Proceedings of the 2022 Winter Simulation Conference
B. Feng, G. Pedrielli, Y. Peng, S. Shashaani, E. Song, C.G. Corlu, L.H. Lee, E.P. Chew, T. Roeder, and P. Lendermann, eds.

APPLYING A HYBRID MODEL TO SOLVE THE JOB-SHOP SCHEDULING PROBLEM WITH PREVENTIVE MAINTENANCE, SEQUENCE-DEPENDENT SETUP TIMES AND UNKNOWN PROCESSING TIMES

Joep Ooms
Alexander Hübl

University of Groningen
Faculty of Science and Engineering
Engineering Systems and Design Group
Nijenborgh 4
Groningen, 9747 AG, THE NETHERLANDS

ABSTRACT

Although, many researchers propose to optimize the job-shop scheduling problem using all processing times initially available, to mimic a more real-life environment, in this paper, processing times are unknown at the beginning of the optimization. The job-shop scheduling problem is considered with sequence-dependent setup times and preventive maintenance constraints. Processing times are revealed when products arrive at a machine. Unknown processing times will give a more real-world representation, where exact processing times aren't available. A hybrid model, combining discrete event simulation and optimization is applied to simulate the production process and to solve the job-shop problem. The hybrid model uses optimization by creating new production schedules when a job is processed, and when the product arrives at the next machine. The meta-heuristics of the genetic algorithm, ant colony optimization, and simulated annealing algorithm are used. The results showed significantly better results for the hybrid optimization than random sequencing of jobs.

1 INTRODUCTION

Manufacturing companies can improve the key performance indicators by adjusting production capacity (Altendorfer et al. 2014). Effective scheduling is a way for companies to improve their production throughput without using extra resources (Topcuoglu et al. 2002). Flexible manufacturing systems emerged as a result of the developments in manufacturing systems (Akturk and Ozkan 2001). The job-shop scheduling problem is one of the most studied problems regarding scheduling (Xiong et al. 2022). The job-shop scheduling problem is defined as a problem that focuses on changing the sequence of jobs that are assigned to specific resources to improve key performance indicators (Błażewicz et al. 1996). The job-shop scheduling problem is described by n amount of jobs, that have to be processed on m number of machines. Each job n_j has a specific routing of operations on the machines that have to be performed, leaving a set of operations for all jobs $O = o_{1j}, o_{2j}, o_{3j}, o_{ij}$. The processing times of the operations of all products are stochastic distributed, but known at the beginning of the scheduling. Effective scheduling of these operations on the machines can reduce the total length of the production schedule. The effective allocation of the jobs on the machines aims to decrease the makespan.

Traditionally, the job-shop scheduling problem consists of five conditions; (I) the following job cannot be started before the previous job on a machine is finished, (II) jobs cannot be interrupted by another job, and the same time, every job can only be performed on one machine, (III) each job can only be performed at one machine at every time instance (IV) sequence of machines which a job visit is completely specified and has a linear precedence structure, and (V) processing times are all known (Błażewicz et al. 1996; Tamilarasi et al. 2010). The last constraint is very unlikely for some industries, where production times cannot be accurately predicted. Examples are make-to-order and engineer-to-order job-shops (Kundu et al. 2018). The difference in this paper is that (V) is relaxed and considered to be initially unknown. This will result in less information available about the processing times. Therefore, local optimization at the machines is chosen as a solution to improve the production schedule of the products.

In real-world production facilities such as raw material handling, chemical production, and assembly departments, the setup times of jobs, such as cleaning times or machine setups, are required between jobs and they depend heavily on the preceding and the current job on the machine. This assumption is based on the fact that when setup delays are explicitly factored into scheduling decisions, higher throughput can be realized (Afshar-Nadjafi 2018). Including sequence-dependent setup times in manufacturing, show higher presence in recent job-shop studies. Sequence-dependent setup times are considered in this problem, because of the importance in real-world manufacturing systems (Naderi et al. 2009).

Sufficient maintenance decreases the hazard rate or depreciation of a system, whereas worse maintenance can increase this, or make the system fail or break down. Maintenance includes operations that begin at predetermined intervals or according to predetermined criteria, and are designed to lessen the likelihood of failure or degradation of an item's functionality. Preventive maintenance is included in this paper to mimic a real-world situation of the job-shop scheduling problem.

This paper includes hybridization of the job-shop scheduling problem, which is a modeling approach using a combination of simulation and optimization, to implement the constraints of the preventive maintenance, sequence-dependent setup times and the unknown processing times in the system. To apply all these constraints for the system, hybrid modeling is applied to the job-shop, combining discrete event simulation and optimization where during the discrete event simulation the optimization function is called whenever a new job arrives at the machine. In Mustafee et al. (2017) the definition of hybrid simulation is stated as the application of two or more simulation techniques combined. The application of hybrid modelling makes a system more easy to solve. In Brailsford et al. (2019) hybrid simulation is primarily used to represent a complex system where different parts of this system are better captured by more than one simulation method. This hybridization is applied to capture all the constraints of the preventive maintenance, unknown processing times and sequence-dependent setup times in the job-shop model.

