International Journal of Production Research

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/tprs20

A simulation-based optimisation framework for process plan generation in reconfigurable manufacturing systems (RMSs) in an uncertain environment

Amirhossein Kazemisaboor , Abdollah Aghaie & Hamed Salmanzadeh

INTERNATIONAL JOURNAL OF PRODUCTION RESEARCH
https://doi.org/10.1080/00207543.2021.1883762

A simulation-based optimisation framework for process plan generation in reconfigurable manufacturing systems (RMSs) in an uncertain environment

Amirhossein Kazemisaboor, Abdollah Aghaie and Hamed Salmanzadeh

Industrial Engineering Department, K. N. Toosi University of Technology, Tehran, Iran

ABSTRACT
Reconfigurable manufacturing system (RMS) is a manufacturing paradigm which is proven to be time and cost-effectively adaptable to a wide range of market changes due to its customisable capacity and functionality. In this paper, a two-step framework is proposed for the process plan generation in RMS. The first step, aimed at solving the multi-objective part-family Single-Unit Process Plan (SUPP) generation problem, involves a comparative approach using three metaheuristics, namely: The Non-Dominated Sorting Genetic Algorithm (NSGA-II), the Archived Multi-Objective Simulated Annealing (AMOSA) and the Multi-Objective Particle Swarm Optimisation (MOPSO) as well as a simulation-based optimisation method. The second step is designed to solve the multi-objective part-family Multi-Unit Process Plan (MUPP) generation problem with unpredictable demands in different periods using a combination of the answers of the algorithms in step 1. The number of units is also optimised using the NSGA-II. Finally, a novel heuristic algorithm named Designed Periods Algorithm (DPA) is proposed in the second step to meet the unpredictable demands in different periods. To illustrate the applicability of the framework, an example is presented, the results of which have shown the superiority of the MUPP over the SUPP in response to unpredictable demands according to the periods designed by DPA.

ARTICLE HISTORY
Received 11 July 2020
Accepted 28 December 2020

KEYWORDS
Reconfigurable manufacturing system (RMS); simulation-based optimisation; process plan generation; uncertain environment; metaheuristics

1. Introduction

Reconfigurable manufacturing system (RMS), first proposed by Koren et al. (1999), is designed to produce parts of a family which, unlike traditional manufacturing paradigms such as Dedicated Manufacturing Lines (DMLs) and Flexible Manufacturing Systems (FMSs), does not require a significant amount of time and cost or a fundamental change in the layout. RMSs are designed using hardware and software modules that can be integrated quickly and reliably in a way that without them, the reconfiguration process will be costly, time-consuming and impractical (Landers, Min, and Koren 2001; Gadalla and Xue 2018). Achieving this design goal requires an RMS, with several key characteristics (Koren 2006) that can effectively generate process plans, especially in uncertain environments. These essential characteristics include modularity, integrability, customisation, scalability, convertibility and diagnosability, all of which are applied to the design of production systems as well as machines along with their controllers, human resources and the entire company (Koren 2006). This manufacturing paradigm together with these characteristics can generate flexible process plans to meet unpredictable demands that may be ordered any time under any deadline to reduce the main costs and time of the system (Dou, Li, and Su 2016; Najid, Castagna, and Kouiss 2020).

Process plan generation in RMS is an $N \times M$ programming problem (N tasks and M machines) which is considered NP-hard (Dou et al. 2020; Khezri, Benderbal, and Benyoucef 2020a). Some scheduling problems, multi-objective ones with different levels of existing parameters in particular, cannot be solved by exact solution methods such as mixed-integer programming and Branch and Bound methods (Bensmaine, Dahane, and Benyoucef 2013a). The advent of heuristic methods, such as genetic algorithms and other evolutionary algorithms in recent years, has made it possible to tackle problems of this kind. To build upon this point a little more, by means of the simulation-based optimisation method, the details of constraints and variables can be easily implemented with one or more simulation modules (Bensmaine, Dahane, and Benyoucef 2013a).

In recent years, RMS has become one of the most attractive research topics (Yelles-Chaouche et al. 2020; Najid, Castagna, and Kouiss 2020). Nevertheless, very few works have addressed the multi-objective multi-unit

CONTACT Abdollah Aghaie aaghaie@kntu.ac.ir

part-family process plan generation problem in RMS. Due to the lack of studies on this critical and challenging problem, the multi-objective multi-unit part-family process plan generation problem has been considered in this paper. In addition to minimising the total manufacturing cost and time, a new heuristic algorithm is employed as a novel optimisation criterion. The heuristic algorithm, called Designed Periods Algorithm (DPA), is proposed to respond to unpredictable demands with deadlines in periods designed through a multi-unit process plan. A two-step framework is developed to solve the problem where over the first step, called Single-Unit Process Plan (SUPP), the multi-objective part-family single-unit process plan generation problem is utilised. Afterwards, three metaheuristic algorithms including The Non-Dominated Sorting Genetic Algorithm (NSGA-II), the Archived Multi-Objective Simulated Annealing (AMOSA) and the Multi-Objective Particle Swarm Optimisation (MOPSO) are employed. A simulation-based optimisation method is also adopted. On the other hand, step two, Multi-Unit Process Plan (MUPP), focuses on solving the multi-objective part-family multi-unit process plan generation problem with unpredictable demands in different periods by combining the answers of algorithms in the first step. Moreover, the optimisation of the number of units is performed using the NSGA-II.

The remaining parts of the paper are organised as follows: Section 2 summarises works related to process plan generation for an RMS. Section 3 describes the simulation-based optimisation framework and the method of responding to the obtained demands based on DPA and according to the SUPPs and MUPPs. Section 4 presents the proposed multi-objective formulation of the simulation module. Section âĂŐ5 implements and evaluates the framework using a numerical example and analyses the obtained results. Ultimately, Section âĂŐ6 concludes the paper using some future research work suggestions.

2. Literature review

In the literature related to RMSs, the main studies have focused on different aspects such as layout design, process plan generation and load balancing. Process plan generation is defined to determine economical methods by which a product can be manufactured (cost aspect) and, at the same time, remains competitive (time aspect) (Touzout and Benyoucef 2019). The existing literature in this field is reviewed in the following.

Oke, Abou-El-Hossein, and Theron (2011) presented a combination of different techniques to generate process plans for molding machines, in which the approach was based on the weight of priority factors applied to the machining process for forming the part. They considered three factors for determining the order of operations including technology factor, geometric factor and economic factor. Some other authors investigated the process plan generation for a product with various features in SUPP by three objective functions of total time and cost and amount of generated greenhouse gasses through the mixed integer programming method (Khezri, Benderbal, and Benyoucef 2020b; Touzout and Benyoucef 2019). In these three studies, the use of exact solution methods has prevented the authors from implementing details for configurations, tools and Tool Approach Directions (TADs), which is while the use of simulation could provide this platform. The authors could also use MUPP to reduce the extra time and cost of switching from the current status to the initial status of the configurations, tools and TADs. Failure to consider the uncertainty of the real world has led them to deviate from the expected realistic outcome.

Musharavati et al. (2008) adjusted the AMOSA to investigate the process plan generation problem in the multipart RMSs. The main goal was to achieve an economical process plan approximately close to the optimum point. Chaube, Benyoucef, and Tiwari (2012) also adapted a simulation-based NSGA-II for process plan generation in the RMS. The main problem considered in their work was to present process plans for different parts on a set of machines concerning the part type, type of operation, machines, machine configurations and tools, and two objectives were considered: the completion time and the total cost. Furthermore, Musharavati and Hamouda (2012) employed the AMOSA for process plan generation in the RMS to improve the manufacturing process. The authors successfully achieved an enhancement in the performance of the metaheuristic algorithm in terms of the obtained results and computational effort. Moreover, Bensmaine, Dahane, and Benyoucef (2014) investigated the process plan generation for a product with various features using three objective functions: total time, cost and the amount of generated greenhouse gasses, where the problem was dealt with through the NSGA-II and the AMOSA. In these four studies, the authors have used simulation-based optimisation and have simply considered the details of the constraints and variables and solved the problems. However, not considering the real-world uncertainty has made the estimated result of time and cost of the problem not comparable to the result of real systems. They could also use MUPP to reduce cost and time due to the elimination of the time and cost of changing the current status to the original status of the configurations, tools and TADs.

Takahashi et al. (2006) presented an uncertain model by simultaneous consideration of the order planning and intended system configuration which provided the maximum profit. The main idea was that configuration of some systems could be devoted to a group of products depending on the profit value (it is not appropriate to use slow and cheap machines when demand is high and vice versa). The authors proposed an algorithm to determine the decision-making threshold based on the presented uncertain model which was then applied to choose between two configurations (fast configuration or slow configuration). Although this study has considered uncertainty and has provided closer and more realistic results, the authors have only used SUPP. This is while they could use MUPP to eliminate the extra time and cost of changing the current status at any production time to the initial status. Bensmaine, Dahane, and Benyoucef (2013a) presented a process plan generation procedure for a product with various features in SUPP by two objective functions of total time and cost. Their proposed approach copes with market uncertainty and demand fluctuations in order to satisfy demands within their deadlines in multiple periods through the Genetic Algorithm. The main advantage of this research work over that of Takahashi et al. (2006) is that it has used simulation, as a result of which not only has it implemented more and better-modeled details, but also it has obtained results in a broader range of time and cost.

