Multi-objective sustainable process plan generation in a reconfigurable manufacturing environment: exact and adapted evolutionary approaches

Faycal Touzout, Lyes Benyoucef

Multi-objective multi-unit process plan generation in a reconfigurable manufacturing environment: A comparative study of three hybrid metaheuristics

Faycal A. Touzout[1] and Lyes Benyoucef[2] [*]
[1]*DISP, INSA-Lyon, Lyon, France*
faycal.touzout@insa-lyon.fr
[2]*LIS UMR 7020, Aix Marseille University, Marseille, France*
lyes.benyoucef@lis-lab.fr

Abstract

Low costs, high reactivity and high quality products are necessary criteria for industries to achieve competitiveness in nowadays market. In this context, reconfigurable manufacturing systems (RMSs) have emerged to fulfill these requirements. RMS is one of the latest manufacturing paradigms, where machines components, software or material handling units can be added, removed, modified or interchanged as needed and when imposed by the necessity to react and respond rapidly and cost-effectively to changing. This research work addresses the multi-objective single-product multi-unit process plan generation problem in a reconfigurable manufacturing environment where three hybrid heuristics are proposed and compared namely: repetitive single-unit process plan heuristic (RSUPP), iterated local search on single-unit process plans heuristic (LSSUPP) and archive-based iterated local search heuristic (ABILS). Single-unit process plans are generated using the adapted non-dominated sorting genetic algorithm (NSGA-II). Moreover, in addition to the minimization of the classical total production cost and the total completion time, the minimization of the maximum machines exploitation time is considered as a novel optimization criterion, in order to have high quality products. To illustrate the applicability of the three approaches, examples are presented and the obtained numerical results are analysed.

Keywords: Reconfigurable manufacturing system (RMS), process plan generation, multi-objective optimization, NSGA-II, iterated local search.

1. Introduction

Manufacturing has always been a pilar of the global economy. In 2016, The World Bank Group stated that manufacturing accounted for 17% of the global gross domestic product (GDP) (WorldBankGroup, 2016). Today, customer satisfaction is a challenge for most manufacturing companies. Mass customization, a product deployment concept that combines low price with extensive variation and adaptation, has emerged due to its potential impact upon customer regarding the perceived value of the product (Tseng & Jiao, 2001). Nevertheless, with the continuous demand for products incorporating new and complex functionalities, there has been a lot of pressure on the manufacturing companies (ElMaraghy, 2008).

---

* corresponding author

On the other hand, it is well known that, the dedicated manufacturing system/line (DMS/DML) involves the production of one product at a time (Koren, 2010). Though it is cost effective in the case of the bulk production but lacks the variety. Where, flexible manufacturing system (FMS) involves production of variety of products at the same time but it involves a lot of initial investments (Koren, 2010). A cost effective response to market changes requires a new manufacturing approach that not only combines the high throughput of DML with flexibility of FMS, but also able to react to market changes quickly and efficiently.

According to Koren , father of reconfigurable manufacturing systems (RMSs), DMLs are inexpensive but their capacities are not fully utilized in several situations especially under the pressure of global competition, thus they engender losses (Koren et al., 1999). Moreover, FMSs respond to product changes, but they are not designed for structural changes. Hence, in both systems, a sudden market variation cannot be countered, like demand fluctuation or regulatory requirements. RMS combines the high flexibility of FMS with the high production rate of DML. It comprises the positive features of both systems, thanks to its adjustable structure and design focus. Thus, in situations where both productivity and system responsiveness to uncertainties or to unpredictable scenarios (e.g. machine failure, market change,...) are of a vital importance, RMS ensures a high level of responsiveness to changes with a high performance. This can be achieved through six main principles respectively customization, convertibility, scalability, integrability, modularity, and diagnosability.

Moreover, Koren suggested that in manufacturing systems, the key to responsiveness in markets as well as to cope with changing market conditions that causes product demand and mix fluctuations, is to adjust the production system capacity (Koren, 2006). He stressed that this adjustment is possible thanks to two types of reconfiguration capabilities in manufacturing systems, which are functionality adjustment and production capacity adjustment. These characteristics are achievable due to reconfigurable machine tool (RMT), which is considered as one of the major components of RMS. With this reconfigurable structure, RMT provides a customized flexibility and offers a variety of alternatives features. However, planning, managing and optimizing, in this context, are an-exponentially more complex tasks.

Recently, RMS became one of the most attractive research topics. Nevertheless, few works addressed the multi-objective process plan generation problem in reconfigurable environments. Due to the lack of works on this critical and challenging problem, in this paper, we consider the multi-objective multi-unit single-product process plan generation problem. In addition to the minimization of the classical total production cost and the total completion time, the minimization of the maximum machines exploitation time is considered as a novel optimization criterion. It consists of minimizing the total processing time of the most used machine in the process in order to prevent reliability issues and to have high quality products. To solve the problem, we develop and compare three hybrid metaheuristics namely: repetitive single-unit process plan heuristic (RSUPP), iterated local search on single-unit process plans heuristic (LSSUPP) and archive-based iterated local search heuristic (ABILS). The developed approaches are inspired by a microscopical study of optimal Pareto front solutions, generated with an iterative integer linear program for small instances of the problem.

The rest of the paper is organized as follows: Section 2 summarizes the related works to process plan

generation for a RMS. Section 3 describes the problem under consideration. Section 4 presents the proposed multi-objective mathematical formulation. Section 5 describes more in details the proposed three approaches. Section 6 analyses the obtained numerical results. Section 7 concludes the paper with some future research work directions.

2. Literature Review

Reconfigurable manufacturing system is a paradigm that answers many of the challenges that the market nowadays imposes. Therefore, it has been a very active research field where multiple state of the arts have been dedicated covering many areas, such as design, layout optimization, reconfigurable control, process planning and production scheduling (Huang et al., 2019; Prasad & Jayswal, 2018; Moghaddam et al., 2018; Gadalla & Xue, 2018; Bortolini et al., 2018; Maganha et al., 2018; Andersen et al., 2015; Renzi et al., 2014; Babu et al., 2013; Bi et al., 2008). However, in this section, we will summarize the most related research works dedicated to the process plan generation problems.

Swamidass (2000) defines the process planning (called also process plan generation) as *"the preparatory step before manufacturing, which determines the sequence of operations or processes needed to produce a part or an assembly. This step is more important in job shops, where one-of-a-kind products are made or the same product is made infrequently"*. Jain & Jain (2001) stated that *"process planning can be defined as the systematic determination of methods and means to manufacture a component economically and competitively"*. ElMaraghy (2007) insisted that *"we need to associate the evolutions, reconfigurations and reconfigurable process plans to changes and evolutions of manufacturing systems and products"*. Nallakumarasamy et al. (2011) considered the process plan as *"the activity that decides the sequence, which the manufacturing process must follow"*. Finally, Chaube et al. (2012) claimed that *"process planning is complex and crucial activity for enterprises, where the planning has to be carried out at both macro and micro levels. The macro-process planning is concerned with selecting the best sequence of multiple different processing steps and set-ups as well as the machines to perform the required operations. And the micro-process planning is concerned with the optimization of each individual operation in order to determine the best process parameters"*. At both levels, the problem is known complex and has attract researches and industrials with very interesting contributions. It becomes very complex when the manufacturing environment is reconfigurable.

