

Available online at www.sciencedirect.com**ScienceDirect**

Procedia CIRP 33 (2015) 76 – 80

www.elsevier.com/locate/procedia

9th CIRP Conference on Intelligent Computation in Manufacturing Engineering - CIRP ICME '14

Modelling the problem of production scheduling for reconfigurable manufacturing systems

Ahmed Azab^{a*}, Bahman Naderi^b^aProduct Lifecycle Management Research Lab, Department of Industrial Engineering, University of Windsor, Windsor, Canada^bDepartment of Industrial Engineering, Faculty of Engineering, Kharazmi University, Karaj, Iran* Corresponding author. Tel.: +1-519-253-3000; fax: +0-000-000-0000. E-mail address: azab@uwindsor.ca

Abstract

Companies are gradually moving towards reconfigurable manufacturing systems (RMS) to achieve both changeable functionality of flexible manufacturing systems (FMS) and scalable capacity of dedicated manufacturing lines (DML) to the extent possible. Despite this expected trend in manufacturing, there is a dearth of literature on scheduling for RMS; papers in the production scheduling literature still mainly focus on either DML or FMS. This paper tackles the problem of scheduling production operations in RMS. After explicitly defining different aspects of the problem, a mathematical model is developed to formally model the problem. Using the model and commercial software of operations research, the small instances of the problem are solved for optimality. To effectively solve large instances of the problem, different simulated annealing metaheuristics are developed. Using numerical experiments, the model and simulated annealing algorithms are evaluated for performance.

© 2014 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license

[\(http://creativecommons.org/licenses/by-nc-nd/4.0/\)](http://creativecommons.org/licenses/by-nc-nd/4.0/).

Selection and peer-review under responsibility of the International Scientific Committee of “9th CIRP ICME Conference”

Keywords: Reconfigurable manufacturing systems, scheduling, mathematical modeling.

1. Introduction

Manufacturing is the transformation of raw material and input parts into finished products using resources such as machines, tools, energy and labor. The configurations and specifications of resources, so-called manufacturing systems, require to be developed so as to perform effectively to meet market's requirements. During the era of economy of scale, 1970s, enterprises aimed at cost advantages obtained due to size. Hence, manufacturing systems included some dedicated machines producing very low variety of products, yet high volume production. This type of manufacturing system is called dedicated manufacturing lines (DML). In 1980s, due to rapid changes in market' demand, the objective is switched from large scale of production for a single product to mass customization and greater responsiveness to changes in products. Therefore, manufacturing systems consisted of flexible machinery producing mid-volume and mid-variety production needs. This era was characterized and labeled as

economy of scope; the enablers for the era was numerical control, programmable logic control, robotics, automated guided vehicles and automated storage and retrieval systems; this compromised what we currently refer to as flexible manufacturing system (FMS).

But, FMS is still not satisfactory due to high cost of production and high cycle time (Koren et al., 1999) since machines are general purpose, not custom-made. They are also a very expensive paradigm, since FMS does provide a generalized flexibility; i.e., an FMS system would have all the functionality possibly needed built into the system ahead of time. They are built before process plan is done. As a result, a new generation of manufacturing system, called reconfigurable manufacturing system (RMS), is developed to yield both changeable functionality of FMS and scalable capacity of DML to extent possible. The key point to have RMS is to design the system and its machines for adjustable structures that enable scalability in both capacity and functionality. In RMS, configurations are designed around an

extended product family and customized flexibility required to produce all given extended product family. In other words, products are first grouped into subfamilies where each requires one configuration of the system. The system is configured to produce the first subfamily of products. Once it is completed, the system is reconfigured for producing the second subfamily, and so on and so forth. Figure 1 positions RMS versus FMS and DML regarding both product flexibility and scalability.