1.1 Previous Research

Naderi et al. (2009) studied the job-shop scheduling problem with sequence-dependent setup times and preventive maintenance. In this study, preventive maintenance is applied to the system to prevent machines from failure. Idle time is applied to machines to simulate preventive maintenance. The appliance of preventive maintenance makes the system more complex and computational harder to solve.

Many kinds of research regarding optimization in the job-shop scheduling problem consider all processing times to be known in advance (Mohan et al. 2019; Mirjalili 2019; Lin et al. 2010). Reduction of setup times and idle time by use of efficient scheduling techniques proved to be very efficient in job-shop scheduling. Choi and Korkmaz (1997) showed to be very efficient and tested five datasets using different scenarios of sequence-dependent setup times and made a makespan reduction of on average 21%. Defersha and Rooyani (2020) also shows an average reduction of makespan of on average 49% over all replications in a flexible job-shop environment. The results show that scheduling, by use of meta-heuristics, when processing times are initially known can seriously improve the makespan. The fact that all processing times are initially known makes it easier to find the reduction in the solutions using heuristics or deterministic methods. The deterministic idea that all processing times are exactly known at the start of production is unrealistic for some industries and makes optimization studies not always applicable to real-world cases. However, in the literature, there is almost no effective scheduling approach of real-world stochastic production environment (Hübl 2018; Egilmez et al. 2012). This paper bridges the gap between mostly unrealistic known information of processing time towards real-world application.

Stochastic job-shop scheduling has been studied in the literature, where the influences of the stochastic processing times on the production schedule is examined (Egilmez et al. 2012). Moreover, stochastic job-shop scheduling is characterized by processing times drawn from a known distribution; this form of scheduling uses expected values for scheduling in advance of the production. This still considers much information known in advance of the production process, which is unrealistic due to many factors that can influence the actual processing times. Sotskov et al. (2020) researched a two-machine job-shop with uncertain processing times but did not consider maintenance or sequence-dependent setup times. Lou et al. (2012) researched reactive scheduling, using an initial schedule, based on unexpected events in the simulation environment. This research was done using a multi-agent-based model, using sequence-dependent setup times and preventive maintenance. Based on our literature research, no papers have been found considering a job-shop with sequence-dependent setup times and preventive maintenance, where the processing times are unknown at the beginning. Previously mentioned Sotskov et al. (2020) shows the most comparable research with having uncertain products times until the product is produced, without sequence-dependent setup times and preventive maintenance. Naderi et al. (2009) investigates sequence-dependent setup times and preventive maintenance in job-shop scheduling, however they considered all processing times to be known at the start.

The real world of some manufacturing industry is deterministic-free, and production attributes are all stochastic (Tavakkoli Moghaddam et al. 2005). These stochastic values and constraints are not available in traditional approaches of the job-shop problem. To approach a real-world manufacturing environment, the stochastic attributes should be considered in the problem of the job-shop scheduling problem.

Previous research has not yet applied the stochasticity is not well in the job-shop scheduling problem. This is caused by the fact that all processing times or distributions of processing and setup times cannot be known in advance and this is assumed in many researches. For example, processing times depend on the current production circumstances, such as the processing quality of the previous working station. The difference in this research is that no information about the processing times is available in advance of the production. The difference with the deterministic models is that not all information is known. This makes it unable to centralize the production scheduling. An initial schedule cannot be made, and the production sequences should be determined at the machine. The production schedule should be adjusted at the moment more information about the processing times will be known. This research aims to close the gap by making all processing times and setup times unknown at the beginning of the simulation of the job-shop. The processing and setup times will only be available when the products arrive at the machines. This lack of information is expected to create a smaller reduction in makespan; however, it will extend the current literature and make scheduling more applicable in a stochastic manufacturing system.

1.2 Sequence-dependent Setup Times

The job-shop scheduling problem can be studied with sequence-dependent setup times. Setup times are defined as the time interval a machine needs to adjust, before it can start operating on the next product or batch of products. In addition to this, for sequence-dependent setup times, the setup times depend on both the task and its preceding task. The sequence-dependent setup times in the job-shop scheduling problem make the problem harder to solve. The job-shop scheduling problem becomes a notoriously difficult problem in combinatorial optimization. This makes even modest sized instances remain of the job-shop computationally intractable.

The hardness is caused by the increase in the problem's computational complexity. Adding sequence-dependent setup times implies that the job-shop scheduling problem can be solved using graph theory. Setting all jobs as vertices and all setup times as an edge on the graph, makes it possible to model the problem on a graph $G(V,E)$. This creates a traveling salesman problem. Modeling the job-shop scheduling problem as a TSP makes the job-shop scheduling problem a NP-hard problem, because the TSP always contains a Hamiltonian Cycle based on the input n and consists of output n. Every solution for the traveling salesman problem will contain a Hamiltonian cycle. The traveling salesman problem is a NP-hard problem, the job-shop scheduling problem is this as well (Sotskov and Shakhlevich 1995).

Three meta-heuristics will be applied to try and optimize the job-shop scheduling problem. The meta-heuristics will attempt to solve it using graph theory as done in the travelling salesman problem. Several meta-heuristics are used to optimize the scheduling problem in the system. The best sequence should be found between the available jobs at the machine. The production schedule will reduce the total amount of setup times in the system. All relevant literature implemented in the model will be discussed in the next subsections.

