Production Engineering
https://doi.org/10.1007/s11740-024-01286-y

ORIGINAL PAPER

Powder bed fusion factory productivity increases using discrete event simulation and genetic algorithm

Ruba Al-zqebah[1] · Matthias Guertler[1] · Lee Clemon[1,2]

Received: 15 March 2024 / Accepted: 3 May 2024

Abstract

Powder bed fusion is importance is growing with uses across industries in both polymer and metallic components, particularly in mass individualization. However, due to the relatively slow mass deposition speed compared to conventional methods, scheduling and production planning play a crucial role in scaling up additive manufacturing productivity to higher volumes. This paper introduces a framework combining discrete event simulation and a genetic algorithm showing makespan improvement opportunities for multiple powder bed fusion factories varying workers, jobs and available equipment. The results show that bottlenecks move among workstations based on worker and capital equipment availability, which depend on the size of the facility indicating a resource-driven constraint for makespan. A makespan reduction of 78% is achieved in the simulation. This shows the trade-off of worker and capital equipment to achieve makespan improvements. The addition of personnel or equipment increases production with further gains achieved by scheduling optimization. Two levels of job demands are analyzed showing productivity gains of 45% makespan improvement when adding the first worker and additional savings with scheduling optimization using a genetic algorithm up to 11%. Most research on additive manufacturing production has focused on the quality of produced parts and printing technology rather than factory level management. This is the first application of this methodology to varying sizes of these potential factories. The method developed here will help decision-makers to determine the appropriate number of resources to meet their customer demand on time, additionally, finding the optimal route for jobs before starting the production process.

Keywords Additive manufacturing · Discrete event simulation · Makespan · Genetic algorithm · Scheduling

1 Introduction

Powder bed fusion (PBF) [1], has become a viable option for industrial and commercial production. It has enabled the production of parts that are more complex, customized, and variable than conventionally manufactured parts. As noted and evaluated in [2] effective planning of factories is a challenging and crucial step for manufacturing competitiveness. In this study, we investigate a set of design choices for such planning for Multi Jet Fusion (MJF). MJF uses infrared lamps and a doping agent deposited by inkjet nozzles for fusing polymers together and is classified in the PBF family of additive manufacturing processes per ISO/ASTM 52900 [1]. MJF is a competitive process with other polymer processes, the closest being Selective laser sintering (SLS). In comparison to FFF, MJF is up to ten times faster with superior surface finish and more isotropic mechanical properties [3] leading to commercial adoption in aerospace, medical, and automotive industries [4]. Examples include tooling [5], production line components [6], and orthotics [7]. Furthermore, the recent announcement of HP's metal system [8], which shares similarities with their existing polymer systems, such as ease of use and scalability, has sparked interest in the industry for factory arranged production systems. However, the scale-up of additive manufacturing for production-level fabrication is challenged by the low

✉ Lee Clemon
lmclemon@illinois.edu

Ruba Al-zqebah
Ruba.Al-zqebah@student.uts.edu.au

Matthias Guertler
Mathias.Guertler@uts.edu.au

[1] School of Mechanical and Mechatronic Engineering, University of Technology Sydney, Ultimo, NSW 2007, Australia

[2] Mechanical Science and Engineering, University of Illinois, Urbana-Champaign, Urbana, IL 61801, USA

Published online: 16 June 2024

mass deposition per unit of time and the classic problem of high mix low volume products [9]. PBF systems allow for individual customization of parts built in the same batch leveraging the geometric freedom of AM and enabling economic lot size 1 production. PBF is a layer-by-layer process, and the deposition rate is limited by the size of the deposition nozzles, the resolution of the print, and the properties of the material being used [10]. This can result in longer production times per unit compared to conventional manufacturing methods like injection molding. At a factory level, additional variables affect the total production time such as machine failure, worker availability, and shift time. When production quantities increase, the impacts of these variables become more significant.