Bruccoleri et al. (2005) proposed an agent-based approach with a negotation model for the production planning activities in reconfigurable enterprises. The multi-agent system supports decentralized decisions in enterprises characterized by complex, articulated and geographically distributed production capacities. However, the main disadvantage of this approach concerns the increasing complexity due to the multi-agent-system. In addition, it does not consider skills of RMS on the lowest planning level. Azab & ElMaraghy (2007) considered reconfigurable process plans, where an existing process plan is reconfigured when a new feature is added to an existing part, in order to avoid the generation of a new process plan. Reconfiguration of process plan consists to include minor modifications to meet the requirements of the new part. A weighted

ILP model is proposed where both the production cost and completion time are minimized. Furthermore, Shabaka & ElMaraghy (2007) developed a new genetic algorithm based model to perform process plan in RMS environment. The model simultaneously considers all process plan parameters such as machine assignment and machine configurations.

Abbasi & Houshmand (2009) and Abbasi & Houshmand (2011) addressed the problem of production planning of product families for a RMS where the orders arrive following the Poisson distribution. The orders are lost if they are not available. To determine the optimum sequence of production tasks, corresponding configurations, and batch sizes, they proposed a mixed integer nonlinear programming (MINLP) model. Since the problem is NP-hard, tabu search and genetic algorithm based procedures are proposed to solve the model. To illustrate the applicability of both the MINLP model and the procedures, numerical examples are presented and the results analysed.

Musharavati & Hamouda (2012a,b) investigated the use of simulated-annealing-based algorithms in solving process planning problem for a reconfigurable manufacturing. They developed several variants of the simulated annealing algorithms respectively a variant of the basic simulated annealing algorithm, a variant of the simulated annealing algorithm coupled with auxiliary knowledge and a variant of the SA algorithm implemented in a quasi-parallel architecture. The obtained experimental results showed the superiority of the variants in comparison to a basic simulated annealing algorithm. Moreover, Maniraj et al. (2014) proposed a two-phase-based ant colony optimisation approach to solve the process plan generation problem of a single product flow-line in a reconfigurable context. In the first phase, priority-based encoding technique is applied to find feasible operation clusters. Where, in the second phase, ant colony technique is used for minimising the total cost of the RMS. A case study is presented to demonstrate the applicability of the developed approach. Recently, Battaïa et al. (2017b) proposed a joint formulation for process planning and system configuration for design of rotary transfer machines for a mixed-model production of different parts. The objective is to minimize the total system cost. Moreover, Battaïa et al. (2017a) developed a decision support tool for the design of reconfigurable rotary machining systems with turrets used for producing several families of parts. The tool captures an exhaustive set of constraints such as precedence, inclusion and exclusion. They modelled the system using mixed integer programming formulation and proposed heuristics to take efficient decisions about part orientations, selection of machining modules, and configuration/reconfiguration of working positions depending on the part families to be produced.

In a multi-objective context, Chaube et al. (2012) and Bensmaine et al. (2013) proposed an evolutionary-based approach to solve the problem. Chaube et al. (2012) adapted the non-dominated sorting genetic algorithm (NSGA-II) where two objectives are considered, respectively, the total completion time and manufacturing cost. Bensmaine et al. (2013) integrated the process plan generation with the design problem using the same approach. They elaborated rich experimental comparisons and analyses based on the obtained Pareto fronts. Furthermore, Bensmaine et al. (2014) proposed a new heuristic to deal with the integrated process plan generation and operations scheduling (IPPS) problem for a RMS. The heuristic takes into ac-

count the multi-configuration nature of machines to integrate both process planning and scheduling. The obtained experimental results showed an advantage of the proposed heuristic over the sequential process planning and scheduling strategy. In the same context, Dou et al. (2016) considered the problem of integrating both the optimal design of physical configuration (called configuration generation) and the proper scheduling for reconfigurable flow line in which multiple parts within the same part family can be produced at the same time. Two objectives are minimized respectively the total cost including capital cost and reconfiguration cost and the total tardiness. To solve the problem, a multi-objective mixed integer programming model (MO-MIP) is established first and an adapted version of the NSGA-II is developed second. They presented an illustrative case study showing the applicability of the proposed NSGA-II.

More recently, Haddou Benderbal et al. (2018) developed an adapted version of the AMOSA to solve the integrated design and process plan generation problem for RMS. In addition to the classical optimization criteria, respectively, cost and time, the authors considered modularity as a third criterion. Xia et al. (2018) extended the concept of reconfigurable process plan to a concept of reconfigurable machining process planning which targets the process plan generation for a part family. Touzout & Benyoucef (2018a) solved the sustainable process plan generation problem for a RMS, where the amount of greenhouse gases (GHG) emitted during the manufacturing process is minimized in addition to the total production cost and completion time. The authors developed an iterative multi-objective integer linear programming (I-MOILP) approach and compared with adapted versions of the archived multi-objective simulated-annealing approach (AMOSA) and the NSGA-II approach. And, they studied the influence of the probabilities of genetic operators on the convergence of the adapted NSGA-II and illustrated the applicability of the three approaches using numerical examples. Moreover, inspired by Touzout & Benyoucef (2018a) and a microscopical study of optimal Pareto front solutions generated by an iterative integer linear program, Touzout & Benyoucef (2018b) developed three hybrid-metaheuristics using an adapted version of AMOSA with $2-opt$ heuristic to solve the multi-objective multi-unit process plan generation problem for a RMS.

Finally, from our literature review and as a natural extension of Touzout & Benyoucef (2018a) and Touzout & Benyoucef (2018b) results, in this paper, we propose and compare three hybrid metaheuristics to solve the multi-objective multi-unit process plan generation problem in a reconfigurable manufacturing environment. The next section details the problem under consideration.

3. Problem Description

In this paper, we attempt to solve the multi-objective multi-unit process plan generation problem for a single-product type in a reconfigurable manufacturing environment. Three optimization criteria are considered:

(i) The total production cost.

(ii) The total completion time.

(iii) The maximum exploitation time. It consists of minimizing the total processing time of the most used machine in the process in order to prevent reliability issues and to have high quality products.

3.1. Single-unit Process Plan

Let us consider a part to be manufactured. A part is composed of features and each feature contains an amount of operations to be achieved. The different operations are linked with each other by precedence constraints (e.g. Figure 1) and are represented by a set of tools as well as a tool approach directions (TADs) (i.e. $x_{\pm}, y_{\pm}, z_{\pm}$) that are needed to perform them. We can see that the resulting graph $G = \{V, A\}$ is an acyclic graph.
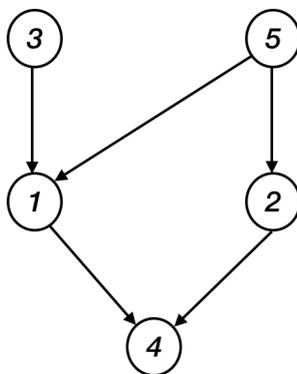

Figure 1: A precedence graph for the required operations

A RMS is composed of a set of RMTs. A machine in this case is represented by, respectively, a set of tools and configurations. Each configuration provides TADs. Table 1 shows the required TADs and tools for, respectively, operations and configurations of our example.

In this context, we can define, for each operation i, a set of triplets K_i. A triplet $k \in K_i$ in this case is defined by, respectively, the index of its machine ($indexM$), the index of its configuration ($indexC$) and the index of its tool ($indexT$).

Therefore, in our case, the process plan generation problem can be defined as the problem of determining a chronological sequence of operations to be performed as well as the triplets to perform each operation of the sequence. Table 2 illustrates a generated process plan for our example.