1.3 Preventive Maintenance

Maintenance and failures can be included as an extra constraint for the job-shop scheduling problem. Preventive maintenance can be implemented in job-shop scheduling to obtain a more realistic representation of the job-shop. This can be done by implementing probability functions for the availability of machines. Preventive maintenance could then be timed when the machine works for an optimal period, with a slight chance of failure. The behavior of failures in machines is different in every production environment. For some production operations, failures are far worse than the implementation of preventive maintenance, because failures are more time consuming and often come with more costs (Naderi et al. 2009). The addition of preventive maintenance makes the problem more realistic and adds extra constraints to the problem, where the machines will be unavailable some moments. Preventive maintenance can be applied using three maintenance policies:

1. Preventive maintenance at fixed predefined time intervals. The preventive maintenance intervals are performed on a machine at the end of a predefined time interval. It does not consider probabilistic functions.
2. Optimum period model for the preventive maintenance maximizing the machines' availability. The preventive maintenance is planned based on the stochasticity distribution of the failure. The optimal time interval which should be available before a new item starts is used to define if the preventive maintenance should start.
3. Maintaining a minimum reliability threshold for a given production period. The preventive maintenance activity is planned when the chance of a failure is expected to create more idle time than the preventive maintenance activity.

1.4 Hybridization

Hybridization is applied in simulation to make use of one system and update the system based on the analysis to optimize the system. In literature it has been argued that a hybrid approach including discrete event simulation and an external optimizing could lead to a more realistic picture and better solving of complex systems with fewer assumptions and less complexity (Brailsford et al. 2019). The earliest framework of hybridization was presented by Chahal and Eldabi (2008), here three modes of hybridization were defined: hierarchical, process environment, and integrated. A sequential model has one model its output to serve as another model its input. An enriching model has a smaller model covered within a more dominant model, where the smaller model has limited use.

Using a model that uses simulation and optimization is a form of hybridization which is called simulation optimization (Amaran et al. 2016). For the classic interpretation of simulation optimization the optimizer calls the simulation models, delivers a set of parameters, those parameters are tested in the simulation, and sent back to the optimizer. In this paper, the simulation model is executed and whenever a new job arrives at the machine an optimization function is called. The optimization is done based on the input parameters from the simulation, as the discrete event simulation model requires an input from the optimizer, where interaction is necessary, the type of hybridization simulation optimization can be defined as sequential (Brailsford et al. 2019). Due to the np-complexity of the optimization of the job-shop problem, and the meta-heuristics are applied to decrease the computing time and still be able to find good solutions. The meta-heuristics in the optimization part of them model are used to find the best solutions for the production schedules. Due to the computational complexity of finding the solution of the production schedules due to the sequence-dependent setup times and the extra constraints considering the preventive maintenance, a sequential hybridization model is applied in this research.

1.4.1 Genetic Algorithm

The genetic algorithm proved to be a powerful method for combinatorial and constrained optimization problems since its development (Wang and Yu 2010). The genetic algorithm is inspired by the theory of evolution and natural selection by Charles Darwin. Therefore, it uses the survival of the fittest approach to select the optimal solution. The genetic algorithm first generates an initial population. Every person chooses a random route along the cities. For every individual, a fitness value is generated according to the costs of traveling. All scores are analyzed, and then individuals will be selected, and the individuals with the best fitness value will be selected. The selection for comparing the fitness values happens in various ways. After the selection, the crossover takes place. A randomly arranged meeting is created for the crossover. After the crossover, a child is produced as a new generation of individuals. At last, the mutation will take place, this is done to make sure a new solution will be created (Mirjalili 2019).

1.4.2 Simulated Annealing Algorithm

The simulated annealing algorithm is a locally-based searching algorithm. It can be used to solve combinatorial problems. The algorithm works like the annealing of metal in metallurgy. It uses the physical properties of the cooling of metal to try to find the optimal solution outcome. The system starts at a high temperature, at which the particles make many irregular movements. During the cooling of the metal, the molecules try to find the optimal place in the crystal structure (Chakraborty and Bhowmik 2013). When the temperature reduces, the movement space of the molecules reduces, which is equivalent the algorithm's search space. Therefore, at the moment, the temperature is very low the optimal solution is found (Cruz-Chávez et al. 2017).

The simulated annealing algorithm starts with setting the starting annealing temperature T and the number of iterations k. The algorithm starts with a random solution ω. The goal is to find the optimal solution ω^* in the solution set of Ω. The algorithm first generates an initial solution. The algorithm starts to find a new solution ω', in the neighborhood of $N(\omega)$, $\omega' \in N(\omega)$. The following equation ω' is accepted as a modified solution based on the new solution.

$$P(\omega') = \begin{cases} 1 & \text{if } \Delta c \leq 0 \\ e^{-\Delta c/t_k} & \text{if } \Delta c > 0 \end{cases} \tag{1}$$

In equation 1 P is defined as the probability that the new solution ω' will be accepted as the current solution ω. Δc is the difference in the costs for the following solution, t_k is the current temperature at the iteration k. If the new solution ω' is better than the previous solution, it is always accepted. Otherwise, it depends on the difference in costs. When the solution is accepted, the cooling schedule applies, the temperature is adjusted, and a new iteration starts.

