Scheduling of Parallel 3D-Printing Machines with Incompatible Job Families: A Matheuristic Algorithm

Mohammad Rohaninejad, Zdeněk Hanzálek, Reza Tavakkoli-Moghaddam

Scheduling of parallel 3D-printing machines with incompatible job families: A Matheuristic algorithm

Mohammad Rohaninejad[1] [0000-0002-0623-4890],
Zdeněk Hanzálek[1] [0000-0002-8135-1296], and
Reza Tavakkoli-Moghaddam[2] [0000-0002-6757-926X]

[1] Industrial Informatics Department, Czech Institute of Informatics Robotics and Cybernetics, Czech Technical University in Prague, Prague, Czech Republic
Mohammad.Rohani.Nezhad@cvut.cz
[2] School of Industrial Engineering, College of Engineering, University of Tehran, Tehran, Iran

Abstract. Additive manufacturing (AM) is a promising technology for the rapid prototyping and production of highly customized products. The scheduling of AM machines has an essential role in increasing profitability and has recently received a great deal of attention. This paper investigates the scheduling of batch processing of parallel 3d-printing machines to minimize the total weighted tardiness. Accordingly, a mathematical model is proposed to formulate the problem considering the sequence-dependent setup time and incompatible job families, where jobs of different families are processed with different materials and desired quality. Due to the high complexity of the problem, an efficient matheuristic algorithm is presented based on the hybridization of a genetic algorithm and a local search method based on mixed integer programming (MIP). Computational results show that the proposed approach is efficient and promising to solve the problem.

Keywords: Additive manufacturing · Batch processing · Scheduling · Matheuristic.

1 Introduction and Literature Review

Additive manufacturing (AM) that also called 3D-printing is a remarkable technology in the context of industry 4.0, which is rapidly developing smart manufacturing systems. various 3D-printing machine types are developed and applied in prototyping, production, and biomedicine [1]. In terms of industrial production, manufacturing companies are employing 3D-printing technology for facilitating the fabrication of highly customized, and lighter weight products. Powder-based, liquid-based, and solid or wire extrusion techniques are the main processing technologies applied in 3D-printing machines [2] that produce parts by depositing material layer upon layer according to a predesigned computer pattern.

This research focuses on the scheduling of a special kind of powder-based 3D-printing machine known as Selective Laser Melting (SLM) machine. We consider several independent machines working in parallel. In the SLM machine, the laser beam hits the metal powder and welds its particles together. Then, a new layer of metal powder is added and this process continues until the final product is reached. In the current study, the scheduling of parallel SLM machines consists of batching a variety of parts with incompatible job families and then determining the allocation and sequencing of the formed batches in such a way that the total cost of tardiness is minimized ($\mathbf{P_m}|batch, incompatible|\sum w_j T_j$).

According to the shift of AM from making the prototype to real parts production, the production planning and scheduling in AM systems has changed to a crucial problem. Special characteristics of AM environments, such as the wide variety of orders, high production cost, high purchasing cost of AM machines in industrial dimension, and dependence on the orientation of parts in machines, etc. have made the scheduling of AM more complex than the other scheduling problems. Li et al. [3] proposed a mathematical model and two different heuristics to solve the problem. Dvorak et al. [4] presented the scheduling of 3D-printing machines in a job shop while minimizing makespan and satisfying deadlines. Li et al. [5] proposed an approach to make decisions simultaneously on the acceptance and scheduling in AM production. Zhou et al. [6] and Maiet al. [7] studied the scheduling of distributed AM in cloud manufacturing [8]. Regarding to SLM machines, Li et al. [3] proposed two heuristic procedures named 'best-fit' and 'adapted best-fit' to minimize the production cost per volume of material on nonidentical SLM machines. Griffiths et al. [9] studied part orientation and 2D bin packing in the SLM machine to minimize the production cost.

There are few studies on parallel batch processing machine scheduling in AM. Zhang et al. [10] have developed an improved evolutionary algorithm for ($\mathbf{P_m}|batch|C_{max}$) in SLA (Stereo Lithography Appearance) 3D printing machines. They have combined a genetic algorithm with a heuristic placement strategy to take into account the allocation and placement of parts integrally. Kucukkoc [11] has addressed scheduling problem of Single, parallel identical and parallel non-identical AM machines to minimize the makespan. He developed an MILP model that can easily be adopted by AM firms. Our study extended Kucukkoc [11] research. In his research, there was only one type of material and desired quality, and the objective function was the makespan. In contrast, we considered parts with different material types and desired quality, sequence-dependent setup times, and total weighted tardiness as the objective function. Hence, a new mathematical model is presented and due to the high complexity of the problem, a novel matheuristic algorithm based on the combination of Genetic algorithm and an efficient MIP-based local search is developed. Regarding to the computational results, it is clear that the proposed algorithm is an effective step forward to solve the proposed problem.

