



# Reconfigurable Manufacturing System Design Using a Genetic Algorithm

Marco Bortolini, Cristian Cafarella, Emilio Ferrari, Francesco Gabriele Galizia<sup>✉</sup>,  
and Mauro Gamberi

Alma Mater Studiorum - University of Bologna, Viale del Risorgimento 2, 40136 Bologna, Italy  
francesco.galizia3@unibo.it

**Abstract.** In the modern context governed by Industry 4.0, Reconfigurable Manufacturing Systems (RMSs) rose as an effective production strategy able to cope with the increased product variety, the dynamic market demand and the need for flexible production batches. The manufacturing environment is usually made of a set of intelligent machines, i.e. Reconfigurable Machine Tools (RMTs), consisting of basic and auxiliary modules, which allow performing different operations. In this context, this paper proposes an optimization model for the dynamic design of RMSs with alternative part routing and multiple time periods, aiming at determining the part routing mix and the auxiliary module allocation best balancing the part flows among RMTs and the effort to install the modules on the machines. The model is solved through the application of a genetic algorithm applying different crossover operators and different threshold values for the occurrence of crossover and mutation processes. Results from the considered instance highlight that the two point crossover operator allows achieving the lowest fitness value, i.e. the lowest value of the defined objective function, getting a manufacturing system configuration characterized by low inter-cell part flow and machine reconfiguration time.

**Keywords:** Reconfigurable manufacturing systems · Reconfigurability · RMS · Genetic algorithm · Optimization model

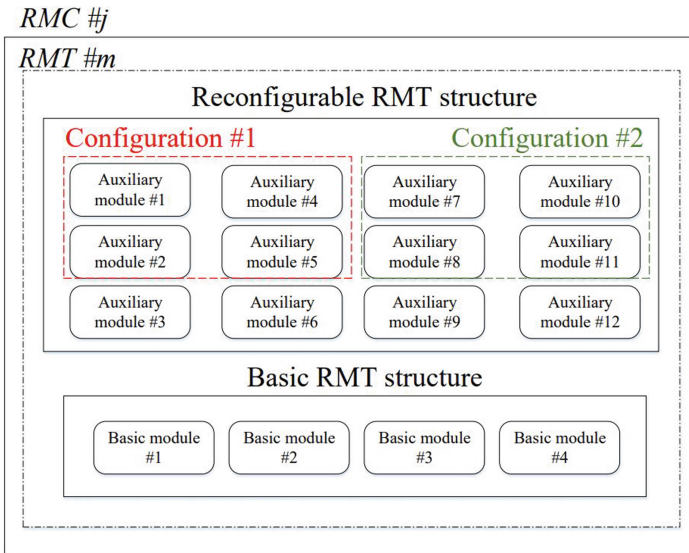
## 1 Introduction and Literature Review

In modern industry, manufacturers are facing a high level of market globalisation, increased product innovation and variety, dynamic customer demand and technological advancements [1, 2]. These trends encourage industrial companies to adopt the mass customisation paradigm to meet every customers' request and satisfy their individual needs. In this dynamic and changeable scenario, reconfigurability is one of the major enablers of changeability and, from the Industry 4.0 perspective, it is an essential element to cope with the ever-increasing complexity of the modern industrial and market scenario [3–5]. Reconfigurable Manufacturing Systems (RMSs) rose in 1999 as a new production system paradigm including changeability attributes at both physical and logical levels [6, 7] joining their core features of modularity, integrability, diagnosibility, convertibility,