1.4.3 Ant Colony Algorithm

The ant colony optimization algorithm is inspired by the foraging behavior of ants, who communicate within colonies to find the optimal route between the colony and a food source in an environment (Zhao et al. 2006). The algorithm works with a starting number of ants walking onto random routes between the cities. Initially, the ants start their search by roaming randomly and selecting the next node to be visited by a probabilistic equation. After that, a stronger pheromone concentration on a route stimulates the ants to move in the direction of that route. Ants that take shorter trails to the food source return faster to the nest than ants that take longer routes. This results in the shorter trail having a higher quantity of pheromone laid down on the path. Subsequently, ants will be biased in choosing the shorter path. Occasionally, ants will be straying away from this trail and exploring the other routes (Dorigo et al. 2006). Every iteration of the ant colony optimization algorithm, the pheromone values are updated for all m number of ants that have found a solution (Dorigo et al. 2006). τ_{ij} is defined as the pheromone associated with the cities i and j. This pheromone is updated as follows in equation 2:

$$\tau_{ij} \leftarrow (1-\rho)\cdot\tau_{ij} + \sum_{k=1}^{m} \Delta\tau_{ij}^{k} \tag{2}$$

ρ is the evaporation rate, $\Delta\tau_{ij}^{k}$ is the pheromone rate on the, which is calculated using equation 3:

$$\Delta\tau_{ij}^{k} = \begin{cases} Q/L_k & \text{if ant } k \text{ used edge } (i,j) \text{ in its tour} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Where Q is a constant and L_k is the length of the tour, made by that ant. To find a solution, the ant selects the next city to be visited throughout a stochastic mechanism, based on his current partial solution s^p. The probability of ant k going to city j from current city i is defined by equation 4:

$$p_{ij}^{k} = \begin{cases} \frac{\tau_{ij}^{\alpha}\cdot\eta_{ij}^{\beta}}{\sum_{c_{il}\in\mathbf{N}(s^p)} \tau_{il}^{\alpha}\cdot\eta_{il}^{\beta}} & \text{if } c_{ij}\in\mathbf{N}(s^p) \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

$\mathbf{N}(s^p)$ is the set of the feasible components. l is a city which yet is not visited. Parameters α and β define the relative importance of the pheromone versus the heuristic information η_{ij}. η_{ij} is defined by equation 5:

$$\eta_{ij} = \frac{1}{d_{ij}} \tag{5}$$

Parameter d_{ij} is the length between cities i and j (Dorigo et al. 2006). The algorithm performance can be adjusted by setting the α, β, and ρ parameters.

2 SIMULATION MODEL

The job-shop scheduling model is applied in a discrete event environment and an optimizer is connected to optimize the production sequence in front of the machines whenever a new job arrives. The idea of a generic simulation model where changing the inputs (e.g. bill of material, routings) and the model adapts the material flow automatically is applied according to Hübl et al. (2011). The simulation model represents a digital twin of a production system where the optimized sequence of production orders by an optimizer is executed. The stochastic processing times are generated when the products arrive at the machines. In the simulation model, the key performance indicators are measured. The job-shop is simulated in Anylogic 8.7 and the for the optimization Python 3.8 is used. The hybridization is made by using the Pypeline connector version 1.4 within the Anylogic model to connect with Python. Therefore, in Python the meta-heuristics are performed to find an optimal sequence of jobs in front of the machine.The model can be easily adapted to any job-shop in any kind of industry.

In this paper the job-shop scheduling problem is non-flexible, and all jobs should be produced in their own specific technological order defined in the routings. This is according to the standard job-shop scheduling problem of Taillard (1993). The job-shop takes into account sequence-dependent setup times and preventive maintenance. The job-shop is set up according to the Taillard benchmark standard, and the preventive maintenance is implemented according to policy 1 in section 1.3.

2.1 Job-shop

The job-shop problem is formulated as a mathematical experiment with a set M machines containing i number of machines and a set of N products containing j in every experiment. Where the M set of machines is defined as $M = \{m_1, m_2, m_3, m_4, \dots, m_i\}$ and the set of jobs as $N = \{n_1, n_2, n_3, n_4, \dots, n_j\}$. Therefore, all experiments of job-shop scheduling are described as an $M \times N$ experiment. Every job which is performed on a machine is described with an operation $O = \{o_{11}, o_{12}, o_{13}, o_{14}, \dots, o_{ij}\}$. In the problem of this job-shop, one machine $m_i \in M$ can only process one job, $n_j \in N$, at the same time. Every $n_i \in N$, has a routing and the number of machines the product n has to visit can be smaller than the number of machines j. The setup times are considered to be asymmetric in the system, so the processing time and thus the lengths of the edges differ between two of the same cities. All experiments are conducted on four configurations of job-shop size, using a different number of jobs and machines (n,m). The job-shop sizes are chosen from the standard job-shops of (Taillard 1993). The simulation time stops if all jobs are processed by the machines.