3.2. Multi-unit Process Plan

Now, let us consider that there are U units of the same product to be manufactured. A multi-unit process plan is a sequence of single-unit process plans. Table 3 presents an example of a generated multi-unit process

Table 1: Required TADs and Tools for our example

OP M — C		x_+	x_-	y_+	y_-	z_+	z_-	Tools
OP1			x	x	x			9
OP2		x		x				1
OP3			x		x			6
OP4			x	x	x			3
OP5		x					x	5
M1	C1	x		x		x	x	
	C2			x		x		1, 5, 6, 9
	C3		x	x		x		
	C4					x		
M2	C1	x	x	x	x			1, 5, 6, 9
M3	C1	x	x	x	x			1, 3, 9
M4	C1		x		x	x		
	C2					x	x	1, 6, 9
	C3					x		
M5	C1	x	x	x			x	1, 3, 5

Table 2: An illustrative example of a process plan

Operations	OP3	OP5	OP1	OP2	OP4
Machines	M4	M1	M2	M3	M3
Configurations	C1	C1	C1	C1	C1
Tools	T6	T5	T9	T1	T3

plan.

For this problem, we can distinguish three types of costs and times:

(i) Costs and times related to operations: the processing cost and the processing time to realise operation i with triplet k.

(ii) Costs and times related to machines: the cost and the time to change a configuration or a tool for a machine j.

(iii) Costs and times of transfer: the cost and the time to transfer a product from one machine to another.

The problem of process plan generation in RMS is NP-Hard. In fact, the proof is simple since by eliminating the aspect where for each operation, an optimal machine, configuration and tool need to be designated, the problem can be reduced to the well-known optimization problem the travelling salesman problem (TSP). Operations in this case will represent the nodes of the network and the objective will be to

Table 3: An illustrative example of a 5-unit process plan

Unit 1	Operations	OP3	OP5	OP1	OP2	OP4
	Machines	M4	M1	M3	M3	M3
	Configurations	C1	C1	C1	C1	C1
	Tools	T6	T5	T9	T1	T3
Unit 2	Operations	OP5	OP2	OP3	OP1	OP4
	Machines	M1	M2	M4	M3	M3
	Configurations	C1	C1	C1	C1	C1
	Tools	T5	T1	T6	T9	T3
Unit 3	Operations	OP5	OP3	OP1	OP2	OP4
	Machines	M1	M4	M3	M3	M3
	Configurations	C1	C1	C1	C1	C1
	Tools	T5	T6	T9	T1	T3
Unit 4	Operations	OP5	OP2	OP3	OP1	OP4
	Machines	M1	M2	M4	M3	M3
	Configurations	C1	C1	C1	C1	C1
	Tools	T5	T1	T6	T9	T3
Unit 5	Operations	OP5	OP3	OP1	OP2	OP4
	Machines	M1	M4	M3	M3	M3
	Configurations	C1	C1	C1	C1	C1
	Tools	T5	T6	T9	T1	T3

find the optimal route. The multi-unit process plan generation is therefore a general case for the single-unit process plan generation problem.

4. Mathematical Formulation

Throughout the next section, the following notations are used:

4.1. Data

Parameters :

O : Set of operations

n : Number of operations

U : Number of units

u : Index of unit

i, i' : Index of operation

P_i : Set of predecessors operations

t, t' : Index of triplet

j, j' : Index of position in the sequence

M : Set of machines

m, m' : Index of machine

T_m : Set of available triplets with machine m

T : Set of triplets, where: $T = T_i \cup T_m$

k, l : Index of configuration

tl, tl' : Index of tool

Production costs :

$CCM_{m,m'}$: Cost of changing machine per time unit

$CCC_{tl,tl'}$: Cost of changing configuration per time unit

$CCT_{tl,tl'}$: Cost of changing tool per time unit

$CP_{i,t}$: Cost of processing per time unit

Time :

$TCM_{m,m'}$: Time of changing machine

$TCC_{tl,tl'}$: Time of changing configuration

$TCT_{tl,tl'}$: Time of changing tool

$TP_{i,t}$: Time of processing

4.2. Decision variables

To formulate our problem, the following decision variables are needed:

$x_{i,j}^{t,u} = 1$ if the ith operation of the uth unit is processed at the jth position using the tth triplet, 0 otherwise.

$y_{j,t}^{m,u} = 1$ if the mth machine is using the tth triplet at the jth position of the uth unit, 0 otherwise.

$cm_{j,m,m'}^{u} = 1$ if between position $j - 1$ and j of the uth unit, there has been a change between machines m and m', 0 otherwise.

$cc_{j,t,t'}^{m,u} = 1$ if between position $j - 1$ and j of the uth, there has been a change between triplet t and t' of machine m, 0 otherwise.

4.3. Objectives

f^c, f^t and f^e are, respectively, the total production cost, the completion time and the maximum exploitation time, where:

$$f^c = \sum_{u=1}^{U} \sum_{j=1}^{n} \sum_{i \in O} \sum_{t \in T_i} x_{i,j}^{t,u} \times CP_{i,t} \times TP_{i,t} + \sum_{u=1}^{U} \sum_{j=1}^{n} \sum_{m \in M} \sum_{m' \in M} cm_{j,m,m'}^{u} \times CCM_{m,m'} \times TCM_{m,m'} +$$

$$\sum_{u=1}^{U} \sum_{j=1}^{n} \sum_{m \in M} \sum_{t \in t^m} \sum_{t' \in t^{m'}} cc_{j,t,t'}^{m,u} \times TCT_{IT^t, IT^{t'}} \times (CCC_{IC^t, IC^{t'}} + CCT_{IT^t, IT^{t'}})$$

$$f^t = \sum_{u=1}^{U} \sum_{j=1}^{n} \sum_{i \in O} \sum_{t \in T_i} x_{i,j}^{t,u} \times TP_{i,t} + \sum_{u=1}^{U} \sum_{j=1}^{n} \sum_{m \in M} \sum_{m' \in M} cm_{j,m,m'}^{u} \times TCM_{m,m'} + \sum_{u=1}^{U} \sum_{j=1}^{n} \sum_{m \in M} \sum_{t \in t^m} \sum_{t' \in t^{m'}} cc_{j,t,t'}^{m,u} \times$$

$$(TCC_{IC^t, IC^{t'}} + TCT_{IT^t, IT^{t'}})$$

$$f^e = \max_{m \in M} \sum_{u \in U} \sum_{t \in T_m} \sum_{i,j \in O} x_{i,j}^{t,u} \times TP_{i,t}$$

4.4. Model

Our problem can be formulated as the following Multi-Objective Integer Linear Program (MOILP). Note that the MOILP can generate both a multi-unit and single unit (i.e. if $U = 1$) process plans.