customization and scalability [4]. A typical RMS structure includes a set of intelligent machines called reconfigurable machine tools (RMTs) with an adjustable and modular structure through a set of basic and auxiliary custom modules, which allow to increase the set of feasible operations to perform [2, 8, 9]. In particular, each RMT incorporates a number of basic modules that are structural elements permanently attached to the machine, and a number of auxiliary modules, which are kinematical or motion-giving. Therefore, a specific combination of such modules provides a particular set of operational capabilities to the RMT. In current literature, a wide set of studies concerning optimization models for RMS design and management has been developed [10–12]. Youssef and ElMaraghy [13, 14] defined a novel algorithm supporting the RMS configuration selection with the goal to find the most suitable configurations for the different demand scenarios over the considered time horizon and to select those that allow minimizing the reconfiguration effort. In the same field, Moghaddam et al. [15] faced the RMS configuration design in presence of dynamic market demand. In such a context, the production system configuration needs to vary accordingly to the demand data at the minimum cost. To face this issue, the Authors developed a mixed integer linear programming formulation to manage the first manufacturing system configuration design as well as the further required configurations according to the dynamic demand rate. Goyal et al. [16] defined a multi-objective model to estimate the reconfigurability potential and task capability of RMTs according to the auxiliary module interactions. Moreover, the proposed mathematical algorithm supported the optimal part-machine assignment in case of a single part flow line allowing the parallel working of similar RMTs. Another wide group of researchers proposed to arrange RMSs in cellular production patterns, leading to the rise of Cellular Reconfigurable Manufacturing Systems (CRMSs) [9, 17, 18]. In conventional cellular manufacturing systems (CMSs), the formation of the manufacturing cells is an activity traditionally performed during the initial setup of the CMS and the layout does not change during the production life cycle. However, the recent trends imposed by Industry 4.0, e.g. mass production, dynamic market demand, etc., make CMSs obsolete because of the manufacturing cells may need to vary their structure throughout the production life cycle. To face this issue, recent studies suggest introducing the modularity attribute in the design of the manufacturing machines to include in the cells, enabling reconfigurability [8, 9]. In this field, Pattanaik et al. [8] proposed a clustering-based approach to design reconfigurable machine cells through adjustable machines. Eguia et al. [19, 20] defined a mathematical optimization model for the design of CRMSs aiming at minimizing the total inter-cell part movements and the overall production costs. According to this background, this paper proposes an optimization model for the dynamic design of CRMSs, best balancing the trade-off between the effort to reconfiguring the RMT hosting the part, in terms of auxiliary module installation and disassembly, versus the inter-cell part travel flows. The model is, then, applied to a numeric operative case study and solved by applying a genetic algorithm (GA). According to this background and the outlined goals, the remainder of this paper is organized as follows. Section 2 introduces the optimization model for CRMSs design while the description of the solving procedure is in Sect. 3. The application of the model to the operative reference case study and the results discussion are in Sect. 4. Finally, Sect. 5 concludes the paper with final remarks and future opportunities for research.

## 2 Optimization Model for the Design and Management of CRMSs

The aim of this section is to introduce and describe the optimization model for the dynamic design of CRMSs. The production context is made of a set of Reconfigurable Machine Cells (RMC) including a number of RMTs. Each RMT is characterized by a library of basic and auxiliary modules. As described in Sect. 1, the basic modules are structural elements permanently attached to the machines, while auxiliary modules are dynamic entities, which can be assembled and disassembled to/from the RMT when needed to provide different operational capabilities. Next Fig. 1 shows a conceptual framework of a typical CRMS structure, derived from [21].



**Fig. 1.** Schematic of a cellular RMS structure, derived from [21].

### 2.1 Problem Description, Assumptions and Notations

The proposed CRMS model relies on an initial RMT-RMC assignment and explores how to best-balance the reconfigurability effort, i.e. assembly and disassembly of the auxiliary modules on/from the RMTs, and the part flow among the RMCs, by using the available information about the operation sequence and the compatibility among the auxiliary modules, operations and RMTs. To this aim, the model minimizes the sum of the inter-cell, i.e. inter-RMC, parts travel time and the reconfiguration time to assemble and disassemble the auxiliary modules defining the part batch flows and the most suitable allocation of the modules to the RMTs. For the sake of brevity, in the following, analytic details about model indices, parameters, variables and the objective function formulation will be provided; while the complete formulation of the logical constraints is omitted.

The following notations are used.

- Indices

- $i$  parts  $i = 1, \dots, M$   
 $j$  RMCs  $j = 1, \dots, N$   
 $k$  modules type  $k = 1, \dots, K$   
 $m$  RMTs  $m = 1, \dots, Z$   
 $o$  operations in part work cycle  $o = 1, \dots, O_i$   
 $t$  time periods  $t = 1, \dots, T$

- Parameters

- $G_{omk}$  1 if operation  $o$  can be performed on RMT  $m$  using an auxiliary module of type  $k$ ; 0 otherwise [binary]  
 $MAC_{mj}$  1 if RMT  $m$  is assigned to RMC  $j$ ; 0 otherwise [binary]  
 $R$  maximum number of modules per RMT and period [#]  
 $r_{it}$  definition of the operation in which the batch of part  $i$  is in period  $t$   
 $t_{ijj_1}$  travel time for batch of part  $i$  from cell  $j$  to cell  $j_1$  [min/batch]  
 $\lambda_{mk}$  assembly time of module type  $k$  on RMT  $m$  [min/module]  
 $\mu_{mk}$  disassembly time of module type  $k$  from RMT  $m$  [min/module]  
 $\tau_{om}$  time to perform operation  $o$  on RMT  $m$  [min/op]  
 $\xi$  available time per RMT [min/machine]  
 $\delta_i$  planned production volume during a predefined period of time for part  $i$  [parts]