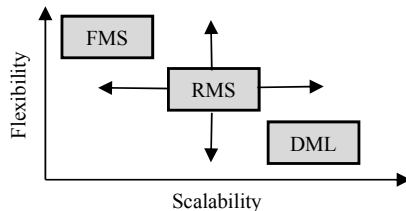


Figure 1. RMS versus FMS and DML regarding both product flexibility and scalability

The scheduling problem in RMS can be defined as follows. There is an extended product family grouped into subfamilies each of which includes several parts/jobs of the extended family. Each part consists of several operations with precedence relations specified. Each subfamily requires a predetermined configuration of system. To perform parts inside each subfamily, minor setups are required, while to switch to a different subfamily, the system has to be reconfigured. This major reconfiguration/setup is sequence dependent, i.e., the magnitude of setup depends on how similar two consecutive configurations/subfamilies are. On the other hand, the minor setup is sequence independent and consequently can be added to processing times.

This problem deals with two different sequencing decisions, sequencing subfamilies as well as parts inside each subfamily. The objective is to minimize makespan (i.e., maximum completion time of parts where a part is completed once all its operations are done). Additionally, we assume both operations (subfamilies) are non-preemptive. That is, when process of an operation (subfamily) starts, it cannot be interrupted before its completion. Additionally, we assume that machines are continuously available. Machine can process at most one operation at a time and each part can be processed by at most one machine at a time.

In brief, this paper considers the problem of scheduling jobs in reconfigurable manufacturing systems. It mathematically formulates the problem using a mixed integer linear programming model. The model is evaluated for both size and computational complexities. Using mathematical programming software of CPLEX, the problem is optimally solved.

The rest of the paper is organized as follows. Section 2 reviews the relevant literature. Section 3 defines and formulates the problem. Section 4 experimentally evaluates the model. Section 5 finally concludes the paper.

2. Literature review

RMS is an active field research for more than one decade thanks to the pioneer work by Koren et al. (1999). They introduce RMS as a solution for manufacturing systems to react to changes rapidly and cost-effectively. For a manufacturing system to be reconfigurable, six key characteristics are required, namely modularity, integrability, convertibility, diagnosability, customization and scalability. ElMaraghy (2006) compares RMS versus FMS. She recognizes in details customized flexibility in RMS versus generalized flexibility in FMS. Galan (2008) studies RMS to group products into families and to schedule the families. Several heuristics and metaheuristics (tabu search and ant colony optimization) are developed. He only considers scheduling of part families and ignores scheduling parts and their operations.

Meng (2010) proposes a model for the reconfiguring process of a manufacturing system by applying colored timed object-oriented Petri nets. This model integrates object-oriented methods, stepwise refinement ideas and Petri nets together. Abbasi and Houshmand (2011) study the utilization stage of an RMS and introduce a methodology to effectively adjust scalable production capacities and the system functionalities to market demands. They propose a mixed integer nonlinear programming model to determine lot sizes, corresponding configurations, and optimum arrangement of production tasks. A genetic algorithm-based procedure is developed to solve the model.

Azab and Gomaa (2011) consider operations sequencing in RMS to minimize changeover time/cost while satisfying a number of precedence constraints. They develop an integer model as well as a variant of the canonical genetic algorithms. Bensmaine et al. (2012) consider RMS (i.e., the selection of candidate reconfigurable machines among an available set) based on products specifications and reconfigurable machines capabilities. Two objectives are the minimization of the total cost and the total completion time. They propose an adapted version of the non-dominated sorting genetic algorithm to solve the problem. Guan et al. (2012) consider RMS to design the layout utilizing automated guided vehicle. The objective is minimization of total material handling cost. They propose a revised electromagnetism-like mechanism.

Musharavati and Hamouda (2012) study process planning problems in RMS. They develop simulated-annealing-based algorithms with other algorithm concepts such as knowledge exploitation and parallelism. Yu et al. (2013) consider scheduling problems in RMS with a limited number of fixtures. They assume that this problem is flexible job shop where each job has multiple process plans. They propose a practical priority rule based approach. Azab et al. (2013) study RMS to identify mechanisms and set points to reconfigure manufacturing systems by considering the supply and return of machines and system modules required for reconfiguration.

Although there are several papers studying RMS, very few papers consider scheduling problems in RMS. Even these few papers suffer from limitations. Galan (2008) only considers scheduling of part families and ignores scheduling of parts and

their operations. Yu et al. (2013) schedule flexible job shops with multi process plans as RMS.