Bensmaine, Dahane, and Benyoucef (2011) studied a hybrid simulation-based approach of process plan generation for a part family within Multi-Unit Process Plan (MUPP) by two objective functions of total time and cost through the NSGA-II. Touzout and Benyoucef (2018) also studied a simulation-based approach for process plan generation within Multi-Unit Process Plan (MUPP) by three objective functions: total time, total cost and the amount of generated greenhouse gasses. The problem is then solved through three hybrid-metaheuristic algorithms. The authors in the two previous studies have preferred simulation to the use of exact solution methods. This has been due to the fact that the use of simulation could provide more intricate details that could not be otherwise considered by the exact solution methods, and that the constraints can be applied in more detail in this way. Although these studies have benefited from the simultaneous use of simulation and MUPP, not considering real-world uncertainty has made their results far from being realistic.

Table 1 illustrates a summary of primary references in the literature on process plan generation based on these criteria: solution method, process plan units and environmental conditions.

In some reviewed papers, either simulation-based optimisation with MUPP or simulation-based optimisation with uncertainty has been discussed along with some of their associated benefits. However, none of them have taken into account the advantage of simulation-based optimisation considering unpredictable demands and MUPP, which utilises the three mentioned optimisation approaches simultaneously. Therefore, there is a research gap in the simultaneous use of these three factors. Hence, the primary goal of this paper is to suggest a simulation-based optimisation framework for process planning and responding to unpredictable demands in periods designed by DPA with deadlines by means of MUPP. The problem under consideration in the present work is detailed in the upcoming section.

3. Methodology

The simulation-based optimisation framework is first explained in section 3.1, which includes metaheuristic algorithms used in the optimisation module and the two steps of generating SUPPs and MUPPs (with the associated flowcharts). Section 3.2 describes the employed heuristic algorithm (DPA) and the related pseudocode, through which the obtained demands can be divided into designed periods and responded by SUPPs and MUPPs. Section 3.3 describes how the best parameters of the metaheuristic algorithms explained in section 3.1 can be determined using the Taguchi Method (TM). Finally,

Table 1. A summary of the examined literature.

	Solution Method		Process Plan		Environment	
Reference	Simulation	Others	Single-Unit	Multi-Unit	Certain	Uncertain
Takahashi et al. (2006)		✓	✓			✓
Musharavati et al. (2008)	✓		✓		✓	
Oke, Abou-El-Hossein, and Theron (2011)		✓	✓		✓	
Bensmaine, Dahane, and Benyoucef (2011)	✓			✓	✓	
Chaube, Benyoucef, and Tiwari (2012)	✓		✓		✓	
Musharavati and Hamouda (2012)	✓		✓		✓	
Bensmaine, Dahane, and Benyoucef (2013a)	✓		✓			✓
Bensmaine, Dahane, and Benyoucef (2014)	✓		✓		✓	
Touzout and Benyoucef (2018)	✓			✓	✓	
Touzout and Benyoucef (2019)		✓	✓		✓	
Khezri et al. (2020b)		✓	✓		✓	

Section 3.4 defines performance metrics to evaluate the answers of the metaheuristic algorithms.

3.1. Simulation-based optimisation framework

In the present study, a simulation model is created for calculating the total time and cost in RMS which is implemented as a computer code using MATLAB programming software (2016). The simulation module is performed based on discrete-event time variation and single unit timing (Bensmaine, Dahane, and Benyoucef 2013a). All functions implemented in this code are thoroughly validated and examined in terms of the efficiency and accuracy of their performances.

The primary approach in this paper is to deploy a simulation-based optimisation in which, in contrast to exact solution methods, the objective function is replaced by one or multiple simulation modules. The main advantage brought by simulation is that more details can be implemented than the exact solution methods ever could (Bensmaine, Dahane, and Benyoucef 2013a). Apart from that, the constraints can be applied in more detail, and even the ones that cannot be taken into account by the exact solution methods can also be explored. Simulation allows users to determine how a system responds to different inputs, which, in turn, determines how well the system operates (Bensmaine, Dahane, and Benyoucef 2013a). The optimisation algorithm utilises the simulation module results via an iterative approach to provide the feedback used for conducting the optimum solution search method. To clarify the calculation method in the simulation module, the operation assignment is performed in the four following steps for machines in this module:

- Identifying the needs of operations (precedence matrix, TADs and required tools)
- Identifying the candidate machines
- Assignment of operations to machines
- Arranging the sequence of operations

The aim of the optimisation module is to use the most appropriate metaheuristic algorithms. Onar et al. (2016) showed in their research that the most commonly used metaheuristic algorithms in the field of manufacturing systems are Genetic Algorithm (GA), Simulated Annealing (SA) and Particle Swarm Optimisation (PSO), respectively. Accordingly, one of the main benefits of the present research is that these three algorithms have been compared in terms of the SUPP (In most studies, GA and SA have been compared, but PSO has rarely been examined). Afterwards, the results of SUPPs obtained from these three multi-objective metaheuristic algorithms have been utilised to generate MUPPs. In this procedure, the initial solution is provided by the optimisation module and sent to the simulation module which runs the computer program and returns the feedback to the optimisation module. This process is repeated until the convergence condition is met.

To clarify the methodology by means of which the calculations are performed in the simulation module with the help of optimisation by the metaheuristic algorithms, a two-step framework is considered as follows:

In step 1 (SUPP), the optimum solutions of the single-unit process plan are obtained considering the objective functions and constraints based on the simulation-based optimisation approach. Then, the quality of solutions obtained by these algorithms is compared with one another. Figure 1 shows both the performance of the optimisation and simulation module and the interaction between them to generate SUPPs.

In step 2 (MUPP), all or some of the obtained solutions of step 1 (SUPP) are put together to form comprehensive process plans. In this procedure, the resources are shared between different products that existed in the manufacturing system. Moreover, machines with reconfiguration capability can perform various operations. As a result, it is possible to create waiting time for different parts in order to be operated by different machines. A method is also applied as follows: first, it is assumed that n number of parts are demanded from a part family and p optimum Pareto solutions have been obtained for process plans from the SUPP step. In this approach, the problem is to determine the best plan with m units of the p process plans, as the comprehensive (m-unit) process plan considers both objectives of the completion time and the total cost. This step is included in the process since the combination of different process plans minimises the risk of bringing about fundamental changes in the existing configurations, TADs and tools status for the next process, which ultimately decreases completion time and total cost values. In this step, metaheuristic algorithms of the optimisation module search the optimum sequence of process plans with m units of the p process plans as the comprehensive (m-unit) process plan. This sequence is then sent to the simulation module to be evaluated it in terms of completion time and total cost. The obtained results are returned to the optimisation module, and the cycle continues until the stop criterion is satisfied. For this optimisation module, NSGA-II has been selected by encoding the integer number solutions. Each gene in the chromosome shows a unit of comprehensive (m-unit) process plan and all units form the intended chromosome. In each gene, each integer number indicates the number of the process plan from p process plans.

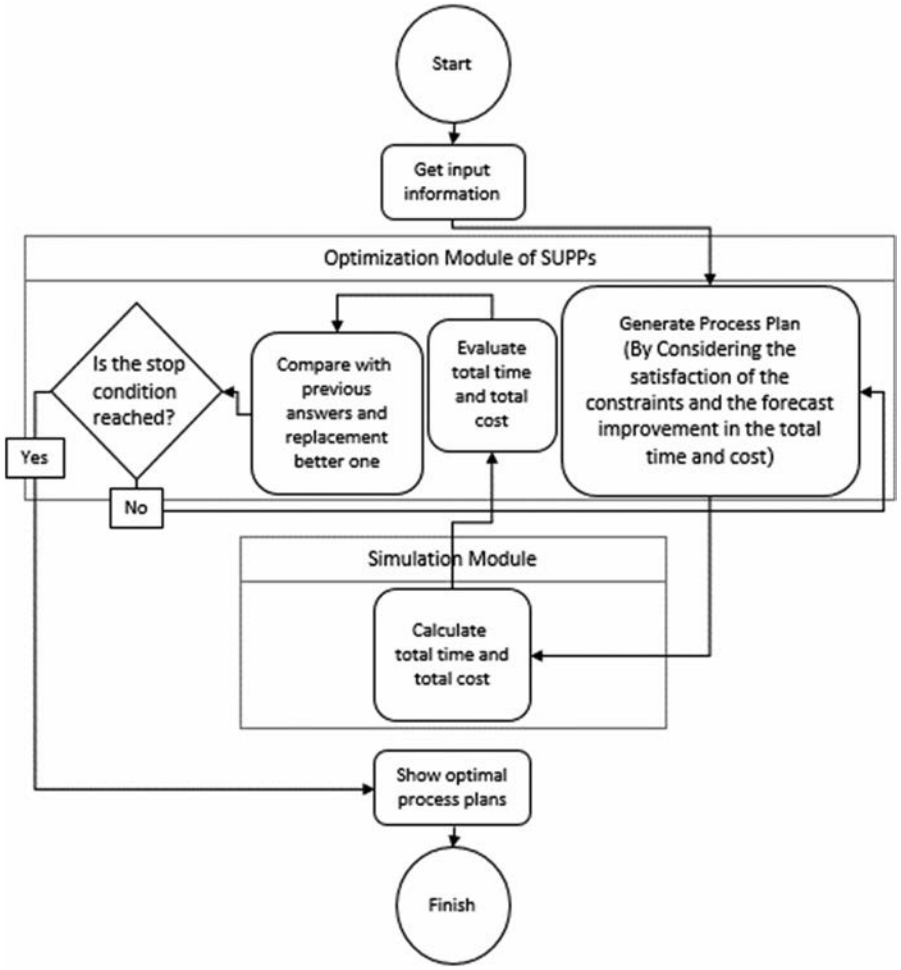

Figure 1. The performance of the optimization and simulation module and feedback to generate SUPPs (Step 1).

