JJMIE

Volume 16, Number 5, December. 2022
ISSN 1995-6665
Pages 793– 809

Jordan Journal of Mechanical and Industrial Engineering

An Optimization Approach for Predictive-Reactive Job Shop Scheduling of Reconfigurable Manufacturing Systems

A.A. Abdul Rahman[a*], O.J. Adeboye[a], J.Y. Tan[a], M.R. Salleh[a], M.A. A.Rahman[b]

[a]*Faculty of Manufacturing Engineering, Universiti Teknikal Malaysia Melaka, Hang Tuah Jaya, 76100 Durian Tunggal, Melaka, Malaysia.*

[b]*School of Mechanical and Aerospace Engineering, Queen's University Belfast, 123 Stranmillis Road, Ashby Building, BT9 5AH Belfast, United Kingdom*

Received 1 Aug 2022 *Accepted 31 Oct 2022*

Abstract

The manufacturing industry is now moving forward rapidly towards reconfigurability and reliability to meet the hard-to-predict global business market, especially job-shop production. However, even if there is a properly planned schedule for production, and there is also a technique for scheduling in Reconfigurable Manufacturing System (RMS) but job-shop production will always come out with errors and disruption due to complex and uncertainty happening during the production process, hence fail to fulfil the due-date requirements. This study proposes a generic control strategy for piloting the implementation of a complex scheduling challenge in an RMS. This study is aimed to formulate an optimization-based algorithm with a simulation tool to reduce the throughput time of complex RMS, which can comply with complex product allocations and flexible routings of the system. The predictive-reactive strategy was investigated, in which Genetic Algorithm (GA) and dispatching rules were used for predictive scheduling and reactivity controls. The results showed that the proposed optimization-based algorithm had successfully reduced the throughput time of the system. In this case, the effectiveness and reliability of RMS are increased by combining the simulation with the optimization algorithm.

Keywords: Scheduling; Simulation; Optimization; Genetic Algorithm; Predictive-Reactive; Reconfigurable Manufacturing System.

1. Introduction

The contemporary market continues to drive all kinds of companies and businesses, particularly manufacturers, towards flexibility. The control of today's production systems is becoming more complex due to an increasing number of product variants, short-time delivery requirements and non-standardized production processes.[1], [2], [3]. The complexity and limitations of production processes cause the products throughput time has greatly increased and unable to achieve due date requirements [4]. Manufacturing sectors are forced to handle demand fluctuations, rapidly adopt new products and order changes to make sure that the products are finished within a specific time [5].

Reconfigurable Manufacturing Systems (RMS) are primarily designed for rapid change in their structure [6], to quickly adjust their production capacity and functionality, within a part family, in response to market changes [7]. Cost, product quality, and market reactivity are the three primary aims of any production system. Designing manufacturing systems with upgradable capacity and adaptable functionality enables responsiveness. The benefits of RMS are highlighted when compared to Dedicated Manufacturing Systems (DMS) & Flexible Manufacturing Systems (FMS) [8] from the standpoint of these objectives [6].

Reconfigurable Manufacturing Systems (RMS) has been quite common in recent research works, and most algorithms, dispatching rules, and strategies have already been developed. However, there is no clear framework or specific strategy developed to get these systems reliable and easy to control [9], [5]. The majority of studies in job scheduling concentrate on static scheduling constraints and do not consider dynamic factors [10]. Conventional approaches suggest a high approximation of real systems and are complex in the formulation; indeed, due to the complexity of a large number of variables and restrictions, most of the current algorithms do not give good results in a reasonable time [11], [12], [13], [14], [15].

The predictive-reactive approach can adapt to rapid changes in the shop floor's execution and provide a flexible schedule [16]. It is very difficult to perform optimization processes analytically during such complicated processes, hence simulation-based optimization is useful [17]. On the other hand, scheduling and controlling with simulation-based optimization can increase the performance and efficiency of the manufacturing systems, and provide an easy and fast evaluation of new layouts and schedules with direct production control [18], [19], [20], [21].

* Corresponding author e-mail: azrulazwan@utem.edu.my.

Due to dynamic job shop scheduling problems being Non-deterministic Polynomial-time – hard (NP-hard) combinatorial optimization problems, heuristic methods are useful for solving these types of problems [10]. Hence, this research focused on modelling the simulation model for RMS that can provide versatility in system layout and product mix with flexible routing and production sequence. The optimization framework was developed based on the model schedule [22], [23]. In this research, the combination of simulation and optimization-based algorithms for scheduling the RMS under various optimization restrictions was studied. For the predictive part, the feasible schedule for the RMS flow shop was predicted and decided. Rule-based simulation and optimization are then implemented into the schedule: first, a rough schedule was determined using an optimization algorithm, and then rule-based simulation systems were used to refine the schedule to obtain the most optimal results. For the reactive phase, the schedule obtained was adjusted and validated by the real-life system.

The application of GA with the dispatching rules can effectively optimize a manufacturing system providing engineers with the needed flexibility and control in the industry 4.0 context. The result obtained can be considered an important contribution to the research community in the field of industrial Engineering and smart manufacturing systems.

2. Literature review

Literature reviews for several papers regarding the different topics which are related to this study are summarized into two categories, which are state-of-the-art research and state-of-the-art implementation. All the summarized papers are research articles for the most recent five years from now (2016 – 2020) and are applied in the field of manufacturing.

Among all these papers related to the simulation field, including the objectives, application of the simulation process and research results, it can be summarized that majority of the simulation process is carried out for experiments purposes in a simple production system without considering the dynamic aspects of the manufacturing systems. However, time becoming one of the most common objectives in research and simulations also proved to obtain satisfactory results in those experiments.

Besides, some papers were studied on several methods to solve related manufacturing problems with evolutionary algorithms, together with the technique chosen, objectives functions, application, and future study. The majority of the study had shown that the genetic algorithm is the most popular technique used in the study, but not the basic Genetic Algorithm (GA). The GA is integrated with other heuristics or methods for experiment purposes. Moreover, the objectives functions are majority based on time, and suggestions for future studies are more to adapting the proposed algorithm or strategy in different situations or on different objectives. However, there is still a lack of research on the parameters for GA used in experiments on simulation optimization problems, and also a lack of studies focusing on RMS which included dynamic aspects and focus on complex routings and product mix.

Besides, according to Doh et al. [19], previous research on the topic of flexible job shop scheduling can be divided into two categories: single process plans with only alternative machines and multiple process plans with both alternative operations and machines. The reconfigurable job-shop scheduling problem consider in this study has a significant difference compared to previous studies in terms of the system's operational characteristics. Part of the difference includes, the number of transportation equipment is limited, the component input sequencing considered, and the material flow in the system is considered based on the flexible routing and the machines for a specific operation.

2.1. State-of-the-art Research

The papers that emphasized proposal, investigation, analysis and study of specific processes in manufacturing systems to establish facts and reach new conclusions are classified into this section. The papers are divided into different fields based on the research.

2.1.1. Simulation and Optimization

In the simulation research area, Mourtzis [24] had done a review focused on scholarly peer-reviewed journals that use modelling in manufacturing-related fields over 58 years. The exponential growth in publications on the subject reveals the ever-increasing importance of simulation in manufacturing. To estimate the total amount of relevant work, a search in the Scopus database using the keywords (a) simulation in manufacturing and (b) simulation in the design and operation of manufacturing systems was conducted to get the exact quantity of the total related work. The first search yielded more than 23,000 publications, while the second yielded more than 10,000. As a result, simulation is an ever-increasing part of manufacturing that can help with a variety of issues.

Figure 1 depicts the progression of research on this subject over time, with research findings classified from 1970 until today. More precisely, the results of this study show that there has been a significant rise in simulation-related publications over the last decade. Hence, there is still space for further research and development in this area.

In recent 5 years from 2016 - 2021, several publications mentioned and emphasized the simulation tool analysis. Siderska [25] analysed a single nail production line by introducing the construction of a simulation model for production and logistic processes conditions using Tecnomatix Plant Simulation and successfully proved that Tecnomatix Plant Simulation is an effective IT tool for increasing the efficiency of the existing system, optimizing resource consumption, limiting stocks and shortening the production time. Yann et al. [26] aimed to optimize the production control system in product customization manufacturing environments by generating and tested on four models with different generic routings with a sample of data from an industrial case using the software Work in Progress Simulation (WIPSIM). The author concluded that the proposed models generated worthwhile generic routings and help manufacturers to make decisions based on their product customization-specific context.

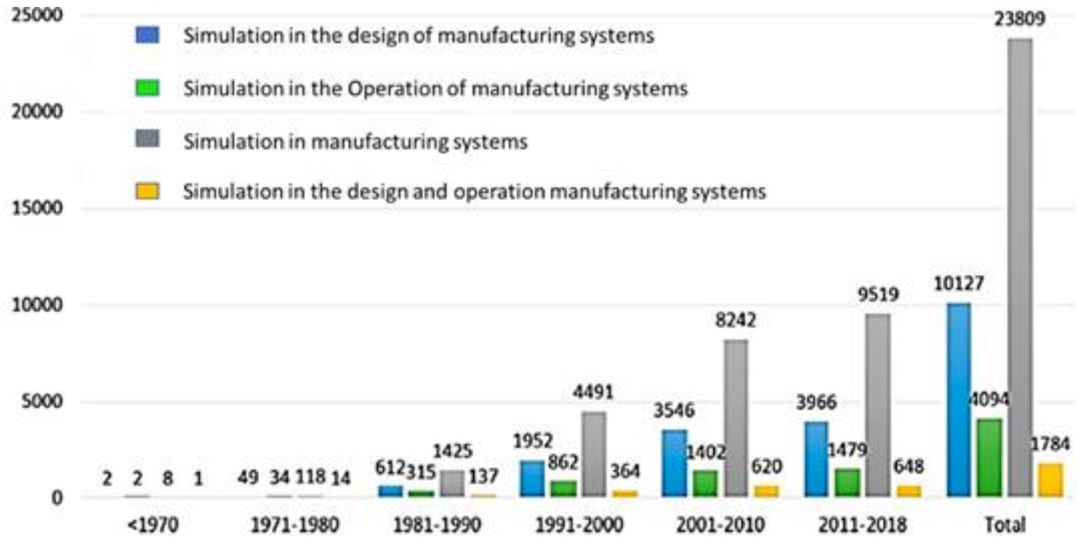


Figure 1. Number of publications based on Scopus database score [19].

