Twentieth International Working Seminar on Production Economics

PRE-PRINTS VOLUME 1

**Papers scheduled for
Tuesday, February 20, 2018
8.00 am to 21.15 pm**

Edited by
*Robert W. Grubbström, Hans H. Hinterhuber
and Janerik E. Lundquist*

CONGRESS INNSBRUCK
INNSBRUCK
AUSTRIA

February 19-23, 2018

The Scientific Field of Production Economics

Production Economics focuses on scientific topics treating the interface between engineering and management. All aspects of the subject in relation to manufacturing and process industries, as well as production in general are covered. The subject is interdisciplinary in nature, considering whole cycles of activities, such as the product life cycle - research, design, development, test, launch, disposal - and the material flow cycle - supply, production, distribution, recycling and remanufacturing.

The ultimate objective is to create and develop knowledge for improving industrial practice and to strengthen the theoretical base necessary for supporting sound decision making. It provides a forum for the exchange of ideas and the presentation of new developments in theory and application, wherever engineering and technology meet the managerial and economic environment in which industry operates.

Tracing economic and financial consequences in the analysis of the problem and solution reported, belongs to the central theme.

◆

The International Working Seminars on Production Economics

The purpose of the *International Working Seminars on Production Economics* is to provide an opportunity for research scientists and practitioners to meet, present and develop their ideas on subjects within the field of Production Economics. A *Discussant* is appointed for each paper. The intention is that models and methods presented, and the discussion of them, will result in concrete ideas for future research and developments in this area. These seminars are *working seminars*, indicating that their main aim is to initiate and improve research results and to provide ample opportunities for interaction between Authors, Discussants, Chairmen and Audience, rather than to publish results. The purpose of these *PrePrints* is to have background working material for the discussion.

This special character of the International Working Seminars on Production Economics, most likely, makes them unique in the international landscape of scientific interaction.

First Seminar, Engelberg, Switzerland
February 20-24, 1978

Second Seminar, Lans/Innsbruck, Austria
February 16-20, 1981

Third Seminar, Igls/Innsbruck, Austria
February 20-24, 1984

Fourth Seminar, Igls/Innsbruck, Austria
February 17-21, 1986

Fifth Seminar, Igls/Innsbruck, Austria
February 22-26, 1988

Sixth Seminar, Igls/Innsbruck, Austria
February 19-23, 1990

Seventh Seminar, Igls/Innsbruck, Austria
February 17-21, 1992

Eighth Seminar, Igls/Innsbruck, Austria
February 21-25, 1994

Ninth Seminar, Igls/Innsbruck, Austria
February 19-23, 1996

Tenth Seminar, Igls/Innsbruck, Austria
February 16-20, 1998

Eleventh Seminar, Igls/Innsbruck, Austria
February 21-25, 2000

Twelfth Seminar, Igls/Innsbruck, Austria
February 18-22, 2002

Thirteenth Seminar, Igls/Innsbruck, Austria
February 16-20, 2004

Fourteenth Seminar, Innsbruck, Austria
February 20-24, 2006

Fifteenth Seminar, Innsbruck, Austria
March 3-7, 2008

Sixteenth Seminar, Innsbruck, Austria
March 1-5, 2010

Seventeenth Seminar, Innsbruck, Austria
February 20-24, 2012

Eighteenth Seminar, Innsbruck, Austria
February 24-28, 2014

Nineteenth Seminar, Innsbruck, Austria,
February 22-26, 2016

Twentieth Seminar, Innsbruck, Austria,
February 19-23, 2018

Pre-Prints

Twentieth International Working Seminar on Production Economics, Innsbruck, Austria, February 19-23, 2018

Contents of Volume 1
Presentations in Sessions Tuesday, February 20, 2018

Invited Keynote Speaker

Regular Sessions

Scheduling of Multiple Additive Manufacturing and 3D Printing Machines to Minimise Maximum Lateness

Ibrahim Kucukkoc[1], Qiang Li[2,3], Naihui He[2] and David Zhang[2,3]

[1] Department of Industrial Engineering, Balikesir University,
Cagis Campus, Balikesir, Turkey
ikucukkoc@balikesir.edu.tr
[2] College of Engineering, Mathematics and Physical Sciences, University of Exeter,
EX4 4QF Exeter, United Kingdom
[3] College of Mechanical Engineering, Chongqing University, 400044 Chongqing, China
q.li@exeter.ac.uk, n.he@exeter.ac.uk, d.z.zhang@exeter.ac.uk

Abstract

Additive manufacturing is now being increasingly utilised in industry especially when the batches are small but the customisation is high. The decisions on the allocation of orders to the machines used for additive manufacturing is important in terms of the completion time related measures (e.g. makespan, flow time) and delivery date related performance measures (e.g. total lateness, maximum lateness, number of tardy jobs), as well as the cost for production. This research focuses on the production scheduling problem in a multiple additive manufaturing/3D printing machine environment. Parts have release dates and due dates. A genetic algorithm approach is developed to minimise the maximum lateness where there are more than one additive manufacturing machines with different capacity specifications (i.e. height and width). The parts are allocated to these parallel additive manufacturing machines in batches considering their release dates, due dates as well as the resource constraints. A detailed numerical example is provided to illustrate the running mechanism of the GA and some improvements are made to increase its performance.