The main point of the simulation module in this step is to extend the process plan to the intended number of units. The process plan is run once with m units, where m is the number of the optimal units obtained. In this simulation, the status of machines, tools and configurations are adjusted based on the first comprehensive (m-unit) process plan where parts could use free machines. For the next stage, the configuration of the machines is considered to have the same arrangement and condition as the previous step. Taking these issues into consideration, the cost and time of all manufacturing units are categorised into two stages, the first stage of which calculates the cost and time in which the configurations, tools and TADs are adjusted based on the first comprehensive (m-unit) process plan; the second stage calculates the cost and time in which the configurations, tools and TADs are adjusted based on the previous comprehensive (m-unit) process plan. To obtain results after the two initial stages, time and cost are considered with weight factors of 0.05 and 0.95, respectively. Figure 2 shows the performance of the optimisation and simulation module and the interaction between them to generate MUPPs.

3.2. Designed periods algorithm (DPA)

To make the results closer to the real world, unpredictable demands are considered with unpredictable deadlines, along with presenting a new procedure called Designed Periods Algorithm (DPA) to deal with them. The applied strategy for responding to these demands is that the created demands are received over different periods, which are then divided into the designed periods. The division is done in a way that the beginning of the first period is at time zero (which is the beginning of the whole division, the same time the first demand is received). The end time of each period is equal to the time the new event occurs. Moreover, the start time of each designed period is equal to the end of the previously designed period. This is how designed periods are formed. Then, the periods in which multiple demands have overlapped will be taken into consideration, where the manufacturing rate must be higher than those of the periods with lower demands. Accordingly, the periods are divided into different groups considering the difference in the number of their demands. Afterwards, the appropriate

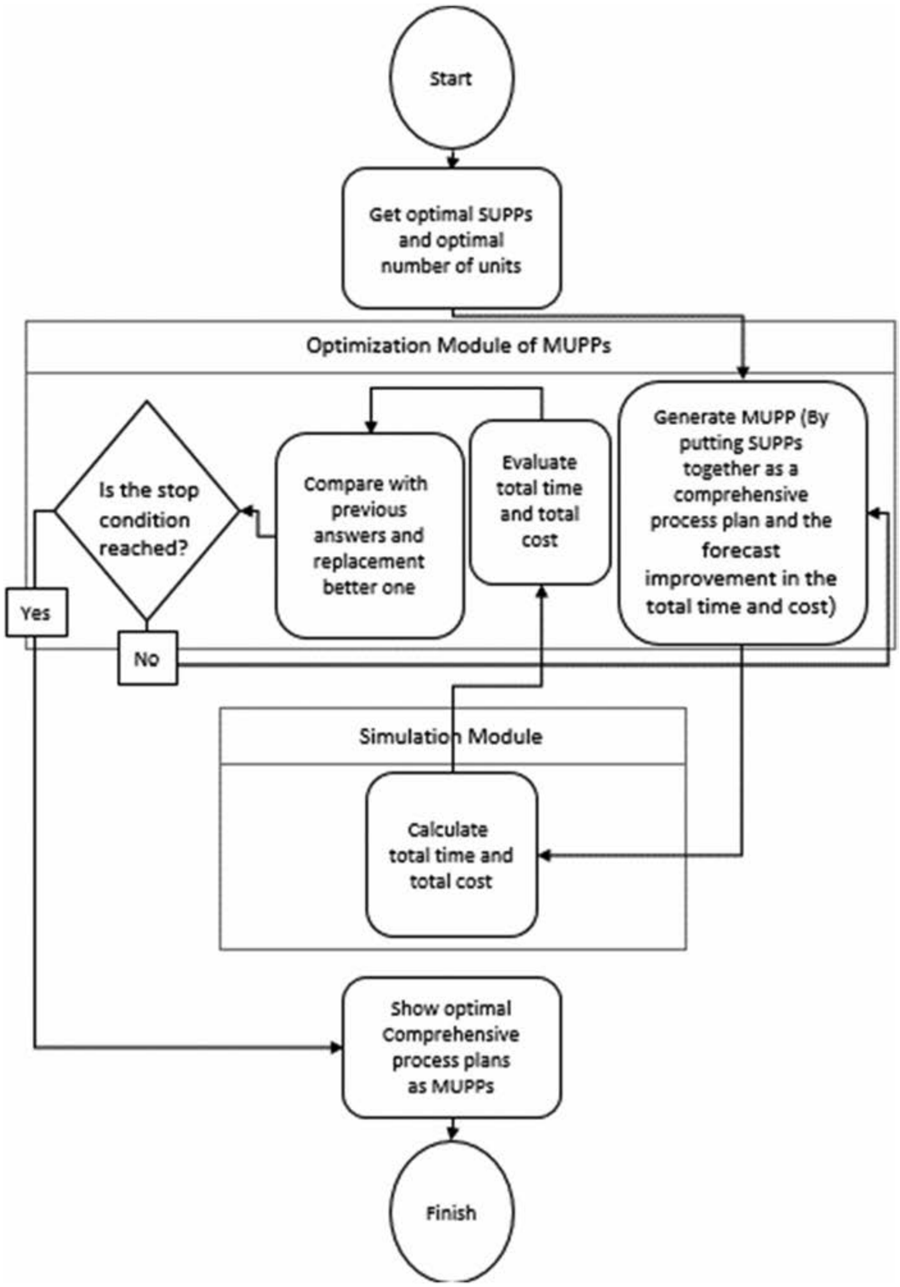

Figure 2. The performance of the optimization and simulation module and feedback to generate MUPPs (Step 2).

value for each demand in each designed period is considered via the summation of overlapped demands. This value is determined based on the ratio of the given period to the available completion time for responding to the demands. Accordingly, using the designed simulation module, the best periods are obtained for responding demands. Eventually, different unpredictable demands at different periods with unpredictable deadlines are dealt with based on the DPA by SUPP and MUPP considering the optimum answers for the process plans, the optimum number of units for the MUPP and also the obtained comprehensive (m-unit) process plans. For example, suppose some demands are received on different days with different deadlines; In order to create the Designed Periods, Algorithm 1 must be followed.

3.3. Determining the optimal parameters of the metaheuristic algorithms by Taguchi method (TM)

The Taguchi Method is applied to cases where multiple parameters influence the problem. This method is one of the most efficient ways in which the effects of controllable and uncontrollable variables on the quality of the problem can be determined together. Using

Algorithm 1: DPA

```
Initialise Total Period Time as TPT, Start times of Periods as StP, Demand of Periods as DP, Deadlines of Periods as FtP, where d is the index of Designed Periods, p is the index of periods, Start times of Designed Periods as StDP, Deadlines of Designed Periods as FtDP, Demand of Designed Period as DDP, Maximum Capacity of DP as MC t = 1, d = 1
Sort (Periods, Based on Start times)
the start time of the first designed period considered zero: StDP(1) = 0
while t < TPT
    if t = = any(StP(p) or FtP(p))
        The end time of a DP is equal to the occurrence of the new event: FtDP(d) = t
        for each p from 1 to P
            if StP(p) is in (StDP(d), FtDP(d))
                logical statement (1 or 0): log1 = (StP(p) < StDP(d))
                logical statement (1 or 0): log2 = (FtP(p) < FtDP(d))
                D = (FtDP(d)×(1–log2)+FtP(p)×(log2))-(StDP(d)×(log1)+StP(p)×(1–log1)) ÷(FtP(p)-StP(p))×Dp(p)
                add the value obtained to this value DP: DDP(d) = DDP(d)+D
                if DDP(d) > MC
                    Subtract the value from this value DP: DDP(d) = DDP(d)–D
                    d” = Index of Minimum(DDP(d’ for each d’ from 1 to d))
                    add the value to the minimum value of DPs: DDP(d”) = DDP(d”)+D
                    if DDP(d”) > MC
                        Display (None of the PPs can meet this amount of demand)
                        get out of the main loop
                    end if
                end if
            end if
        end for
        the start time of the next DP is equal to the end of the current DP: StDP(d+1) = t
        the start time and deadline of a DP have been set; go to the next period: d = d+1
    end if
    end if
    take the time one unit forward: t = t + 1
end while
```

the Taguchi Method, the effect of many variables on the problem can be examined at a lower cost within a shorter time (Güvenç, Çakır, and Mıstıkoğlu 2019). All of the following tasks must be performed in the Taguchi Method: determining factors and interactions, determining the level of each factor, selecting an appropriate orthogonal matrix, transferring factors and interactions to columns of orthogonal matrices, conducting experiments, analyzing data and determining optimum levels and finally confirming the validity of all mentioned steps (Güvenç, Çakır, and Mıstıkoğlu 2019). In the current paper, Minitab 17 software, which provides two graphs of Mean of Means (MM) and Mean of Signal to Noise ratios (MSN), has been utilised for the implementation of the Taguchi Method. The minimum value of the MM graph and the maximum values of different levels in the MSN have been selected as the values of each of the parameters at different levels (Güvenç, Çakır, and Mıstıkoğlu 2019).

3.4. Performance evaluation and parameter calibration of metaheuristic algorithms

In this paper, four performance metrics are used to assess the performances of the three proposed algorithms (NSGA-II, AMOSA and MOPSO) regarding the problem under consideration (Nemati-Lafmejani, Davari-Ardakani, and Najafzad 2019). The multi-objective performance metrics applied in this research are as follows.