- Decisional Variables

- $F_{ijj_1t}$  1 if batch of part  $i$  moves from RMC  $j$  to RMC  $j_1$  in period  $t$ ; 0 otherwise [binary]  
 $W_{mit}$  1 if batch of part  $i$  is processed by RMT  $m$  in period  $t$ ; 0 otherwise [binary]  
 $\sigma_{mkt}$  1 if module type  $k$  is on RMT  $m$  in period  $t$ , 0 otherwise [binary]  
 $X_{mkt}$  1 if module type  $k$  is assembled on RMT  $m$  in period  $t$ , 0 otherwise [binary]  
 $Y_{mkt}$  1 if module type  $k$  is disassembled from RMT  $m$  in period  $t$ , 0 otherwise [binary]

- Objective function

$\min \psi$  Part travel time and module assembly/disassembly time [min], as in (1)

$$\psi = \sum_{t=1}^T \sum_{m=1}^Z \sum_{k=1}^K X_{mkt} \cdot \lambda_{mk} + \sum_{t=1}^T \sum_{m=1}^Z \sum_{k=1}^K Y_{mkt} \cdot \mu_{mk} + \sum_{i=1}^M \sum_{j=1}^N \sum_{j_1=1}^N \sum_{t=1}^{T-1} F_{ijj_1t} \cdot t_{ijj_1} \quad (1)$$

The first and the second terms are for the module assembly and disassembly time on/from RMTs, respectively, while the third term is for the part travel time. Next Sect. 3 introduces the procedure used to solve the model, based on GA.

### 3 Solving Procedure

The goal of the model is to determine the part batch flow among RMTs and RMCs as well as the best allocation of the auxiliary modules to the RMTs for part processing, considering the part work cycle and the compatibility information among auxiliary modules, operations and RMTs. Due to the model complexity, a heuristic method, i.e. GA, is chosen as solving method. GA relies on the concept of evolutionary computation reproducing the natural selection and biological reproduction of animal species. In fact, it originates from Darwin's "survival of the fittest" concept, meaning that a good parent produces better offspring, and it has been successfully applied over the time for flexible job-shop and flow-shop scheduling [22]. Prior to its application, GA requires to design the genetic representation, e.g. chromosome, of the candidate solutions. A chromosome represents each solution in the initial solution set of the population and it evolves through a crossover and a mutation operator to produce offspring, with the aim to improve the current set of solutions. The chromosomes are then evaluated through a fitness function, and the less fit chromosomes are replaced with better children. Such process of crossover, evaluation and selection is repeated for a number of iterations, usually up to the point in which the system ceases to improve. To summarize, Table 1 lists the main steps to follow for GA implementation to determine the optimum or near to optimum manufacturing configurations.

**Table 1.** Steps for GA implementation.

<i>Genetic algorithm (GA) implementation steps</i>
Step A: Parent selection
Step B: Crossover
Step C: Mutation
Step D: Fitness evaluation
Step E: Termination criterion

In this study, the step A, i.e. parent selection, is implemented according to the *roulette* method. Therefore, the probability of selecting a string is closely related to its fitness value. As reference example, in case of objective function to minimize, the lower the fitness of a string, the higher the probability that it will be selected as a parent. The step B, i.e. crossover, aims at combining the genetic information of the parents to generate new offspring. Initially, the algorithm defines a random value in the range 0–1, called *cross*, which is then compared to a value, called *pcross*, which corresponds to the probability of implementing a crossover operator and its value is set by the user. If the relation

$cross < p_{cross}$  is verified, the crossover process will occur. Therefore, the higher the value of  $p_{cross}$ , the more strings generated will be the result of a genetic exchange. Several crossover operators exist: the most widespread as well as those applied in this study are the *one point* crossover, the *two point* crossover and the *uniform* crossover. The step C, i.e. mutation, is performed to maintain genetic diversity from one generation of a population of GA chromosomes to the next. The probability of occurrence of this operator is given by the value  $p_{mut}$ . Traditionally, this variable assumes low values as it expresses the probability that errors can occur during the genetic exchange. Therefore, for each allele, the system will define a random value in the range 0–1, called  $mut$ , which is compared to  $p_{mut}$ . If the relation  $mut \leq p_{mut}$  is verified, the mutation operator will occur, i.e. the allele value changes from 0 to 1 and vice-versa. Once concluded the mutation phase, the evaluation of the generated strings occurs, i.e. step D. In particular, it is verified that each child satisfies all the model constraints: if even one constraint is not satisfied, the string is automatically discarded, otherwise the process moves toward the fitness function evaluation according to Eq. (1). The algorithm ends, i.e. step E, once a specific number of strings has been generated.

