic method based on permutations for the blocking job-shop scheduling problem. Their research shows the potential of heuristic optimization methods for solving complex scheduling problems, especially in the context of resource blocking constraints. Luo et al. (2020) have presented an improved genetic algorithm to solve the scheduling problem in a flexible job shop. Their research shows how advanced optimization techniques can be used to improve scheduling efficiency in complex manufacturing environments. Chen et al. (2021) have addressed real-time order acceptance and flow-shop scheduling with permutation, using two-level interactive optimization with nonlinear integer programming. Their research highlights the importance of advanced optimization techniques in order management and production scheduling.

3. Methodology

This article will present an approach using a discrete event simulator with 3D visualization. The advantage of this approach is the ability to modify classical tasks by extending/modifying the problem in a 3D environment. The experiment was run on a model matched to a typical production hall for which a layout was prepared and shown on Figure 1.

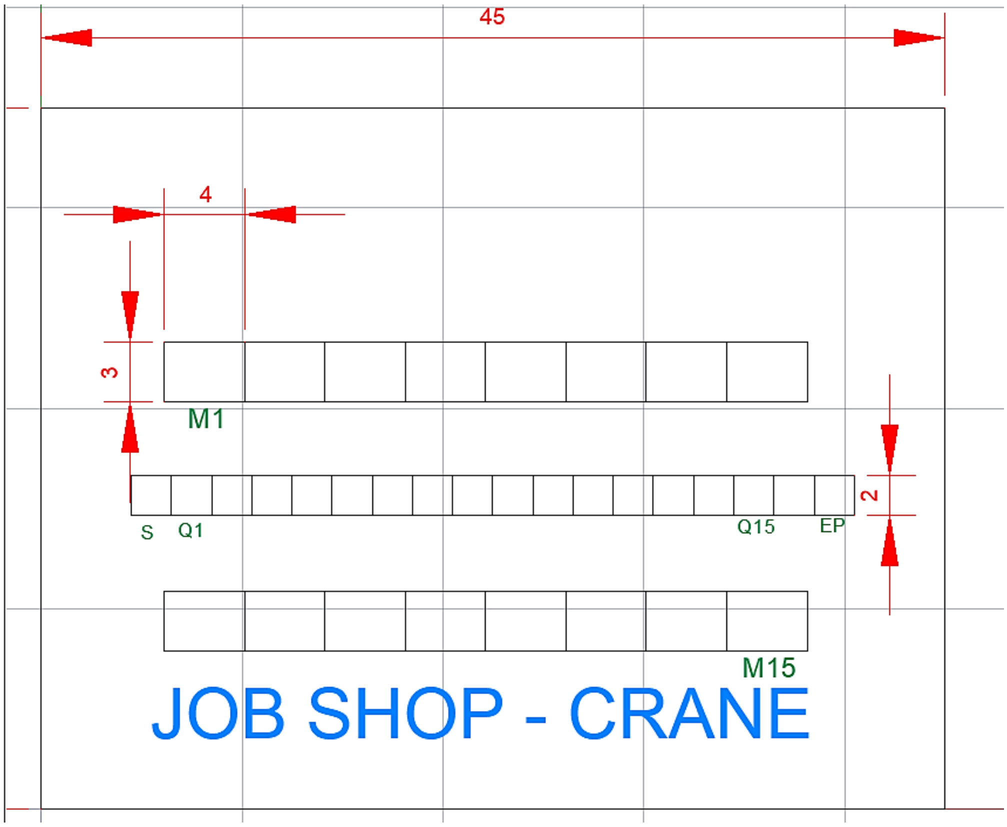

Figure 1. Layout of the production hall.

Source: Own work.

Prepared in Figure 1, the hall is 35x45 meters. The basic dimensions are shown on a separate layer marked in red and expressed in metres too. The machine stations are numbered from M1 to M15 in two rows. The middle row starts from the generated source through the Q1 to Q15 storage fields and ends with the EP field of end products.

The basic parameters of the crane setup (dimensions 40.35x26x7.69) are presented in Table 1 with the sequence L>XY>D[1].