MOILP

$min\ f^c$

$min\ f^e$

$min\ f^t$

$s.t.$

$$\sum_{i\in O}\sum_{t\in T_i} x_{i,j}^{t,u} = 1 \qquad \forall j=1...n, \forall u=1...U \qquad (1)$$

$$\sum_{j=1}^{n}\sum_{t\in T_i} x_{i,j}^{t,u} = 1 \qquad \forall i\in O, \forall u=1...U \qquad (2)$$

$$\sum_{t\in T_i} x_{i,j}^{t,u}\times|P_i| \leq \sum_{i'\in P_i}\sum_{j'=1}^{j-1}\sum_{t'\in T_{i'}} x_{i',j'}^{t',u} \qquad \forall i\in O, \forall j=1...n, \forall u=1...U \qquad (3)$$

$$\sum_{t\in T_m} y_{j,t}^{m,u} = 1 \qquad \forall j=1...n, \forall m\in M, \forall u=1...U \qquad (4)$$

$$y_{j,t}^{m,u} \geq x_{i,j}^{t,u} \qquad \forall j=1...n, \forall m\in M, \forall t\in T_m, \forall u=1...U \qquad (5)$$

$$\sum_{i\in O} x_{i,j}^{t,u} + x_{i,j-1}^{t',u} \leq ct_{j,IM^t,IM^{t'}}^{u} + 1 \qquad \forall j=2...n, \forall t,t'\in T, \forall u=1...U \qquad (6)$$

$$y_{j,t^t}^{m,u} + y_{j-1,t'}^{m,u} \leq cc_{j,t,t'}^{m,u} + 1 \qquad \forall j=2...n, \forall m\in M, \forall t,t'\in T_m, \forall u=1...U \qquad (7)$$

$$\sum_{t,t'\in T_m} cc_{j,t,t'}^{m,u} = 1 \qquad \forall j=1...n, \forall m\in M, \forall u=1...U \qquad (8)$$

$$\sum_{i\in O} x_{i,1}^{t,u} + x_{i,n}^{t',u-1} \leq ct_{1,IM^t,IM^{t'}}^{u} + 1 \qquad \forall t,t'\in T, \forall u=2...U \qquad (9)$$

$$y_{1,t^t}^{m,u} + y_{n,t'}^{m,u-1} \leq cc_{1,t,t'}^{m,u} + 1 \qquad \forall m\in M, \forall t,t'\in T_m, \forall u=2...U \qquad (10)$$

$$\sum_{u\in U}\sum_{t\in T_m}\sum_{i,j\in O} x_{i,j}^{t,u}\times TP_{i,t} \leq f^e \qquad \forall m\in M \qquad (11)$$

Constraint (1) states that one operation is processed at each position of the process plan. Constraint (2) states that each operation is processed once. Constraint (3) states that an operation is processed if and only if all its predecessors operations are already processed. Constraint (4) states that each machine is using one configuration and one tool at once. Constraint (5) states which configuration and tool are used at position j for machine m. Constraints (6),(7),(9) and (10) state, respectively, if there's a change of machine and a change of configuration and/or tool between positions $j-1$ and j. Constraint (8) states that there's only one

change of configuration between positions $j-1$ and j. Finally, constraint (11) states the maximal exploitation time.

The MOILP will be excluded from the numerical experiments because of its high complexity for the multi-unit case. However, we used it to generate optimal solutions and study them microscopically. From our observations, an optimal multi-unit process plan can be:

(i) A repetition of an optimal single-unit process plan

(ii) A sequence of optimal single-unit process plans

(iii) A sequence of neighbours of optimal single-unit process plans

5. Proposed Approaches

In the following, inspired from the above observations, we will describe in details the developed approaches as well as the approach used to generate single-unit process plans.

5.1. Adapted Non Dominated Sorting Genetic Algorithm II (NSGA-II)

The non dominated sorting genetic algorithm (NSGA-II) is a population-based evolutionary algorithm proposed by Deb et al. (2002). Starting with a randomized initial population called the *parent population* of a given size, for each iteration of NSGA-II, a new population called *child population* is generated by applying genetic operators (i.e. mutation, crossovers...) with specified probabilities. The parent population of iteration $iter + 1$ is the result of an elitist procedure applied to $parentPopulation_{iter} \cup childPopulation_{iter}$. This elitist procedure is ensured by a fast non-dominated sorting algorithm, as well as a crowding distance sorting.

A description of our approach is proposed in Algo. 1. Furthermore, for clear descriptions of the used coded process plan as well as the mutation and the crossover operators, refer to Touzout & Benyoucef (2018a). More detailed descriptions of the fast non dominated sorting and the crowding distance sorting algorithms are also presented in Deb et al. (2002).

5.2. RSUPP: A repetitive single-unit process plan metaheuristic

RSUPP is a hybrid-metaheuristic that uses NSGA-II to generate a single unit Pareto front. A Pareto front of multi-unit process plans is then generated by the repetition of each single-unit process plan for each unit. A description of RSUPP is presented in Algo. 2.

Algorithm 1: Adapted NSGA-II
_____

1: input data

2: initialize $populationSize, iteration, p_{mutation}, mutationRatio, p_{crossover}$

3: randomize $parentPopulation$

4: **for** $iter = 1 : iteration$ **do**

5:   generate $childPopulation$ from $parentPopulation$

6:   $population = parentPopulation \cup childPopulation$

7:   $F = fastNonDominatedSorting(population)$

8:   **for** $l = 1 : size(F)$ **do**

9:     **if** size($newPopulation$)+size(F_l) $< populationSize$ **then**

10:       $newPopulation+ = F_l$

11:     **else**

12:       $crowdingDistanceSorting(F_l)$

13:       **for** $k = 1 : size(F_l)$ **do**

14:         **if** size($newPopulation$) $< populationSize$ **then**

15:           $newPopulation+ = F_l^k$

16:         **else**

17:           break;

18:         **end if**

19:       **end for**

20:     **end if**

21:   **end for**

22:   $parentPopulation = newPopulation$

23: **end for**

24: **return** $parentPopulation$
_____

Algorithm 2: RSUPP

1: input data

2: set an empty *archive*

3: solve the single-unit problem using Algo. 1

4: **for** $current - p$ in $archive(single - unit)$ **do**

5:     set MU: a multi-unit process plan

6:     **for** $u = 1...U$ **do**

7:         $MU_u = current - p$

8:     **end for**

9:     **if** MU is non dominated and $\notin archive$ **then**

10:         add MU

11:         remove dominated MUs in regards to MU

12:     **end if**

13: **end for**

14: **return** *archive*

5.3. ILSSUPP: An iterated local search on single-unit process plans metaheuristic

Similarly to the first step of RSUPP, ILSSUPP generates an optimal or near optimal Pareto front for the single-unit process plan problem. A multi-unit process plan in this case will be a combination of several single-unit process plans. An adaptation of the well-known traveling salesman problem heuristic $2 - opt$ (Croes, 1958) will be performed in order to find the best combinations. A more detailed description of the algorithm is presented in Algo. 3.

5.4. ABILS: An archive-based iterated local search metaheuristic

The idea behind ABILS is to utilize the archives of RSUPP and ILSSUPP, which are potentially a subset of the optimal Pareto front from our observations, to either generate better solutions or enlarge the size of the Pareto front. This goes through applying a re-adaptation of the $2 - opt$ algorithm. The difference between the $2 - opt$ of ABILS and ILSUPP is that for the former, it is applied at a microscopical level (i.e., for each unit, the neighbours of the single-unit process plan are generated). A simple illustration of the algorithm is presented in Algo. 4.

It is important to state that, for large-sized instances, ABILS and ILSSUPP can in fact be very time consuming. Although the enumeration of the $2 - opt$ neighbours can be done in a polynomial time, the approaches are still exponential because there's no guarantee for when they will stop improving the solutions of the archive. In this context, a time limit or a number of iterations as stopping conditions can be imagined.