For using simulation for solving scheduling problems, Doh et al. [19] researched simulating the scheduling problem in a flexible job shop. The flexible job shop is equipped with a conventional job shop and a reconfigurable manufacturing cell, with the objectives, fixed on minimizing makespan, mean flow time and mean tardiness, and the best rule combinations were identified for each of the three performance measures. Nasiri et al. [15] developed a simulation-based real-time scheduling composite dispatching rule approach in open job shop scheduling to minimize the mean waiting time of jobs, resulting in the optimal composite dispatching rule dominating the known dispatching rules. Xiong et al. [27] had done a simulation-based analysis of dispatching rules for scheduling in a dynamic job shop with batch release to minimize the total tardiness and the percentage of tardy jobs. The four new proposed dispatching rules effectively minimise the tardiness of jobs and the relative performance of dispatching rules can be affected by some model parameters. Lin et al. [28] carried out experiments and comparisons of statistical tests for several search methods for Automated Guided Vehicle (AGV) and machines in Flexible Manufacturing Systems (FMS) to evaluate the performance of scheduling decisions by proposing a simulation-based optimization to address the simultaneous scheduling of vehicles and machines. The proposed method successfully enhanced solution quality and search efficiency.

2.1.2. Algorithms and Heuristics

The research involving the development of algorithm or heuristics were popular in these 5 years from 2016 – 2021, and most of them used simulation to validate the algorithms. Few papers use the GA approach together with other techniques for investigation. Kundakci and Kulak [10] proposed a GA, a new Karmarkar-Karp (KK) heuristic and dispatching rules to approach solving job-shop scheduling problems with dynamic events to minimize the makespan. The authors wish to apply to different environments and use different performance measures for further study. Deng et al. [29] conducted experiments and compared the proposed Bee Evolutionary Guiding Nondominated Sorting Genetic Algorithm II (BEG-NSGA-II) with benchmark problems to minimize the maximal completion time, the workload of the most loaded machine, and the total workload of all machines. However, this study did not concentrate on dynamic and real-time scheduling problems. Piroozfard et al. [30] had proposes an improved multi-objective evolutionary algorithm with GA for solving the newly extended bi-objective problem with considerations of environmental objectives, which is to minimize the total carbon footprint and total late work criterion. In future, the author would like to consider other scheduling criteria, objectives and heuristics approach. Lin et al. [28] proposed a Local search Genetic Algorithm Optimal computing budget allocation (L-GAOCBA) algorithm to address the simultaneous scheduling of vehicles and machines in FMS to evaluate the performance of scheduling decisions including stochastic elements, such as vehicle congestion, deadlock, and uncertain processing time, and decided to consider multi-objectives in future. Zan et al. [31] proposed a new Pareto-based GA for solving the multi-objective scheduling problem of deadlock-prone Automated Manufacturing Systems (AMSs) with limited resource capacity, aimed to optimize makespan, mean of earliness and tardiness, and mean completion time. Sivarat and Apichat [32] develop the simulation-optimization approach using artificial neural networks and GA to support observational data-driven manufacturing capacity planning for Small and Medium-sized Enterprises (SMEs) by reducing the amount of work involved in the exploitation of the data.

There were also some papers which used techniques other than GA, and the majority of the research was focused on time as an objective. Valledor et al. [33] proposed a rescheduling architecture for solving the problem based on a predictive-reactive strategy and a new method to calculate the reactive schedule in each rescheduling period to evaluate the dispatching rules for analysis of makespan, total weighted tardiness and stability. The result proved that the random rule provides better behaviour compared to other evaluated rules and a lower ratio of non-dominated solutions compared to Apparent Tardiness Cost (ATC) and First-in-first-out

(FIFO) rules, however, this approach could be tested in a dynamic system in future. Touzout and Benyoucef [34] aimed to reduce the total production cost, total completion time and maximum exploitation time by proposing and comparing three hybrid heuristics which are the Repetitive Single-Unit Process Plan (RSUPP) heuristic, Iterated Local Search on Single-Unit Process Plans heuristic (LSSUPP) and Archive-Based Iterated Local Search heuristic (ABILS) using the generated numerical results in RMS. The author would like to compare and analyse other local search-based metaheuristics with RMS. Zheng and Jin [35] proposed an improved Back and Forth Nudging algorithm (IBFN) to use in single-machine lot scheduling problems of indivisible jobs for minimizing the total completion time of jobs. However, the proposed algorithm required enhancements in future. Amir et al. [36] created a metamodel to replace the simulation experiments aimed at reducing the computation and test the proposed method on Stochastic Job Shop Scheduling Problem (SJSSP) by presenting a new Evolutionary Learning Based Simulation Optimization (ELBSO) method embedded within the Ordinal Optimization using Genetic Programming (GP). The author wanted to apply this improved method in solving existing production planning and scheduling problem. Gheisariha et al. [32] proposed an enhanced multi-objective algorithm which is the Enhanced Multi-Objective Harmony Search (EMOHS) algorithm and a Gaussian mutation. [32] also designed a simulation-optimization framework for implementing the rework process and compare the algorithm with the well-known 5 types of algorithms for minimizing both maximum completion time and mean tardiness.

2.2. State-of-the-art Implementation

The papers that emphasized the realization of an application, execution of algorithms and model, are classified into this section. The papers are divided into different fields based on the papers.

2.2.1. Simulation and Optimization

The studies that were mainly based on the implementation of the simulation process into real industry cases were included by experimenting and analysing the results generated. Niehues et al. [14] proposed a WIP regulating method for production control for job-shop productions in the automotive industry to reduce the impacts of control activities on orders by experimenting with the production system. The results shown in the simulation model demonstrated the suitability and effectiveness of production control in manual job-shop production systems. Kuck et al. [38] proposed an adaptive simulation-based optimization approach for individual selection of dispatching rules in production control by conducting experiments on a scenario from the semiconductor industry which resulted in improved solution quality at the beginning of the optimization process (local optimum), but not very good in global optimum. Wang et al. [39] analysed a semiconductor packaging facility for enhancement of the sustainability of a factory simulation model by proposing five strategies, while the result proved that the three strategies proposed successfully reduced the requirement of money, time, and effort in building the factory simulation model. However, the sustainability of a simulation model is still uncertain for several years after it is built. Grabowik et al. [40] analysed a single-car manufacturing line to examine production efficiency based on a few proposed changes in system organization. The author declared that simulation is useful in checking different models of organizational solutions and following the long-term behaviour of the system simply and effectively.

2.2.2. Algorithms and Heuristics

There are only several papers that implemented the algorithms, especially GA into the real situation system. Niehues et al. [21] proposed a new approach in sequence scheduling for a job shop control system and verify its effectiveness through simulation with Tecnomatix Plant Simulation and MATLAB using GA. The author aimed to improve the adherence to delivery dates was fulfilled by improving due date compliance. Sobottka et al. [41] developed a hybrid simulation optimization module for use in a novel production optimization tool using GA in a food processing facility by considering the material flow and thermal-physical behaviour for the improvement of the energy efficiency of the production system. Wang et al. [42] proposed a two-stage energy-saving optimization method for Flexible Job Shop Scheduling Problem (FJSSP) in the metal-production industry using GA and Particle Swarm Optimization (PSO) to reduce energy consumption and production cost. The author would like to integrate the proposed method with big data technology in future studies.

All these studies concluded that in the literature, there is still a gap between the flexibility of the system and the complexity of the product mix and intelligence decision-making in optimized results, considering the design complexity of the manufacturing system which can be implemented into the dynamic condition. Besides, the majority of the research is taking makespan or completion time as objectives. Hence, makespan become the most common performance measure to be studied among the others. This lead gave proper direction to the experimentation part of this research paper.

3. Method

To close the gap, an integration approach is proposed. The combination of simulation and optimization-based algorithms with the predictive-reactive approach for scheduling the RMS under various optimization restrictions was studied through experiments. For the predictive part, the feasible schedule for the RMS job shop is predicted and decided. Rule-based simulation and optimization are then implemented into the schedule: first, a rough schedule was determined using an optimization algorithm, and then rule-based simulation systems were used to refine the schedule to obtain the most optimal results. For the reactive phase, the schedule obtained is adjusted and validated by the MONTRAC monorail system. The results obtained from the experiments are compared and analysed to find out the effectiveness of the proposed method structure, algorithms and architecture.

3.1. Real System

This research is using MONTRAC, manufactured by the company MONTRACTEC GmbH, Germany as an example. This system is a modular monorail system that allows the interlink of production processes between robots and workplaces more flexibly. Pioneering medical institutions and manufacturing firms in automobile, chemicals, household products, optics, food, medicinal and pharmaceutical markets using this system to increase their product throughput and reduce cycle times. In conjunction with this system, the MONTRAC shuttles are the main components of the MONTRAC system, which are intelligent single or twin-axle conveyors mounted with an onboard power supply. The shuttles are moving self-centred on the monorail, fitted with state-of-the-art sensors that avoid possible collisions with barriers or other shuttles. Each shuttle is operated by an axle-located, maintenance-free, low-voltage engine. Shuttle velocity and stopping positions are defined by cams on the T-grooves along the track [43].