- Quality Metric (QM)

One of the most crucial criteria to compare answers obtained by multi-objective metaheuristic algorithms is the Quality Metric. Pareto set of solutions for all algorithms, in QM, are compared to determine the metric, and those dominated by solutions of the other algorithm are removed. Afterwards, quality metric associated with each algorithm is determined by dividing the cardinal of the remaining set of Pareto-optimal solutions to the cardinal of the original set of Pareto optimal solutions. The higher the quality metric, the better the performance of the algorithm (Nemati-Lafmejani, Davari-Ardakani, and Najafzad 2019).

- Mean Ideal Distance (MID)

The mean ideal distance metric measures the relative distance of Pareto-optimal solutions from the ideal solution. The ideal solution is defined as the solution that individually optimises all objective functions of the problem. Equation (1) is used to calculate this performance metric.

$$MID = \frac{\sum_{i=1}^{n}\sqrt{\left(\frac{f_1^i - f_1^{best}}{f_{1,total}^{max} - f_{1,\,total}^{min}}\right)^2 - \left(\frac{f_2^i - f_2^{best}}{f_{2,total}^{max} - f_{2,\,total}^{min}}\right)^2}}{n} \tag{1}$$

where, $f_{k,total}^{max}, f_{k,total}^{min}$ represent the maximum and minimum values of kth objective functions, respectively. For maximising/minimising objectives, f_k^{best} is equal to $f_{k,total}^{max}$, $f_{k,total}^{min}$, respectively. Also, f_k^i represents the value of the kth objective function for ith Pareto-optimal solution, and n represents the main set of Pareto-optimal solutions. Lower values of MID indicate better performance of the algorithm in terms of the MID metric (Nemati-Lafmejani, Davari-Ardakani, and Najafzad 2019).

- Diversification Metric (DM)

The diversification metric shows the diversity of solutions obtained by metaheuristic algorithms. Equation (2) is used to calculate this performance metric for a bi-objective optimisation problem.

$$DM = \sqrt{(f_{1,total}^{max} - f_{1,total}^{min})^2 - (f_{2,total}^{max} - f_{2,\,total}^{min})^2} \tag{2}$$

Higher values of DM are indicative of a superior performance (Nemati-Lafmejani, Davari-Ardakani, and Najafzad 2019).

- Number of Pareto-optimal Solutions (NPS)

This metric is defined as the main set of Pareto-optimal solutions according to which, higher NPS values are preferable.

Due to different dimensions of the aforementioned performance metrics, the concept of Relative Performance Deviation (RPD) is utilised such that, for each experiment, the response variables (QM, MID, DM and NPS) are calculated for each metaheuristic algorithm. Then, Equation (3) is used to calculate the $\mathrm{RPD_i}$ value for each response variable i (Nemati-Lafmejani, Davari-Ardakani, and Najafzad 2019).

$$RPD_i = \frac{(RV_{ik} - Best_i)}{(Worst_i - Best_i)} \tag{3}$$

where $\mathrm{RPD_i}$, $\mathrm{RV_{ik}}$, $\mathrm{Best_i}$ and $\mathrm{Worst_i}$ represent the RPD value for the ith response variable (performance metric), the value of the ith response variable for the kth experiment, the best obtained value and the worst obtained value of the ith response variable among all experiments, respectively. Finally, the weighted mean of $\mathrm{RPD_i}$ values represents the RPD value. The obtained solutions from all algorithms were examined by the Taguchi Method (TM) to calibrate parameters of metaheuristic algorithms (Nemati-Lafmejani, Davari-Ardakani, and Najafzad 2019).

4. Formulation of the simulation module

In this section, the formulation method of the considered simulation module is explained. The calculation performance of the simulation module is explained as a mathematical and understandable model in section 4.1, followed by Section 4.2 which describes the encoding and decoding method for solution generation by the metaheuristic algorithms. Pseudocode of proposed metaheuristics algorithms for step 1 are described in Section 4.3.

4.1. Mathematical calculation in the simulation module

To formulate the first step (SUPP) of the problem, the following notations are used:

In the present research, both total cost and completion time objective functions are minimised simultaneously. The objective functions and constraints are stated in the following section.

- Total Cost objective function

The imposed costs are: Machine Usage Cost (MUC), Machine Change Cost (MCC), Configuration Change Cost (CCC), Tools Usage Cost (TUC), Tools Change Cost (TCC), all of which are obtained by Equations (4) to (8) and Equation (9) shows the total cost objective function:

$$MUC = \sum_{i=1}^{NP}\sum_{j=1}^{NOP_i} CM(M(i,j),C(i,j)) \times PrTime(\mathrm{M(i,j),C(i,j),T(i,j)}) \tag{4}$$

Indexes	
i	1, . . . , NP {Number of Parts}
j,j’	1, . . . , NOP(i) {Number of Operations of Part i}
d	1{+x},2{-x},3{+y},4{-y},5{+z},6{-z}
t,t’	1, . . . , NT {Number of Tools}
k	1, . . . , NM {Number of Machines}
l	1, . . . , NC(k) {Number of Configuration for machine k}
Nomenclature	
D(i,j)	Selected TAD to perform operation j of part i
T(i,j)	Selected Tool to perform operation j of part i
M(i,j)	Selected Machine to perform operation j of part i
C(i,j)	Selected Configuration to perform operation j of part i
Inputs	
OPP(i,j,j’)	Operation Precedence matrix for Part i {1 if OP(j) must be done before OP(j’), 0 otherwise}
OPTAD(i,j,d)	Matrix of the TADs required for operation j of part i {1 if operation j of part i required TAD d, 0 otherwise}
OPT(i,j,t)	Matrix representing the Tool required for Operation j of part i {1 if operation j of part i required tool t, 0 otherwise}
MT(k,t)	Matrix showing the Tools available with Machine k {1, if tool t available with machine k,0 otherwise}
CTAD(k,l,d)	Matrix representing the TAD offered by Configuration l of machine k {1, if TAD d available with configuration l of machine k}
Cost Information Inputs	
CM(k,l)	Cost of using configuration l of Machine k per time unit
TrC(k,k’)	Cost of Transfer between machine k and machine k’
CCCost(k,l,l’)	Configuration Change Cost from configuration l to configuration l’ of machine k
CT(k,t)	Cost of using Tool t of machine k per time unit
TCCost(k,t,t’)	Tool Change Cost from tool t to tool t’ of machine k
Time Information Inputs	
CCTime(k,l,l’)	Configuration Change Time from configuration l to configuration l’ of machine k
TrTime(k,k’)	Time of Transfer between machine k and machine k’
TCTime(k,t,t’)	Tool Change Time from tool t to tool t’ of machine k
PrTime(k,l,t)	The Processing Time for a particular operation by configuration l of machine k, by tool t
Decision Variables	
$S_{ijM(i,j)}$	Start time of operation j for part i on machine M(j) (the first activity of the manufacturing plan starts at time instant 0).
$Com_{ijM(i,j)}$	Completion time of operation j for part i on machine M(j)
Outputs	
Ϝ cost	The total cost
Ϝ time	The completion time

$$MCC = \sum_{i=1}^{NP} \sum_{j=1}^{NOP_i} TrC(M(i,j),M(i,j')) \quad OP_j < OP_j', \quad \forall M(j) \neq M(j') \tag{5}$$

$$CCC = \sum_{i=1}^{NP} \sum_{j=1}^{NOP_i} CCCost \times (M(i,j),C(i,j),C(i,j')) \quad OP_j < OP_{j'}, \quad \forall M(j) = M(j') \tag{6}$$

$$TUC = \sum_{i=1}^{NP} \sum_{j=1}^{NOP_i} CT(T(i,j)) \times PrTime(\mathrm{M(i,j),C(i,j),T(i,j)}) \quad (7)$$

$$TCC = \sum_{i=1}^{NP} \sum_{j=1}^{NOP_i} TC\mathrm{Cost(M(i,j),T(i,j),T(i,j'))} \quad (8)$$

$$f_{\mathrm{cost}} = MUC + MCC + CCC + TUC + TCC \quad (9)$$

- Completion Time objective function

Completion time, which is the maximum time needed for completing each operation of each part on the machine, is calculated using Equation (10) and (11):

$$Com_{ijM(\mathrm{i,j})} = S_{ijM(\mathrm{i,j})} + PrTime \times (M(i,j),C(i,j),T(i,j)) \quad \forall i \in 1,\ldots,NP, \; \forall j \in 1,\ldots,NOP_i \quad (10)$$

$$S_{ijM(i,j)} = Max(Com_{ij-1M(i,j-1)} + TrTime(M(i,j-1), M(\mathrm{i},j), Com_{i'j'M(i,j)} + TCTime(\mathrm{M(i,j), T(i,j'), T(i,j)}) + CCTime(M(i,j), C(i,j'), C(i,j))) \quad \forall i \in 1,\ldots,NP, \; \forall j \in 1,\ldots,NOP_i \quad (11)$$

where:

$$OP_{j\prime} < OP_j, \forall M(i,j') = M(\mathrm{i},j)$$

Equation (12) shows the total time objective function:

$$f_{time} = Max(Com_{ijM(\mathrm{i,j})}) \quad \forall i \in 1, \ldots,NP, \; \forall j \in 1,\ldots,NOP_i \quad (12)$$

- Model constraints

The proposed model optimisation is subject to the constraints expressed in the following:

Equation (13) ensures that a machine assigned to an operation is able to perform this operation in terms of the degree of freedom.

$$OPTAD(i,j,D(i,j)) < CTAD(M(i,j),C(i,j),D(\mathrm{i},j)) \quad \forall i \in 1, \ldots,NP, \; \forall j \in 1,\ldots,NOP_i \quad (13)$$

Equation (14) guarantees that the precedence constraints are satisfied (All operations should satisfy the precedence cluster assigned).

$$OPP(i,j,j_\prime) < 1 \quad \forall j_\prime < j, \; \forall i \in 1,\ldots,NP, \; \forall j \in 1,\ldots,NOP_i \quad (14)$$

Equation (15) ensures that the tools associated with a certain operation are able to perform it.

$$OPT(i,j,T(i,j)) < \mathrm{MT(M(i,j)},T(i,j)) \quad \forall i \in 1,\ldots,NP, \; \forall j \in 1,\ldots,NOP_i \quad (15)$$

4.2. Pseudocode of the proposed metaheuristics algorithms for step 1

The pseudo-algorithm of NSGA-II is proposed in Algorithm 2. However, for clear descriptions of the coded process plan as well as the mutation and the crossover operators, refer to (Bensmaine, Dahane, and Benyoucef 2013b).