The rest of this paper proceeds as follows. Section 2 illustrates the problem description and presents the corresponding mixed integer linear programming (MILP) model. In Section 3, the proposed matheuristic algorithm is described.

Section 4 presents the computational results and evaluation of the proposed method. Finally, Section 5 concludes the paper and presents some directions for future research.

2 Problem Description

This section describes the investigated 3D-printing scheduling problem, its assumptions, and the mathematical model. There is a set of parts ($i \in I$) with specific properties that must be produced on a set of parallel SLM machines ($m \in M$), while machines have different area of build platform (CA_m) and height of build platform (CH_m). The characteristics of the parts include material type ($Mt_i \in K$), area (ap_i), height (hp_i), volume (vp_i), desired quality ($Qu_i \in Q$), as well as the due date (dd_i) and tardiness penalty per time unit (tc_i). There is a set of batches ($b \in B$), and the parts with different families (different material types or different desired quality) cannot be assigned to the same batch. The processing speed of the machine depends on the material type and the desired quality of its allocated batch, and the processing time of each batch depends on the total volume and the maximum height of its assigned parts. In other words, the total volume of parts affects the total time required to melt the metal powder and the maximum height of assigned parts affects the number of times to add a new layer of metal powder. After processing each batch, the cleaning and setting of machines should be performed for starting the next batch, while the time required for the new setup depends on the material type of the previous batch. Other parameters and variables and the corresponding mathematical model (Model 1) are as follows.

Parameters

- vt_m^{kq} Time for melting material k with quality q on machine m per volume unit
- ht_m^k Time required for powder layering of material type k on machine m
- σ_m^{0k} Setup time to start the first batch with material type k on machine m
- $\sigma_m^{kk'}$ Setup time required to start the batch with material type k on machine m when the material type of the previous batch on the machine was k'
- G Big positive number

Variables

- x_{ibm} 1 if part i is processed in batch b by machine m; 0, otherwise
- y_{mb}^{kq} 1 if material k is employed for batch b on machine m to produce the parts with quality q; 0, otherwise
- p_{mb} Processing time of batch b on machine m
- tr_i Tardiness of part i

C_{mb} Completion time of batch b on machine m

c_i Completion time of part i

$$Min \sum_{i \in I} tc_i \cdot tr_i \qquad \forall i \tag{1}$$

$$s.t. \sum_{m \in M} \sum_{b \in B} x_{ibm} = 1 \qquad \forall i \tag{2}$$

$$\sum_{i \in I} ap_i \cdot x_{ibm} \leq CA_m \qquad \forall m, \forall b \tag{3}$$

$$hp_i \cdot x_{ibm} \leq CH_m \qquad \forall i, \forall m, \forall b \tag{4}$$

$$y_{mb}^{kq} \cdot G \geq \sum_{\substack{i \in I \,|Mt_i = k \\ \& \, Qu_i = q}} x_{ibm} \qquad \forall m, \forall k, \forall q, \forall b \tag{5}$$

$$\sum_{k \in K} \sum_{q \in Q} y_{mb}^{kq} \leq 1 \qquad \forall m, \forall b \tag{6}$$

$$y_{mb}^{kq} \leq \sum_{\substack{i \in I|Mt_i = k \\ \& \, Qu_i = q}} x_{ibm} \qquad \forall m, \forall k, \forall q, \forall b \tag{7}$$

$$\sum_{\substack{i' \in I \\ |Mt_{i'} = k \\ \& \, Qu_{i'} = q}} x_{i'bm} \leq G \cdot (1 - x_{ibm}) \quad \forall i, \forall m, \forall k, \forall q, \forall b \mid (Mt_i \neq k \ or \ Qu_i \neq q) \tag{8}$$

$$p_{mb} \geq vt_m^{kq} \sum_{i \in I} vp_i \cdot x_{ibm} + ht_m^k \cdot \max_{i \in I} \{hp_i \cdot x_{ibm}\} - G \cdot (1 - y_{mb}^{kq}) \quad \forall m, \forall k, \forall q, \forall b \tag{9}$$