Figure 2 shows the schematic illustration of the MONTRAC system. The real system's structure is modelled using Tecnomatix Plant Simulation V12 and consists of rails, buffers and workstations. Firstly, parts will be loaded into the system, transported by rail-guided vehicles and unloaded in buffers near workstations for further processing. After the specific process had been done at the workstation, the product will be loaded in another buffer and waiting for pick up by the rail-guided vehicles again to transport further. After the product had finished all the operations, the product will be unloaded at the final buffer for further processing.

3.2. Computer Simulation Software

The real system is modelled using computer simulation software which is Tecnomatix Plant Simulation V12 by Siemens. Tecnomatix Plant Simulation is an object-oriented 3D program founded by German company Siemens PLM Software, the leading global supplier of software for Product Lifecycle Management (PLM) and Manufacturing Operations Management (MOM), which specifically designed for discrete production simulation process and modelling into a digital model. The models built in the Plant Simulation can be run by experiments and scenarios for analysis of causes and effects in the current production systems or the newly designed systems without disturbing the working process of production systems. Plant simulation is provided with well-developed analytical tools that promote the analysis of a system's bottlenecks, together with the illustration of diagrams and statistic configurations, and even can import 3D geometrical models from CAD systems, visualise the entire manufacturing system including workstations and transportations [25], [40].

Job Shop Scheduling Problem

In this study, the job shop scheduling problem is described as a set of n jobs, J_i where i = 1, 2, . . ., n which are going to operate on a set of m machines, M_k where k = 1, 2, . . ., m. Every single job contains a set of operations, each of which needs to be processed during an uninterrupted period of a given length on a given machine. Operation of the i_{th} job on the k_{th} machine will be denoted by O_{ik}. There are several constraints and assumptions set on jobs and machines as follows [44], [45], [1].

1. The job release dates are time T = 0.
2. All the machines are available at time T = 0.
3. The number of machines and jobs is finite and constant in time (with respect to their characteristics).
4. The machine breakdowns and the setup times are statistically included in the processing times.
5. Each machine can process only one operation at a time and each job can be processed by only one machine at a time.
6. Once a job begins processing, it cannot be interrupted until it is completed, and no precedence constraints exist among jobs.
7. The due dates are specified.
8. The time to put the parts on or to take them off the material handling vehicles is negligible.

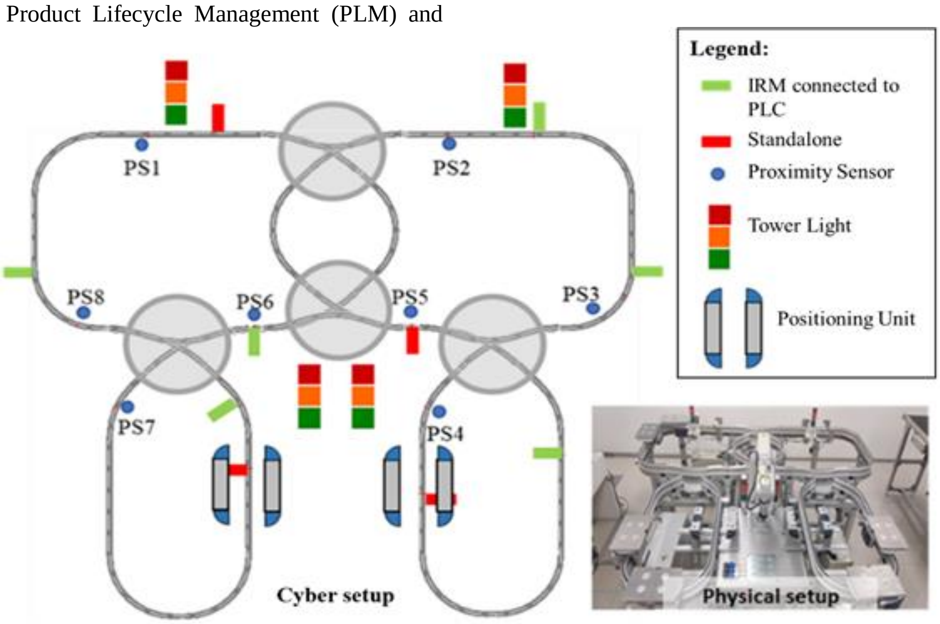


Figure 2. Schematic MONTRAC monorail material handling system

3.2.1. Scheduling Procedure

Before the scheduling process starts, there are some parameters required to be decided, such as the objective functions. The selection of objective function is decided by the industry itself, while in this study, the results will be reported for the objectives of minimizing the makespan denoted by Cmax which is defined as the time when the last job leaves the system:

$$C_{max} = \max (C_1, C_2, \ldots, C_{max}) \quad (1)$$

where C_i is the completion time of the job, J_i.

Besides objective functions, the initial data and some related settings are required to be provided in the model, including the schedules of the workstations, speed of transporters, the number of parts to be entered into the system etc. The schedules provided then undergo process simulation, together with the application of optimization methods including priority dispatching rules and GA to obtain an optimized solution.

The reconfigurable job shop considered in this study has both operations and routing flexibilities, which can be interpreted in the form of a multiple-process plan, for example, each component can be processed through alternative machines, and each component can be transported through alternative routes. The scheduling problem contained three decision variables: (a) selection of operation machines for each part; (b) sequencing of parts to be entered into the RMS, and (c) routing of the parts transported to each machine. These three decisions are made at the same time by combining operation machine selection rules, input sequencing rules and part sequencing rules with genetic algorithm. The rules combination performances will be tested through simulation experiments, where the dispatching rules that are considered in this study as recommended by Zeestraten [46] are shown in Table 1.

After achieving a suitable schedule, the simulation is run with the logging option activated to obtain logs that are used to control the real system. The logs obtained contains information including the start and stop of processing time at every station, all transportation activities, and the sequence of products and paths for every part. During the control stage of the real system, rescheduling requests might be triggered due to disturbances such as station failure, planned maintenance or sudden changes in order. Therefore, the reactive loop represents the transfer of system status information to the model and the reactivation of the scheduling procedure with new inputs. The scheduling procedure is illustrated in Figure 3.

Table 1. The dispatching rules that have been considered for optimization purposes in the model.

Dispatching rule	Explanation
Earliest Due Date (EDD)	Select the job that has the earliest due date first.
First Come First Served (FCFS)	Select the operation that is available first.
Shortest Processing Time (SPT)	Select the job that has the shortest processing time of the first process.
Longest Processing Time (LPT)	Select the job that has the longest processing time of the first process.
Fewest Operations Remaining (FOPR)	Select the job that has the smallest number of successive operations.
Most Operations Remaining (MOPR)	Select the job that has the largest number of successive operations.
Shortest Remaining Processing Time (SRPT)	Select the job that has the shortest sum of processing times
Longest Remaining Processing Time (LRPT)	Select the operation that has the longest sum of processing times.

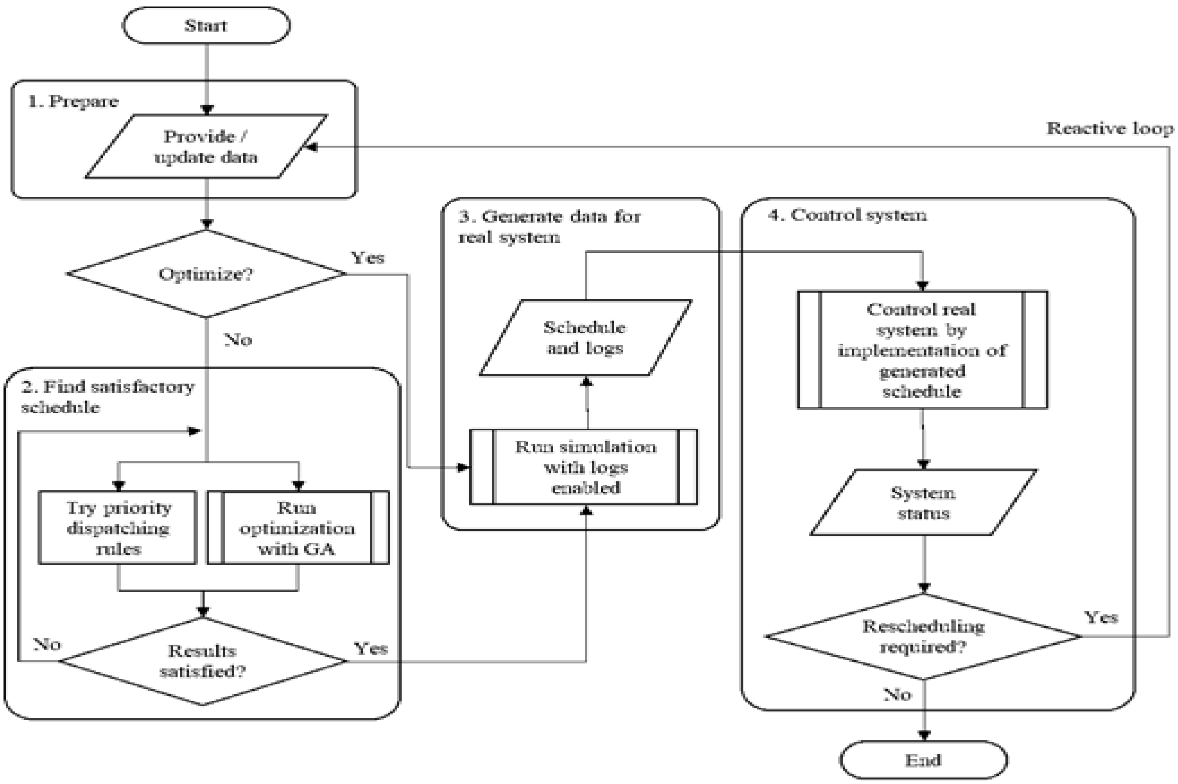


Figure 3: Algorithm of scheduling procedure.