While the pseudocode of AMOSA are proposed in Algorithm 3, more detailed descriptions of this metaheuristic algorithm can be found in Touzout and Benyoucef (2019).

Also, the pseudocode of MOPSO are proposed in Algorithm 4; for further details about which, refer to (Coello Coello, Pulido, and Lechuga 2004).

Algorithm 2: NSGA-II

```
1: initialise Tmax, t = 1, newPopulation archive
2: while t < Tmax
3:     T = fast-nondominated-sort (newPopulation)
4:     parentPopulation = 0
5:     while |parentPop| < N
6:         T = crowding-distance-assignment (T)
7:         parentPopulation = parentPopulation + T
8:     end while
9:     Sort (parentPopulation)
10:     ParentPopulation = first N element in
   parentPopulation
11:     childPopulation = generateNewPopulation (use
   selection, crossover and mutation)
12:     newPopulation = ParentPopulation ∪
   childPopulation
13:     t = t + 1
14: end while
15: return archive
```

Algorithm 3: AMOSA

```
initialise Tmax, Tmin, iter, α, temp = Tmax, perturbationRatio archive
current = random(archive)
while temp > Tmin
    for i = 0: iter
        new = perturb(current)
        depending on the dominance status: new replace current and it is added to the archive
        delete dominated process plans from the archive
        end for
        temp = α × temp
end while
return archive
```

Algorithm 4: MOPSO

```
initialise the position and velocity of each particle, Tmax, t = 1
evaluate the particles based on the cost function
archive = select non-dominated solutions from the swarm
create the grid and find the grid index of particles in the archive
while t < Tmax
    for each particle
        select the leader (gbest) from the archive for each particle based on grid index
        update the velocity and position
        evaluate the particle, based on the cost function
        update pbest of each particle
    end for
    add non-dominated particles to archive
    determine the domination of new archive members
    keep only non-dominated members in the archive
    update grid and grid index
    t = t+1
end while
return archive
```

4.3. Description of the encoding and decoding method for solution generation by the metaheuristic algorithms

To understand the process plan generated using the metaheuristic algorithms in the present study, introduction of an instance of a process plan seems to be necessary. Accordingly, a process plan is presented by an M×N matrix. Table 2 illustrates an example of a process plan, which is interpreted from left to right in a column by column order. For example, the first column is read as follows: operation Op1 for part P2 is conducted in machine M2, with configuration C2 and using tool T4. Other columns are interpreted in the same way.

Table 2. An example of a process plan in tabular form.

Part	P2	P1	P3	P1	P2	P3	P1
Operation	Op1	Op1	Op1	Op3	Op2	Op2	Op2
Machine	M2	M3	M1	M1	M3	M2	M2
Configuration	C2	C1	C2	C1	C3	C1	C1
Tool	T4	T1	T2	T3	T2	T2	T1

Solution encoding, which includes the movement from the real form presentation of the solution to an encoded presentation form, is a crucial step in the execution process of metaheuristic algorithms. The main goal of this step is the generation of more solutions by the search algorithm, thereby clarifying unnecessary details and allowing the search algorithm to browse the search space readily. Manipulating the encoded solution throughout the evolution of the search process can affect some of the values. If the encoding/decoding method is not selected accurately, changes of this kind can lead to infeasible solutions, to prevent which a method of encoding is presented in the space of real numbers in this paper. The encoded solution is always in the same form as the data structure mentioned for the process plan, the so-called M×N matrix form. The encoded string has five groups of variables, including parts, operations, machines, configurations and tools, each of which occupies one row of the matrix. The number of columns shows the number of all operations. Every element in the encoded solution is a real number in the range of 0–1 (0 excluded). This encoded solution should be decoded in the space of the real numbers in a process plan to be evaluated. The decoding process is conducted from left to right and row by row using a similar principle to the 'Roulette Wheel Selection Method' in the genetic algorithm. Table 3 shows an example of a process encoded by real numbers.

There are several methods for decoding different components of process plans. Decoding the parts in this paper consists of the following steps (Bensmaine, Dahane, and Benyoucef 2013b).

Table 3. Process Plans encoded in real numbers.

Part	0.66	0.06	0.23	0.32	0.35	0.63	0.05
Operation	0.04	0.31	0.82	0.81	0.15	0.21	0.86
Machine	0.20	0.04	0.53	0.51	0.90	0.90	0.89
Configuration	0.84	0.21	0.36	0.09	0.32	0.00	0.04
Tool	0.18	0.33	0.52	0.90	0.66	0.39	0.08

(a) Identifying parts that have not yet been completed (if there is precedence, those that have met the precedence).
(b) Making a list, such that n is the sum of the remaining operations for all candidate parts. For example, if two parts P1 and P2, are candidates, with 4 operations left (not yet done) for P1 and 3 operations left for P2, the list would be $n = 3 + 4 = 7$.
(c) Putting the candidate parts in the list. The number of positions of each part in the list is equal to the number of operations remaining from that part. Therefore, this list will look like this: P1 – P1 – P1 – P1 – P2 – P2 – P2.
(d) Assuming that the classification in the list starts with the lower index parts (P1 to P2).
(e) Multiplied n by its real number in the encoded matrix in the cell corresponding to the decoded cell. For example, to decode the second cell from the first row for the parts, the real number in the first row / second column is multiplied by n $(0 \leq p < n)$.
(f) Rounding up p. So p will be between 1 and n $(1 \leq p \leq n)$.
(g) Selecting the part in p position of the list in order to be placed in the corresponding cell.

For example, suppose the third cell in the row of parts is to be placed; The cell is read from the first row, and the third column from the encoded matrix, which is 0.23, then multiplied by the number of remaining multiples, which is 7, and rounds up; The number obtained is 2. The second part is read from the order of the remaining parts, which is P1. Accordingly, in the third cell of the row of parts, P1 is placed. Decoding the operations, machines, configurations and tools are also performed in the same way.

5. Illustrative example

In this section, a numerical example is given to examine the applicability of the proposed framework.

5.1. Input data

Input data related to the examples of the present work are precedence graphs between different operations, tools required to perform each operation and TADs. This RMS has been designed for a product built by a simultaneous assembly of 3 parts (parts from a family), the first, second and third parts of which need 3, 5 and 4 operations, respectively. The input information for the problem are stated in the following. Table 4 shows the TADs, tools required to perform each operation and precedence operations of each part.

Table 4. the TADs, tools, and the precedence required of each operation.

		TAD							
Part	Operation	−x	+x	−y	+y	−z	+z	Required Tools	Precedence
P1	Op1	✓	✓					T1	-
	Op2					✓	✓	T2	Op3
	Op3		✓					T1	Op1
P2	Op1	✓						T2	-
	Op2					✓	✓	T1 or T5	Op1
	Op3	✓				✓		T4	Op1
	Op4						✓	T1	Op1,Op3
	Op5			✓	✓			T1	Op1,Op3
P3	Op1				✓			T3	-
	Op2	✓						T4	Op1
	Op3			✓				T4	Op1
	Op4		✓		✓			T2	Op1,Op2,Op3

Table 5. TADs and the available tools for each machine.

		TAD						
Machines	Configurations	−x	+x	−y	+y	−z	+z	Available Tools
M1	C1	✓	✓		✓			T1,T4
	C2		✓		✓		✓	
	C3	✓	✓	✓		✓		
M2	C1		✓		✓	✓	✓	T1,T2,T3,T6
	C2	✓		✓	✓			
M3	C1					✓	✓	T1,T5,T7
	C2	✓	✓	✓	✓			

Details of 3 machines are defined in Table 5, where the existing TADs in each machine resulting from each of the configurations (first, second and third machines have 3, 2 and 2 configurations, respectively) and the available tools for each machine (Total number of tools = 7) have been included.

5.2. Optimisation of the parameters of the proposed metaheuristic algorithms in step 1 using the Taguchi method

To determine the most appropriate parameters of NSGA-II, AMOSA and MOPSO using the performance metrics (mentioned in section 3.4), Taguchi Method and Minitab 17 software are used.