[1] X-Move Gantry, Y-Move Troley, L-Lift Hoist, D-Drop Hoist.

Table 1.
Crane parameter

	Max_Speed	Acceleration	Deceleration
Gantry	2	1	1
Trolley	2	1	1
Hoist_Lift	2	1	1
Hoist_Drop	2	1	1

Source: FlexSim Crane properties.

Table 1 shows the crane parameters in terms of velocity and accelerations and decelerations at manipulations in meters per second[m/s] and [m/s^2]. These values are the defaults for crane-class transport in the FlexSim simulator.

3.1. Calculation environment

A computing environment of 300 cores was prepared for Experiment I calculations. The laboratory was equipped with desktop computers with parameters: 13th Gen Intel(R) Core(TM) i5-13500 2.50 GHz, 32.0 GB, RTX3070.

In the experiment, 15 nodes and one computer were configured to supervise the calculations. Each node had the FlexSim 2023 update 1 application version installed, along with an OptQuest software license. Cluster configurations are shown in Figure 2.

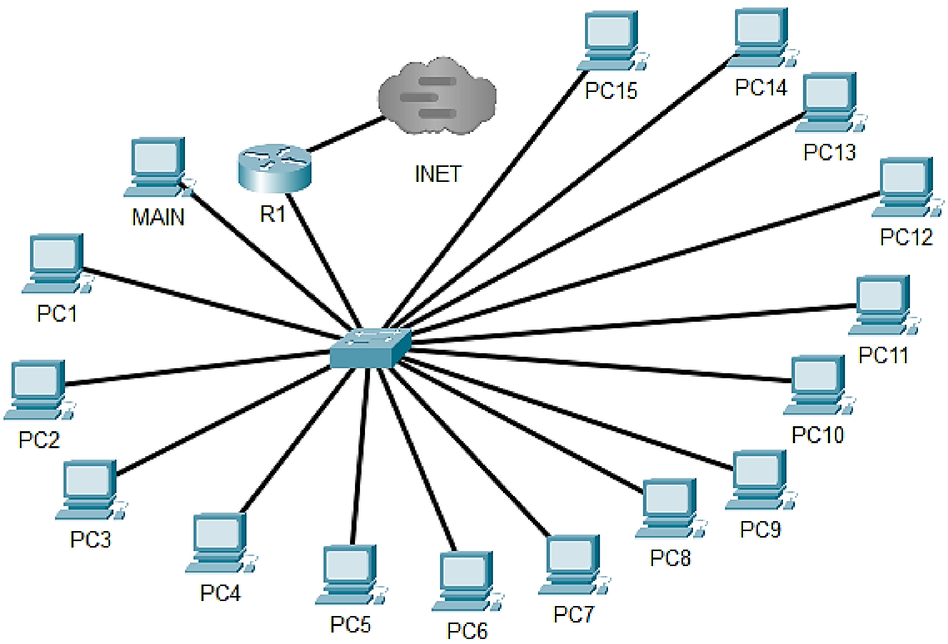

Figure 2. Experiment cloud architecture.

Source: Own work.

The nodes of the cluster are located on the same local network connected to a Layer 2 switch. The hosts are addressed with Class C private pool addresses. Each node provides 20 cores.

3.2. Data sets

The article uses the time and sequence tables presented by Prof. Eric Taillard (Taillard, 1993). Imported data for the smallest nesting problem from a list of test fields (benchmarks) of size 15x15.

The first array is a time array, in which machine operations for each task are expressed in seconds. Table 2 is shown below.