3.3. Construction of Feasible Schedule

A schedule defines the execution sequence of all operations for all jobs on machines [47]. Before the simulation process begins, several feasible schedules are required as primary input data for the construction of the simulation model. In this study, these feasible schedules are built with a predictive approach, which included the process plan for every type of product, the time schedule that specified the durations of every operation, the workstation's plan that indicated which type of process to be operated and also the due date schedule together with the quantity required for the customers.

3.3.1. Process Plan

A process schedule is constructed for the process sequence required to be done for each product type as illustrated in Table 2.

There is a total of 10 types of product types and each of the product types has different process sequences. In this study, a maximum of 5 processes for each product type will be considered, but the simulation model is built to fit a maximum of 9 processes for each product type to reach the dynamic aspects of market requirements.

The sequence of each product type can also be decided whether or not to follow the sequences during the operations. In this case, the schedule had been set to follow the partial order, where the product that set to keep the process sequence for all product types except product Type _6, Type_7 and Type_8, which means that these product types can proceed to any operations first depends on the system.

Table 2. The process plan for every product type.

Name	Keep Sequence	1st Process	2nd Process	3rd Process	4th Process	5th Process
Type_1	true	Proc_1	Proc_3	Proc_4		
Type_2	true	Proc_2	Proc_5	Proc_1	Proc_4	
Type_3	true	Proc_5	Proc_2	Proc_4	Proc_3	
Type_4	true	Proc_3	Proc_4	Proc_5	Proc_1	Proc_2
Type_5	true	Proc_2	Proc_4	Proc_3	Proc_1	
Type_6	false	Proc_2	Proc_3	Proc_4		
Type_7	false	Proc_1	Proc_4	Proc_5		
Type_8	false	Proc_2	Proc_4			
Type_9	true	Proc_5	Proc_3	Proc_1		
Type_10	true	Proc_2	Proc_3	Proc_5		

3.3.2. Workplace Operation Time Schedule

There is a total of 4 workplaces or stations to be included in the system in this study, which are given the names H1, H2, H3 and H4. Each workstation has a specific operation time schedule that indicates different operations with different processing times. Each workstation is also fixed with a different type of process, and in this study, the maximum amount of operation to be done in a specific workstation is fixed with 2 types of operations. Table 3 shows the time needed for every operation (Proc_1, Proc_2, Proc_3, Proc_4) based on the product type (Type_ 1 to Type_10), and also the workstations that are eligible to carry out the specific process.

Table 3. The workplace operation time schedule for every product type.

	H1	H2		H3		H4	
	Proc_1 (s)	Proc_1 (s)	Proc_2 (s)	Proc_3 (s)	Proc_5 (s)	Proc_4 (s)	Proc_5 s)
Type_1	1200	1200	-	1800	-	420	-
Type_2	1500	1500	900	-	240	600	240
Type_3	-	-	660	510	420	735	420
Type_4	180	180	120	300	210	135	210
Type_5	1830	1830	1200	915	-	495	-
Type_6	-	-	600	720	-	1200	-
Type_7	600	600	-	-	600	900	600
Type_8	-	-	840	-	-	1500	-
Type_9	1080	1080	-	480	1200	-	1200
Type_10	-	-	2100	660	600	-	600

3.3.3. Due Date Schedule

Different product has different quantity and due date to meet customers' requirements, therefore due date schedule is constructed with the desired amount of product and the due date as well, as shown in Table 4.

Table 4: The due date schedule for every product type.

Product Name	Qty (unit)	Due Date (date, time)
Type_1	5	01.12.21, 13:10
Type_2	10	01.12.21, 11:35
Type_3	12	02.12.21, 10:55
Type_4	4	01.12.21, 14:55
Type_5	8	02.12.21, 16:15
Type_6	10	01.12.21, 11:40
Type_7	3	01.12.21, 09:40
Type_8	6	01.12.21, 20:40
Type_9	15	02.12.21, 11:40
Type_10	2	01.12.21, 18:40

The due date is written in the format of the date (dd.mm.yy), then followed by the time. All the products have to be finished before the due date with the specific amounts, for example, Type_1 product is required to produce 5 quantities before the 1st of December 2021, 1.10 pm.

3.4. Construction of Simulation Model

The simulation model is built to represent the real system, including the position of the buffers and workstations. There is a total of 11 buffers, 6 rail-guided vehicles and 4 workstations included. The model is built with high flexibility where the layout and the position of buffers and workstations can be modified or added easily. The construction of a simulation model is mainly divided into two categories, one is for controlling the simulation process of the model and the other one is for the preparation of necessary input data for the simulation process.

3.4.1. Path Generations

The most important methods in this model are the path generations method, which indicated the generations of the products' paths along the whole system. Every path generated consists of a sequence of objects including tracks, buffers and workstations, which implies the process plan and allocation. This method written aimed to obtain all the information regarding all the possible transportation routes for every single job and product type intelligently without needing to enter the information manually into the settings for the optimization process. This method also acted as preparation for the process of population initialization and evaluation required for the GA. All the output generated from this method is stored in a table named "PathsTable".

The path generations are started with product type by generating a sequence of processes with the permutation method. After that, for every sequence of processes, a sequence of workstations is created. The concept of generations of all these outputs is by checking every station table to ensure whether the specific stations provide the process. After all possible station sequences are generated, then search for paths beings. The method follows the tracks from one station to the next station until all possible ways are discovered. The application of complexity level is also included in this method by eliminating long paths and keeping the shorter paths.

3.4.1.1. Permutation

Especially when a product is not necessary to strictly follow the process plan, possible sequences of processes are generated through this method. This method is programmed to receive a string of characters and return a list of permuted strings. The algorithm of this method is illustrated in Figure 4 by taking characters 1, 2, 3, and 4 as an example.

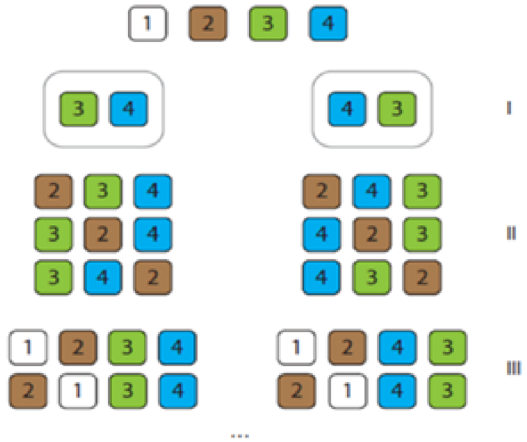


Figure 4. Permutation algorithm illustration.

The algorithm of permutation started with the last character selected which is 4. Then, the preceding character 3 is inserted into every available position, which is before and after 4. This resulted in two strings as shown in stages 1, that is 34 and 43. After this stage, the same procedure continues with inserting the next character into every string. Taking the first string 34 as an example, character 2 is inserted before, between and after characters 3 and 4 respectively, thus the result obtained are 234, 324 and 342. The action is performed until the first character is inserted into all strings. The total number of obtained permuted strings is N! where N is the number of characters. For this given example, the total number of obtained permuted strings is 24.

3.5. GA Setup and Control

In this study, GA is used to find the optimum solution by choosing one of the paths for each instance of a product, combining all products in different sequences and running simulations to assess their objective function values. The GA in this model is run by an object named "GAV12V" which is a frame that contains a lot of methods and other objects that are used to control the GA optimization process. GAWizard provided some settings or options related to the GA process for the user to choose and enter manually, and most of these settings are transferred with a method named "SetGA" between the user interface dialogue and the wizard.

The GA process run by GAWizard which is a user interface created by the author in the simulation software program and is divided into a few steps, which involved the definition of chromosomes in the initial population, the selection of parent's chromosomes, and the generation of offspring chromosomes. The process of GA will be terminated when there is no improvement in the fitness value of the best individual during N generations. The termination of the GA process will also be triggered when the set time limit had reached.

3.5.1. Initialization of the Population of Chromosomes

The GA is started by defining chromosomes, each chromosome stands for one entity with its respective work order number. Every chromosome encodes with a sequence of operations. In the generation of the initial population, each chromosome is initialized by following the entity arrangement in "Release_List" where the entities are generated in random sequences. Table 5 shows the example of chromosome representation for a population of chromosomes generated at random.

There are a total of 75 entities to be processed, therefore the chromosomes generated for each generation and each individual will be a total of 75 chromosomes. The generation of chromosome populations are depending on the size of the generations and the number of generations. For example, if the generation size is set to 10, then 10 genes or 10 individuals will be generated with 75 chromosomes with random path variants and random entity sequences. If the number of generations is set to 5, then the 10 genes and individuals generated with the 75 chromosomes will be created until 5 generations. During a generation process, the genetic operator's processes will be performed on chromosomes to obtain better solutions.

Table 5: Example of a population of chromosome representation based on product type and work order number.

Chromosome	1	2	3	4	5	6	7	8	9	10
Work order number (Entity)	10	51	26	67	29	55	43	28	3	56
Part Type	2	7	3	9	4	8	6	4	1	8

3.5.2. Selection

Before the genetic operators are performed, GA will authorize a population composed of a large number of individuals to evolve under specified selection rules to a state that maximizes the fitness value. The fitness of an algorithm is a measure of how effectively it has learned to anticipate outputs from inputs. A fitness evaluation aims to provide information to the learning algorithm on which individuals should be allowed to multiply and reproduce and which should be eliminated from the population [48]. In this case, the process of selection is done by Roulette Wheel selection to select the parents. According to Kofjač and Kljajić [49], the Roulette Wheel selection method is the most common method used in GA selection. The Roulette Wheel selection is depending on the fitness values assigned to the chromosomes by fitness functions, while the fitness value is used to relate the possibility of selection with each chromosome. The probability of being selected is denoted as:

$$\mathbf{P_i} = \frac{\mathbf{f_i}}{\sum_{\mathbf{j=1}}^{\mathbf{n}} \mathbf{f_j}} \quad (2)$$

where f_i is the fitness of an individual i in the population, n is the number of individuals in the population and j is the job.