To determine the parameters of NSGA-II by the four given factors (Population Size, Number of Iterations, Crossover Rate and Mutation Rate) and three levels (Population Size: 200, 300 and 400; Number of Iterations: 300, 500 and 700; Crossover Rate: 0.7, 0.8 and 0.9; Mutation Rate: 0.1, 0.2 and 0.3), a nine-some fractional factorial design is created. Based on the acceptable answers obtained from both MM and MSN graphs, proper parameters of the NSGA-II are determined. Using both graphs, the best values for the first, second, third and fourth parameters are set equal to 200, 700, 0.8 and 0.1,

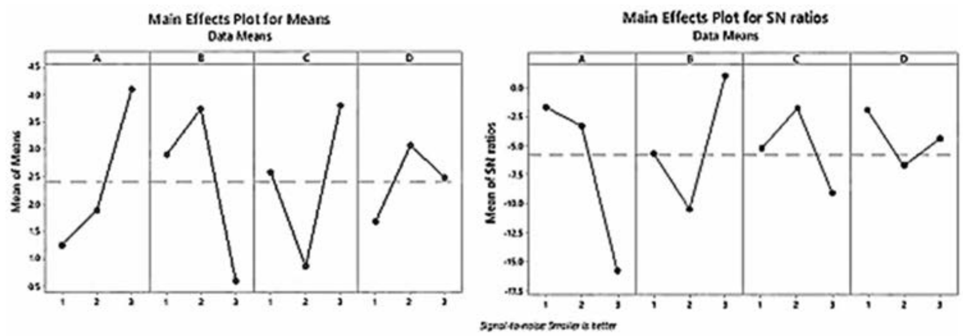

Figure 3. MM and MSN graphs for parameters of NSGA-II for step 1 (SUPP).

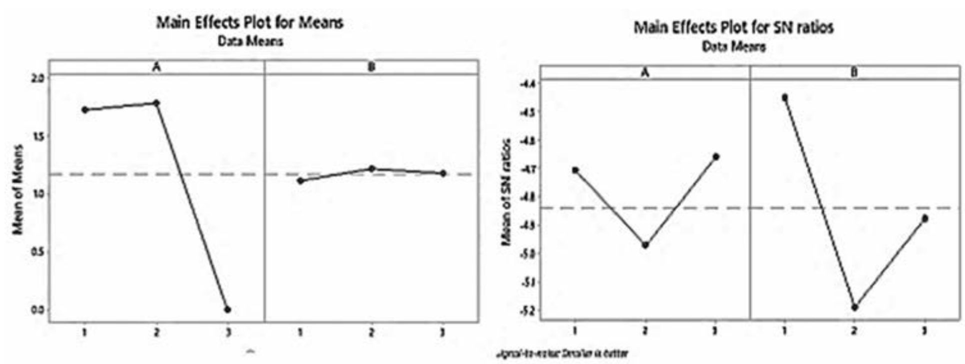

Figure 4. MM and MSN graphs for parameters of MOPSO for step 1 (SUPP).

respectively. The MM and MSN graphs for the NSGA-II are shown in Figure 3.

In a similar process, a nine-some fractional factorial design is created to determine the parameters of MOPSO based on the Taguchi Method by the two given factors (Particle Swarm and Iterations Number) and three levels (Particle Swarm: 50, 75 and 100; Iterations Number: 100, 150 and 200). The appropriate parameters for the MOPSO are determined considering the acceptable answers obtained from both MM and MSN graphs. Based on these graphs, the best values of the first and second parameters are set equal to 100. Figure 4 indicates the MM and MSN graphs for MOPSO.

In order to determine the parameters of AMOSA based on the Taguchi Method using the two given factors (Neighborhood Search in Each Temperature and Value of the Desired Temperature) and three levels (Neighborhood Search in Each Temperature: 3000, 5000 and 7000; Value of the Desired Temperature: 100, 200 and 300), a nine-some fractional factorial design is generated. The best parameters for the AMOSA are determined based on the acceptable answers obtained from both MM and MSN graphs. By using both graphs, the best values for the first and second parameters are obtained as 5000 and 100, respectively. The MM and MSN graphs for the AMOSA are shown in Figure 5.

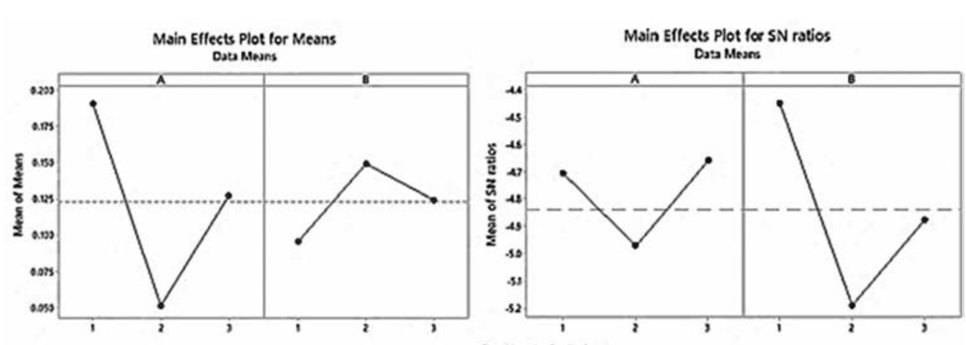

Figure 5. MM and MSN graphs for parameters of AMOSA for step 1 (SUPP).

Without the proposed encoding procedure, the search process would have spent an unreasonably long time on modifying the generated solutions instead of exploring the search space and improving the Pareto exponent.

5.3. Comparison of answers of the metaheuristic algorithms in SUPP using analysis of variance (ANOVA)

To make a general comparison between these algorithms, the weighted factor method is applied in order to convert the two objective functions to a single objective function. This allows a convenient comparison between the answers obtained from the algorithms. As a result, different weights are considered for each criterion and the final answer is obtained for each of the algorithms. Afterwards, all answers are standardised through the Relative Percentage Deviation (RPD) method. Finally, the experiment is conducted by using the ANOVA method in the Minitab 17 software.

Table 6 demonstrates the variance analysis for comparing the Pareto answers with the null hypothesis: 'identical quality for Pareto answers'.

Given that the P-value is greater than 0.05, there is no evidence to reject the null hypothesis. Therefore, the quality of the obtained Pareto answers is approximately the same.

Table 6. Variance analysis for Pareto answers of the intended algorithms.

Analysis of Variance					
Source	DF	Adj SS	Adj MS	F-Value	P-Value
Factor	2	563,312	281,656	3.07	0.068
Error	21	1,925,798	91,705	–	–
Total	23	2,489,111	–	–	–

5.4. Optimal answers of step 1 (SUPP)

The Pareto answers for the first step obtained through the three intended metaheuristic algorithms are demonstrated in Table 7.

5.5. Optimisation of units for step 2 (MUPP)

Now that the Pareto answers of the first step have been obtained, the second step can be implemented. For executing step 2 (MUPP) using the results shown in Table 6 (Identical quality of the Pareto answers for metaheuristic algorithms), all Pareto answers are considered the input data in this step, such that: Pareto answers from number 1–14, 15–19 and 20–23 are associated with NSGA-II, AMOSA and MOPSO, respectively. Then, the optimum number of units is selected for the comprehensive (m-unit) process plan, for which the plan is run by starting from one unit, followed by comparing the results. Finally, the optimum number of units is determined. In this step, the run-time criterion is considered in addition to the four mentioned criteria. Figure 6 indicates the optimality for the number of manufacturing units.

Owing to the existing relationship between the answers of units shown in Figure 6, plan execution with a high number of units leads to inappropriate values. Therefore, 3-unit is the most suitable answer based on the obtained results. For this reason, the number of units is considered 3 in the following sections.

5.6. Optimisation of NSGA-II parameters using the Taguchi method for step 2 (MUPP)

Since the NSGA-II is used for optimisation in this section, the most appropriate parameters must be determined using the Taguchi Method similar to the previous

Table 7. Values of Pareto answers for each of the meta-heuristic algorithms in step 1.

	NSGA-II		MOPSO		AMOSA	
	Completion Time (h)	Total Cost ($)	Completion Time (h)	Total Cost ($)	Completion Time (h)	Total Cost ($)
Run 1	0.4139	34.6523	0.4139	34.6523	0.3944	35.4142
Run 2	0.4125	34.6551	0.4069	35.0180	0.3917	35.7266
Run 3	0.3972	35.0192	0.3972	35.0192	0.3694	37.4676
Run 4	0.3944	35.0197	0.3903	35.6477	0.3653	37.9236
Run 5	0.3903	35.6477	0.3597	38.1297	–	–
Run 6	0.3875	36.0427	–	–	–	–
Run 7	0.3778	36.6907	–	–	–	–
Run 8	0.3750	36.8192	–	–	–	–
Run 9	0.3722	36.9312	–	–	–	–
Run 10	0.3694	37.1052	–	–	–	–
Run 11	0.3681	37.4492	–	–	–	–
Run 12	0.3653	37.5612	–	–	–	–
Run 13	0.3625	37.7352	–	–	–	–
Run 14	0.3597	38.1297	–	–	–	–

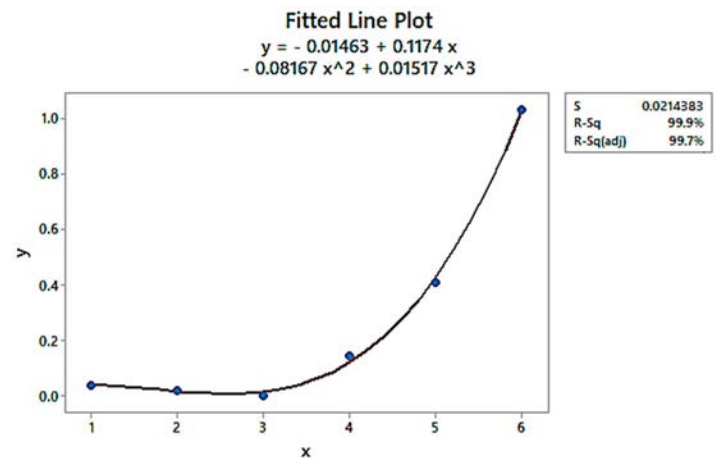

Figure 6. Optimality value graphs for the number of units for step 2 (MUPP).