Scheduling and production planning methods in AM are still developing with a recent review indicating limited investigation into this topic [11]. Existing studies on production planning in AM treated this challenge as a nesting and scheduling problem using heuristic optimization methods to allocate workers and parts to a static representation of the production system. Their objectives varied including minimizing production cost [12, 13], minimizing lateness [14], minimizing makespan [15], or maximizing profit and resource utilization [16]. Kucukkoc et al. [17] examined the scheduling problem in additive manufacturing for the purpose of optimizing the processing time by assigning parts in job batches and scheduling them on multiple machines based on multi-integer linear programming models. Similarly, Dvorak et al. [18] used heuristic algorithms to minimize makespans such as hill climbing (HC), tabu search (TS), and simulated annealing (SA). Zhang, et al. [19] developed an algorithm for additive manufacturing based on combining the genetic algorithm with a heuristic placement strategy to minimize the makespan, they found that the developed algorithm showed better performance than the other heuristic algorithms such as a genetic algorithm alone and a particle swarm algorithm. Oh et al. [20] investigated the production planning and scheduling problem within multiple AM machines, they proposed a heuristic algorithm to optimize the build orientation, 2D packing, and scheduling based on the longest cycle time. Akram et al. [21] examined the scheduling of identical parallel AM machines, aiming to meet distinct order deadlines while minimizing overall tardiness. They introduced both the mathematical framework for this challenge and a heuristic approach. The problem was deconstructed into two subproblems: first, the allocation of parts/jobs via part clustering linked to due dates, followed by job scheduling centered on job tardiness. However, this body of research focused on reducing the makespan analytically or with a static representation of the system and does not consider the stochastic and dynamic nature of a factory floor. Stittgen et al. [22] notes a strong impact of job allocation and selection within builds on laser-PBF throughput highlighting this challenge needs further investigation. Prior methods focused on specific snapshots in time, i.e. were static in their implementation and unable to update with changes in worker or machine availability and setups. Thus, these methods did not adjust for dynamic changes in the factory, like machine failures or worker absence. More dynamic methods of simulating production flows are needed to explore factory level implications of operating PBF systems and provide better insight into factory planning and management. We apply Discrete Event Simulation integrated with a genetic algorithm to solve complex problems allowing stochastic variation to address this gap for MJF.

To overcome these challenges, using multiple approaches from in scheduling optimization and production planning for conventional manufacturing could be beneficial. Conventional manufacturing systems have been significantly improved using computer simulation and various optimization algorithms for design and operation. A seminal work in dynamic system simulation introduced Discrete Event Simulation (DES) to capture real-world variability for production scheduling [23]. DES is a powerful method for simulating solving complex queuing problems. DES has been successfully applied for bottleneck identification [24], and for manual assembly [25], where it was combined with operation management to solve a lot size problem to reduce the production cost. DES may be suitable for analyzing and improving AM process planning. Despite the general advantages of DES, it does not iteratively perform optimization in isolation. Instead a suitable optimization method is needed to provide further improvements.

In this respect, several optimization methods have improved productivity in factories by simplifying the problem and either solving it analytically or using a heuristic optimization method, simulated annealing (SA) [26], and ant colony optimization (ACO) [27]. Genetic algorithms (GA) represent an established and successful optimization search heuristic for a wide variety of problems [28]. A genetic algorithm is a heuristic optimization technique based on the natural process of evolution [29]. An advantage to GA is it is not myopic and can escape local minima [30]. GA has been successfully applied to optim izing and planning production systems, such as job scheduling [31], layout optimization [32], assembly line balancing [33], and resource utilization [34]. Conventional manufacturing settings have demonstrated the benefits of combining DES with optimization tools to solve some complex production problems. Shi et al. [35] used a genetic algorithm with DES to minimize the overall energy consumption, carbon emission, and makespan in a flow shop. Nili et al. [36] used DES and GA for maintenance plan schedules for repairing parts which led to minimizing the total cost of repairing projects. Rashid et al. [37] combined DES and GA in a construction factory to determine the number of workers on workstations that minimize total production time. Fumagalli, et al. [38] used this combination in aerospace manufacturing, they used DES to evaluate the solutions created by GA for solving the scheduling problem. Integrating DES with the GA is a useful approach that takes the benefits of both capturing dynamic variation in the system and a non-convex optimization search method. However, these methods have not yet been applied to flow-shop scheduling in additive manufacturing, which may provide insight into very high mix low volume (lot size 1) production systems. Given the success in other manufacturing settings, combining DES and GA approach may similarly improve planning and production system design in AM.

This work presents a simulation-optimization approach to solve a scheduling and bottleneck identification problem in additive manufacturing for a powder bed fusion-based process which can inform factory planning and operations. This research extends productivity improvement methods from conventional manufacturing into in the field of additive manufacturing showing the effectiveness of integrating GA with DES. Application of this method can enable companies to reduce their makespan and make informed decisions about equipment and staffing arrangements while reducing costs and improving customer service. This approach helps systematically analyze and identify the specific trade-off point between intended output performance and required machines and personnel as well as the contribution of intelligent scheduling. Additionally, it enhances the overall decision making process by capturing the dynamic environment which associated with high level of uncertainties [39]. A summary of the related literature of production planning in additive manufacturing and the using of DES and GA in conventional manufacturing are listed in Tables 1 and 2.