Keywords: Additive manufacturing, production planning, scheduling, genetic algorithm, 3D printing.

1. Introduction

Additive manufacturing (AM), also known as 3D printing (3DP), is used to describe the technologies that build 3D objects by adding layer-upon-layer of material. It is now being increasingly utilised in industries requiring small-batch but high-customisation manufacturing. Among its variants, the metallic powder-bed fusion processes (such as Selective Laser Melting and Electron Beam Melting) have been widely adopted as an advanced direct rapid manufacturing method particularly in aerospace and orthopaedic implant sectors (Calignano et al., 2017; Schmidt et al., 2017).

The production with metallic powder-bed fusion process is job-based where the machine can handle one job at a time and the job can consist of a batch of parts which will be processed simultaneously (Kucukkoc et al., 2016). Both the production cost and the production time of an AM/3DP job dynamically depend on the combination of parts allocated to this job (Li et al., 2017). Therefore, the decision on the allocation of parts to the machines is crucial for the scheduling in AM/3DP production, while the uncertainties caused by variations in the production cost and time make it more challenging in decision-making. This is especially important when the delivery time related performance measures and cost based objectives must be considered together.

The considered production planning problem of AM/3DP machines in this paper is a kind of batch scheduling problem in such an environment with multiple machines. However, it is significantly important to note that the production time of jobs in AM/3DP production planning problem changes based on the total volume and maximum height of parts assigned to the job. The decision maker needs to make a decision based on the allocation of parts to jobs on machines with different specifications, e.g. speed, maximum area, maximum height, the time needed to set-up and cleaning etc. In that sense, this paper introduces the production planning

of AM/3DP problem with the aim of minimising maximum lateness. The paper also contributes to the literature considering the part release dates. So that, a job containing a set of parts to be produced cannot start before the release date of any of those parts. Also, when a job consisting a set of parts is started, it is not possible to take out any of those parts until all parts finish. The first genetic algorithm (GA) approach, whose parameters have been tuned with preliminary tests, is proposed to solve the problem. Note that the detailed 2D or 3D nesting problem is not considered in this research.

This paper is organised as follows. The problem is defined and corresponding formulations are provided in Section 2. The proposed GA is described illustratively through examples in Section 3. A medium sized numerical example consisting of three machines and eighteen parts are solved using the proposed approach in Section 4 and the paper is concluded in Section 5 with key research results and future research directions.

2.   Problem Statement

AM/3DP problem aims to decide the allocation of part orders ($i = 1,2, \dots, in$ and $i \epsilon I$) received from customers to job batches ($j = 1,2, \dots, jn$ and $j \epsilon J$) to be utilised on more than one AM/3DP machines ($m = 1,2, \dots, mn$ and $m \epsilon M$) with different specifications with the aim of optimizing one or more performance criterion. The performance criterion considered in this research is the maximum lateness. The lateness of a part (L_i) is the time difference between the completion time of a part (CT_i) and its due date (d_i). As known, a job is considered late if it is completed later than its due date ($CT_i > d_i$).

Parts have different specifications, i.e. height (h_i), area (a_i), volume (v_i), release date (r_i) and due date (d_i). Machines also have maximum supported production area (MA_m) and height (MH_m) specifications. Thus, it is required to ensure that the height of parts to be assigned a machine must be smaller or equal to the maximum height supported by that machine. It is also necessary to ensure that the total area of parts assigned to a machine cannot exceed the production area supported by that machine.

The completion time of a part (CT_i) is characterised by the completion time of the job (CT_{mj}) in which the part is allocated. CT_{mj} is calculated accumulating the earliest start time of the job (EST_{mj}) and its production time (PT_{mj}) as seen in the following expression.

$$CT_{mj} = EST_{mj} + PT_{mj} \tag{1}$$

where, $EST_{mj} = \max\{CT_{m(j-1)}, \max_{i \in I_{mj}}\{r_i\}\}$. The first expression ($CT_{m(j-1)}$) here corresponds to the completion time of the previous job on the same machine. If j is the first job on machine m, $CT_{m(j-1)}$ is considered to be zero as there is no job scheduled earlier than job j. I_{mj} is the set of jobs allocated in job j on machine m. This denotes that $EST_{mj} \geq r_i$ for all $i \in I_{mj}$. So that the job cannot start before the release date of any job to be produced in that job.

PT_{mj} is calculated using the following formula:

$$PT_{mj} = SET_m + VT_m \cdot \sum_{i \in I_{mj}} v_i + HT_m \cdot \max_{i \in I_{mj}}\{h_i\} \tag{2}$$

where, SET_m is the set-up time for machine m, VT_m is the time needed to release per volume material for machine m, and HT_m is the accumulated interval time per unit height for machine m. Thus, the production time of a job basis on the total volume and maximum height of parts assigned to that job as well as the set-up time of the machine on which the job is scheduled.

One of the basic assumptions of the problem is that, a job cannot stop until it is fully completed. Therefore, even a small and short part is completed before the completion of the whole job, it cannot be taken out until the job completely finishes. Also, it is not possible to add a part into a job after it starts. Another assumption is that all parts must be assigned to exactly one job. It is not possible to split parts or jobs on to more than one AM machine. It is also necessary to indicate that all AM machines work in parallel, independently from other(s). They may have different speed and require different amount of time for set-up.