$$\sum_{i \in I} x_{ib+1m} \leq G \cdot \sum_{i \in I} x_{ibm} \qquad \forall m, \forall b \leq B - 1 \tag{10}$$

$$C_{m1} \geq p_{m1} + \sigma_m^{0k} - G \cdot (1 - \sum_{q \in Q} y_{m1}^{kq}) \qquad \forall m, \forall k, b = 1 \tag{11}$$

$$C_{mb} \geq C_{mb-1} + p_{mb} + \sigma_m^{k'k} + G \cdot (\sum_{q \in Q} y_{mb-1}^{k'q} + \sum_{q \in Q} y_{mb}^{kq} - 2) \quad \forall m, \forall k, k', b \neq 1 \tag{12}$$

$$c_i \geq C_{mb} - G \cdot (1 - x_{ibm}) \qquad \forall i, \forall m, \forall b \tag{13}$$

$$tr_i \geq c_i - dd_i \qquad \forall i \tag{14}$$

$$x_{ibm}, y_{mb}^{kq} \in \{0, 1\}; \ C_{mb}, c_i, tr_i, p_{mb} \geq 0 \qquad \forall i, \forall m, \forall k, \forall q, \forall b \tag{15}$$

The relation (1) indicates the objective function of the problem, which is the minimization of the total tardiness cost. Constraint (2) ensures that each part is assigned to one batch. The capacities of SLM machines in terms of area and height of building platform is observed by constraints (3) and (4). Constraints (5)-(7) determine the material type and quality of each formed batch. Constraint (8) prevents the assignment of parts with different material and desired quality to the same batch. The production time of each batch based on the total material volume and the maximum height of its assigned parts is determined by constraint (9). This constraint can be linearized by using variable $\gamma_{m,b}$ instead of $\max_{i \in I}\{hp_i \cdot x_{ibm}\}$ while $\gamma_{m,b} \geq (hp_i \cdot x_{ibm})$ for all i, m and b. Constraint (10) ensures that the parts cannot be assigned to a specific batch while its previous batch is not formed. This constraint, along with constraints (11) and (12) are necessary to determine the completion time of the batches. The tardiness of each part is computed by constraints (13) and (14). Finally, Constraint (15) specifies the ranges for the variables of Model 1.

3 Solution procedure

In this section, a hybrid algorithm called GA-MLS-α% is proposed based on a hybridization of the genetic algorithm (GA) and a local search based on mixed-integer programming (MIP-based local search). In this hybrid algorithm, the GA is used to optimize the sub-problems related to determining the sequence of parts, and allocation of parts to the machine. Then the assignment of parts to the batches is performed by an effective heuristic named batching heuristic. Finally, the MIP-based local search is implemented on the α% of the best solutions in the current population to exchange the batch of parts respecting their sequence. This process continues until the termination condition is met. This procedure is terminated by reaching one of the cases i) a given number of iterations or ii) a computational time limit. Fig. 1 illustrates the procedure of the proposed algorithm.

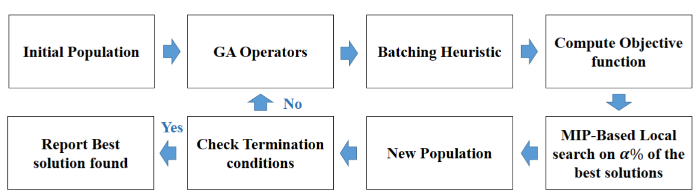

Fig. 1. Schematic pattern of proposed Matheuristic algorithm

3.1 Solution representation and initial population

The utilized solution representation consists of a matrix S with two rows and $|I|$ columns. In the S matrix, the elements of the first row indicate the set of parts ($i \in I$) while their arrangement delineates the relative execution sequence of parts. The components in the second row determine the assigned machines to their corresponding part in the first row. Fig. 2 shows a solution for a problem with 10 parts and 2 machines. In this figure, the columns with the same color show parts that have the same material type and desired quality (same family). For generating the initial population different rules are applied. Sequence of parts is obtained by three rules using: i) random generation ii) shortest processing time (SPT) first and iii) earliest due date (EDD) first. Moreover, the assignment of machines to parts is obtained i) randomly and by ii) earliest time of machine availability (ETA) rule [12].