Table 2.
Taillard job-shop 15x15 time table

job	o1	o2	o3	o4	o5	o6	o7	o8	o9	o10	o11	o12	o13	o14	o15
1	94	66	10	53	26	15	65	82	10	27	93	92	96	70	83
2	74	31	88	51	57	78	8	7	91	79	18	51	18	99	33
3	4	82	40	86	50	54	21	6	54	68	82	20	39	35	68
4	73	23	30	30	53	94	58	93	32	91	30	56	27	92	9
5	78	23	21	60	36	29	95	99	79	76	93	42	52	42	96
6	29	61	88	70	16	31	65	83	78	26	50	87	62	14	30
7	18	75	20	4	91	68	19	54	85	73	43	24	37	87	66
8	32	52	9	49	61	35	99	62	6	62	7	80	3	57	7
9	85	30	96	91	13	87	82	83	78	56	85	8	66	88	15
10	5	59	30	60	41	17	66	89	78	88	69	45	82	6	13
11	90	27	1	8	91	80	89	49	32	28	90	93	6	35	73
12	47	43	75	8	51	3	84	34	28	60	69	45	67	58	87
13	65	62	97	20	31	33	33	77	50	80	48	90	75	96	44
14	28	21	51	75	17	89	59	56	63	18	17	30	16	7	35
15	57	16	42	34	37	26	68	73	5	8	12	87	83	20	97

Source: (Taillard, 1993).

The column is a column defining jobs for which the execution of jobs on machines is defined by the operations O1 through O15. Another important table for the job-shop problem is the table of the order of operations on the machines. In this case, too, the original layout of the 15x15 test field was retained and is presented in Table 3.

Table 3.
Taillard job-shop 15x15 operation order

job	o1	o2	o3	o4	o5	o6	o7	o8	o9	o10	o11	o12	o13	o14	o15
1	7	13	5	8	4	3	11	12	9	15	10	14	6	1	2
2	5	6	8	15	14	9	12	10	7	11	1	4	13	2	3
3	2	9	10	13	7	12	14	6	1	3	8	11	5	4	15
4	6	3	10	7	11	1	14	5	8	15	12	9	13	2	4
5	8	9	7	11	5	10	3	15	13	6	2	14	12	1	4
6	6	4	13	14	12	5	15	8	3	2	11	1	10	7	9
7	13	4	8	9	15	7	2	12	5	6	3	11	1	14	10
8	12	6	1	8	13	14	15	2	3	9	5	4	10	7	11
9	11	12	7	15	1	2	3	6	13	5	9	8	10	14	4
10	7	12	10	3	9	1	14	4	11	8	2	13	15	5	6
11	5	8	14	1	6	13	7	9	15	11	4	2	12	10	3
12	3	15	1	13	7	11	8	6	9	10	14	2	4	12	5
13	6	9	11	3	4	7	10	1	14	5	2	12	13	8	15
14	9	15	5	14	6	7	10	2	13	8	12	11	4	3	1
15	11	9	13	7	5	2	14	15	12	1	8	4	3	10	6

Source: (Taillard, 1993).

Similarly, the first column of Table 3 indicates the number of production orders for which the order of execution of operations on the machines has been predefined. The intersection of job1 and o2 means that order 1 in operation 2 is executed on the 13th machine. The minimized objective function is the so-called Makespan i expressed by formula 1. This is the time to complete the last job on the last machine in the schedule. This time in the task is minimized.

The number of all solutions for this task including unacceptable solutions is expressed by formula 2.

$$C_{max} \rightarrow min \tag{1}$$

$$n!^{m} = 15!^{15} = 5.59106e + 181 \tag{2}$$

Example permutation is shown as formula 3.