In GAWizard, the parent selection settings can be chosen either deterministic or random. If the option "deterministic" is chosen, then the parents will be selected randomly according to their fitness values with roulette wheel selection, and individuals with good fitness values will be used more often as parents for creating the next generation. However, individuals with bad fitness values also have a chance to be used as parents. While for the "random" option, the fitness values are not used and all individuals are having the same likelihood to be used as parents.

In this model, the option deterministic is chosen and the fitness value is to be set as minimized since the objective function is to minimize the makespan. After a specific number of individuals had been generated based on the generation size, GA will calculate the fitness value for all the individuals, and the individuals with the best fitness value will be selected as parents for further processes.

3.5.3. Crossover

The crossover mechanism is a random process with a probability of crossover and is used to create a new generation of a pair of children's chromosomes from a pair of parent chromosomes via the crossover operation. The crossover operator's average probability ranged between 0.6 and 1.0 [50]. In this model, the crossover processes are done randomly between 75 entities in every generation. There are two types of crossover processes generated by GAWizard, which are order crossover (OX) and partially matched crossover (PMX). OX crossover preserves the relative position or neighbour relation of the items of the solution to each other, while PMX crossover stresses the absolute position of the objects. The crossover algorithm for OX crossover and PMX crossover is illustrated in Figures 5 and 6 respectively.

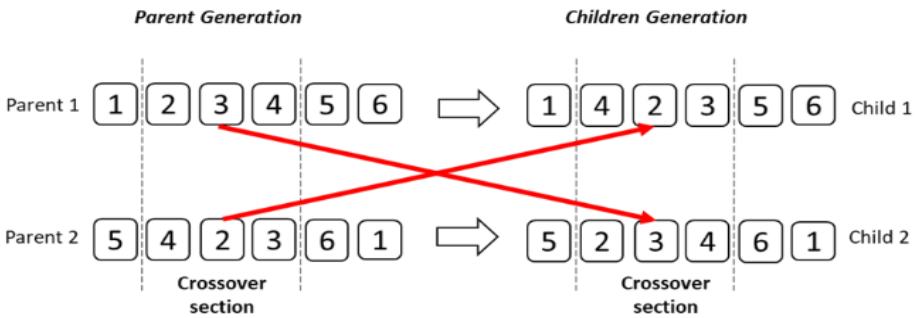


Figure 5. Example of OX crossover between two parents and two children.

For OX crossover, two parents and two offspring are considered as an example. Firstly, the genes (1 5 6) are copied from Parent 1 to Child 1 and located outside the crossover section following Parent 1's sequences, while genes (5 6 1) are duplicated from Parent 2 to Child 2 with the same order and location. The gene position inside the crossover section for both Child 1 and Child 2 will remain empty. The missing genes gap in both children is filled by duplicating the genes from the crossover section from both parents in opposite manner, which means that the genes (2 3 4) from Parent 1 will be duplicated to place in the crossover section in Child 2, while the genes (4 2 3) from Parent 2 will be duplicated to locate in crossover section in Child 1.

For PMX crossover, the first step is the same as in OX crossover where the genes outside the crossover section in both parents will be copied and placed in both children to the gene's original position and sequence. However, the genes in the crossover section in both parents will not be duplicated to the children in the opposite manner and followed the sequences during the process. To produce a feasible schedule, the gap in each child must be filled with the missing genes by taking in order each valid gene from the parent. For example, the genes (2 3 4) in Parent 1 are used to fill the crossover section gap in Child 1 while the genes (4 2 3) in Parent 2 are used to fill the crossover section gap in Child 2 but the sequences are different from the original sequences of the genes in parent's generation.

3.5.4. Mutation

After the crossover process is done, the mutation process will follow. The mutation process is crucial to the GA's success because it diversifies the search directions and prevents convergence to local optima. This process has only involved some offspring randomly. The size is decided by the probability of mutation which the value is typically between 0.0015 and 0.03 [48]. The mutation rate is calculated as:

$$\mathbf{P_m = 1 - \frac{f_{best}}{f_{child}}} \tag{3}$$

where f_{best} is the fitness function value of the chromosome that has yielded the best result, while f_{child} is the fitness function value of the child that requires mutation. The chromosome with the fitness value closer to the f_{best} would have a lower pm than the one with a fitness value closer to the worst value [49].

In this model, the mutation operations are done randomly between 75 entities in every generation. The mutation process is taking part in the allocation task by swapping the locations of genes that were chosen randomly to produce a feasible solution, or by determining a value from the allocation set or chosen from the defined interval randomly and then allocated to the selected gene in sequence task. The mutation algorithm for the mutation process is illustrated in Figure 7.

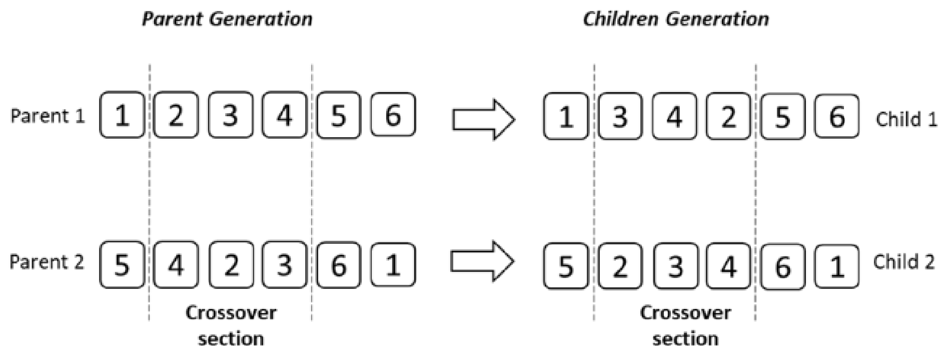


Figure 6: Example of PMX crossover for two parents and two children.

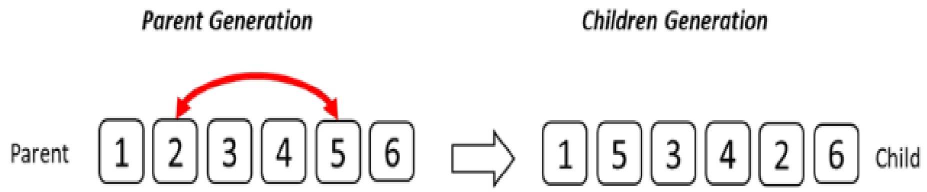


Figure 7: Example of EX mutation in a parent and a child.

4. Results and Discussion

The experiments were conducted with a 2.3 GHz Intel Core i3 processor and 6GB RAM and the heuristics were implemented together with GA using SimTalk which is the programming language used in Tecnomatix Plant Simulation. Due to the lack of benchmarks in the literature related to process plan generation in a reconfigurable manufacturing environment, the experiments are performed with randomly generated instances using GA for the reactive scheduling situation.

The parameters used for analysis and comparison purposes are the number of generations and the generation size, which refer to the parameters in the previous study of Gibbs et al. [50] as follows: generation size = 6, 10, 25 and number of generations = 5, 10. Besides, the makespan, release control with various path generator complexity levels and dispatching rules are being compared. The parameters and conditions included in the experiments are summarized as shown in Table 6.

Table 6: The parameters involved in the experiment.

Parameter / Condition	Values
Maximum number of MUs	18
Number of transporters	6
Transporter speed (m/s)	0.226
Intersection transfer time (s)	0.2
Transporter load/ unload time (s)	5
Keep process sequence	Selective
Use buffer after workplace	Yes
Pick up on the way to buffer	Yes
Pick up after unloading	Yes
Release control	Default, Option 1, Option 2, Option 3
Path generator complexity level	Level 1, Level 2, Level 3
Number of generations	5, 10
Generation size	6, 10, 25
Observations per individual	1

4.1. Makespan

The main objective of this study is to reduce the makespan of the overall process of a reconfigurable manufacturing system, thus the system was run without using GA first to obtain the initial makespan for every type of dispatching rules, then with GA and for a different number of generations and different generation sizes. For both the default and first option (Op1) release control option during the generation of the initial makespan, the system resulted in block condition, while the second and third options (Op2 and Op3) with three levels of path generators generated the results.

4.1.1. Number of Generations

The experiment was run with fix generation size which is 6 with a different number of generations which is 5 and 10 respectively. Figure 8 shows the makespan for every type of dispatching rule to a different number of generations with a generation size of 6 for the reactive scheduling situation. The initial makespan for the situation resulted in the longest duration, while for both the number of generations of 5 and 10 it resulted in a shorter makespan. However, for the reactive scheduling situation, the difference between both generation numbers 5 and 10 is larger. The percentage of reduction in makespan is calculated and tabulated in Table 7.

The percentage of reduction in makespan for reactive scheduling cases is successfully reduced by more than 15% except for the reactive scheduling case of generation number 5. However, generation number 10 for the reactive scheduling case had successfully proven to have a percentage of reduction of more than 15%. It can be concluded that the highest percentage reduction of 38% for the reactive scheduling case. The combination of GA and dispatching rules in finding the optimized schedule of solutions for reconfigurable manufacturing systems is effective even in the small number of generations, also higher generation numbers denoted to better results.