Algorithm 3: ILSSUPP

1: input data

2: set an empty archive

3: solve the single-unit problem using Algo. 1

4: $current =$ a random multi-unit process plan

5: $archive \mathrel{+}= current$

6: **for** $current$ in $archive$ **do**

7:     **for** new in $2 - opt$ neighbours of $current$ **do**

8:         **if** new is non dominated and $\notin archive$ **then**

9:             add new

10:             remove dominated MUs in regards to new

11:         **end if**

12:     **end for**

13: **end for**

14: **return** archive

Algorithm 4: ABILS

1: input data

2: set an empty $archive$

3: solve $RSUPP$ and $ILSSUPP$

4: $archive = archive(RSUPP) + archive(ILSSUPP)$

5: **while** still improving **do**

6:     **for** $current$ in $archive$ **do**

7:         **for** new in $2 - opt$ neighbours of $current$ **do**

8:             **if** new is non dominated and $\notin archive$ **then**

9:                 add new

10:                 remove dominated MUs in regards to new

11:             **end if**

12:         **end for**

13:     **end for**

14: **end while**

15: **return** $archive$

6. Experimental Results & Analyses

The experiments were conducted with a 2.7 GHz Intel Core i5 processor and 8GB RAM and the meta-heuristics were implemented with Java. 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 and for $U = 5$. An instance is identified by the number of operations, the number of machines and its index. It is denoted by: $nbOperations\_nbMachines\_indexInstance$.

Based on the results of Touzout & Benyoucef (2018a), where the sensitivity of the parameters of NSGA-II for the single unit case were compared, we use the parameters that showed the best performance overall, and they are as follows: mutation = 90%, crossover = 10%, population-size = 40 and iteration=1000.

Although the optimality or the enumeration of the whole pareto fronts of the single-unit process plans generated with NSGA-II is not proven, we can see that the results of these experiments are in accordance to the phenomena observed in Section 4 when optimal pareto fronts generated with MOILP were microscopically studied.

We can distinguish three analyses from Figures 2:12.

Observation 1: From Figure 7, we can see that ABILS didn't produce any new process plans that dominates or are non dominated by the ones generated by RSUPP and ILSSUPP. This doesn't necessarily mean that the latter process plans (i.e. a repetition of an optimal single-unit process plan or a sequence of optimal process plans) are optimal but it strengthens however the idea that they are.

Observation 2: From Figures 3, 5, 6, 9, 10 and 11, we observe that ABILS produced new process plans that are non dominated by the ones generated by RSUPP and ILSSUPP. This means that ABILS is able to enrich the pareto fronts.

Observation 3: Finally, from Figures 2, 3, 4, 5, 6, 8, 11 and 12, we notice that ABILS produced new process plans that dominates either some process plans from the pareto front (e.g. Figures 3, 5, 6, 8 and 11) or the whole pareto front (e.g. Figures 2, 4 and 12) generated by RSUPP or ILSSUPP. Therefore, in this case, ABILS have two roles: generating good process plans and cleaning the pareto fronts from bad ones.
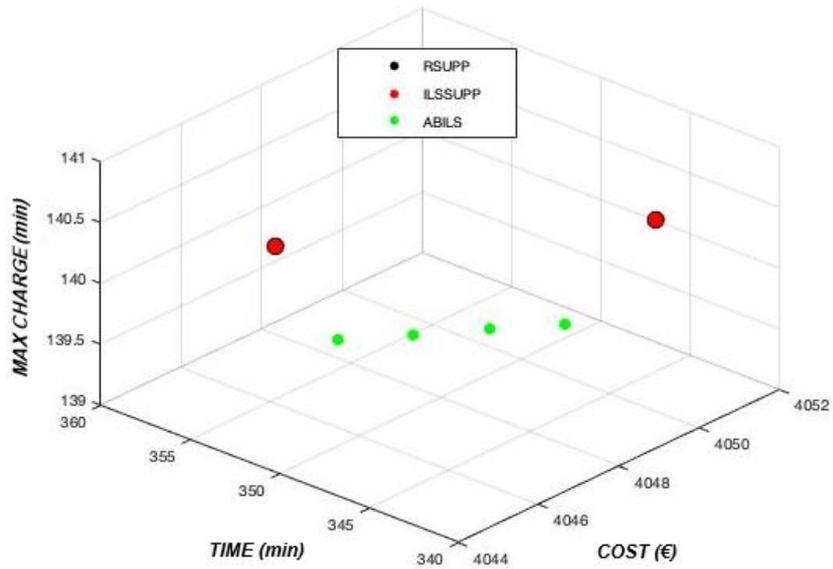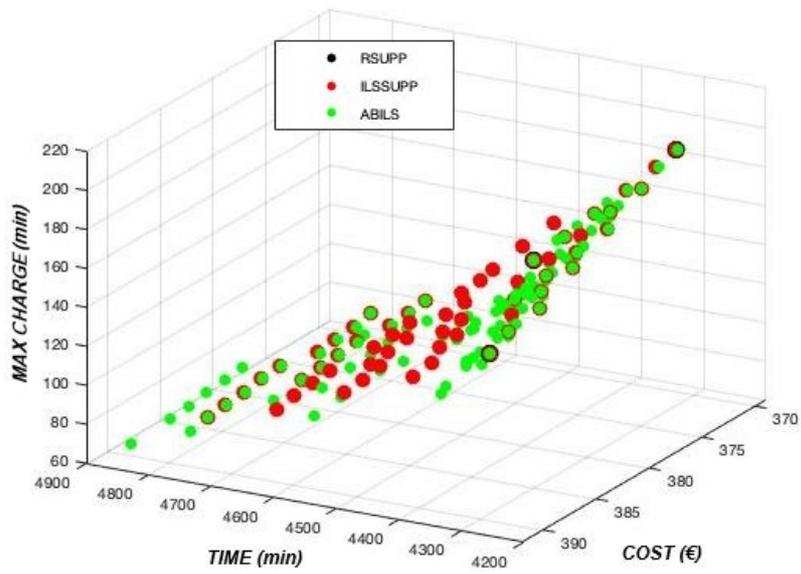
Figure 2: Parento fronts of instance 5_5_1

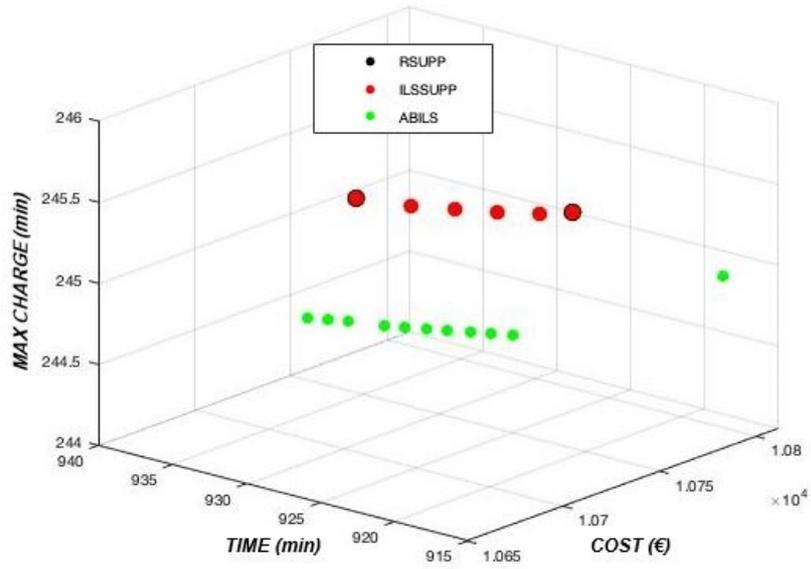Figure 3: Parento fronts of instance 5_5_4