Parts	8	6	5	2	10	1	3	7	9	4
Machines	2	1	1	2	1	2	1	2	1	1

Fig. 2. Solution representation for a problem with 10 parts and 2 machines

3.2 Crossover and mutation operators

The crossover and mutation operators are performed in the same way that proposed by Rohaninejad et al. [13] In the crossover operator, first, ρ $(1 \leq \rho < I)$ parts are randomly selected. Then, all selected parts are transferred to the first offspring in the same sequence and position related to the first parent. The assigned machines for these parts are selected from the second parent. Next, the remaining parts are transferred to the offspring respecting their sequence in the second parent, and their assigned machines are determined according to the first parent. A reverse procedure of the first offspring is used for the second offspring. In the mutation operator, first, 50% of the parts are selected randomly, and their sequence and assigned machines are determined randomly as well. The remained parts are copied to the mutated individual according to their order of placement and machine assignment in the previous individual.

3.3 Batching heuristic method

This section presents an effective heuristic for the assignment of parts to batches. As shown in Fig. 2, the proposed solution representation lacks any information regarding this decision variable. This pattern of solution representation contributes to a faster search in the solution space and provides feasible solutions in any condition in combination with the proposed batching heuristic. In this method, the parts are assigned to an opened batch as much as possible with maximum observance of their sequence. Algorithm 1 shows the pseudo code of the proposed batching heuristic method.

3.4 MIP-based local search

In each iteration of the solution algorithm, an MIP-based local search is performed on the $\alpha\%$ of the best solutions in the current population. The MIP-based

Algorithm 1: Batching (assignment of parts to the batches) heuristic

```
Result: The corresponding schedule of solution representation
input  : The solution representation matrix (S); nb = 0; MC=[ ]; MK=[ ]; RC=[ ]
for ρ = 1 to |I| do
    i = S[1, ρ] and m = S[2, ρ] and k = Mt_i
    if ∄ batch b ≤ nb while MC[b] = m and MK[b] = k and RC[b] ≥ ap_i then
        Open new batch (b = nb + 1) and b ← i
        MC[b] = m ; MK[b] = k ; RC[b] = CA_m − ap_i
        nb = nb + 1
    else
        Find the smallest b which MC[b]=m and MK[b]=k and RC[b]≥ ap_i then
        b ← i and RC[b] = RC[b] − ap_i
```

local search explores the neighborhood of the original solution. According to this local search, the batching decision variables will be optimized again by a new MIP model (Model 2) with respect to the sequence of parts and their assigned machines corresponding to the original solution. The proposed local search steps in each iteration of the GA are as follows respectively.

- Create set $\mathbb{E}$ including $\alpha\%$ of the best solutions in the current population.
- Set the model 2 for each solution in $\mathbb{E}$.
- Solve model 2 for each solution in $\mathbb{E}$ and determine the batching variables again.

In Model 2, the objective function (1) and constraints (4), (6), (11), (12), (14), and (15) are repeated without any change. Constraints (3), (5), (7), (8), (9), (10) and (13) just needs to be written for every combination of m and i that $\omega[m,i] > 0$. The constraint (2) is replaced by constraint (16) and the constraints (17) and (18) must be added in Model 2. Constraint (17) ensures that the order of the parts with the same material and quality is according to their order on the original solution. Constraint (18) guarantees that the parts can be processed before a part with a lower sequence number just to fill the remaining capacity of the previous batches.

$\varphi[m]$ Number of parts that assigned to the machine m

$\omega[m,i]$ Sequence number of part i in solution representation if m is assigned to i; 0, otherwise

$$\sum_{b \in B \,|\, b \leq \varphi[m]} x_{ibm} = 1 \qquad \forall i, \forall m \mid \omega[m,i] > 0 \tag{16}$$

$$\sum_{b'=1 \,|\, b' \leq \varphi[m] \,\&\, b' > b} x_{ib'm} \leq G.(1 - x_{i'bm})$$
$$\forall i, \forall i', \forall m, \forall b \mid \omega[m,i] > 0 \,\&\, \omega[m,i'] > 0 \,\&\, \omega[m,i] < \omega[m,i'] \tag{17}$$
$$\& \, Mt_i = Mt_{i'} \,\&\, Qu_i = Qu_{i'}$$

$$\sum_{\substack{i' \in I \\ \mid \omega[m,i']>0 \\ \omega[m,i']>\omega[m,i] \\ \& Mt_{i'} \neq Mt_i \\ \& Qu_{i'} \neq Qu_i}} x_{i'bm} \leq G.(1 - x_{ib'm}) + G. \sum_{\substack{i' \in I \\ \mid \omega[m,i']>0 \\ \omega[m,i']<\omega[m,i]}} x_{i'bm} \tag{18}$$

$$\forall i, \forall m, \forall b \leq \varphi[m], \forall b' \leq \varphi[m] \mid b < b' \;\& \, \omega[m,i] > 0$$