3. The Proposed GA

GA is a well-known nature inspired optimisation technique inspired by the natural selection process in the nature (Goldberg and Holland, 1988). GA was selected in this research due to its successful implementations on many combinatorial optimisation problems from manufacturing to transportation and design. It is a powerful method for solving sophisticated problems, and has been applied to many scheduling problems, see for example Cheng et al. (1996) for a survey of job-shop scheduling problems using genetic algorithms; Gonçalves et al. (2005) and Pezzella (2008) for GAs developed for job-shop and flexible job-shop scheduling problems; Kucukkoc and Zhang (2015) and Kucukkoc and Zhang (2016) for GA and hybrid GA algorithms for assembly line balancing problems; and Woo et al. (2017) for GA developed to solve parallel machine scheduling problem.

The general outline of the proposed GA is depicted in Figure 1. The algorithm starts with generating $popSize$ number of chromosomes to form the initial population. Each chromosome is made up with a randomly permuted string of numbers corresponding to part numbers. Therefore, the length of the chromosome is equal to the number of parts. All individuals in the population are evaluated one-by-one using the procedure which will be given in Section 3.1. Genetic operators are applied to the chromosomes selected through tournament selection, with a tournament size of $popSize/6$. After the application of the genetic operators, the new generation is formed and genetic operators are applied again. This cycle continues until the maximum number of iterations ($maxIter$) is exceeded with no improvement.
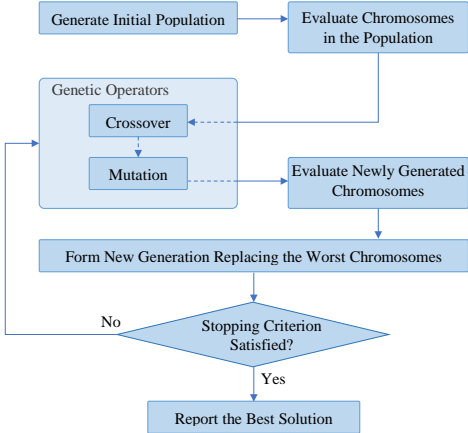
Figure 1. The general outline of the proposed algorithm.

3.1.   Decoding Procedure

The procedure used when evaluating the chromosomes is shown in Figure 2. As seen in the figure, chromosomes are decoded allocating parts to the machines in the order of their appearance on the chromosome. The decoding procedure starts with the first job on the first machine and assigns as many parts as possible starting from the first gene on the chromosome while the machine is high enough to produce the part and the remaining production area (RA_{mj}) is large enough. If any part is not assignable to a job on a specific machine, it is skipped and considered for the next job for the next machine. This procedure is continued until all parts have been assigned to a job.
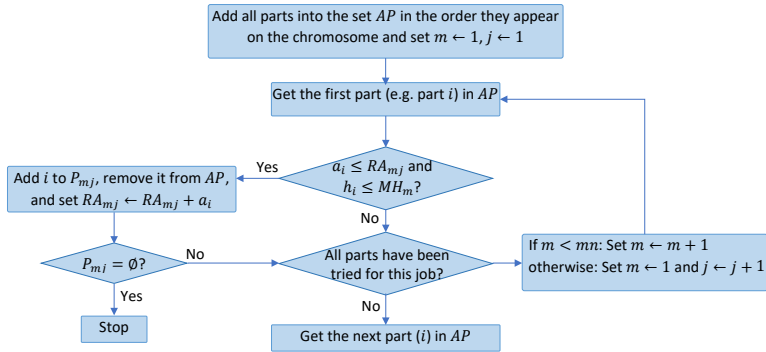
Figure 2. The flow chart of the decoding procedure.

Let us assume an AM/3DP problem with two machines and ten parts, for which the detailed specifications are given in Table 1 and Table 2. Figure 3 presents a chromosome sample and illustrates its decomposition based on the decoding procedure considering the data for the numerical example given in Table 1 and Table 2. As seen in the figure, after the allocation of P3, P6 and P1 to job 1 on machine 1 (i.e. $P_{11} = \{3,6,1\}$), none of the other parts can be assigned due to the insufficient remaining production area. Therefore, the machine is changed and P4, P2 and P5 are assigned to the first job on machine 2. As the remaining production area (326.9 cm^2) is not large enough to assign P8, it is skipped and P7, P10 and P9 are also assigned in the same job. Upon filling up the first two jobs on both of the machines, a new job is opened on the first machine again and P8 is assigned. Please see Figure 4 for the steps of the decoding procedure in details. The EST_{mj}, PT_{mj} and CT_{mj} values of the jobs are also calculated and presented in the same figure.

Parameters	M1	M2
VT_m, the time consumption to form per unit volume	0.030864 (hour/cm³)	0.030864 (hour/cm³)
HT_m, the accumulated time per unit height	1 (hour/cm)	1 (hour/cm)
SET_m, the time consumption for setting up a new	2 (hour)	1 (hour)
H_m, the maximum height supported	32.5 (cm)	32.5 (cm)
A_m, the maximum production area supported	625 (cm²)	625 (cm²)

Table 1. Specifications of the AM machines.