17

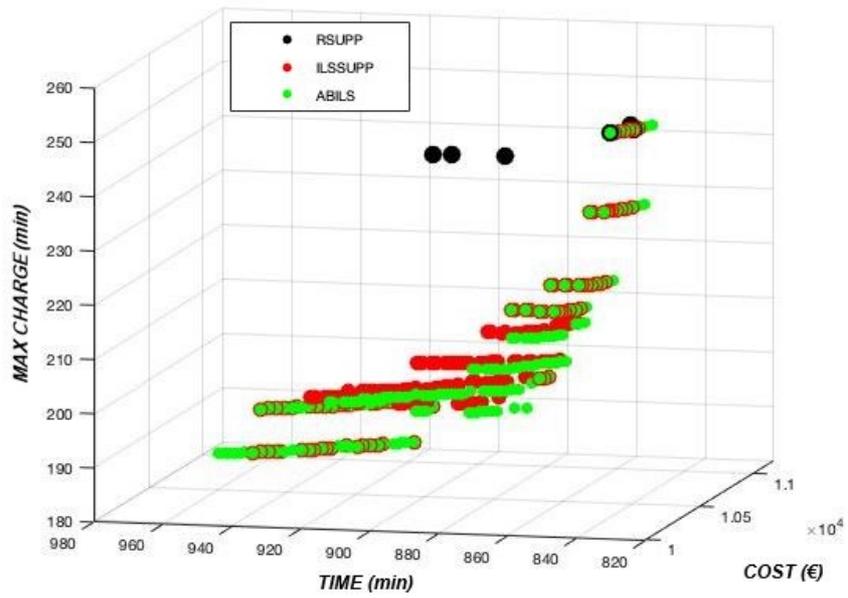Figure 4: Parento fronts of instance 10_5_1

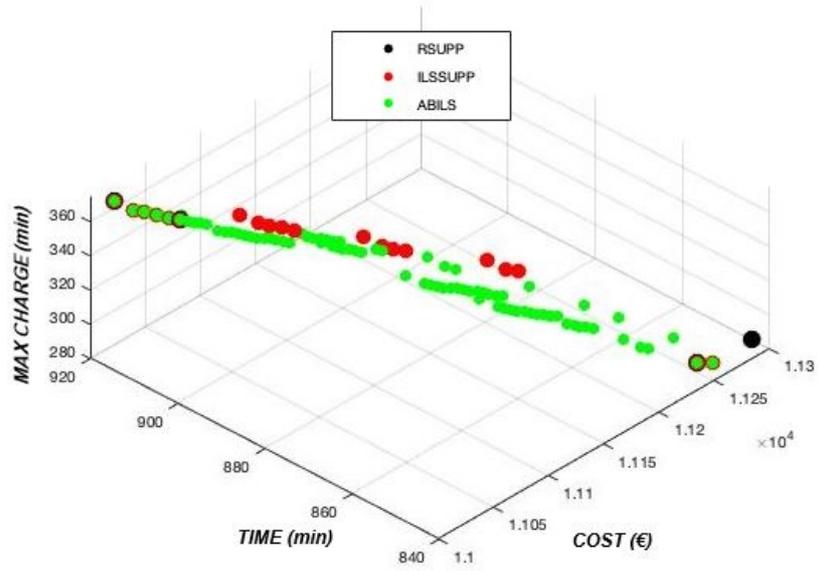
Figure 5: Parento fronts of instance 10_5_2

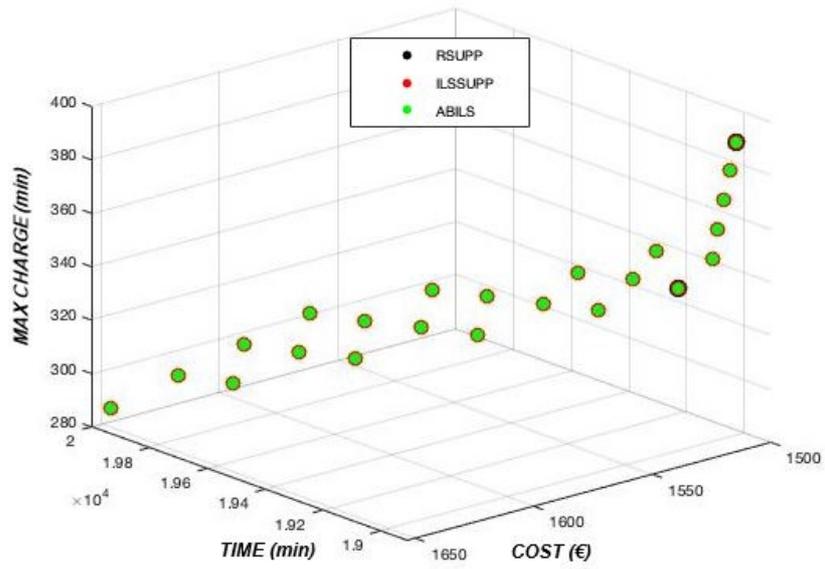Figure 6: Parento fronts of instance 10_5_5
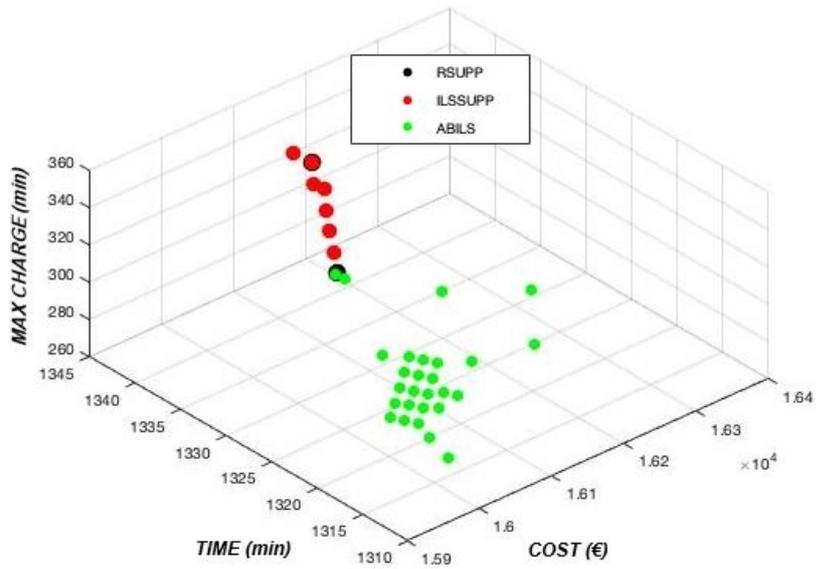