2.2 Data Generation

For all experiments a $M \times N$ job-shop environment is simulated. At the start of the simulation, all jobs are directly generated. All jobs get three information matrices: (i) the routing, (ii) the processing times, and (iii) the setup times. The matrices of the processing and setup times of the jobs are initially empty. The routing of the job is randomly generated according to the m number of machines. For the machines, only the time between the preventive maintenance activities and the duration of the preventive maintenance is known. For each instance, when a job arrives at a machine, the processing and setup times of the job at that machine are generated. Based on the jobs present at that machine, the optimizer determines based on the meta-heuristic the best sequence of products at that machine.

The distributions used are based on Taillard (1993) and Naderi et al. (2009). This means that the processing times for all operations (o_{ij}) are uniformly generated over a distribution of $U = (1,99)$ time units. For each machine in the experiment, a seed value is given to get comparable results in between the experiments. The setup times (s_{ij}) are set relative to the processing times with 50% and 100% ratios. Two scenarios are set for the relative level of setup times to the processing times. The relative level of the processing times to the setup times is defined as the variable L_{st}. This results in the following two scenarios for the setup times of $U = (1,50)$ for $L_{st} = 50\%$, and $U = (1,99)$, for $L_{st} = 100\%$.

During each experiment, all values are independently randomly generated based on the defined distributions. For each job, a random routing is created along all machines of the job-shop. For each replication of the experiments, routings along the machines are generated with the same seed value, to obtain comparable results for different experiments. This is done to create identical routing for the jobs on the same replication between the different experiments.

2.2.1 Preventive Maintenance

Preventive maintenance is applied during the production in job-shop scheduling for each machine. The time interval between two consecutive preventive maintenance activities is defined as the time between preventive maintenance (TPM). The first preventive maintenance method of the section 1.3 is applied in this research. This means that the following job will start if the TPM is longer than the processing time of the following job. The PM activity starts, if the processing time of the following product that should be produced is greater than the time until the following preventive maintenance activity starts. The time distance beteween the initial sets of PM is defined as the TPM. The time between preventive maintenance is calculated with equation 6, according to Naderi et al. (2009), where $\bar{p}$ and $\bar{s}$ are the average processing times and setup times. The expected number of maintenance breaks is set as parameter r. The duration of the preventive maintenance is also generated according to the paper of Naderi et al. (2009) over a uniform distribution of $U = (1,25)$.

$$T_{PM} = \frac{(\overline{p}+\overline{s})*n}{r} \tag{6}$$

2.3 Experiments

Several experiments are tested on the job-shop scheduling. The following experiments are conducted:

1. All experiments are conducted on 4 different job-shop sizes (20×15), (50×15), (50×20), and (100×20). All job-shop experiments are conducted using the three different meta-heuristics (genetic algorithm, simulated annealing algorithm, and ant colony algorithm).
2. The experiment are repeated using different expected number of maintenance stops using the genetic algorithm. The experiments are conducted with three different number of expected maintenance stops, $r = 3$, $r = 5$, $r = 7$, as done in Naderi et al. (2009), to compare if there is any difference in maintenance sensitivity.
3. At last, the results experiment are run against two different distribution of setup times; $L_{st} = 50\%$ and $L_{st} = 100\%$, to determine the sensitivity of the setup times for (batches of) products.

2.4 Key Performance Indicators

Several key performance indicators are measured to obtain the performance of the job-shop. The key performance indicators and how they are measured are stated below.

2.4.1 Makespan

The makespan defines how efficiently the production schedule is set up. The makespan is defined as the time that elapses from the beginning to the end of the production sequence, and it is measured in time units (Afshar-Nadjafi 2018). The makespan is used to compare the efficiency of the method which is applied on the job-shop scheduling problem. If the makespan decreases, higher efficiency of the resources is obtained. The average reduction in makespan (δ_{av}) by use of optimization is measured by comparing the makespan results. The benchmark is chosen as a random run of products through the system because no information about the processing times is available at the simulation's beginning. Equation 7 shows the calculation of δ_{av}, which is the decrease in makespan between the random sequencing of jobs and the epitomized sequence.

$$\delta_{av} := \frac{Sol_{rand} - Sol_{alg}}{Sol_{rand}} \tag{7}$$

2.4.2 Utilization

Utilization is defined as the amount of time the machines are not idle (Pachpor et al. 2017). In this research, this is the moment when products are produced and when the preventive maintenance activity is active. If products have to wait for entering the machine, the machine is utilized due to a preventive maintenance activity. The long-term goal is to maximize machine utilization, because this ensures high use of resources (Hübl 2018). However, this makes the system inflexible to fluctuating demand. Equation 8 shows the calculation for the utilization statistic for a machine. T_{tot} is defined as the total time, T_{pm} is defined as the time for preventive maintenance activities, and T_{prod} is defined as the time that the machine is operating on products.

$$U := \frac{T_{pm} + T_{prod}}{T_{tot}} \tag{8}$$

2.4.3 Work In Progress

Work in progress(WIP) describes the partially finished goods in the job-shop that still needs to be finished. Work in progress can refer to raw materials, labor, and overhead costs incurred for products at various stages of the production process (Huang et al. 2008). In this research, the work in progress is measured as the number of jobs present in the system. The work in progress starts as the n number of jobs, and decreases when the system finishes a product.