M1 [9, 4, 2, 1, 8, 15, 10, 14, 6, 3, 5, 7, 11, 13, 12]
M2 [2, 11, 12, 6, 3, 10, 13, 5, 8, 9, 7, 15, 1, 14, 4]
M3 [9, 8, 7, 2, 11, 14, 12, 5, 6, 1, 4, 13, 3, 15, 10]
M4 [6, 7, 13, 2, 15, 5, 14, 9, 8, 11, 12, 4, 3, 10, 1]
M5 [13, 3, 2, 8, 9, 7, 14, 1, 6, 10, 12, 4, 11, 15, 5]
M6 [7, 15, 14, 11, 2, 12, 13, 3, 6, 10, 8, 1, 4, 9, 5]
M7 [5, 14, 10, 1, 8, 15, 6, 13, 7, 12, 2, 9, 4, 3, 11]
M8 [10, 7, 14, 11, 6, 4, 8, 12, 2, 9, 15, 3, 1, 5, 13] (3)
M9 [11, 10, 9, 14, 3, 6, 12, 1, 5, 8, 4, 15, 13, 2, 7]
M10 [14, 3, 6, 1, 7, 10, 9, 13, 2, 11, 5, 12, 15, 8, 4]
M11 [3, 8, 14, 2, 9, 12, 13, 11, 4, 10, 15, 1, 5, 6, 7]
M12 [13, 2, 6, 11, 12, 9, 14, 3, 4, 1, 15, 8, 5, 10, 7]
M13 [13, 9, 7, 14, 1, 8, 12, 15, 5, 11, 10, 3, 2, 6, 4]
M14 [2, 14, 5, 4, 6, 13, 9, 11, 7, 15, 3, 8, 10, 1, 12]
M15 [14, 9, 10, 11, 5, 8, 6, 2, 7, 3, 1, 15, 4, 12, 13]

An example solution is a set of 15 independent permutations. Each permutation is assigned to a machine. The set of all permutations from all machines is the solution to the task.

4. Results

First, a working model was prepared according to the data presented in the previous section.

Correct operation of the model required taking into account the experimenter's operation at a later stage, for this appropriate references for downloading parameters were prepared, such as: Model.parameters.m1[current.mo], where.mo is a column storing information about the current order operation on machine m1. ‘Pull’ strategies were used for this. In addition, other table references were created using labels. such as Table ('Time') [item.job] [item.co + 1]. The working 3D model is shown in the figure.

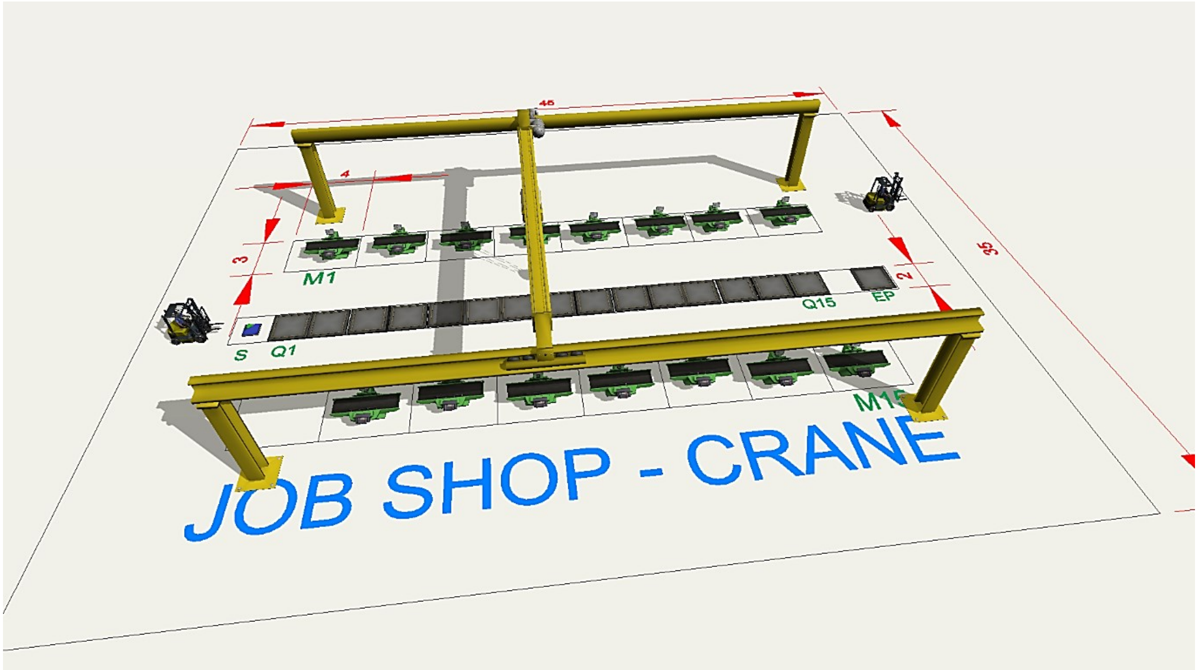

Figure 3. Experiment working 3D model without connections.