Figure 7: Parento fronts of instance 15_5_2
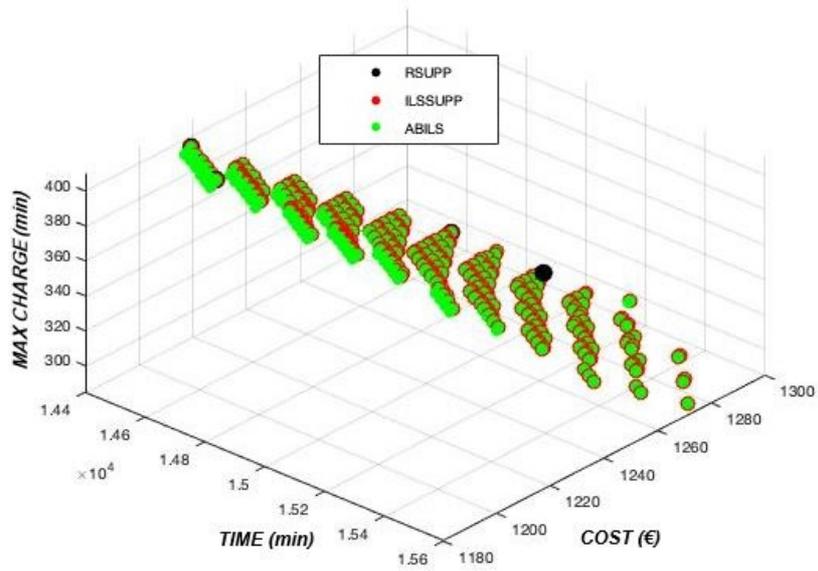
19

Figure 8: Parento fronts of instance 15_5_3
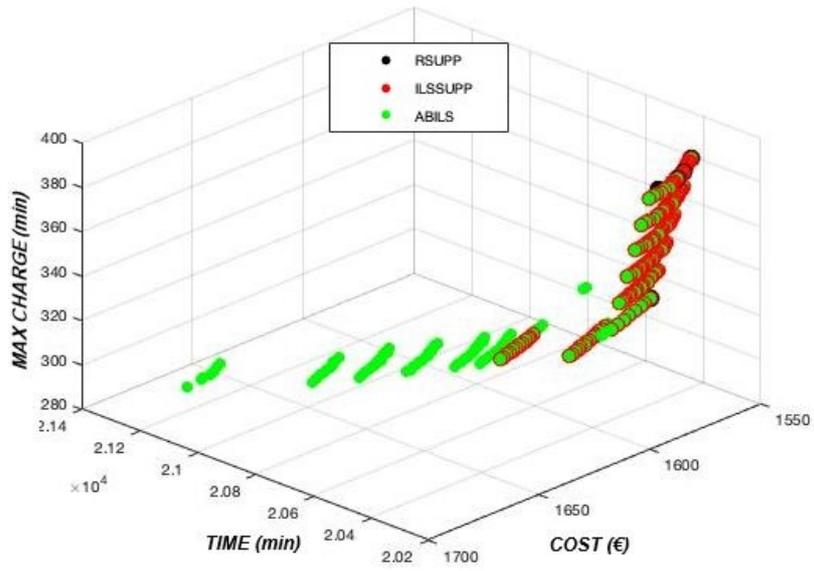

Figure 9: Parento fronts of instance 15_5_5

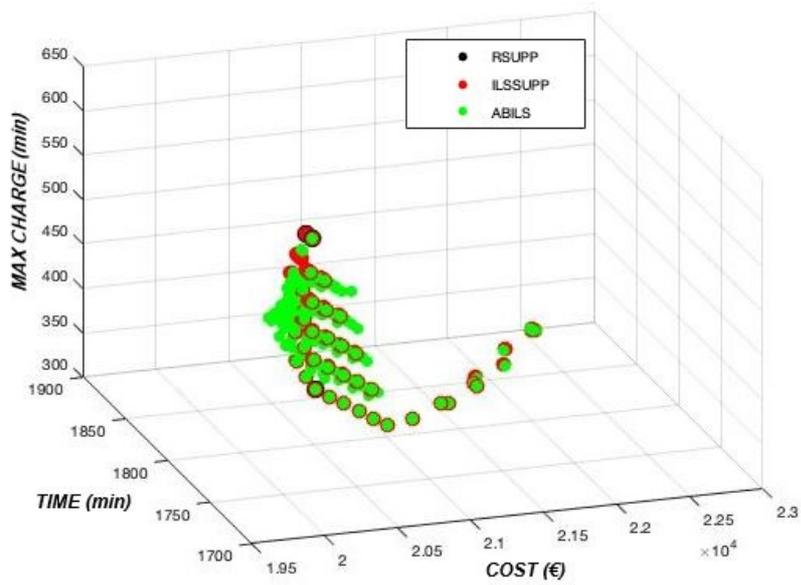Figure 10: Parento fronts of instance 20_5_1

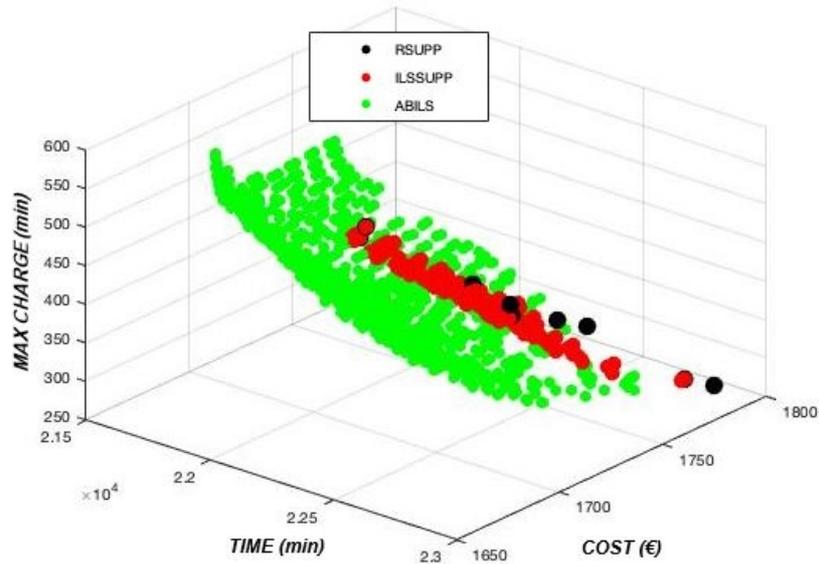Figure 11: Parento fronts of instance 20_5_3

Figure 12: Parento fronts of instance 20_5_4

7. Conclusions

Reconfigurable manufacturing system is a paradigm that answers many of the challenges that the market nowadays imposes. Moreover, RMS is designed to overcome the shortcomings of traditional manufacturing systems in a rapid and cost-effective way. In this paper, we presented and compared three hybrid metaheuristics for the multi-objective single-product multi-unit process plan generation in a reconfigurable manufacturing environment. Three criteria were considered, respectively, the total production cost, the completion time and the maximum exploitation time for machines. We presented a rich panel of experimental results and analyses to demonstrate the quality of the developed approaches. The obtained results are promising and can be viewed as a preliminary study for the integrated process plan and scheduling problem (IPPS) in reconfigurable manufacturing environment.

For future works, shortly, we expect to use other local search-based metaheuristics such as tabu search or an adapted version of the travelling salesman problem heuristic k-Opt instead of the adapted 2-opt. Furthermore, metaheuristics, such as ant colony optimization and particle swarm optimization, can be also used and compared to analyse and discuss possible results improvements for the considered problem. Finally, other criterion such as greenhouse gas (GHG) emission for sustainability purposes is a promising research direction.

References

Abbasi, M., & Houshmand, M. (2009). Production planning of reconfigurable manufacturing systems with stochastic demands using tabu search. *International Journal of Manufacturing Technology and Management*, *17*, 125–148.

Abbasi, M., & Houshmand, M. (2011). Production planning and performance optimization of reconfigurable manufacturing systems using genetic algorithm. *The International Journal of Advanced Manufacturing Technology*, *54*, 373–392.