3. Problem definition and formulation

In RMS, we have a set of n jobs and a set of m machines for processing. Based on processing route jobs are divided into c sets. Jobs within each set have the same processing route. To prepare machines for each set, machines are reconfigured. Therefore, we need c different configurations. The objective is to determine both configuration sequence and job sequence within each configuration to minimize makespan.

Looking at such a system, one could conclude the following. Since the jobs within each configuration have the same processing route, each configuration (i.e., job sequence) is a permutation flow shop problem. Note that we need to setup up the system for a new configuration (performing reconfiguration), where the magnitude of this setup depends on the two consecutive configurations (going from which configuration to which configurations). Reconfiguring from a configuration to a completely different one requires more extensive setup. Therefore, the configuration sequence is also a single machine problem with sequence dependent setup times.

All in all, the RMS problem actually is a combined problem of (permutation flow shop and single machine problems). Let us further describe the problem using an example. Consider a problem with $n=6$ and $m=2$. Jobs 1, 2 and 3 belong configuration 1 and Jobs 4, 5 and 6 belong configuration 2. The processing times are shown in Table 1. The sequence dependent setup times are shown in Table 2.

Recon.	Jobs	Machines	
		1	2
1	1	5	3
	2	9	4
	3	4	6
2	4	5	9
	5	3	7
	6	7	5

Conf.	Conf.	
	1	2
0	10	15
1	-	20
2	5	-

For this example there are $2!$ different configuration sequences, and $3!$ for job sequence within each configuration. It sums up to $2!(3!)^2$ different solutions. One possible solution is configuration sequence $\{2,1\}$; jobs in conf. 1 $\{3,1,2\}$; jobs in conf. 2 $\{4,6,5\}$. Figure 2 shows final Gantt chart of solution.

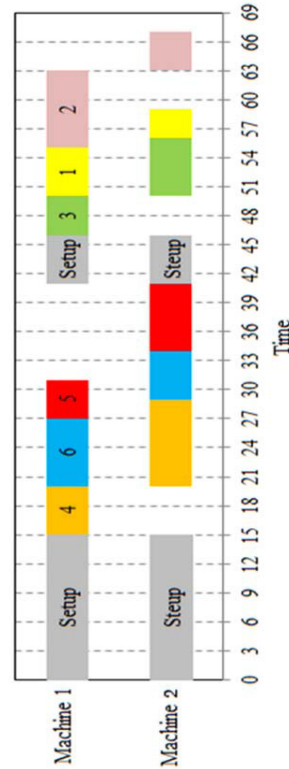


Figure 2. The final Gantt chart of the example

Before presenting the model, the following notations are considered. First, parameters and indexes are presented and then decision variables

Parameters and indexes

q	The number of job sets
r, l	Index for job sets $\{1, 2, \dots, q\}$
n_r	The number of job in set r
m	The number of machines
j, k	Index for jobs $\{1, 2, \dots, n_h\}$
i	Index for machines $\{1, 2, \dots, m\}$
$p_{r,j,i}$	The processing times of job j of set r on machine i
$s_{l,r}$	The setup times of set r after set l
M	A large positive number

Decision variables

$X_{r,k,j}$	Binary variable taking value 1 if job j of set r is processed after job k ; and 0 otherwise. $k = \{1, 2, \dots, n_r - 1\}, j > k$
$Y_{l,r}$	Binary variable taking value 1 if set r is processed immediately after set l ; and 0 otherwise. $l = \{0, 1, 2, \dots, q\}, r \neq l$
$C_{r,j,i}$	The continuous variable for completion time of job j of set r on machine i
F_r	The continuous variable for completion time of set r
B_r	The continuous variable for starting time of set r

The mathematical model is as follows.