3 EXPERIMENTAL RESULTS

Two scenarios were simulated in the system. First, the jobs are randomly scheduled at the machines. Afterward, the system has locally been optimized by the use of meta-heuristics. Furthermore, parameter variations are used to see how the system interacts in different situations, for the number of expected maintenance stops and the relative importance of the processing times. During the simulation, in both cases the processing times are revealed at the machines, the results of local optimization showed a significant reduction in makespan for three out of four sizes of job-shops on which it was tested, in comparison to the random allocation of jobs through the system. The reduction of makespan (δ_{av}) became relatively more prominent when the number of jobs increased, and the number of machines stayed the same or decreased. This can be explained by the fact that if the ratio between jobs and machines becomes bigger, the number of possibilities schedules between on the machines will increase. This increase in possibilities per iteration will make a higher deduction in makespan possible. δ_{av} is defined as the average decrease in makespan by using local optimization, on average over all replications. Table 1 shows the result of local optimization using the genetic algorithm in comparison with the benchmark of random allocation. The table shows 99% confidence intervals for the results. For the job-shop sizes of 50×15, 50×20, and 100×20, significant reductions in the makespan where found over 25 replications. On the confidence intervals(CI) of 99% no significant reductions in makespan were found on the 20×15 job-shop size.

Table 1: The results of the local optimization versus random allocation using the genetic algorithm, $n = 25$.

	Optimization		Random allocation		
Job-shop size	Makespan	CI 99%	Makespan	CI 99%	δ_{av}
20×15	2419.5	2341.3 - 2497.8	2478.5	2401.5 - 2555.5	2.38%
50×15	4147.5	4086.2 - 4208.9	4537.5	4415.6 - 4659.4	8.59% *
50×20	4589.6	4500.8 - 4678.3	4837.1	4693.7 - 4980.5	5.11% *
100×20	7393.1	7293.3 - 7493.0	8157.7	8157.7 - 8422.2	10.82% *

Table 2 shows the results of all three KPIs for a different number of maintenance stops. The higher the number of maintenance stops, the makespan increases slightly; this is due to the higher idle time of the system for the preventive maintenance activities. Due to the preventive maintenance policy chosen in the model the makespan on average only increases with the average duration of the preventive maintenance relative to the added or deducted number of maintenance stops.

The average machine utilization does not show differences if the number of maintenance stops is adjusted. The utilization was expected to increase slightly because maintenance stops add busy time to the machines. The maintenance stops would give extra processing time and thus a higher utilization. This is not visible in the results, this could be clarified by the fact it only slightly influences the results of the average of all machines and has a higher influence on individual machines in the system.

At last, the results of the experiments do not show a relation between the WIP with the number of expected maintenance stops. The higher the number of expected maintenance stops does not influence the average number of jobs in the system significantly. Furthermore, a regression analysis was ran the WIP and the makespan or utilization, and in the experiment no relation between can be observed.

Table 2: The KPIs for different number of maintenance stops.

	r=3			r=5			r=7		
Size	Makespan	Util	WIP	Makespan	Util	WIP	Makespan	Util	WIP
20×15	2367.1	0.6273	14.63	2419.5	0.6218	14.57	2471.5	0.6179	14.52
50×15	4106.3	0.7987	34.42	4147.6	0.7932	34.63	4214.7	0.7923	34.62
50×20	4567.5	0.7302	34.99	4589.6	0.7294	35.35	4546.6	0.7399	35.34
100×20	7285.5	0.8558	70.69	7352.3	0.8534	70.46	7408.0	0.8495	69.83

The results of three meta-heuristics are compared to see if there is a difference in the performance of the different meta-heuristics. For all three meta-heuristics, a significant reduction was found in comparison to the random allocation of jobs through the system. Figure 1 shows the results of all three meta-heuristic in comparison to the benchmark with a 99% confidence interval on the error bars. For all three meta-heuristics the reduction in makespan is significant for the job-shop sizes of 50×15, 50×20, and 100×20, using 25 replication ($n = 25$) and 5 expected maintenance stops ($r = 5$).

No significant differences between the three meta-heuristics can be found for any of the job-shop sizes. From this, it can be concluded that the local optimization does have a significant effect, but the type of meta-heuristic, which is applied does not influence the results. Improving the production schedule locally on a machine can significantly decrease the makespan of the whole system. Significant improvements in a system lacking processing times information are still possible.

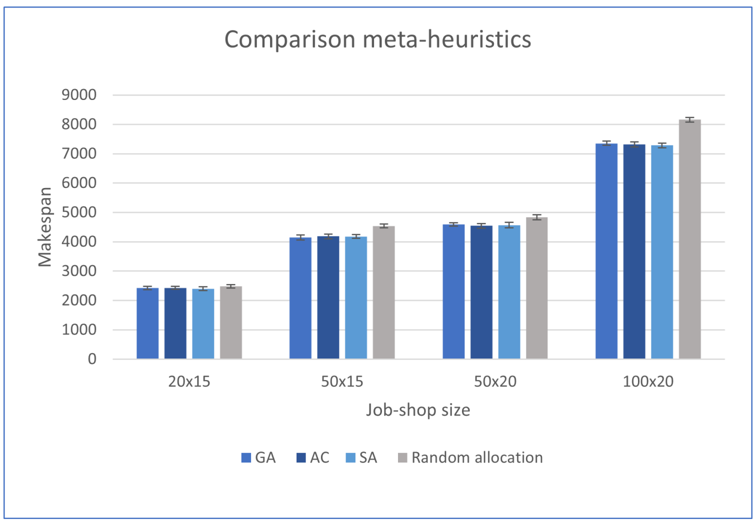

Figure 1: The results of the meta-heuristic comparison, using $r = 5$.