Source: Own work.

Set the crane in terms of the position of the machines and the buffer fields according to the coordinates in the layout. List of configured OptQuest optimizer settings:

constraint1: [jobs]=15 - all jobs must be completed,

Maximum number of iterations: 400000,

Wall time: 300000 (seconds),

Replications: 1,

Seed: 78453245786.

Each iteration contained only one replication due to the lack of a random component in the experiment source. All tasks were available immediately as for the original problem. The course of the experiment is illustrated in Figure 4.

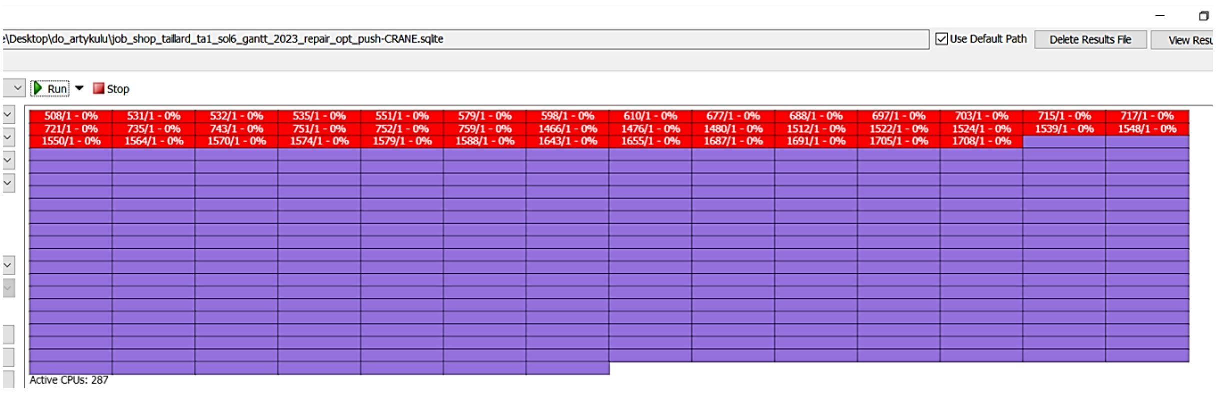

Figure 4. Running calculations.

Source: Own work.

The illustration shows an experiment with optimization of tasks. Although the cloud computation was configured for 300 cores, the illustration shows 287 cores. This is due to the difficulty of taking a screenshot during the live experiment.

The experiment was completed after 24 hours of calculations over 130,000 iterations and nearly 2 million permutations were checked. The group of best solutions is presented in the Figure 5.