Minimize C_{max}
 Subject to:

$$\sum_{l=0, l \neq r}^q Y_{l,r} = 1 \quad \forall_{r \in \{1, 2, \dots, q\}} \quad (1)$$

$$\sum_{r=1, r \neq l}^q Y_{l,r} \leq 1 \quad \forall_{l \in \{1, 2, \dots, q\}} \quad (2)$$

$$\sum_{r=1}^q Y_{0,r} = 1 \quad (3)$$

$$B_r \geq F_l + s_{l,r} - M(1 - Y_{l,r}) \quad \forall_{l \in \{0, 1, \dots, q\}, r \in \{1, 2, \dots, q\}, l \neq r} \quad (4)$$

$$C_{r,j,1} \geq B_r + p_{r,j,1} \quad \forall_{r \in \{1, 2, \dots, q\}, j \in \{1, 2, \dots, n_r\}} \quad (5)$$

$$C_{r,j,i} \geq C_{r,j,i-1} + p_{r,j,i} \quad \forall_{r \in \{1, 2, \dots, q\}, j \in \{1, 2, \dots, n_r\}, i \in \{2, 3, \dots, m\}} \quad (6)$$

$$C_{r,j,i} \geq C_{r,k,i} + p_{r,j,i} - M(1 - X_{r,k,j}) \quad \forall_{r \in \{1, 2, \dots, q\}, k \in \{1, \dots, n_r - 1\}, j \in \{k+1, \dots, n_r\}, i \in \{1, 2, \dots, m\}} \quad (7)$$

$$C_{r,k,i} \geq C_{r,j,i} + p_{r,k,i} - M(X_{r,k,j}) \quad \forall_{r \in \{1, 2, \dots, q\}, k \in \{1, \dots, n_r - 1\}, j \in \{k+1, \dots, n_r\}, i \in \{1, 2, \dots, m\}} \quad (8)$$

$$F_r \geq C_{r,j,m} \quad \forall_{r \in \{1, 2, \dots, q\}, j \in \{1, 2, \dots, n_r\}} \quad (9)$$

$$C_{max} \geq F_r \quad \forall_{r \in \{1, 2, \dots, q\}} \quad (10)$$

$$C_{r,j,i}, B_r, F_r \geq 0 \quad \forall_{r \in \{1, 2, \dots, q\}, j \in \{1, 2, \dots, n_r\}, i \in \{1, 2, \dots, m\}} \quad (11)$$

$$X_{r,k,j} \in \{0, 1\} \quad \forall_{r \in \{1, 2, \dots, q\}, k \in \{1, 2, \dots, n_r - 1\}, j \in \{k+1, \dots, n_r\}} \quad (12)$$

$$Y_{f,r} \in \{0, 1\} \quad \forall_{f \in \{0, 1, \dots, q\}, r \in \{1, 2, \dots, q\}, f \neq r} \quad (13)$$

Constraint sets (1-3) are to determine the configuration sequence. Constraint set (4) specifies the starting time of each configuration (after performing the sequence dependent setup). Note that $F_0 = 0$. Constraint sets (5-8) are to determine the completion time of jobs within each configuration. Constraint set (5) ensures that each job is processed after starting time of its corresponding configuration. Constraint set (6) assures that each job is processed by at most one machine at a time. Constraint sets (7-8) ensures that each machine can process at most one job at a time. Constraint set (9) is to determine the finishing time of each configuration. Constraint set (10) specifies makespan. Constraint sets (11-13) define the decision variables.

4. Experimental evaluation

This section evaluates the model for both size and computational complexities. The size complexity counts the number of continuous and binary variables as well as the number of constraints the model needs to formulate a problem with q configurations, n jobs within each configuration (note that we assume there is the same number of jobs within each configuration), m machines. The computational complexity estimates the time the model needs to solve the problem with q, n and m using mathematical programming software.

The size complexity of the model is presented in Table 3. Regarding the number of binary variables, the model is quadratic in both q and n. Yet, it is independent of m. Regarding the number of continuous variables, the model is linear in all q, n and m. Regarding the number of constraints, the model is quadratic in both q and n while linear in m. The

first conclusion is that the complexity of the model comes from both q and n, while m has linear impact on size complexity.

Table 3. The size complexity of the model

Factor	Quantity
Binary variables	$q^2 + q \frac{n(n-1)}{2}$
Continuous variables	qnm
Constraints	$3q + q^2 + qn(1 + nm) + 1$

To evaluate the computational complexity, a numerical experiment with 12 instances generated is designed as follows.

$$q = \{3, 4, 5\}, n = \{4, 6\}, m = \{3, 5\}$$

The processing and setup times are generated from uniform distributions of [5 50] and [5 100], respectively. Table 4 shows the results. The model can solve all instances up to $q + n = 9$ (i.e., $nq = 20$ jobs). It obtains the average optimality gap of 13.5% for instances with $q + n = 10$, and 22.5% for instances with $q + n = 11$.