## 4 Case Study

In this section, a numeric operative case study is presented to evaluate the efficiency of the proposed model and its solving procedure. The instance considers the manufacturing of 5 products through a global set of 6 tasks. Moreover, the production environment includes 3 RMCs and 5 RMTs, i.e. RMT #1 and RMT #4 in RMC #1, RMT #2 in RMC #2 and RMT #3 and RMT #5 in RMC #3, while the equipment library has a set of 5 auxiliary module types. The compatibility matrix among tasks, RMTs and auxiliary modules is in Table 2.

**Table 2.** Compatibility data among tasks, RMTs and modules.

Tasks (o)	(Auxiliary modules) - [unitary processing times in minutes]				
	RMT #1	RMT #2	RMT #3	RMT #4	RMT #5
1	(1, 3) - [0.012]		(1) - [0.007]		
2		(1) - [0.01]			(1, 4) - [0.011]
3			(2, 5) - [0.008]		
4				(4) - [0.009]	(4, 5) - [0.009]
5		(2, 5) - [0.012]		(2, 5) - [0.011]	
6	(3, 4) - [0.007]				

This matrix shows the task execution modes, i.e. the RMT/RMTs needed for their processing, the required modules (in round brackets) and the unitary processing times (in squared brackets). Additional data about part work cycles, daily production volumes and auxiliary modules assembly and disassembly time are not detailed for the sake of

brevity. Other relevant data concern the parameter  $R$ , set to a value equal to 6 units, and parameter  $T$  equal to 120 periods.

#### 4.1 Genetic Algorithm Application

The GA algorithm used to solve and validate the proposed model is implemented in Microsoft Excel software using the Visual Basic for Applications (VBA) tool. The procedure starts with the generation of five parent strings considering the most relevant variable, i.e.  $W_{mit}$ , which specifies the RMT on which the part is located in each time period. Each parent has 3'000 binary values, i.e.  $5 \text{ parts} \times 5 \text{ RMT} \times 120 \text{ time periods}$ , and because of each part has to be processed by one RMT in each time period, 600 alleles of each string will take the value 1 while the remaining the value 0. Next Table 3 lists the five parents, satisfying all the model constraints, and their fitness value.

**Table 3.** Parent selection.

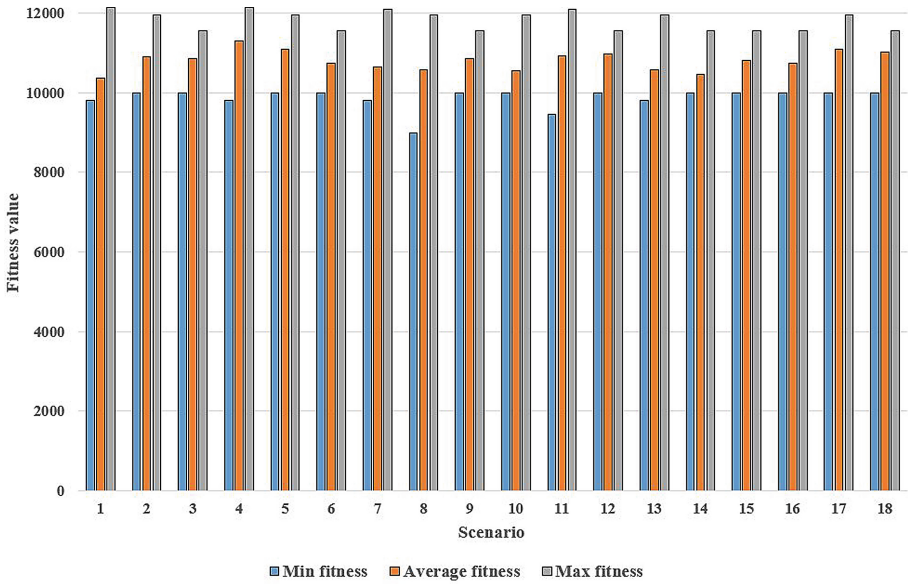
String	Fitness value [min]
Parent 1: $W_{mit1}$	10860.12
Parent 2: $W_{mit2}$	10826.18
Parent 3: $W_{mit3}$	10432.82
Parent 4: $W_{mit4}$	11556.42
Parent 5: $W_{mit5}$	9988.17