Part (i)	Height-cm (h_i)	Area-cm² (a_i)	Volume-cm³ (v_i)	Release Date-hr (r_i)	Due Date-hr (d_i)
1	16.7	300.8	1573.8	6.3	305.9
2	8.8	152.8	421.5	7.3	282.2
3	20.3	19.5	147.8	9.8	378.3
4	7.4	84.2	285.2	20.9	214.7
5	27.3	61.1	583.3	36.5	149.0
6	25.8	299.3	3282.5	51.5	211.6
7	14.5	148.7	1265.5	56.3	240.4
8	3.5	376.4	723.3	69.9	576.2
9	20.4	20.5	278.5	75.0	330.9
10	23.3	91.1	1051.8	86.0	388.0

Table 2. Part specifications.
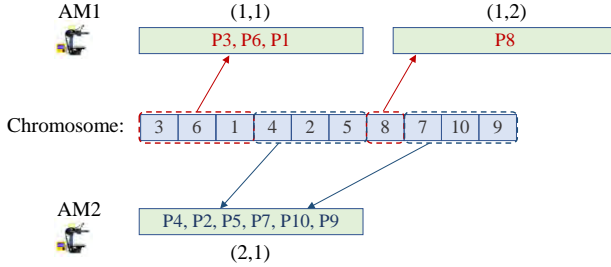
Figure 3. A chromosome sample and its decoding procedure.

(m,j)	$CT_{m(j-1)}$	Remaining Area (RA_{mj})	Considered Part	h_i cm	a_i cm²	Fits	Assigned Parts	Total Volume cm³
(1,1)	0	625.0	3	20.3	19.5	**Yes**	{3}	147.8
		605.5	6	25.8	299.3	**Yes**	{3,6}	3430.3
		306.2	1	16.7	300.8	**Yes**	{3,6,1}	5004.1
		5.4	4	7.4	84.2	No	{3,6,1}	5004.1
		5.4	2	8.8	152.8	No	{3,6,1}	5004.1
		5.4	5	27.3	61.1	No	{3,6,1}	5004.1
		5.4	8	3.5	376.4	No	{3,6,1}	5004.1
		5.4	7	14.5	148.7	No	{3,6,1}	5004.1
		5.4	10	23.3	91.1	No	{3,6,1}	5004.1
		5.4	9	20.4	20.5	No	**{3,6,1}**	**5004.1**

$P_{11} = \{3,6,1\}$, $PT_{11} = 2 + 0.030864 \cdot (5004.1) + 1 \cdot max\{20.3, 25.8, 16.7\} = 182.2465$ hr $\cong$ 182.2 hr
$EST_{11} = max\{CT_{10}, max\{9.8, 51.5, 6.3\}\} = 51.5$ hr, $CT_{11} = EST_{11} + PT_{11} = 51.5 + 182.2 = 233.7$ hr

(2,1)	0	625.0	4	7.4	84.2	**Yes**	{4}	285.2
		540.8	2	8.8	152.8	**Yes**	{4,2}	706.7
		388.0	5	27.3	61.1	**Yes**	{4,2,5}	1290.0
		326.9	8	3.5	376.4	No	{4,2,5}	1290.0
		326.9	7	14.5	148.7	**Yes**	{4,2,5,7}	2555.5
		178.2	10	23.3	91.1	**Yes**	{4,2,5,7,10}	3607.3
		87.1	9	20.4	20.5	**Yes**	**{4,2,5,7,10,9}**	**3885.8**

$P_{21} = \{4,2,5,7,10,9\}$, $PT_{21} = 1 + 0.030864 \cdot (3885.8) + 1 \cdot max\{7.4, 8.8, 27.3, 14.5, 23.3, 20.4\} = 148.2313$ hr $\cong$ 148.2 hr
$EST_{21} = max\{CT_{20}, max\{20.9, 7.3, 36.5, 56.3, 86.0, 75.0\}\} = 86.0$, $CT_{21} = 86.0 + 148.2 = 234.2$ hr

(1,2)	233.7	625.0	8	3.5	376.4	**Yes**	**{8}**	**723.3**

$P_{12} = \{8\}$, $PT_{12} = 2 + 0.030864 \cdot (723.3) + 1 \cdot max\{3.5\} = 27.8239$ hr $\cong$ 27.8 hr
$EST_{12} = max\{CT_{11}, max\{69.9\}\} = 233.7$ hr, $CT_{12} = EST_{12} + PT_{12} = 233.7 + 27.8 = 261.5$ hr

Figure 4. The detailed decoding procedure of the sample chromosome.

The calculations on the lateness values of parts are provided in Table 3. As the GA aims to minimise the maximum lateness, which can be formulated as $Min\ Z = \max_{i \in I}\{L_i\}$, the objective function value of the sample chromosome can simple be found as 85.2 hr.

(m, j)	CT_i (hr)	i	d_i (hr)	L_i (hr)
(1,1)	233.7	3	378.3	-144.6
		6	211.6	22.1
		1	305.9	-72.2
(2,1)	234.2	4	214.7	19.5
		2	282.2	-48.0
		5	149.0	85.2
		7	240.4	-6.2
		10	388.0	-153.8
		9	330.9	-96.7
(1,2)	261.5	8	576.2	-314.7

Table 3. The lateness calculations for parts.