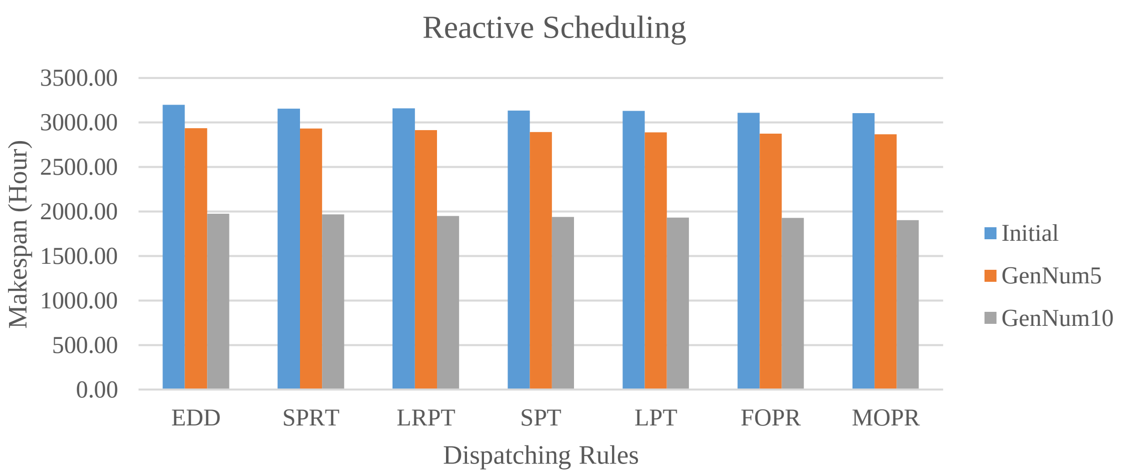


Figure 8. Makespan for rescheduling for the number of generations = 5 and 10 with generation size = 6.

4.1.2. Generation Size

The experiment was then run with a fixed number of generations which is 5 with different generation sizes which are 6, 10 and 25 respectively.

Figure 9 show the makespan for every type of dispatching rule regarding different generation sizes with a generation number of 5 for reactive scheduling situation. The initial makespan for this situation resulted in the longest duration, while the other generation sizes resulted in a shorter makespan compared to the initial results. However, for the reactive scheduling situation, the difference between all three types of generation sizes is more consistent. The percentage of reduction in makespan based on these situations is calculated and tabulated in Table 8.

Table 7: The percentage reduction in makespan for reactive scheduling compared with initial makespan for the number of generations = 5 and 10 with generation size = 6.

Reactive Schedule	Makespan (Hour)						
	EDD	SPRT	LRPT	SPT	LPT	FOPR	MOPR
Initial	3199.043	3155.293	3159.888	3134.600	3128.705	3107.003	3104.078
GenNum5	2935.663	2931.593	2914.122	2894.285	2889.378	2873.448	2867.085
% Reduction	8.233	7.090	7.778	7.667	7.649	7.517	7.635
Initial	3199.043	3155.293	3159.888	3134.600	3128.705	3107.003	3104.078
GenNum10	1974.555	1969.215	1950.927	1937.278	1930.105	1928.515	1902.098
% Reduction	38.277	37.590	38.260	38.197	38.310	37.930	38.723

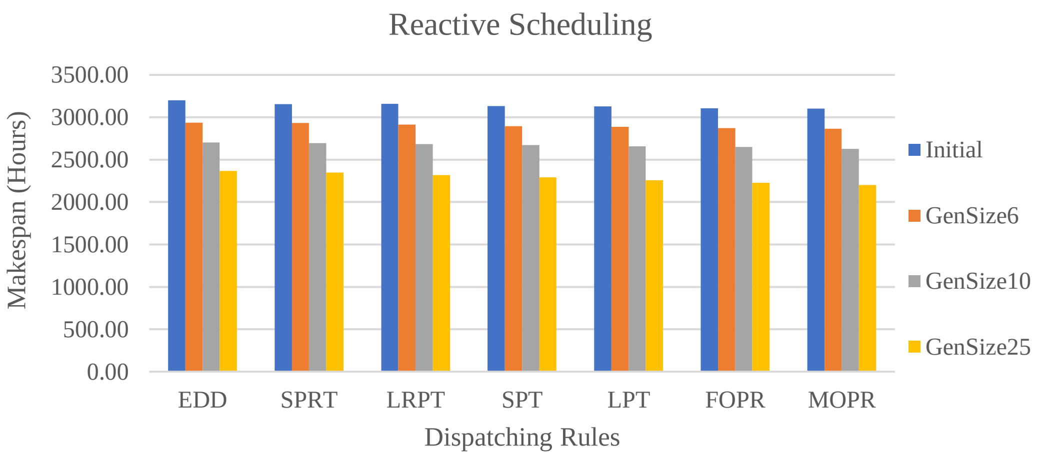


Figure 9. Makespan for reactive scheduling for generation size = 6, 20 and 25 with a number of generations = 5.

Table 8. The percentage reduction in makespan for reactive-scheduling compared with initial makespan for generation size = 6, 10 and 25 with generation number = 5.

Reactive Schedule	Makespan (Hour)						
	EDD	SPRT	LRPT	SPT	LPT	FOPR	MOPR
Initial	3199.043	3155.293	3159.888	3134.600	3128.705	3107.003	3104.078
GenSize6	2935.663	2931.593	2914.122	2894.285	2889.378	2873.448	2867.085
% Reduction	8.233	7.090	7.778	7.667	7.649	7.517	7.635
Initial	3199.043	3155.293	3159.888	3134.600	3128.705	3107.003	3104.078
GenSize10	2703.058	2694.205	2682.417	2671.153	2658.500	2650.988	2628.122
% Reduction	15.504	14.613	15.110	14.785	15.029	14.677	15.333
Initial	3199.043	3155.293	3159.888	3134.600	3128.705	3107.003	3104.078
GenSize25	2369.228	2347.748	2318.100	2291.793	2259.630	2227.118	2201.882
% Reduction	25.939	25.593	26.640	26.887	27.777	28.319	29.065

Based on Table 8, the percentage of reduction in makespan reactive scheduling cases is successfully reduced by more than 15% except for rescheduling cases of generation size = 6. However, other generation sizes = 10 and 25 for reactive scheduling cases had successfully proven to have a percentage of reduction of more than 15%. Again, it can be concluded that, with the highest percentage reduction of 29% for reactive scheduling cases, the combination of GA and dispatching rules in finding the optimized schedule of solutions for reconfigurable manufacturing systems is effective even in small generation size, while higher generation size contributed to better results.

4.2. Dispatching Rules

Bajpai and Kumar [51] stated that combining other approaches with GA can improve effectiveness and efficiency. Hence, the dispatching rules were experimented with GA to identify the results and comparisons of the performances. The performance of the dispatching rules was compared by using the fitness values obtained after each GA run. The fitness value is derived from Rastrigin's function which is defined by Bajpai and Kumar [51] as:

$$\mathbf{Ras(x) = 20 + x_1^2 + x_2^2 - 10(\cos 2\pi x_1 + \cos 2\pi x_2)} \quad (4)$$

Where x_1 & x_2 represent the values of independent variables

Since the objective function is set to be a shorter makespan, hence the direction of optimization is set to be minimum. Therefore, for this study, the smaller the fitness value indicated the better result. The fitness function value for all dispatching rules was tested with a different number of generations and different generation sizes as well.

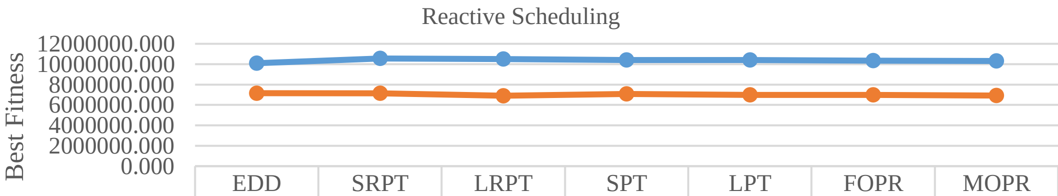
4.2.1. Number of Generations

The experiment was run with fix generation size which is 6 with a different number of generations which is 5 and 10 respectively for the reactive scheduling conditions. The results are illustrated in Figure 10. As a result, for the reactive scheduling, the generation number = 10 resulted in a lower value of fitness compared to the generation number = 5 for all dispatching rules. The dispatching rules LRPT with the number of generations = 10 resulted in the smallest value of fitness function, which is 6906506.463 while dispatching rules SRPT resulted in the highest value of fitness function in generation number = 5.

Conclusively, higher generations number resulted in better results, however, there are no great differences in a better result for any dispatching rules to be selected.

4.2.2. Generation Size

The experiment was then run with a fixed number of generations which is 5 with different generation size which is 6, 10 and 25 respectively. Based on Figure 11, there is an obvious difference between each generation size, while the dispatching rules MOPR indicated the lowest fitness value with generation size = 25.

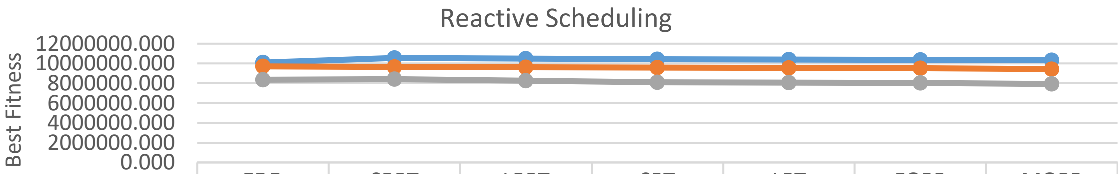


	EDD	SRPT	LRPT	SPT	LPT	FOPR	MOPR
GenNum5	10094008.7	10558704.1	10498487.4	10421616.5	10404727.7	10351359.4	10325614.8
GenNum10	7155271.88	7150250.21	6906506.46	7074937.71	6970238.43	6978293.68	6921106.84

Dispatching rules

GenNum5 GenNum10

Figure 10: The best fitness value in terms of dispatching rules of the reactive scheduling for the number of generations = 5 and 10 with generation size = 6.