Table 4. Computational time of the model

q	n	m	Cmax	Time (sec)	Gap (%)
3	4	3	566	2.11	0
3	4	5	837	2.53	0
3	6	3	714	455	0
3	6	5	813	188	0
4	4	3	771	5.79	0
4	4	5	932	10	0
4	6	3	786	2000	12%
4	6	5	1013	2000	15%
5	4	3	964	1942	0
5	4	5	1261	165	0
5	6	3	899	2000	22%
5	6	5	1135	2000	23%

5. Conclusion

Since both DML and FMS have serious shortcomings attributed to their lack of support for variety of product mix and their high production cost, designing a new manufacturing system with scalable flexibility/functionality and capacity is needed. The solution to this need is RMS.

This paper studies problem of scheduling of operations in reconfigurable manufacturing systems (RMSs). Although it is inevitable that the modern industry need to approach the RMS concept, no paper has comprehensively studied how to schedule operations in such a complex system. In this problem, we assume that we have a manufacturing system with a set of machines. The system is designed to manufacture an extended product family by splitting that wide range of product variety into several product subfamilies. That is, although specified for an extended product family, the system requires reconfiguration to switch from one subfamily to another. To change one configuration to another, some setup must be done. The magnitude of setup highly depends on the two consecutive configurations/subfamilies. The system

within each configuration is a flow shop. The objective is to both schedule subfamilies and products within each subfamily so as to minimize makespan.

As an early attempt on studying this problem, we first mathematically formulate the problem considering both family sequencing and operations sequencing inside each family since there is no mathematical model for the problem. By this model, we can formulate all different aspects of the problem. Using advance OR software such as CPLEX, we can optimally solve the small sized problems.

References

- Abbasi M., Houshmand M., (2011). Production planning and performance optimization of reconfigurable manufacturing systems using genetic algorithm, *International Journal of Advanced Manufacturing Technology* 54, 373–392.
- Azab A., ElMaraghy H., Nyhuis P., Pachow-Frauenhofer J., Schmidt M., (2013). Mechanics of change: A framework to reconfigure manufacturing systems, *CIRP Journal of Manufacturing Science and Technology* 6, 110–119.
- Azab A., Gomaa A.H., (2011). Optimal Sequencing of Machining Operations for Changeable Manufacturing. 4th International Conference on Changeable, Agile, Reconfigurable and Virtual Production (CARV2011), Montreal, Canada.
- Bensmaine A., Dahane M., Benyoucef L., (2013). A non-dominated sorting genetic algorithm based approach for optimal machines selection in reconfigurable manufacturing environment. *Computers and Industrial Engineering*, 66(3), 519-524.
- ElMaraghy, H.A. (2006). Flexible and reconfigurable manufacturing systems paradigms. *International Journal of Flexible Manufacturing Systems*. 17(4), 261-276.
- Galan R., (2008). Hybrid heuristic approaches for scheduling in reconfigurable manufacturing systems, *Studies in Computational Intelligence*, 128, 211–253.
- Koren, Y., Heisel U., Jovane F., Moriwaki T., Pritschow G., Ulsoy G., Brussel H.V. (1999). Reconfigurable manufacturing systems. *CIRP Annals – Manufacturing Technology*, 48(2), 527–540.
- Meng X., (2010). Modeling of reconfigurable manufacturing systems based on colored timed object-oriented Petri nets. *Journal of Manufacturing Systems* 29 (2010) 81–90.
- Musharavati F., Hamouda A.S.M., (2012). Enhanced simulated-annealing-based algorithms and their applications to process planning in reconfigurable manufacturing systems, *Advances in Engineering Software*, 45, 80–90.
- Yu J.M., Doh H.H., Kim J.S., Kwon Y.J., Lee D.H., Nam S.H., (2013). Input sequencing and scheduling for a reconfigurable manufacturing system with a limited number of fixtures. *International Journal of Advanced Manufacturing Technology*, 67, 157–169.