Once defined the parents, the algorithm selects two of them among the five available. Then, the crossover operator takes place, considering as values of  $pcross$  0.90, 0.95 and 0.98 and the one point, two point and uniform as crossover operators. The mutation phase follows the crossover, performed setting as values of  $pmut$  0.0002, 0.0001 and 0.002. Once these steps have been completed, the algorithm verifies that the two children satisfy the model constraints. In case of success, the child fitness is evaluated and it will be included in the set of the available parents. Finally, the algorithm checks whether the generated string is the hundredth, i.e. termination criterion: if not, it proceeds generating other offspring and repeating the process, otherwise it ends.

#### 4.2 Experimental Results and Discussion

A multi-scenario analysis is performed varying, in each scenario, the crossover operator method and the  $pcross$  and  $pmut$  values according to the data discussed in Sect. 4.1, getting a total of 18 scenarios. Aggregated results are in next Fig. 2.

Results mark that the best scenario, in terms of lower fitness value, is the ID. 8, with a fitness value equal to 8995.91 min. Such scenario corresponds to the application of the two point crossover operator, while the selected  $pcross$  and  $pmut$  values are 0.95 and 0.0001, respectively. Conversely, the scenario characterized by the highest fitness is the ID. 1, with a fitness value equal to 12128.82 min. Such scenario corresponds to



**Fig. 2.** Multi-scenario analysis results.

the application of the one point crossover operator, while the selected  $pcross$  and  $pmut$  values are 0.95 and 0.0002, respectively. Indeed, moving from such two scenarios, the fitness undergoes an increase of about 34%. Another relevant aspect to highlight is that the minimum and maximum fitness values are the same for all the scenarios in which the uniform crossover has been applied, i.e. scenario ID. 3, 6, 9, 12, 15, 18 with minimum fitness equal to 9988.17 min and max fitness equal to 11556.42 min. Moreover, such values correspond to the fitness of the parents. Therefore, the uniform method does not generate offspring with lower fitness values than the initial ones. Additional scenarios need to be assessed in the future, considering a wider set of  $pcross$  and  $pmut$  values as well as a more complex instance.

## 5 Conclusions and Future Research

In the last years, Reconfigurable Manufacturing Systems (RMS) emerged as an efficient manufacturing solution able to cope with the emerging industrial and market trends, e.g. dynamic market demand, short product life cycles, and flexible batches. To reach this goal, such systems use intelligent machines made by fixed modules and auxiliary custom modules, which can be assembled and disassembled when needed to provide different operational opportunities. In this scenario, this paper proposes a mathematical optimization model for the dynamic design of Cellular Reconfigurable Manufacturing Systems (CRMS) with the aim to best-balance the trade-off between the effort to reconfigure the manufacturing machine hosting the parts, in terms of auxiliary module assembly and disassembly, versus the inter-cell part travel flows. The model is, then, applied to an



operative case study and solved by applying a genetic algorithm. Moreover, a multi-scenario analysis is performed applying different crossover operators, i.e. one point, two point and uniform, and different threshold values for the occurrence of crossover and mutation processes. Results from the considered instance highlight that the two point crossover operator allows achieving the lowest fitness value, i.e. the lowest value of the defined objective function. Future research deal with the inclusion of other relevant dimensions in the model formulation, e.g. economic, and the application of the model to larger instances.