3.2. Genetic Operators

Crossover and mutation are two basic operators of GA used to search the solution space through differentiating the chromosomes chosen stochastically from the population. The parents are chosen using the tournament selection (Miller and Goldberg, 1995), *i.e.* a total of $tourSize$ (where $tourSize = popSize/6$) candidates are selected from the population and the one which has the lowest objective value win the tournament. The determination of the tournament size is important to give a chance to all chromosomes in the population while favouring the better ones. This is important to keep the diversity in the population while sustaining the convergence capacity of the GA.

Crossover is applied using two chromosomes selected from the population with tournament selection. A random cutting point between $[1, in-1]$, including boundaries, is determined and two children are built as shown in when the randomly determined cutting point is assumed to be four.
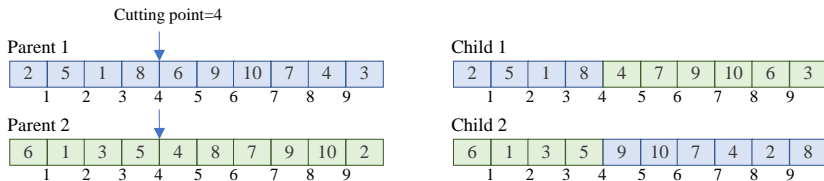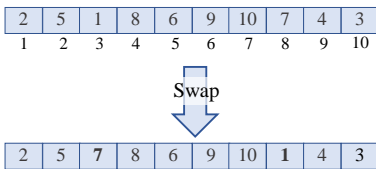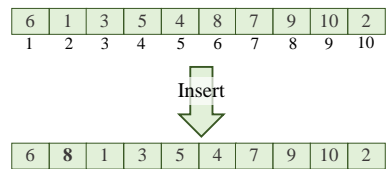

Figure 5. The application of the crossover procedure

Figure 6. The application of the swap and insert mutations.

Mutation helps algorithm avoid getting stuck in local optima. It plays crucial role to explore the different regions of the search space. Therefore, mutation is applied in two ways in this research, i.e. swap and insert. A random number (rnd_0) is determined between [0,1). If $rnd_0 < 0.5$, swap mutation is applied; otherwise, insert mutation is applied. Figure 6 illustrates the application of the mutation operator.

To apply the swap mutation, two random numbers (rnd_1 and rnd_2) are determined between [1,in] and the genes corresponding to these numbers are exchanged (see Figure 6a). To apply the insert mutation, the gene located at the location rnd_1 is removed and located at the location rnd_2, where rnd_1 and rnd_2 are generated randomly between [1,in], again (see Figure 6b).

3.3. Forming the New Generation
The new generation is formed replacing the worst in the population by a cyclic manner. For this aim, the fitness values of all newly generated solutions are calculated after the application of the genetic operators. The worst chromosome in the population is replaced with the best chromosome among the new individuals. This procedure is continued until there is no newly generated chromosome better than any individual in the population. It should be noted here that the duplication of the chromosomes is not allowed during the replacement process. This will be exemplified in the following subsection.

Parameters	M1	M2	M3
VT_m (hr/ cm^3)	0.030864	0.030864	0.030864
HT_m (hr/cm)	0.7	0.7	0.7
SET_m (hr)	1.2	1.0	1.2
H_m (cm)	32.5	32.5	32.5
A_m (cm^2)	625	625	625

Table 4. Specifications of the AM machines.

i	h_i (cm)	a_i (cm^2)	v_i (cm^3)	r_i (hr)	d_i (hr)
1	20.3	19.5	147.8	9.8	378.3
2	7.4	84.2	285.2	21.0	214.7
3	27.3	61.1	583.3	36.6	149.0
4	25.8	299.3	3282.5	51.5	576.2
5	14.5	148.7	1265.5	56.4	240.4
6	3.5	376.4	723.3	69.9	211.6
7	20.4	20.5	278.5	75.1	330.9
8	23.3	91.1	1051.8	86.0	388.0
9	26.3	20.0	201.9	93.0	447.1
10	12.8	123.9	866.1	93.1	445.1
11	28.4	333.5	7347.6	97.2	634.6
12	10.9	74.2	333.6	97.3	177.6
13	14.1	268.5	2956.0	101.8	355.3
14	3.9	11.2	32.5	107.0	258.8
15	3.2	138.9	265.6	107.6	295.6
16	24.6	92.0	1387.4	115.6	509.6
17	7.1	424.0	1086.0	128.4	294.3
18	25.2	181.4	3559.2	132.3	575.7

Table 5. Parts data

4. Detailed GA Solution of a Numerical Example

A numerical example consisting of three AM machines and eighteen parts is solved using the GA proposed in this research. Table 4 and Table 5 present the data related to the machine specifications and part details.

The parameters of the algorithm have been determined as $maxIter = 5000$, $popSize = 30$, $crossover\ rate\ (cr) = 0.6$, and $mutation\ rate\ (mr) = 0.1$ based on the common tendency in the literature and some preliminary tests as shown in Figure 7. The tournament size has been set to $popSize/6$.