	EDD	SRPT	LRPT	SPT	LPT	FOPR	MOPR
GenSize6	10094008.7	10558704.1	10498487.4	10421616.5	10404727.7	10351359.4	10325614.8
GenSize10	9709063.73	9645327.17	9614383.60	9584172.03	9546933.87	9507325.78	9438284.90
GenSize25	8354940.49	8417315.24	8276956.26	8095122.99	8054011.31	8035525.06	7927747.53

Dispatching rules

GenSize6 GenSize10 GenSize25

Figure 11: The best fitness value in terms of dispatching rules of the reactive scheduling for generation sizes = 6, 10 and 25 with the number of generations = 5.

For the experiments in terms of dispatching rules regarding different generation sizes and numbers, the results indicated that the dispatching rules to be selected are mainly depending on the objective functions. For the objective function makespan, there are no big differences.

4.3. Improvement Rate

There are durations when the experiments are running, and those readings are classified as optimization time, which indicated how long a specific experiment needs to run until the result was obtained. For stochastic simulation, the individuals should be evaluated by several simulation runs. However, due to limited time issues, the observation per individual is set to only 1. In Tecnomatix Plant Simulation, when the GA Wizard executes, it will calculate the number of simulations to be run based on the formula below and run the simulation:

$$\textbf{Number of simulation runs} = \mathbf{O_i} \times (\mathbf{GS} + \mathbf{2} \times \mathbf{GS} \times (\mathbf{GN} - \mathbf{1})) \quad (5)$$

where O_i = Observations per individual, GS = Generation Size, GN = Number of Generations.

Hence, the larger the number of generations or generation size, the larger the number of simulations runs, therefore resulting in longer optimization time.

4.3.1. Number of Generations

To analyse the relationship of improvement rate with the generation size, a formula is used to calculate the rate from the data generated from the experiment as follows:

$$\boldsymbol{Rate} = \frac{\boldsymbol{Best\ fitness\ value}}{\boldsymbol{Optimization\ time}} \quad (6)$$

The experiment was run with the fixed number of generations = 5 and varied generation sizes = 6, 10 and 25. Table 9 shows the results from the calculation of the formula (6) for the reactive scheduling conditions with different dispatching rules.

Table 9. The improvement rate for the reactive scheduling of dispatching rules with the number of generations = 5 and generation sizes = 6,10 and 25.

	Gen Size 6	Gen Size 10	Gen Size 25
	Reactive Scheduling	Reactive Scheduling	Reactive Scheduling
EDD	17830.482	10755.993	3444.372
SRPT	19668.227	10477.326	3518.048
LRPT	19115.259	9735.380	3378.730
SPT	20799.089	10367.003	3344.148
LPT	17140.189	10125.550	3306.326
FOPR	18432.581	10539.054	3378.890
MOPR	17802.480	14580.917	3488.352

The data from Table 9 were generated into Figure 12 for analysis. From Figure 12, firstly the improvement rate for generation size = 6 is the highest for the situation, and when the generation sizes increase, then the improvement rate decrease linearly for the reactive scheduling case. Results show that the performances are getting better when the generation size is larger.

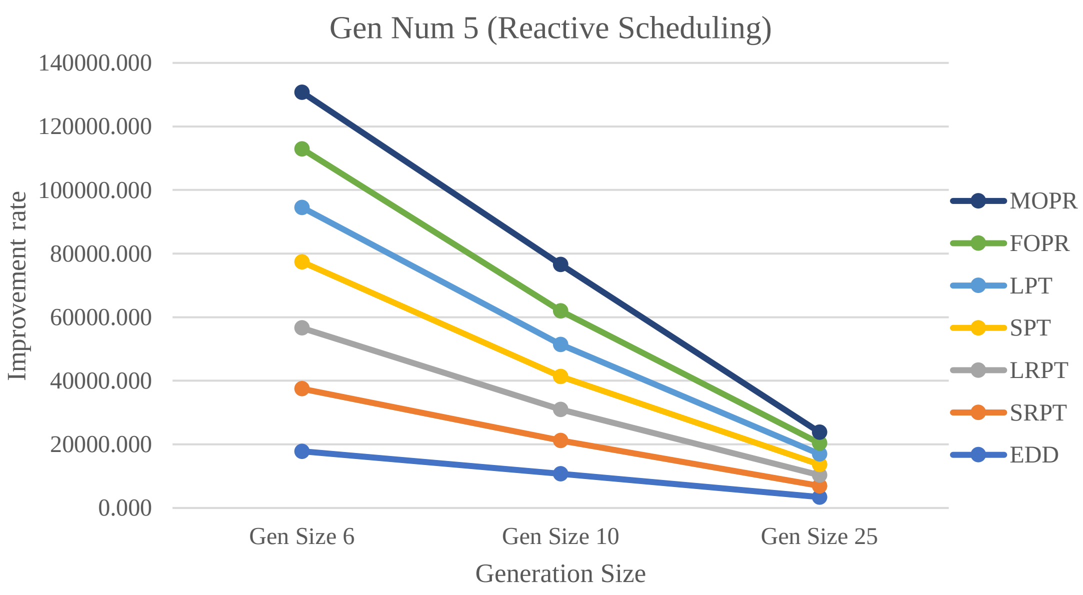


Figure 12. The improvement rate for reactive scheduling with the generation number = 5 and varied generation sizes.

5. Conclusion

This study is to analyze several aspects of combining simulation and optimization-based algorithms for job-shop scheduling of reconfigurable manufacturing systems with a predictive-reactive approach using priority dispatching rules and GA. A predicted feasible schedule will be first determined and tested from a developed model where the reconfigurable production system in real-life is taken as a reference and case study. The simulations were run with GA together with dispatching rules, together with different conditions and settings of the reconfigurable manufacturing system to identify the results in a different environment.

The parameters such as the number of generations and generation size have also been analyzed to identify the effect towards the results, however, due to limited time, only some parameters were tested based on the previous study in the literature. The result showed that the model built had demonstrated good efficiency and the ability to find an effective schedule in a specified period and the algorithm can tackle the complicated scheduling issue successfully and in lesser time.

The results obtained from this simulation run with GA using Tecnomatix Plant Simulation included the makespan, the best fitness value, the optimization running time, the best parameter of the allocation of the products, the best sequences of the products, the evolution and performance of the fitness value during the generations, the details included the children and parents’ genetics data and so on. However, the performance and schedule optimization degree cannot be compared to other scheduling methods, because this model only provided the optimization possibility using GA and dispatching rules. The series of optimization runs do not provide an adequate collection of data to conclude the recommendations for the best selection of GA options and dispatching rules of the reconfigurable manufacturing system. Overall, the model proved the efficacy of integrating simulation and optimization with a genetic algorithm, providing engineers with the needed flexibility and control.

Conclusively, although various methods and algorithms have been created in the literature, only a few comparisons have been made. Other research suggests that the proposed approaches perform well under certain assumptions, but not so well or even poorly under others. Part of future works to be carried out will involve benchmark problems of a certain type of production system operation must be established with specific test objective functions. Reasonable comparisons between different approaches may also be made to validate the effectiveness of the suggested solutions more precisely.

Acknowledgement

The authors would like to thank Universiti Teknikal Malaysia Melaka (UTeM) and UTeM Zamalah Scholarship Scheme 2.0, for the support and opportunity to conduct this research study.

References

[1] L. Asadzadeh, “A local search genetic algorithm for the job shop scheduling problem with intelligent agents”. Computers and Industrial Engineering, Vol. 85, 2015, 376–383.

[2] M. Niehues, F. Buschle, G. Reinhart, “Adaptive job-shop control based on permanent order sequencing”. Procedia CIRP, Vol. 33, 2015, 127–132.

[3] A. Allahverdi, E. Pesch, M. Pinedo, F. Werner, “Scheduling in manufacturing systems: new trends and perspectives”. International Journal of Production Research, Vol. 56, No. 19, 2018, 6333–6335.

[4] B. Scholz-Reiter, D. Lappe, S. Grundstein, “Capacity adjustment based on reconfigurable machine tools – Harmonising throughput time in job-shop manufacturing”. CIRP Annals, Vol. 64, No. 1, 2015, 403–406.

[5] R. Angkiriwang, Pujawan, I. Nyoman, B. Santosa, “Managing uncertainty through supply chain flexibility: reactive vs. proactive approaches. Production & Manufacturing Research”. Vol. 2, No. 1, 2014, 50–70.

[6] Y. Koren, X. Gu, W. Guo, “Reconfigurable Manufacturing Systems: Principles, design, and future trends,” Frontiers of Mechanical Engineering, Vol. 13, No. 2, 2017, 121–136.

[7] R. Pansare, G. Yadav, M. R. Nagare, “Reconfigurable Manufacturing System: A systematic review, meta-analysis and future research directions,”. Journal of Engineering, Design and Technology, Vol. 33, 2021, 543-574.

[8] P.L. Mareddy, S.R. Narapureddy, V.R. Dwivedula, S.V. Prayagi, "Simultaneous Scheduling of Machines, Tool Transporter and Tools in a Multi-Machine Flexible Manufacturing System Without Tool Delay Using Crow Search Algorithm". Jordan Journal of Mechanical and Industrial Engineering, Vol. 16, No. 3, 2022, 403-419.

[9] D.Y. Lee, F. DiCesare, “Integrated Scheduling of Flexible Manufacturing Systems Employing Automated Guided Vehicles”. IEEE Transactions on Industrial Electronics, Vol. 41, No. 6, 1994, 602–610.

[10] N. Kundakci, and O. Kulak, “Hybrid genetic algorithms for minimizing makespan in dynamic job shop scheduling problem”. Computers and Industrial Engineering, Vol. 96, 2016, 31–51.