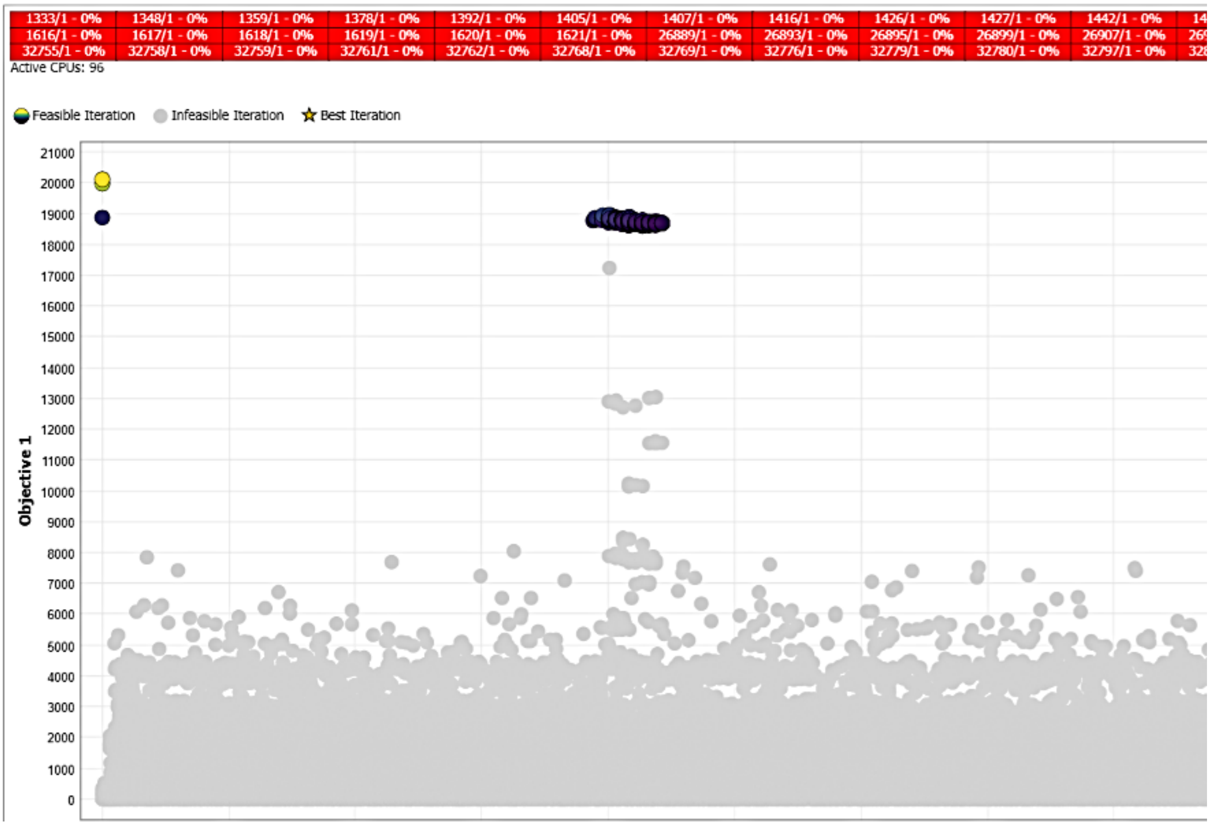

Figure 5. Best findings after 24 hours (slice).

Source: Own work.

The results database occupied ~500MB in a sqlite file. Data analysis required viewing the results outside the OptQuest optimizer in SQLite DB Browser. The best solution obtained is 18598. It was decided to compare the results in Table 4 obtained with the best results for the original problem without and with transport.

Table 4.
Results

Original job-shop Taillard 15x15 on FlexSim layout with best known solution		Job-shop Taillard 15x15 witch crane transport and FlexSim layout with best known solution		Best findings after 24 hours with cluster cloud computing	
job	time	job	time	job	time
1	742	1	10339.30	1	1861.48
2	959	2	10688.64	2	2799.75
3	1074	3	11995.35	3	4899.48
4	1114	4	13042.93	4	8374.05
5	1136	5	13489.75	5	9610.85
6	1164	6	13563.58	6	10250.79
7	1180	7	13700.40	7	10398.53
8	1181	8	13789.22	8	11125.07
9	1189	9	13840.67	9	11180.96
10	1190	10	14025.13	10	12392.89
11	1204	11	14051.15	11	13432.46
12	1212	12	14271.68	12	14978.85
13	1220	13	14312.82	13	16431.12
14	1222	14	14358.93	14	17859.84
15	1231	15	14414.49	15	18593.96

Source: Own work.

The value of 1231 Makespan job-shop for the Taillard test is identical to published values[2]. Transport has been removed from the model, and the tasks on the machines will appear immediately according to the assumptions in the tables (time, sequence). Thus, we are sure that the logic of the model coincides with the original task published by Professor Taillard. The solution applied directly to the built model with an overhead crane, due to the need to move orders between machines, is a whole order of magnitude larger (longer times for each order). The third column presents the completion tables to obtain each order for the best experimental result. A Gantt chart of all production orders and crane work is shown in Figure 6.