Table 3 shows the average simulation time over 10 runs for all job-shop sizes. The time per iteration describes the average simulation time per application of the meta-heuristic at the machines. All simulations of different job-shop sizes have low simulation time per iteration. Therefore, the simulation-optimization model is applicable in real-world job-shops.

Table 3: The simulation times for the different job-shop sizes.

Size	Simulation time(s)	Number of iteration(#)	Time per interation(s)
20x15	3.54	300	0.011
50x15	23.36	750	0.031
50x20	22.65	1000	0.022
100x20	88.79	2000	0.044

4 CONCLUSION

A hybrid simulation model is proposed in this research to apply the job-shop scheduling with sequence-dependent setup times and preventive maintenance while the processing times are unknown. Every time a job enters the machine the optimizer is executed to identify the optimal sequence in front of the machine. The results showed significant improvements on makespan when the meta-heuristics were locally applied compared to the benchmark of a random allocation through the job-shop with unknown processing times. This shows that an increase can be obtained in job-shop departments by effective scheduling without the use of extra resources. The improvement is reached by applying the simulation-optimization model and shows the effectiveness of applying meta-heuristics to the simulation of the job-shop. The system shows a real-world imitation by not knowing all processing times in advance. The research extends the current literature of applying stochastic in the job-shop scheduling environment, in contrast to previous researches, where all processing times are known in advance. The application of sequence-dependent setup times and preventive maintenance made the simulation model as close as possible to imitating a real-world manufacturing process.

The results lead to insights and limitations in the implementation of scheduling optimization. Currently, all the jobs are known in the beginning. Simulating the job-shop with job arrivals would give more interesting results in the light of mimicking a real-world environment. It is more common that all jobs arrive with inter-arrival times in real-world applications. Simulating the environment with the addition of job arrivals and possible due dates could be considered to make the research even more applicable in real-world systems.

REFERENCES

Afshar-Nadjafi, B. 2018. “A Solution Procedure for Preemptive Multi-mode Project Scheduling Problem With Mode Changeability To Resumption”. *Applied Computing And Informatics* 14(2):192–201.

Akturk, M. S., and S. Ozkan. 2001. “Integrated Scheduling And Tool Management In Flexible Manufacturing Systems”. *International Journal of Production Research* 39(12):2697–2722.

Altendorfer, K., A. Hübl, and H. Jodlbauer. 2014. “Periodical Capacity Setting Methods for Make-to-order Multi-machine Production Systems”. *International Journal of Production Research* 52(16):4768–4784.

Amaran, S., N. V. Sahinidis, B. Sharda, and S. J. Bury. 2016. “Simulation Optimization: A Review Of Algorithms And Applications”. *Annals of Operations Research* 240(1):351–380.

Błażewicz, J., W. Domschke, and E. Pesch. 1996. “The Job Shop Scheduling Problem: Conventional And New Solution Techniques”. *European Journal Of Operational Research* 93(1):1–33.

Brailsford, S. C., T. Eldabi, M. Kunc, N. Mustafee, and A. F. Osorio. 2019. “Hybrid Simulation Modelling In Operational Research: A State-of-the-art Review”. *European Journal of Operational Research* 278(3):721–737.

Chahal, K., and T. Eldabi. 2008. “Applicability of Hybrid Simulation to Different Modes of Governance In UK Healthcare”. In *Proceeding of the 2008 Winter Simulation Conference*, edited by S. Mason, R. Hill, L. Mönch, and O. Rose, 1469–1477. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Chakraborty, S., and S. Bhowmik. 2013. “Job Shop Scheduling Using Simulated Annealing”. In *Proceedings of First International Conference on Computation and Communication Advancement*. January 11th-12th, West Bengal, India, 69-73.

Choi, I.-C., and O. Korkmaz. 1997. “Job Shop Scheduling With Separable Sequence-dependent Setups”. *Annals of Operations Research* 70:155–170.

Cruz-Chávez, M. A., M. G. Martínez-Rangel, and M. H. Cruz-Rosales. 2017. “Accelerated Simulated Annealing Algorithm Applied to the Flexible Job Shop Scheduling Problem”. *International Transactions in Operational Research* 24(5):1119–1137.

Defersha, F. M., and D. Rooyani. 2020. “An Efficient Two-stage Genetic Algorithm for A Flexible Job-shop Scheduling Problem With Sequence Dependent Attached/detached Setup, Machine Release Date And Lag-time”. *Computers & Industrial Engineering* 147:106605.

Dorigo, M., M. Birattari, and T. Stutzle. 2006. “Ant colony Optimization”. *Institute of Electrical and Electronics Engineers, Computational Intelligence Magazine* 1(4):28–39.

Egilmez, G., G. A. Süer, and J. Huang. 2012. “Stochastic Cellular Manufacturing System Design Subject To Maximum Acceptable Risk Level”. *Computers & Industrial Engineering* 63(4):842–854.