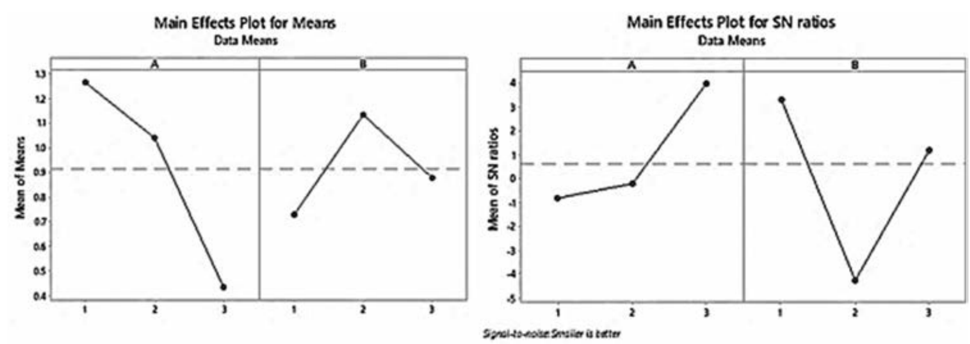

Figure 7. MM and MSN graphs for the parameters of NSGA-II for step 2 (MUPP).

section. Based on the Taguchi Method, with the two given factors (Population Size and Iterations Number) and three considered levels (Population Size: 20, 30 and 40; Iterations Number: 20, 30 and 40), the Taguchi Method is implemented using Minitab 17 software for determining the best parameters at the specified levels. Based on the obtained answers, the appropriate parameters for the NSGA-II are determined by comparing four evaluation criteria and concerning the acceptable answers resulting from both MM and MSN graphs. According to these two graphs, the best values for the first and second parameters are set equal to 40 and 20, respectively. The MM and MSN graphs for this algorithm are shown in Figure 7.

5.7. Optimal answers of MUPP and selected SUPP

Based on the Pareto answers in Table 7, the selected SUPP and MUPP (3-unit) with optimum time and cost are indicated in Table 8.

Table 8. Completion Time and Total Cost of the selected SUPP and 3-unit process plans.

	Process plans	Completion Time of the First Stage (h)	Total Cost of the First Stage ($)	Completion Time of the Second Stage (h)	Total Cost of the Second Stage ($)
SUPP1	[1]	0.4139	34.6523	0.4167	34.6541
SUPP2	[21]	0.3972	35.0192	0.4139	35.0231
SUPP3	[4]	0.3875	36.0427	0.4042	36.0466
SUPP4	[10]	0.3778	36.6907	0.3944	36.6936
SUPP5	[23]	0.3597	38.1297	0.3764	38.1326
MUPP1	[19,6,14]	1.1417	104.3309	1.1778	104.3370
MUPP2	[20,6,14]	1.0875	104.6961	1.0931	104.6973
MUPP3	[21,6,20]	1.0236	105.0602	1.0347	105.0627
MUPP4	[21,3,20]	1.0000	105.6887	1.0111	105.6912
MUPP5	[13,16,6]	0.9963	108.4815	0.9972	108.4791

Table 9. Unpredictable demands with start time and deadlines.

Main Period	Start	Period (d)	Deadline	Demand Quantity
1	0	35	35	500
2	14	12	26	100
3	22	48	70	200
4	37	49	86	100
5	46	44	90	300

5.8. Unpredictable demands and designed periods (DP)

Different demands at different periods are unpredictably obtained using MATLAB 2016 software. Then, the obtained periods are divided into the designed periods. The unpredictable obtained demands at different times with different deadlines are reported in Table 9.

The main periods are indicated in Figure 8.

The designed periods based on DPA are reported in Table 10.

The designed periods for responding to the unpredictable demands are indicated in Figure 9.

5.9. Responding to the demand of each DP by SUPP and MUPP

As shown in Tables 9 and 10, five unpredictably obtained periods have been converted to nine designed periods. The procedure of responding to the designed periods is obtained in two ways: first, the simulation module and SUPP, and second, the simulation and 3-unit process plan. Table 11 demonstrates the process plans, the optimal total time and cost for responding to the demands of the designed periods based on the SUPP and MUPP (3-unit).

Table 12 illustrates the Total Cost and Total Time obtained by SUPP and MUPP (3-unit).

Table 10. Designed periods for responding to the unpredictable demands.

Designed Period	Start	Period (d)	Finish	Demand Quantity
1	0	14	14	285
2	14	8	22	126
3	22	4	26	67
4	26	9	35	147
5	35	2	37	8
6	37	9	46	47
7	46	24	70	280
8	70	16	86	213
9	86	4	90	27

Focusing on Table 12 and Figure 6, which shows the optimum number of units to be considered in step 2 (comprehensive process plan), it is observed that the MUPP (3-unit in this problem) provides a more optimised answer (26 h in time and $807.2 in cost) compared to the SUPP. This optimisation is attained due to two reasons:

(1) DPA divided unpredictable demands into the designed periods, based on which, it is possible to respond to them efficiently by the obtained process plans, resulting in more optimal answers.
(2) There will be no additional cost and time of changing the status from the current status to the initial status of configurations, tools and TADs in each production series using MUPP. Since several units are placed together and the units are filled with the process plans, the extra cost and time to change the current status to the initial status are eliminated, as a result of which, a more optimal answer is obtained.

Since the implementation of this framework is not time-consuming and can create optimality in both time and cost objective functions, it can be very beneficial in the RMSs.

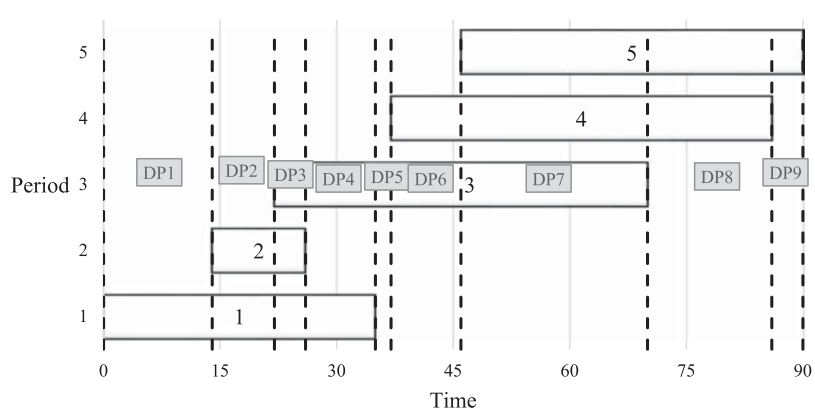

Figure 8. Unpredictable demands at different periods with unpredictable deadlines.

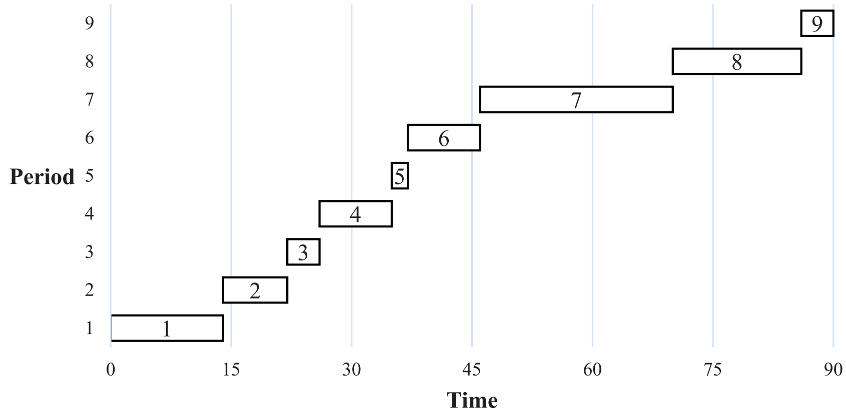

Figure 9. Designed periods for responding to the unpredictable demands.

Table 11. Responding to the demands of designed periods based on SUPP and MUPP.

	Start	Finish	DP Demand Quantity	Selected single- Unit	Total Time of DP (h)	Total Cost of DP ($)	Selected 3-units	Total Time of DP (h)	Total Cost of DP($)
1	0	1	285	[23]	108	10,867.8023	[20,6,14]	104	9,946.2423
2	14	22	126	[1]	53	4,366.4148	[19,6,14]	50	4,382.1499
3	22	26	67	[1]	28	2,321.8229	[19,6,14]	27	2,330.1879
4	26	35	147	[1]	62	5,094.1509	[19,6,14]	58	5,112.5093
5	35	37	8	[1]	4	277.2310	[19,6,14]	4	278.2259
6	37	46	47	[1]	20	1,628.7409	[19,6,14]	19	1,634.6076
7	46	70	280	[1]	117	9,703.1462	[19,6,14]	110	9,738.1185
8	70	86	213	[1]	89	7,381.3215	[19,6,14]	84	7,407.9244
9	86	90	27	[1]	12	935.6589	[19,6,14]	11	939.0273

Table 12. Total Time and Total Cost based on the SUPP and MUPP (3-unit).