[11] T. Nehzati, “Research Outline on Reconfigurable Manufacturing System Production Scheduling Employing Fuzzy Logic”. International Journal of Information and Electronics Engineering, Vol. 2, No. 5, 2012, 812–816.

[12] Y.C. Choi, P. Xirouchakis, “A holistic production planning approach in a reconfigurable manufacturing system with energy consumption and environmental effects”. International Journal of Computer Integrated Manufacturing, Vol. 28, No. 4, 2015, 379–394.

[13] X.-Q. Wan, H.-S. Yan, “Integrated scheduling and self-reconfiguration for assembly job shop in knowledgeable manufacturing”. International Journal of Production Research, Vol. 53, No. 6, 2015, 1746–1760.

[14] M. Niehues, P. Sellmaier, T. Steinhaeusser, G. Reinhart, “Adaptive Job-Shop Control Using Resource Accounts”. Procedia CIRP, Vol. 57, 2016, 351–356.

[15] M.M. Nasiri, R. Yazdanparast., F. Jolai, “A simulation optimisation approach for real-time scheduling in an open shop environment using a composite dispatching rule”. International Journal of Computer Integrated Manufacturing, Vol. 30, No. 12, 2017, 1239–1252.

[16] L. Tang, X. Wang, “A predictive reactive scheduling method for color-coating production in steel industry”. International Journal of Advanced Manufacturing Technology, Vol. 35, No. 7–8, 2008, 633–645.

[17] P. Korytkowski, T. Wiśniewski, S. Rymaszewski, “An evolutionary simulation-based optimization approach for

dispatching scheduling." Simulation Modelling Practice and Theory, Vol. 35, 2013, 69–85.

[18] M. Fera, F. Fruggiero, A. Lambiase, G. Martino, M. Elena, "Production Scheduling Approaches for Operations Management. In: Massimiliano Schiraldi, editor. In Operations Management, United Kingdom: InTech; 2013, p. 113-139.

[19] H.H. Doh, J.M. Yu, Y.J. Kwon, D.H. Lee, M.S. Suh, "Priority scheduling for a flexible job shop with a reconfigurable manufacturing cell". Industrial Engineering and Management Systems, Vol. 15, No. 1, 2016, 11–18.

[20] M. Leusin, E. Frazzon, M.U. Maldonado, M. Kück, M. Freitag, "Solving the Job-Shop Scheduling Problem in the Industry 4.0 Era". Technologies, Vol. 6, No. 4, 2018, 107.

[21] M. Niehues, M. Blum, U. Teschemacher, G. Reinhart, "Adaptive job shop control based on permanent order sequencing: Balancing between knowledge-based control and complete rescheduling". Production Engineering, Vol. 12, No. 1, 2018, 65–71.

[22] O. Bataineh, R.A. Aomar, A.A. Shakra, "Simulation-Based Optimization for Performance Enhancement of Public Departments". Jordan Journal of Mechanical and Industrial Engineering, Vol. 4, No. 3, 2010, 346-351.

[23] A. Azab, A. Ziout, W. Elmaraghy, "Modeling and Optimization for Disassembly Planning". Jordan Journal of Mechanical and Industrial Engineering, Vol. 5, No. 1, 2011, 1-8.

[24] D. Mourtzis, "Simulation in the design and operation of manufacturing systems: state of the art and new trends". International Journal of Production Research, Vol. 58, No. 7, 2020, 1927–1949.

[25] J. Siderska, "Application of Tecnomatix Plant Simulation for Modeling Production and Logistics Processes". Business, Management and Education, Vol. 14, No. 1, 2016, 64–73.

[26] J. Yann, J. Anicia, Z.M. Fatima, T. Damien, B. Patrick, "A new methodological support for control and optimization of manufacturing systems in the context of product customization". Journal of Industrial and Production Engineering, Vol. 38. No. 5, 2021, 341-355.

[27] H. Xiong, H. Fan, G. Jiang, G. Li, "A simulation-based study of dispatching rules in a dynamic job shop scheduling problem with batch release and extended technical precedence constraints". European Journal of Operational Research, Vol. 257, No. 1, 2017, 13–24.

[28] J.T. Lin, C.C. Chiu, Y.H. Chang, Simulation-based optimization approach for simultaneous scheduling of vehicles and machines with processing time uncertainty in FMS". Flexible Services and Manufacturing Journal, Vol. 31, No. 1, 2019, 104–141.

[29] Q. Deng, G. Gong, X. Gong, L. Zhang, W. Liu, Q. Ren, "A Bee Evolutionary Guiding Nondominated Sorting Genetic Algorithm II for Multiobjective Flexible Job-Shop Scheduling". Computational Intelligence and Neuroscience, Vol. 2017, 2017, 1-20.

[30] H. Piroozfard, K.Y. Wong, W.P. Wong, "Minimizing total carbon footprint and total late work criterion in flexible job shop scheduling by using an improved multi-objective genetic algorithm" Resources, Conservation and Recycling, Vol. 128, 2018, 267–283.

[31] X. Zan, Z. Wu, C. Guo, Z. Yu, "A Pareto-based genetic algorithm for multi-objective scheduling of automated manufacturing systems". Advances in Mechanical Engineering, Vol. 12, No. 1, 2020, 1–15.

[32] T. Sivarat, S. Apichat, "A simulation-optimization approach for adaptive manufacturing capacity planning in small and medium-sized enterprises". Expert Systems with Applications, Vol. 168, No. 114451, 2021, 1-13.

[33] P. Valledor, A. Gomez, P. Priore, J. Puente, "Solving multi-objective rescheduling problems in dynamic permutation flow shop environments with disruptions". International Journal of Production Research, Vol. 56, No. 19, 2018, 6363–6377.

[34] F.A. Touzout, L. Benyoucef, "Multi-objective multi-unit process plan generation in a reconfigurable manufacturing environment: a comparative study of three hybrid metaheuristics". International Journal of Production Research, Vol. 57, No. 24, 2019, 7520–7535.

[35] F. Zheng, K. Jin, "An improved heuristic for single machine lot scheduling problem". IFAC-PapersOnLine, Vol. 52, No. 13, 2019, 217–222.

[36] G. Amir, A. Amir, H. Cathal, "Evolutionary Learning Based Simulation Optimization for Stochastic Job Shop Scheduling Problems". Applied Soft Computing Journal, Vol. 106, 2021, 1-19.

[37] E. Gheisariha, M. Tavana, F. Jolai, M. Rabiee, "A simulation–optimization model for solving flexible flow shop scheduling problems with rework and transportation". Mathematics and Computers in Simulation, Vol. 180, 2021, 152–177.

[38] M. Kuck, B. Eike, M. Freitag, "Towards Adaptive Simulation - Based Optimization To Select Individual. Winter simulation conference, Las Vegas, Nevada, 2017.

[39] Y.C. Wang, T.C.T. Chen, L.C. Wang, "Simulating a Semiconductor Packaging Facility: Sustainable Strategies and Short-time Evidences". Procedia Manufacturing, Vol. 11, 2017, 787–795.

[40] C. Grabowik, K. Kalinowski, G. Cwikla, K. Niemiec, I. Paprocka, "A computer simulation as a tool for a production system analysis and optimization". IOP Conference Series: Materials Science and Engineering, Vol. 400, No. 2, 2018, 1-15.

[41] T. Sobottka, F. Kamhuber, M. Rössler, W. Sihn, "Hybrid simulation-based optimization of discrete parts manufacturing to increase energy efficiency and productivity". Procedia Manufacturing, Vol. 21, 2018, 413–420.

[42] H. Wang, Z. Jiang, Wang, Yan, H. Zhang, Wang, Yanhong, "A two-stage optimization method for energy-saving flexible job-shop scheduling based on energy dynamic characterization". Journal of Cleaner Production, Vol. 188, 2018, 575–588.

[43] B. Wally, J. Vyskocil, P. Novak, C. Huemer, R. Sindelar, P. Kadera. "Flexible Production Systems: Automated Generation of Operations Plans Based on ISA-95 and PDDL". IEEE Robotics And Automation Letters, Vol. 4, No. 4, 2019, 4062-4069.

[44] Yang, B., and Geunes, J., 2008. Predictive-reactive scheduling on a single resource with uncertain future jobs. European Journal of Operational Research, Vol. 189, No. 3, 1267–1283.

[45] E. S. Nicoară, F. G. Filip, N. Paraschiv, "Simulation-based optimization using genetic algorithms for multi-objective flexible JSSP". Studies in Informatics and Control, Vol. 20, No. 4, 2011, 333–344.

[46] MJ, Zeestraten, "Scheduling Flexible Manufacturing Systems. PhD Thesis, Technische Universiteit Delft (The Netherlands)". 1991.

[47] P. Ning, "An Adaptive Scheduling Method for Resources in Used Automobile Parts Recycling". Jordan Journal of Mechanical and Industrial Engineering, Vol. 14, No. 1, 2020, 53-60.

[48] K. Mesghouni, "Evolutionary algorithms for job-shop scheduling" Vol. 14, No. 1, 2004, 91–103.

[49] D. Kofjač, M. Kljajić, "Application of genetic algorithms and visual simulation in a real-case production optimization". WSEAS Transactions on Systems and Control, Vol. 3, No. 12, 2008, 992–1001.

[50] M.S. Gibbs, H.R. Maier, G.C. Dandy, J.B. Nixon, "Minimum number of generations required for convergence of genetic algorithms". 2006 IEEE Congress on Evolutionary Computation, CEC, Vancouver, Canada, 2006.

[51] P. Bajpai, M. Kumar, "Genetic algorithm–an approach to solve global optimization problems". Indian Journal of computer science and engineering, Vol. 1, No. 3, 2010, 199–206.