4 Computational results

In this section, 10 random instances are solved to evaluate the validation of Model 1 and efficiency of the proposed algorithm. The instances are labeled with $(I - M - F)$, which represent the number of parts, machines, and job families, respectively. The proposed algorithm with different $\alpha\%$ (GA_MLS_$\alpha\%$) is compared with two different metaheuristic algorithms that named GA_BH and GA_ATC and the mathematical formulation (Model 1). The GA_BH algorithm is developed based on combination of proposed genetic algorithm and batching heuristic. The GA_ATC is a custom version of the genetic algorithm that presented by Balasubramanian et al. [14]. They proposed a GA-based algorithm for $(\mathbf{P_m}|batch, incompatible|\sum w_j T_j)$ while first assigns jobs to machines using a GA, then forms batches on each machine and sequences them by a dispatching rule called Apparent Tardiness Cost (ATC). In this study the GA algorithms and mathematical models (Models 1 and 2) are coded by Python. Also, we have used the CPLEX solver for solving the mathematical models. For each algorithm, we have set the run-time limit to 1800 seconds. In Table 1, a detailed results of proposed algorithms are given. In order to analyze the results of the table, first the RPD% criteria is calculated for each algorithm. The RPD% specifies the Relative Percentage Deviation from mean of the objective functions that obtained by each algorithm. Accordingly, an efficient algorithm has a lower value of RPD%. Based on this criteria, it can be found that the GA_MLS_10% is the best algorithm with the average of RPD% equal to -6.6%. The average of RPD% for all instances are equal to -3.3%, -2.9%, 3.6%, 2.9% and 6.4% for GA_MLS_5%, GA_MLS_15%, GA_BH, GA_ATC and CPLEX, respectively.

Fig. 3 shows the computational time of different proposed algorithms. According to this figure, the GA_MLS_$\alpha\%$ algorithms are defensibly able to solve medium-size problems in a reasonable time.

The box plot in Fig. 4 is employed and depicted based on RPD% criteria of GA_BH, GA_ATC and GA_MLS_10% as the best of GA_MLS_$\alpha\%$ algorithms. According to Fig. 4 the GA_MLS_10% has significantly better performance so that 3/4 of its RPD values are at least smaller than 3/4 of the RPD values related to other methods.

5 Conclusion

This research addresses a scheduling problem in an AM environment with unrelated SLM machines and incompatible job families. A new mathematical model is

Table 1. Compare performance of the proposed algorithms

Instances	CPLEX		GA-BH		GA-ATC		GA-MLS-5%		GA-MLS-10%		GA-MLS-15%	
	Best Obj	Time (s)	Best Obj	Time (s)	Best Obj	Time (s)	Best Obj	Time (s)	Best Obj	Time (s)	Best Obj	Time (s)
8-2-2	839	50	839	6	885	8	839	11	839	18	839	26
10-2-2	504	127	732	9	732	11	732	17	504	32	504	44
12-2-4	1805	>1800	1688	12	1956	15	1688	35	1688	72	1688	106
15-3-4	2994	>1800	2712	16.6	2740	22	2728	57	2642	102	2606	152
20-2-6	4779	>1800	4627	18	5074	34	4472	149	4472	330	4413	492
25-3-6	10683	>1800	10472	23	11027	55	10521	285	9860	498	10412	729
30-3-6	8840	>1800	8316	26	8532	86	8467	401	8320	784	8145	1144
35-3-6	7768	>1800	7477	28	6430	164	5925	722	5860	1365	5925	>1800
40-3-6	15318	>1800	10548	36	12919	123	9155	1640	10441	>1800	12370	>1800
45-4-6	13664	>1800	11203	56	10036	238	9712	>1800	10285	>1800	12421	>1800
Average	**6689**	**>1457**	**6098**	**23**	**5795**	**76**	**5423**	**>512**	**5491**	**>680**	**5932**	**>810**