(a) $maxIter = \{100, 2000, 5000, 10000\}$

(b) $popSize = \{20, 30, 40, 50\}$

(c) $cr = \{0.2, 0.4, 0.6, 0.8\}$

(d) $mr = \{0.01, 0.05, 0.1, 0.2\}$
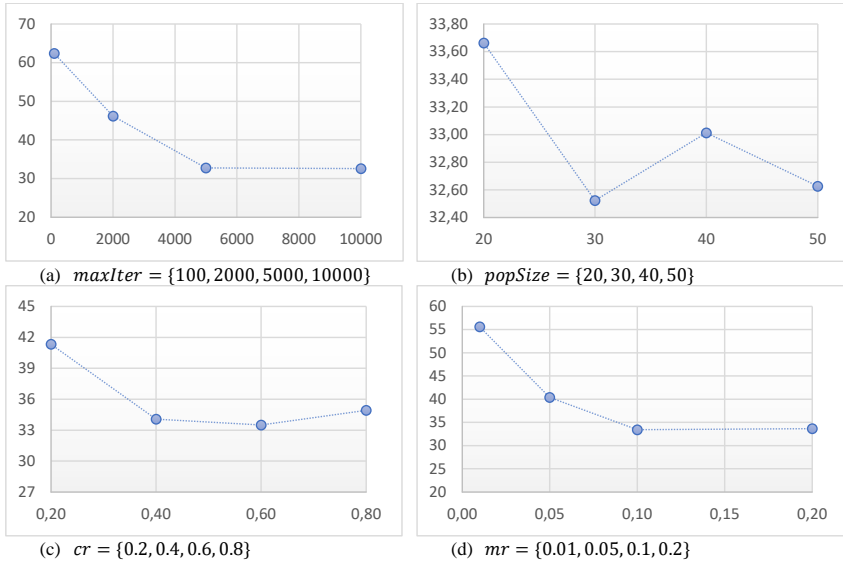
Figure 7. Parameter tuning.

The algorithm was coded in Java and run on Intel Core ® i7 6700HQ CPU@2.6GHz with 16GB of RAM using the parameters determined above. Table 6 shows (a) the chromosomes in the initial population, (b) new individuals obtained from the genetic operators, and (c) the new population after forming the new generation with fitness values. The sign "(-)" indicates that the chromosome has been replaced with that one indicated with (+) in the next generation. As seen from the table, while the best fitness value has not changed, the average fitness value was reduced from 282.30 to 231.69. This shows the convergence of the overall population in only one iteration.