## References

1. Shou, Y., Li, Y., Park, Y.W., Kang, M.: The impact of product complexity and variety on supply chain integration. *Int. J. Phys. Distrib. Logist. Manag.* **47**(4), 297–317 (2017)
2. Bortolini, M., Galizia, F.G., Mora, C.: Reconfigurable manufacturing systems: literature review and research trend. *J. Manuf. Syst.* **49**, 93–106 (2018)
3. Mehrabi, M.G., Ulsoy, A.G., Koren, Y., Heytler, P.: Trends and perspectives in flexible and reconfigurable manufacturing systems. *J. Intell. Manuf.* **13**(2), 135–146 (2002). <https://doi.org/10.1023/A:1014536330551>
4. Koren, Y., Shpitalni, M.: Design of reconfigurable manufacturing systems. *J. Manuf. Syst.* **29**(4), 130–141 (2010)
5. Biswas, P., Kumar, S., Jain, V., Chandra, C.: Measuring supply chain reconfigurability using integrated and deterministic assessment models. *J. Manuf. Syst.* **52**, 172–183 (2019)
6. Molina, A., et al.: Next-generation manufacturing systems: key research issues in developing and integrating reconfigurable and intelligent machines. *Int. J. Comput. Integr. Manuf.* **18**(7), 525–536 (2005)
7. Esmailian, B., Behdad, S., Wang, B.: The evolution and future of manufacturing: a review. *J. Manuf. Syst.* **39**, 79–100 (2016)
8. Pattanaik, L.N., Jain, P.K., Mehta, N.K.: Cell formation in the presence of reconfigurable machines. *Int. J. Adv. Manuf. Technol.* **34**(3–4), 335–345 (2007). <https://doi.org/10.1007/s00170-006-0592-5>
9. Eguia, I., Molina, J.C., Lozano, S., Racero, J.: Cell design and multi-period machine loading in cellular reconfigurable manufacturing systems with alternative routing. *Int. J. Prod. Res.* **55**(10), 2775–2790 (2017)
10. Azevedo, M.M., Crispim, J.A., de Sousa, J.P.: A dynamic multi-objective approach for the reconfigurable multi-facility layout problem. *J. Manuf. Syst.* **42**, 140–152 (2017)
11. Ashraf, M., Hasan, F.: Configuration selection for a reconfigurable manufacturing flow line involving part production with operation constraints. *Int. J. Adv. Manuf. Technol.* **98**(5–8), 2137–2156 (2018). <https://doi.org/10.1007/s00170-018-2361-7>
12. Brahimi, N., Dolgui, A., Gurevsky, E., Yelles-Chaouche, A.R.: A literature review of optimization problems for reconfigurable manufacturing systems. *IFAC-PapersOnLine* **52**(13), 433–438 (2019)
13. Youssef, A.M.A., ElMaraghy, H.A.: Optimal configuration selection for reconfigurable manufacturing systems. *Int. J. Flex. Manuf. Syst.* **19**(2), 67–106 (2007). <https://doi.org/10.1007/s10696-007-9020-x>
14. Youssef, A.M., ElMaraghy, H.A.: Availability consideration in the optimal selection of multiple-aspect RMS configurations. *Int. J. Prod. Res.* **46**(21), 5849–5882 (2008)
15. Moghaddam, S.K., Houshmand, M., Fatahi Valilai, O.: Configuration design in scalable reconfigurable manufacturing systems (RMS); a case of single-product flow line (SPFL). *Int. J. Prod. Res.* **56**(11), 3932–3954 (2018)

16. Goyal, K.K., Jain, P.K., Jain, M.: Optimal configuration selection for reconfigurable manufacturing system using NSGA II and TOPSIS. *Int. J. Prod. Res.* **50**(15), 4175–4191 (2012)
17. Pattanaik, L.N., Kumar, V.: Multiple level of reconfiguration for robust cells formed using modular machines. *Int. J. Ind. Syst. Eng.* **5**, 424–441 (2010)
18. Bortolini, M., Galizia, F.G., Mora, C., Pilati, F.: Reconfigurability in cellular manufacturing systems: a design model and multi-scenario analysis. *Int. J. Adv. Manuf. Technol.* **104**(9–12), 4387–4397 (2019). <https://doi.org/10.1007/s00170-019-04179-y>
19. Eguia, I., Lozano, S., Racero, J., Guerrero, F.: Cell design and loading with alternative routing in cellular reconfigurable manufacturing systems. *IFAC Proc. Vol.* **46**(9), 1744–1749 (2013)
20. Eguia, I., Racero, J., Guerrero, F., Lozano, S.: Cell formation and scheduling of part families for reconfigurable cellular manufacturing systems using Tabu search. *SIMULATION* **89**, 1056–1072 (2013)
21. Bortolini, M., Ferrari, E., Galizia, F.G., Regattieri, A.: An optimization model for the dynamic management of cellular reconfigurable manufacturing systems under auxiliary module availability constraints. *J. Manuf. Syst.* **58**, 442–451 (2021)
22. Sivanandam, S.N., Deepa, S.N.: Genetic algorithms. In: Sivanandam, S.N., Deepa, S.N. (eds.) *Introduction to genetic algorithms*, pp. 15–37. Springer, Heidelberg (2008). [https://doi.org/10.1007/978-3-540-73190-0\\_2](https://doi.org/10.1007/978-3-540-73190-0_2)