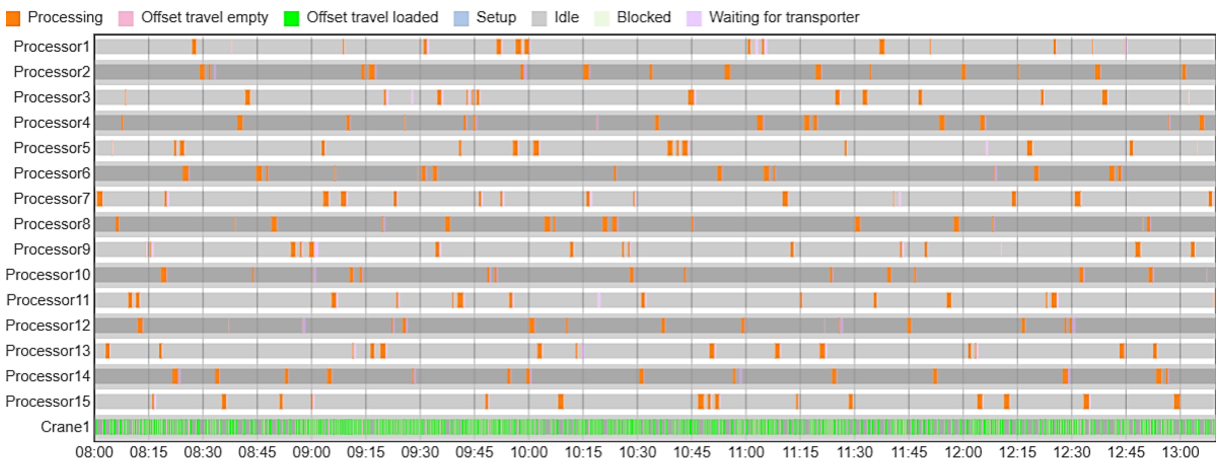

Figure 6. Gantt chart for the machines and crane states.

Source: Own work.

[2] Scheduling instances URL: http://mistic.heig-vd.ch/taillard (stan na 30.09.2024).

The vast majority of the work is related to the operation of the overhead crane. The total number of operations to pass all orders through the overhead crane was 450, double the number of all operations. This was caused by putting orders on declining fields The average idle time of the crane oscillated at 16 seconds and the total distance traveled 19 km.

5. Discussion

Each of the solutions allowed in this experiment will have a longer execution time than each solution for the original 15x15 job-shop Taillard's benchmark because each machine change operation will take into account the time required for the crane to operate. This time is not fixed and depends directly on the order of orders and conditions (e.g., the distance between machines and the order in which machines are placed under the crane) on the production floor. Of additional importance is the lack of a common lay-down area for this scenario. The crane must move between machines as in well as queues where orders are waiting for their turn. The results obtained in the article are not optimal. Figure 5 shows a slice of the solution space with a marked group which may suggest that the optimizer is stuck at the local minimum for a longer period of time.

6. Conclusions

The paper presents an approach to solving a transport nesting task using FlexSim computer simulation on which optimization was performed in OptQuest software in a computational cluster system. The approach to the problem presented in this publication circumvents the need for advanced mathematical apparatuses to describe modified classical situations in a practical situation in real production and beyond. The solution space for such a model and a job-shop problem is enormous. The mechanism for generating new permutations in the OptQuest software is unknown. The literature in this area, as for flow-shop tasks, presents solutions involving meta heuristics such as genetic algorithms, simulated annealing, particle swarming, tabu search, and others. One possible solution is to combine the FlexSim simulator with an external engine for parameter optimization. Such an approach has been presented in other works by the author (Janke, 2023; Janke, Owczarek, 2023). It presented optimization in an external engine and checked the results in a simulation model as well as a real-time connection of an external script.