The best solution was found in the 15th iteration with the fitness value of 28.03 and the algorithm was terminated after 5000 iterations as no improvement was observed in the fitness value. The best solution obtained is shown in Table 7.

```
(a)   Initial population                                                 Fitness Value
[14, 15, 4, 11, 3, 13, 16, 2, 17, 8, 9, 1, 18, 10, 6, 5, 12, 7]          340.65(-)
[18, 5, 1, 7, 16, 15, 10, 8, 4, 14, 9, 13, 11, 17, 3, 6, 2, 12]          249.07
[10, 15, 11, 3, 8, 6, 14, 16, 13, 7, 12, 18, 5, 17, 9, 1, 2, 4]          234.91
[4, 6, 14, 13, 5, 8, 9, 3, 2, 7, 18, 10, 1, 17, 15, 11, 12, 16]          262.44
[11, 8, 3, 2, 4, 17, 1, 18, 13, 5, 12, 10, 7, 15, 16, 6, 14, 9]          320.08(-)
[11, 16, 18, 1, 4, 7, 17, 14, 12, 3, 2, 15, 8, 5, 9, 10, 6, 13]          409.19(-)
[12, 3, 11, 10, 16, 4, 2, 7, 6, 13, 5, 9, 14, 18, 8, 15, 17, 1]          273.28
[16, 15, 14, 17, 12, 8, 9, 1, 13, 5, 4, 11, 2, 18, 6, 7, 3, 10]          110.83
[1, 13, 3, 15, 12, 8, 6, 18, 5, 14, 10, 4, 2, 16, 7, 9, 11, 17]          360.81(-)
[17, 1, 5, 3, 16, 10, 6, 15, 12, 11, 9, 2, 13, 8, 18, 7, 4, 14]          216.17
...                                                                      ...
[9, 16, 13, 12, 17, 18, 1, 15, 4, 14, 3, 8, 5, 2, 10, 7, 11, 6]          368.24(-)
[10, 7, 3, 18, 1, 17, 9, 13, 11, 14, 16, 8, 5, 12, 6, 4, 15, 2]          274.91
[16, 2, 3, 13, 17, 8, 7, 4, 18, 12, 10, 11, 5, 14, 9, 15, 1, 6]          392.54(-)
[15, 4, 7, 5, 18, 12, 8, 9, 6, 17, 16, 14, 2, 13, 1, 10, 11, 3]          369.02(-)
[15, 10, 7, 13, 12, 17, 11, 5, 14, 2, 16, 3, 9, 18, 6, 1, 8, 4]          221.08

Best Fitness: 110.83, Average Fitness: 282.30
```

```
                                    Iteration #1
(b) New chcromosomes after crossover and mutation                        Fitness Value
[16, 15, 14, 17, 12, 8, 9, 1, 13, 5, 4, 11, 2, 18, 6, 7, 3, 10]          110.83
[16, 15, 14, 17, 12, 8, 9, 1, 13, 5, 4, 11, 2, 18, 6, 7, 3, 10]          110.83
[15, 10, 7, 13, 12, 17, 14, 16, 18, 5, 9, 1, 2, 4, 11, 3, 8, 6]          211.78
...                                                                      ...
[2, 16, 12, 18, 3, 11, 13, 8, 6, 5, 9, 14, 15, 17, 1, 10, 4, 7]          233.08
[4, 6, 14, 13, 5, 8, 9, 3, 2, 7, 18, 10, 1, 17, 16, 15, 12, 11]          277.50
[16, 15, 14, 17, 12, 8, 9, 1, 13, 5, 4, 11, 2, 18, 6, 3, 7, 10]          110.83
[15, 10, 7, 13, 12, 17, 11, 5, 14, 2, 16, 3, 9, 18, 6, 1, 4, 8]          221.08
[14, 9, 4, 16, 15, 1, 2, 5, 11, 17, 13, 3, 12, 6, 7, 10, 18, 8]          253.21

(c) The new generation after iteration #1                                Fitness Value
[16, 15, 14, 17, 12, 8, 9, 1, 13, 5, 4, 11, 2, 18, 6, 3, 7, 10]          110.83(+)
[18, 5, 1, 7, 16, 15, 10, 8, 4, 14, 9, 13, 11, 17, 3, 6, 2, 12]          249.07
[10, 15, 11, 3, 8, 6, 14, 16, 13, 7, 12, 18, 5, 17, 9, 1, 2, 4]          234.91
[4, 6, 14, 13, 5, 8, 9, 3, 2, 7, 18, 10, 1, 17, 15, 11, 12, 16]          262.44
[15, 10, 7, 13, 12, 17, 14, 16, 18, 5, 9, 1, 2, 4, 11, 3, 8, 6]          211.78(+)
[15, 10, 7, 13, 12, 17, 11, 5, 14, 2, 16, 3, 9, 18, 6, 1, 4, 8]          221.08(+)
[12, 3, 11, 10, 16, 4, 2, 7, 6, 13, 5, 9, 14, 18, 8, 15, 17, 1]          273.28
[16, 15, 14, 17, 12, 8, 9, 1, 13, 5, 4, 11, 2, 18, 6, 7, 3, 10]          110.83
[2, 16, 12, 18, 3, 11, 13, 8, 6, 5, 9, 14, 15, 17, 1, 10, 4, 7]          233.08(+)
[17, 1, 5, 3, 16, 10, 6, 15, 12, 11, 9, 2, 13, 8, 18, 7, 4, 14]          216.17
...                                                                      ...
[10, 15, 11, 3, 8, 6, 5, 14, 2, 16, 9, 18, 1, 4, 7, 13, 12, 17]          247.89(+)
[10, 7, 3, 18, 1, 17, 9, 13, 11, 14, 16, 8, 5, 12, 6, 4, 15, 2]          274.91
[14, 9, 4, 16, 15, 1, 2, 5, 11, 17, 13, 3, 12, 6, 7, 10, 18, 8]          253.21(+)
[12, 3, 11, 10, 16, 4, 2, 7, 14, 6, 9, 15, 17, 1, 5, 18, 13, 8]          270.48(+)
[15, 10, 7, 13, 12, 17, 11, 5, 14, 2, 16, 3, 9, 18, 6, 1, 8, 4]          221.08

Best Fitness: 110.83, Average Fitness: 231.69
```

Table 6. (a) The initial population, (b) new individuals obtained from the genetic operators, and (c) the new population after forming the new generation

(m,j)	P_{mj}(hr)	PT_i (hr)	CT_i (hr)
(1,1)	17, 5, 14, 7, 9	108.01	236.41
(2,1)	8, 13, 16, 15, 1	197.49	313.09
(3,1)	6, 2, 3, 12	79.73	177.03
(1,2)	11, 18	357.70	594.12
(2,2)	10, 4	147.10	460.19
Chromosome: [17, 5, 8, 14, 13, 6, 7, 16, 9, 15, 2, 11, 3, 12, 1, 18, 10, 4]			

Table 7. The best solution obtained

(m,j)	CT_i (hr)	i	d_i (hr)	L_i (hr)
(1,1)	236.41	17	294.3	-57.89
		5	240.4	-3.99
		14	258.8	-22.39
		7	330.9	-94.49
		9	447.1	-210.69
(2,1)	313.09	8	388.0	-74.91
		13	355.3	-42.21
		16	509.6	-196.51
		15	295.6	17.49
		1	378.3	-65.21
(3,1)	177.03	6	211.6	-34.57
		2	214.7	-37.67
		3	149.0	28.03
		12	177.6	-0.57
(1,2)	594.12	11	634.6	-40.48
		18	575.7	18.42
(2,2)	460.19	10	445.1	15.09
		4	576.2	-116.01

Table 8. The calculation of lateness values of tasks based on the best solution obtained.