Andersen, A.-L., Brunoe, T. D., & Nielsen, K. (2015). Reconfigurable manufacturing on multiple levels: literature review and research directions. In *IFIP International Conference on Advances in Production Management Systems* (pp. 266–273). Springer.

Azab, A., & ElMaraghy, H. (2007). Mathematical modeling for reconfigurable process planning. *CIRP Annals-Manufacturing Technology*, *56*, 467–472.

Babu, A. S. et al. (2013). Reconfigurations of manufacturing systems—an empirical study on concepts, research, and applications. *The International Journal of Advanced Manufacturing Technology*, *66*, 107–124.

Battaïa, O., Dolgui, A., & Guschinsky, N. (2017a). Decision support for design of reconfigurable rotary machining systems for family part production. *International Journal of Production Research*, *55*, 1368–1385.

Battaïa, O., Dolgui, A., & Guschinsky, N. (2017b). Integrated process planning and system configuration for mixed-model machining on rotary transfer machine. *International Journal of Computer Integrated Manufacturing*, *30*, 910–925.

Bensmaine, A., Dahane, M., & Benyoucef, L. (2013). A non-dominated sorting genetic algorithm based approach for optimal machines selection in reconfigurable manufacturing environment. *Computers & Industrial Engineering*, *66*, 519–524.

Bensmaine, A., Dahane, M., & Benyoucef, L. (2014). A new heuristic for integrated process planning and scheduling in reconfigurable manufacturing systems. *International Journal of Production Research*, *52*, 3583–3594.

Bi, Z. M., Lang, S. Y., Shen, W., & Wang, L. (2008). Reconfigurable manufacturing systems: the state of the art. *International Journal of Production Research*, *46*, 967–992.

Bortolini, M., Galizia, F. G., & Mora, C. (2018). Reconfigurable manufacturing systems: Literature review and research trend. *Journal of manufacturing systems*, *49*, 93–106.

Bruccoleri, M., Nigro, G. L., Perrone, G., Renna, P., & La Diega, S. N. (2005). Production planning in reconfigurable enterprises and reconfigurable production systems. *CIRP Annals-Manufacturing Technology*, *54*, 433–436.

Chaube, A., Benyoucef, L., & Tiwari, M. K. (2012). An adapted nsga-2 algorithm based dynamic process plan generation for a reconfigurable manufacturing system. *Journal of Intelligent Manufacturing*, *23*, 1141–1155.

Croes, G. A. (1958). A method for solving traveling-salesman problems. *Operations research*, *6*, 791–812.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, *6*, 182–197.

Dou, J., Li, J., & Su, C. (2016). Bi-objective optimization of integrating configuration generation and scheduling for reconfigurable flow lines using nsga-ii. *The International Journal of Advanced Manufacturing Technology*, *86*, 1945–1962.

ElMaraghy, H. A. (2007). Reconfigurable process plans for responsive manufacturing systems. In *Digital enterprise technology* (pp. 35–44). Springer.

ElMaraghy, H. A. (2008). *Changeable and reconfigurable manufacturing systems*. Springer Science & Business Media.

Gadalla, M., & Xue, D. (2018). An approach to identify the optimal configurations and reconfiguration processes for design of reconfigurable machine tools. *International Journal of Production Research*, *56*, 3880–3900.

Haddou Benderbal, H., Dahane, M., & Benyoucef, L. (2018). Modularity assessment in reconfigurable manufacturing system (rms) design: an archived multi-objective simulated annealing-based approach. *The International Journal of Advanced Manufacturing Technology*, *94*, 729–749.

Huang, S., Wang, G., & Yan, Y. (2019). Delayed reconfigurable manufacturing system. *International Journal of Production Research*, *57*, 2372–2391.

Jain, N., & Jain, V. (2001). Computer aided process planning for agile manufacturing environment. *Agile Manufacturing: The 21st Century Competitive Strategy, Elsevier, Oxford*, (pp. 515–534).

Koren, Y. (2006). General rms characteristics. comparison with dedicated and flexible systems. In *Reconfigurable manufacturing systems and transformable factories* (pp. 27–45). Springer.

Koren, Y. (2010). *The global manufacturing revolution: product-process-business integration and reconfigurable systems* volume 80. John Wiley & Sons.

Koren, Y., Heisel, U., Jovane, F., Moriwaki, T., Pritschow, G., Ulsoy, G., & Van Brussel, H. (1999). Reconfigurable manufacturing systems. *CIRP annals*, *48*, 527–540.

Maganha, I., Silva, C., & Ferreira, L. M. D. (2018). Understanding reconfigurability of manufacturing systems: An empirical analysis. *Journal of Manufacturing Systems*, *48*, 120–130.

Maniraj, M., Pakkirisamy, V., & Parthiban, P. (2014). Optimisation of process plans in reconfigurable manufacturing systems using ant colony technique. *International Journal of Enterprise Network Management*, *6*, 125–138.

Moghaddam, S. K., Houshmand, M., & Fatahi Valilai, O. (2018). Configuration design in scalable reconfigurable manufacturing systems (rms); a case of single-product flow line (spfl). *International Journal of Production Research*, *56*, 3932–3954.

Musharavati, F., & Hamouda, A. (2012a). Enhanced simulated-annealing-based algorithms and their applications to process planning in reconfigurable manufacturing systems. *Advances in Engineering Software*, *45*, 80–90.

Musharavati, F., & Hamouda, A. (2012b). Simulated annealing with auxiliary knowledge for process planning optimization in reconfigurable manufacturing. *Robotics and Computer-Integrated Manufacturing*, *28*, 113–131.

Nallakumarasamy, G., Srinivasan, P., Raja, K. V., & Malayalamurthi, R. (2011). Optimization of operation sequencing in capp using superhybrid genetic algorithms-simulated annealing technique. *ISRN Mechanical Engineering*, *2011*.

Prasad, D., & Jayswal, S. (2018). Reconfigurability consideration and scheduling of products in a manufacturing industry. *International Journal of Production Research*, *56*, 6430–6449.

Renzi, C., Leali, F., Cavazzuti, M., & Andrisano, A. (2014). A review on artificial intelligence applications to the optimal design of dedicated and reconfigurable manufacturing systems. *The International Journal of Advanced Manufacturing Technology*, *72*, 403–418.

Shabaka, A., & ElMaraghy, H. A. (2007). Generation of machine configurations based on product features. *International Journal of Computer Integrated Manufacturing*, *20*, 355–369.

Swamidass, P. M. (2000). *Encyclopedia of production and manufacturing management*. Springer Science & Business Media.

Touzout, F. A., & Benyoucef, L. (2018a). Multi-objective sustainable process plan generation in a reconfigurable manufacturing environment: exact and adapted evolutionary approaches. *International Journal of Production Research*, (pp. 1–17).

Touzout, F. A., & Benyoucef, L. (2018b). Sustainable multi-unit process plan generation in a reconfigurable manufacturing environment: A comparative study of three hybrid-meta-heuristics. In *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)* (pp. 661–668). IEEE volume 1.

Tseng, M. M., & Jiao, J. (2001). Mass customization, handbook of industrial engineering. *Technology and Operation Management*, .

WorldBankGroup (2016). World bank group, manufacturing, value added (% of gdp). URL: https://data.worldbank.org/indicator/NV.IND.MANF.ZS.

Xia, Q., Etienne, A., Dantan, J.-y., & Siadat, A. (2018). Reconfigurable machining process planning for part variety in new manufacturing paradigms: Definitions, models and framework. *Computers & Industrial Engineering*, *115*, 206–219.