References

1. Alizadeh, R., Rezaeian, J., Abedi, M., Chiong, R. (2020). A modified genetic algorithm for scheduling non-emergency outpatient appointments with highly demanded medical services considering patient priorities. *Computers & Industrial Engineering*, *139*, 106106. https://doi.org/https://doi.org/10.1016/j.cie.2019.106106
2. An, Y., Chen, X., Zhang, J., Li, Y. (2020). A hybrid multi-objective evolutionary algorithm to integrate optimization of the production scheduling and imperfect cutting tool maintenance considering total energy consumption. *Journal of Cleaner Production*, *268*, 121540. https://doi.org/10.1016/j.jclepro.2020.121540
3. Barzanji, R., Naderi, B., Begen, M.A. (2020). Decomposition algorithms for the integrated process planning and scheduling problem. *Omega (United Kingdom)*, *93*, 102025. https://doi.org/10.1016/j.omega.2019.01.003
4. Bayu, F., Panda, D., Shaik, M.A., Ramteke, M. (2020). Scheduling of gasoline blending and distribution using graphical genetic algorithm. *Computers and Chemical Engineering*, *133*. https://doi.org/10.1016/j.compchemeng.2019.106636
5. Boufellouh, R., Belkaid, F. (2020). Bi-objective optimization algorithms for joint production and maintenance scheduling under a global resource constraint: Application to the permutation flow shop problem. *Computers and Operations Research*, *122*, 104943. https://doi.org/10.1016/j.cor.2020.104943
6. Brum, A., Ruiz, R., Ritt, M. (2022). Automatic generation of iterated greedy algorithms for the non-permutation flow shop scheduling problem with total completion time minimization. *Computers and Industrial Engineering*, *163*. https://doi.org/10.1016/ j.cie.2021.107843
7. Chen, W., Gong, X., Rahman, H.F., Liu, H., Qi, E. (2021). Real-time order acceptance and scheduling for data-enabled permutation flow shops: Bilevel interactive optimization with nonlinear integer programming. *Omega (United Kingdom)*, *105*. https://doi.org/10.1016/ j.omega.2021.102499
8. Faraji Amiri, M., Behnamian, J. (2020). Multi-objective green flowshop scheduling problem under uncertainty: Estimation of distribution algorithm. *Journal of Cleaner Production*, *251*, 119734. https://doi.org/10.1016/j.jclepro.2019.119734
9. Ghaleb, M., Taghipour, S., Sharifi, M., Zolfagharinia, H. (2020). Integrated production and maintenance scheduling for a single degrading machine with deterioration-based failures. *Computers and Industrial Engineering*, *143*, 106432. https://doi.org/10.1016/j.cie. 2020.106432
10. Janke, P. (2023). Testing algorithms for quick rescheduling flow shop problems with FlexSim based simulation and R engine. *Scientific Papers of Silesian University of*

Technology Organization and Management Series, *168*. https://doi.org/10.29119/1641-3466.2023.168.12
11. Janke, P., Owczarek, T. (2023). Real-time data processing in simulation models with 3D visualization. *Scientific Papers of Silesian University of Technology Organization and Management Series*, *171*. https://doi.org/10.29119/1641-3466.2023.171.2
12. Kaczmar, I., Bányai, T. (2022). Optimisation of flow shop scheduling problem: simulation system vs. evolutive solver. *Advanced Logistic Systems - Theory and Practice*, *16*(1), 31-40. https://doi.org/10.32971/als.2022.003
13. Lange, J., Werner, F. (2019). A permutation-based heuristic method for the blocking job shop scheduling problem. *IFAC-PapersOnLine*, *52*(13), 1403-1408. https://doi.org/10.1016/j.ifacol.2019.11.395
14. Luo, X., Qian, Q., Fu, Y. F. (2020). Improved Genetic Algorithm for Solving Flexible Job Shop Scheduling Problem. *Procedia Computer Science*, *166*, 480-485. https://doi.org/https://doi.org/10.1016/j.procs.2020.02.061
15. Pinedo, M. (2012). *Scheduling: Theory, Algorithms, and Systems*. Springer Science & Business Media.
16. Sobaszek, Ł., Gola, A., Świć, A. (n.d.). *Analiza problemu job-shop z uwzględnieniem,* 658-669.
17. Taillard, E. (1993). Benchmarks for basic scheduling problems. *European Journal of Operational Research*, *64*(2), 278-285. https://doi.org/10.1016/0377-2217(93)90182-M