	Total Time (h)	Total Cost ($)
SUPP	493 h	$42,576.2894
MUPP (3-unit)	467 h	$41,768.9933

6. Conclusion

The present paper was aimed at providing a two-step framework for indicating the ability of simulation-based optimisation methods to generate process plans and respond to unpredictable demands using a heuristic algorithm called DPA in a reconfigurable environment by SUPP (considering three metaheuristic including NSGA-II, AMOSA and MOPSO) and MUPP. The results of the numerical problem showed that in the designed periods (DP), the MUPP could more effectively respond to the unpredictable demands than SUPP. This superiority of answers is firstly due to the optimisation of the number of units of the MUPP. A second reason is that considering the possibility of putting several units together in an MUPP and filling the units with process plans and proper distribution of demands in the periods designed with the DPA, eliminates the extra cost and time to change the current status to the initial status. A suggestion for future studies could be to use other metaheuristic algorithms to obtain the optimal number of units and, consequently, the final answers with this number of units by MUPP. Another recommendation is to investigate the procedure of responding to unpredictable demands which cannot be responded in a specific time by any of the process plans. For example, a warehouse with some parts can be defined as storage to be responsive in case of shortage. As another future research, it is also possible to combine the RMS with the context of Industry 4.0. The reconfigurable machine tool can automatically select the number of units and process plans according to the required operations of the parts.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Notes on contributors

Amirhossein Kazemisaboor is an M.Sc. graduate in Industrial Engineering at K. N. Toosi University of Technology in Tehran, Iran. He received his BSc in Industrial Engineering from the same university (K. N. Toosi University of Technology in Tehran, Iran.). His main research interests are in Operations Research, Operations Management and Business Analytics.

Abdollah Aghaie is a professor of Industrial Engineering at K. N. Toosi University of Technology in Tehran, Iran. He received his BSc from Sharif University of Technology in Tehran, MSc from New South Wales University in Sydney, Australia and PhD from Loughborough University in the U.K. His main research interests are in Modeling and Simulation, Queuing System, Quality Management and Control, Supply Chain and Data Science.

Hamed Salmanzadeh is an assistant professor of Industrial Engineering at K. N. Toosi University of Technology in Tehran, Iran. He received his BSc in Mechanical Engineering from Urmia University in West Azarbaijan province, Iran and PhD and MSc in Mechanical and Process Engineering from Technische Universität Darmstadt in Darmstadt, Germany. His main research interests are in Ergonomics, Human factors, Productivity, Process Scheduling and Operations Management.

References

Bensmaine, A., Mohammed Dahane, and Lyes Benyoucef. 2011. "Simulation-Based NSGA-II Approach for Multi-Unit Process Plans Generation in Reconfigurable Manufacturing System." *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA* v. doi:10.1109/ETFA.2011.6059047.

Bensmaine, A., M. Dahane, and L. Benyoucef. 2013a. "A Non-Dominated Sorting Genetic Algorithm Based Approach for Optimal Machines Selection in Reconfigurable Manufacturing Environment." *Computers and Industrial Engineering* 66 (3): 519–524. doi:10.1016/j.cie.2012.09.008.

Bensmaine, A., M. Dahane, and L. Benyoucef. 2013b. "A Simulation-Based Genetic Algorithm Approach for Process Plans Selection in Uncertain Reconfigurable Environment." *IFAC Proceedings Volumes* 46 (9): 1961–1966. doi:10.3182/20130619-3-RU-3018.00458.

Bensmaine, A., M. Dahane, and L. Benyoucef. 2014. "A New Heuristic for Integrated Process Planning and Scheduling in Reconfigurable Manufacturing Systems." *International Journal of Production Research* 52 (12): 3583–3594. doi:10.1080/00207543.2013.878056.

Chaube, A., L. Benyoucef, and M. K. Tiwari. 2012. "An Adapted NSGA-2 Algorithm Based Dynamic Process Plan Generation for a Reconfigurable Manufacturing System." *Journal of Intelligent Manufacturing* 23 (4): 1141–1155. doi:10.1007/s10845-010-0453-9.

Coello Coello, Carlos A., Gregorio Toscano Pulido, and Maximino Salazar Lechuga. 2004. "Handling Multiple Objectives with Particle Swarm Optimisation." *IEEE Transactions on Evolutionary Computation* 8 (3): 256–279. doi:10.1109/TEVC.2004.826067.

Dou, Jianping, Jun Li, and Chun Su. 2016. "Bi-Objective Optimisation of Integrating Configuration Generation and Scheduling for Reconfigurable Flow Lines Using NSGA-II." *International Journal of Advanced Manufacturing Technology* 86 (5–8): 1945–1962. doi:10.1007/s00170-015-8291-8.

Dou, Jianping, Jun Li, Dan Xia, and Xia Zhao. 2020. "A Multi-Objective Particle Swarm Optimisation for Integrated Configuration Design and Scheduling in Reconfigurable Manufacturing System." *International Journal of Production Research* 1–21. doi:10.1080/00207543.2020.1756507.

Gadalla, Moustafa, and Deyi Xue. 2018. "An Approach to Identify the Optimal Configurations and Reconfiguration Processes for Design of Reconfigurable Machine Tools." *International Journal of Production Research* 56 (11): 3880–3900. doi:10.1080/00207543.2017.1406674.

Güvenç, Mehmet Ali, Mustafa Çakır, and Selçuk Mıstıkoğlu. 2019. "Experimental Study on Optimisation of Cutting Parameters by Using Taguchi Method for Tool Vibration and Surface Roughness in Dry Turning of AA6013." *Proceedings of 10th International Symposium on Intelligent Manufacturing and Service Systems*, no. September 2019: 1032–1040.

Khezri, Amirhossein, Hichem Haddou Benderbal, and Lyes Benyoucef. 2020a. "Towards a Sustainable Reconfigurable Manufacturing System (SRMS): Multi-Objective Based Approaches for Process Plan Generation Problem." *International Journal of Production Research* 1–26. doi:10.1080/00207543.2020.1766719.

Khezri, A., H. Haddou Benderbal, and Lyes Benyoucef. 2020b. "Sustainable Multi-Objective Process Plan Generation in RMS Through Modelling Energy Consumption." In *Reconfigurable Manufacturing Systems: From Design to Implementation*, 161–177. Springer. doi:10.1007/978-3-030-28782-5_8.

Koren, Y. 2006. "General RMS Characteristics. Comparison with Dedicated and Flexible Systems." In *Reconfigurable Manufacturing Systems and Transformable Factories*, 27–45. Springer. doi:10.1007/3-540-29397-3_3.

Koren, Y., U. Heisel, F. Jovane, T. Moriwaki, G. Pritschow, G. Ulsoy, and H. Van Brussel. 1999. "Reconfigurable Manufacturing Systems." *CIRP Annals* 48 (2): 527–540.

Landers, R. G., B. K. Min, and Y. Koren. 2001. "Reconfigurable Machine Tools." *CIRP Annals* 50 (1): 269–274. doi:10.1016/S0007-8506(07)62120-9.

Musharavati, Farayi, and A. S. M. Hamouda. 2012. "Enhanced Simulated-Annealing-Based Algorithms and Their Applications to Process Planning in Reconfigurable Manufacturing Systems." *Advances in Engineering Software* 45 (1): 80–90. doi:10.1016/j.advengsoft.2011.09.017.

Musharavati, Farayi, Napsiah Ismail, Abdel Majid S Hamouda, and Abdul Rahman Ramli. 2008. "A Metaheuristic Approach to Manufacturing Process Planning in Reconfigurable Manufacturing Systems." *Jurnal Teknologi* 48 (1): 55–70.

Najid, N.-M., P. Castagna, and K. Kouiss. 2020. "System Engineering-Based Methodology to Design Reconfigurable Manufacturing Systems." In *Reconfigurable Manufacturing Systems: From Design to Implementation*, 29–55. Springer. doi:10.1007/978-3-030-28782-5_3.

Nemati-Lafmejani, Reza, Hamed Davari-Ardakani, and Hamid Najafzad. 2019. "Multi-Mode Resource Constrained Project Scheduling and Contractor Selection: Mathematical Formulation and Metaheuristic Algorithms." *Applied Soft Computing* 81: 105533. doi:10.1016/j.asoc.2019.105533.

Oke, A. O., K. Abou-El-Hossein, and N. J. Theron. 2011. "Process Planning for Reconfigurable Manufacturing System for Mould/Die Making." In *Presented at the 4th Robotics and Mechatronics Conference of South Africa (ROBMECH 2011)*, 23:25.

Onar, Sezi Çevik, Başar Öztayşi, Cengiz Kahraman, Seda Yanık, and Özlem Şenvar. 2016. "A Literature Survey on Metaheuristics in Production Systems." In *Operations Research/Computer Science Interfaces Series*, Vol. 60, 1–24. Springer. doi:10.1007/978-3-319-23350-5_1.

Takahashi, K., K. Morikawa, T. Myreshka, T. Ohiro, and A. Takubo. 2006. "Selecting an Optimal Production Order and Its Corresponding Configuration in a Reconfigurable Manufacturing System with Multiple Product Groups." In *Reconfigurable Manufacturing Systems and Transformable Factories*, 609–626. Springer. doi:10.1007/3-540-29397-3_30.

Touzout, Faycal A., and Lyes Benyoucef. 2018. "Sustainable multi-unit process plan generation in a reconfigurable manufacturing environment: A comparative study of three hybrid-meta-heuristics." *IEEE* 1: 661–668. doi:10.1109/ETFA.2018.8502551.

Touzout, Faycal A., and Lyes Benyoucef. 2019. "Multi-Objective Sustainable Process Plan Generation in a Reconfigurable Manufacturing Environment: Exact and Adapted Evolutionary Approaches." *International Journal of Production Research* 57 (8): 2531–2547. doi:10.1080/00207543.2018.1522006.

Yelles-Chaouche, Abdelkrim R., Evgeny Gurevsky, Nadjib Brahimi, and Alexandre Dolgui. 2020. "Reconfigurable Manufacturing Systems from an Optimisation Perspective: A Focused Review of Literature." *International Journal of Production Research* 1–19. doi:10.1080/00207543.2020.1813913.