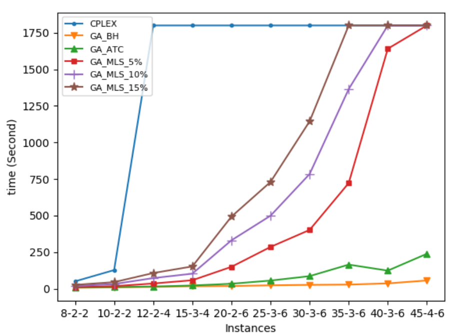

Fig. 3. Comparing computational time

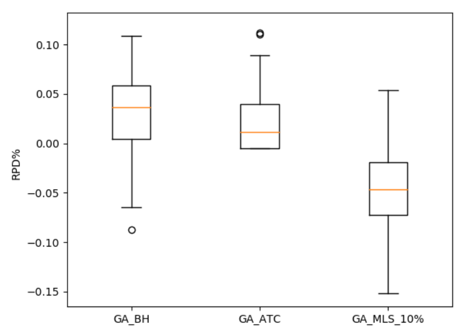

Fig. 4. Comparing the RPD% (Box plot)

presented and due to the high complexity of the problem, an efficient matheuristic method based on the combination of genetic algorithm and a MIP-based local search was developed. Computational results showed the efficiency of the proposed matheuristic method especially for medium-sized problems.

Combination of scheduling and bin packing of parts in 3D-printing machines can be an interesting topic for further research. Besides, studying the given problem with stochastic parameters (e.g, setup time, demand, and available time of machines) brings the problem closer to more realistic conditions.

Acknowledgement

The research has been supported by the Ministry of Education, Youth and Sports within the dedicated program ERC CZ under the project POSTMAN with reference LL1902.

References

1. Berman, B.: 3D printing: the new industrial revolution. Business Horizons **55**(2):155-–162 (2012)
2. Rajaguru, K., Karthikeyan, T., Vijayan, V.: Additive manufacturing–State of art. Materials Today: Proceedings **21**: 628–633 (2020)
3. Li, Q., Kucukkoc, I., Zhang, D. Z.: Production planning in additive manufacturing and 3D printing. Computers & Operations Research **83**: 157–172 (2017)
4. Dvorak, F., Micali, M., Mathieug, M.: Planning and scheduling in additive manufacturing. Inteligencia Artificial, **21**(62), 40–52 (2018)
5. Li, Q., Zhang, D., Wang, S., Kucukkoc, I.: A dynamic order acceptance and scheduling approach for additive manufacturing on-demand production. The International Journal of Advanced Manufacturing Technology, **105**(9), 3711–3729 (2019)
6. Zhou, L., Zhang, L., Laili, Y., Zhao, C., Xiao, Y.: Multi-task scheduling of distributed 3D printing services in cloud manufacturing. The International Journal of Advanced Manufacturing Technology, **96**(9), 3003–3017 (2018)
7. Mai, J., Zhang, L., Tao, F., Ren, L.: Customized production based on distributed 3D printing services in cloud manufacturing. The International Journal of Advanced Manufacturing Technology, **84**(1-4), 71–83 (2016)
8. Vahedi-Nouri, B., Tavakkoli-Moghaddam, R., Rohaninejad, M.: A multi-objective scheduling model for a cloud manufacturing system with pricing, equity, and order rejection. IFAC-PapersOnLine, **52**(13), 2177-2182 (2019)
9. Griffiths, V., Scanlan, J. P., Eres, M. H., Martinez-Sykora, A., Chinchapatnam, P.: Cost-driven build orientation and bin packing of parts in Selective Laser Melting (SLM). European Journal of Operational Research, **273**(1), 334-352 (2019)
10. Zhang, J., Yao, X., Li, Y. : Improved evolutionary algorithm for parallel batch processing machine scheduling in additive manufacturing. International Journal of Production Research, **58**(8), 2263-2282 (2020)
11. Kucukkoc, I.: MILP models to minimise makespan in additive manufacturing machine scheduling problems. Computers & Operations Research, **105**, 58-67 (2019)
12. Rohaninejad, M., Sahraeian, R., Nouri, B. V.: Multi-objective optimization of integrated lot-sizing and scheduling problem in flexible job shops. RAIRO-Operations Research, **50**(3), 587-609 (2016)
13. Rohaninejad, M., Kheirkhah, A., Fattahi, P., & Vahedi-Nouri, B.: A hybrid multi-objective genetic algorithm based on the ELECTRE method for a capacitated flexible job shop scheduling problem. The International Journal of Advanced Manufacturing Technology, **77**(1-4), 51-66 (2015)
14. Balasubramanian, H., Mönch, L., Fowler, J., Pfund, M.: Genetic algorithm based scheduling of parallel batch machines with incompatible job families to minimize total weighted tardiness. International Journal of Production Research, **42**(8), 1621-1638 (2004)