Huang, G. Q., Y. Zhang, and P. Jiang. 2008. “RFID-based Wireless Manufacturing for Real-time Management of Job Shop WIP Inventories”. *The International Journal of Advanced Manufacturing Technology* 36(7):752–764.

Hübl, A. 2018. *Stochastic Modelling in Production Planning*. Wiesbaden, Germany: Springer.

Hübl, A., K. Altendorfer, H. Jodlbauer, M. Gansterer, and R. F. Hartl. 2011. "Flexible Model for Analyzing Production". In *Proceedings of the 2011 Winter Simulation Conference*, edited by S. Jain, R. R. Creasey, J. Himmelspach, K. White, and M. Fu, 1554–1565. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Kundu, K., M. Rossini, and A. Portioli-Staudacher. 2018. "Analysing The Impact of Uncertainty Reduction on WLC Methods in MTO Flow Shops". *Production & Manufacturing Research* 6(1):328–344.

Lin, T.-L., S.-J. Horng, T.-W. Kao, Y.-H. Chen, R.-S. Run, R.-J. Chen, J.-L. Lai, and I.-H. Kuo. 2010. "An Efficient Job-shop Scheduling Algorithm Based On Particle Swarm Optimization". *Expert Systems with Applications* 37(3):2629–2636.

Lou, P., Q. Liu, Z. Zhou, H. Wang, and S. X. Sun. 2012. "Multi-agent-based Proactive–reactive Scheduling For a Job Shop". *The International Journal of Advanced Manufacturing Technology* 59(1):311–324.

Mirjalili, S. 2019. *Genetic Algorithm*, 43–55. Cham: Springer International Publishing.

Mohan, J., K. Lanka, and A. N. Rao. 2019. "A Review of Dynamic Job Shop Scheduling Techniques". *Procedia Manufacturing* 30:34–39.

Mustafee, N., S. Brailsford, A. Djanatliev, T. Eldabi, M. Kunc, and A. Tolk. 2017. "Purpose and Benefits of Hybrid Simulation: Contributing To The Convergence of Its Definition". In *Proceedings of the 2017 Winter Simulation Conference*, edited by V. Chan, A. D'Ambrogio, G. Zacharewicz, N. Mustafee, G. Wainer, and E. Page, 1631–1645. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Naderi, B., M. Zandieh, and S. F. Ghomi. 2009. "Scheduling Sequence-dependent Setup Time Job shops With Preventive Maintenance". *The International Journal of Advanced Manufacturing Technology* 43(1-2):170.

Pachpor, P. S., R. Shrivastava, D. Seth, and S. Pokharel. 2017. "Application of Petri Nets Towards Improved Utilization of Machines in Job Shop Manufacturing Environments". *Journal of Manufacturing Technology Management*.

Sotskov, Y. N., N. M. Matsveichuk, and V. D. Hatsura. 2020. "Schedule Execution For Two-machine Job-shop To Minimize Makespan With Uncertain Processing Times". *Mathematics* 8(8):1314.

Sotskov, Y. N., and N. V. Shakhlevich. 1995. "NP-hardness of Shop-scheduling Problems With Three Jobs". *Discrete Applied Mathematics* 59(3):237–266.

Taillard, E. 1993. "Benchmarks for Basic Scheduling Problems". *European Journal of Operational Research* 64(2):278–285.

Tamilarasi, A. et al. 2010. "An Enhanced Genetic Algorithm With Simulated Annealing for Job-shop Scheduling". *International Journal of Engineering, Science and Technology* 2(1):144–151.

Tavakkoli Moghaddam, R., F. Jolai, F. Vaziri, P. Ahmed, and A. Azaron. 2005. "A Hybrid Method For Solving Stochastic Job Shop Scheduling Problems". *Applied Mathematics and Computation* 170:185–206.

Topcuoglu, H., S. Hariri, and M.-Y. Wu. 2002. "Performance-effective and Low-complexity Task Scheduling for Heterogeneous Computing". *Institute of Electrical and Electronics Engineers, Transactions on Parallel and Distributed Systems* 13(3):260–274.

Wang, S., and J. Yu. 2010. "An Effective Heuristic for Flexible Job-shop Scheduling Problem with Maintenance Activities". *Computers & Industrial Engineering* 59(3):436–447.

Xiong, H., S. Shi, D. Ren, and J. Hu. 2022. "A Survey of Job Shop Scheduling Problem: the Types and Models". *Computers & Operations Research*:105731.

Zhao, P., P. Zhao, and X. Zhang. 2006. "A New Ant Colony Optimization for The Knapsack Problem". In *2006 7th International Conference on Computer-Aided Industrial Design and Conceptual Design*. November 17th-19th, Hangzhou, China, 1-3.

AUTHOR BIOGRAPHIES

JOEP OOMS is a graduate master student in Industrial Engineering and Management at the University of Groningen. He is specializing in the track of Production Logistics Engineering. His e-mail address is j.j.j.ooms@student.rug.nl.

ALEXANDER HÜBL is a member of the Engineering Systems and Design Group at the University of Groningen (Netherlands). He holds a PhD in logistics and operations management from the University of Vienna, Austria. His research interests include discrete-event simulation, agent-based simulation, queuing theory, hybrid modeling and their applications in logistics and operations management. His email address is a.hubl@rug.nl.