As seen from the table, the late parts are P15 in job (2,1), P3 in job (3,1), P18 in job (1,2) and P10 in job (2,2). Among those, the maximum lateness belongs to P3 with 28.03 hr, which determines the objective value. The convergence of the best fitness and average fitness values are depicted in Figure 8 for the first 511 iterations.
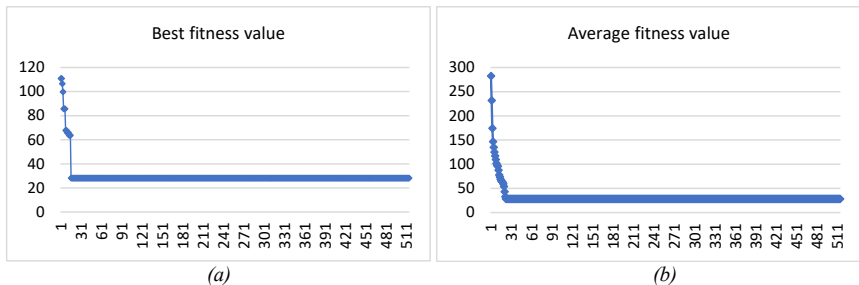

(a)   *(b)*

Figure 8. The convergence of the best fitness and average fitness values.

5. Conclusions

This paper introduced the problem of scheduling additive manufacturing machines considering part release dates and due dates in a multiple machine environment with the aim of minimising maximum lateness, i.e. the maximum positive gap between the completion time and due date of jobs. The machines may have different specifications, e.g. dimensions, set up time, speed,

etc. The problem has been defined and a numerical example is presented to show the production time and lateness calculations based on an example allocation of parts to machines. A GA approach, which employs a machine oriented decoding mechanism, was developed for solving the problem efficiently. Some preliminary tests have been conducted to determine the parameters of the algorithm. The steps of the algorithm have been illustrated through examples and a numerical example consisting of three machines and eighteen parts was solved using the proposed GA. The convergence of the best fitness and average fitness values were plotted and the best solution was reported. The preliminary results show that the proposed GA has a promising performance. The authors' ongoing work aims to enhance the decoding mechanism and compare the performance of the algorithm to other heuristics and/or metaheuristics through a comprehensive computational study.

6. References

Calignano, F., Manfredi, D., Ambrosio, E., Biamino, S., Lombbardi, M., Atzeni, E., Salmi, A., Minetola, P., Iuliano, L., Fino, P., 2017. Overview on Additive Manufacturing Technologies. Proc. IEEE 105, 593–612. https://doi.org/10.1109/JPROC.2016.2625098

Cheng, R., Gen, M., Tsujimura, Y., 1996. A tutorial survey of job-shop scheduling problems using genetic algorithms—I. representation. Comput. Ind. Eng. 30, 983–997. https://doi.org/10.1016/0360-8352(96)00047-2

Goldberg, D.E., Holland, J.H., 1988. Genetic Algorithms and Machine Learning. Mach. Learn. 3, 95–99. https://doi.org/10.1023/A:1022602019183

Gonçalves, J.F., de Magalhães Mendes, J.J., Resende, M.G.C., 2005. A hybrid genetic algorithm for the job shop scheduling problem. Eur. J. Oper. Res. 167, 77–95. https://doi.org/10.1016/J.EJOR.2004.03.012

Kucukkoc, I., Li, Q., Zhang, D.Z., 2016. Increasing the utilisation of additive manufacturing and 3D printing machines considering order delivery times, in: 19th International Working Seminar on Production Economics. Innsbruck, Austria, pp. 195–201.

Kucukkoc, I., Zhang, D.Z., 2016. Integrating ant colony and genetic algorithms in the balancing and scheduling of complex assembly lines. Int. J. Adv. Manuf. Technol. 82, 265–285. https://doi.org/10.1007/s00170-015-7320-y

Kucukkoc, I., Zhang, D.Z., 2015. A mathematical model and genetic algorithm-based approach for parallel two-sided assembly line balancing problem. Prod. Plan. Control 26, 874–894. https://doi.org/10.1080/09537287.2014.994685

Li, Q., Kucukkoc, I., Zhang, D.Z., 2017. Production planning in additive manufacturing and 3D printing. Comput. Oper. Res. 83, 157–172.

Miller, B.L., Goldberg, D.E., 1995. Genetic Algorithms, Tournament Selection, and the Effects of Noise. Complex Syst. 9, 193–212. https://doi.org/10.1.1.30.6625

Pezzella, F., Morganti, G., Ciaschetti, G., 2008. A genetic algorithm for the Flexible Job-shop Scheduling Problem. Comput. Oper. Res. 35, 3202–3212. https://doi.org/10.1016/J.COR.2007.02.014

Schmidt, M., Merklein, M., Bourell, D., Dimitrov, D., Hausotte, T., Wegener, K., Overmeyer, L., Vollertsen, F., Levy, G.N., 2017. Laser based additive manufacturing in industry and academia. CIRP Ann. 66, 561–583. https://doi.org/10.1016/j.cirp.2017.05.011

Woo, Y.-B., Jung, S., Kim, B.S., 2017. A rule-based genetic algorithm with an improvement heuristic for unrelated parallel machine scheduling problem with time-dependent deterioration and multiple rate-modifying activities. Comput. Ind. Eng. 109, 179–190. https://doi.org/10.1016/J.CIE.2017.05.007