The structure of this paper is as follows. Section 2 described the focal AM problem setting, which is to be analysed and improved. Section 3 explains the research approach and how it led to the results in Sect. 4. Section 5 discusses the results and their implications to academia and industry. Section 6 concludes the paper and provides an outlook onto next steps.

2 Problem statement

The production flow in an additive manufacturing facility is defined as a scheduling problem where n jobs, $j \in J = 1, 2, \dots \dots, n$, enter the facility in a processing window and need to be completed. Each job is comprised of p parts of varying dimensions and numbers. For example j_1 contains 5 parts j_2 has two parts j_n has p parts as $p_{j_1} = 5$ and $p_{j_2} = 2$. Each job is restricted to a single batch. A batch is defined as a group of parts that are produced together in the 3D printer as a group at one time. The batch (or job) is then processed in subsequent stages as a collective unit.

Each job must progress through K processing stages. The sequence of stages is the same for each job, but the processing time for each stage varies depending on the features of the job (p parts, part dimensions, and part quantities). No restrictions have been made on the types of parts or their features other than those imposed by the machine used for fabrication. The processing time, $P_{j,k}$, of a given job, j at

Table 1 Summary of the related literature in additive manufacturing (papers 12-33) and conventional manufacturing (papers from 35 to 38)

Paper	Objectives	Methods
[12]	Minimizing production cost	Mixed integer linear programming
[13]	Minimizing production cost	Heuristic: ‘best-fit’ and ‘adapted best-fit’
[14]	Minimizing lateness	GA
[15]	Minimizing makespan	GA
[16]	Maximizing profit and resource utilization	Heuristic method
[17]	Minimize processing time	Multi-integer linear programming
[18]	Minimizing makespan	Heuristic algorithms: HC, TS, and SA
[19]	Minimizing makespan	GA combined with a heuristic
[20]	Optimize build orientation	Heuristic method
[21]	Minimizing lateness	Heuristic method
[22]	Optimizing throughput	Simulation model built on MATLAB
[24]	Bottleneck identification	DES
[25]	Minimizing cost	DES
[26–28]	Optimizing scheduling	Heuristic methods: SA, ACO, GA
[32, 33]	Optimizing layout, Assembly line balancing	GA
[35]	Minimizing energy, carbon emission, makespan	GA with DES
[36]	Minimizing cost of repairing projects	GA with DES
[37]	Minimizing total production time	GA with DES
[38]	Optimizing schedule	GA with DES

stage $k \in K$ is dependent on the number of parts and their design properties, such that $P_{j,k} = f_k(j)$. Processing time also includes any waiting time for worker availability, job setup, material transfer, and buffering.

The facility has m_k parallel machines in stage k such that any available machine, $M(k, i)$, in stage k can be assigned the next job. Define $M(k, i)$ as the machine with individual reference i in stage k. The presence and number of workers in the facility have an effect on the production speed. For some stages, a worker must be present to initialize the operation and for others, the worker must perform tasks throughout the duration of that processing stage. Thus, the layout of the facility and transit to and from each machine across all stages is included. Figure 1 illustrates a general problem structure, depicting multiple stages and multiple machines within each stage. Additionally, the figure depicts multiple jobs that need to be allocated to these machines.

The goal is to minimize the total processing time for all n jobs $j \in J = 1, 2, \ldots \ldots, n$ with processing time $(P_{j,k,i} = \sum_k P_{j,k,i})$ on $M(k, i)$ parallel machines $i \in I = 1, 2, \ldots m_k$, where the machine assigned to the job at each stage may vary. This minimization considers the uncertainty and the dynamic nature of the additive manufacturing environment such as workers' availability, shift time, facility layout, and traveling distances.

To assign jobs to parallel machines, define the binary variable $X_{ijk} = [0, 1]$ that takes the value 1 if the job is assigned to the machine in stage k and is 0, otherwise. Consider the following objective function to minimize makespan (w) [40]:

$$w \leq \sum_{k=1}^{K} \sum_{j=1}^{n} P_{jk} X_{ijk} \tag{1}$$

In Eq. 1 the makespan is defined as the completion time for all jobs where each job must be assigned to exactly one machine in each stage and must progress through all processing stages sequentially.

3 Methodology

The production simulation task is divided into two parts: (A) facility resourcing and (B) job scheduling optimization. For (A), the impacts of the number of workers and the number of capital equipment at the bottleneck location is sought. Then for each facility resourcing, a genetic algorithm is used to allocate jobs to minimize makespan addressing (B). This work examines an example facility that exclusively uses the powder bed fusion (PBF) machines produced by Hewlett Packard (HP). HP calls their PBF system multi-jet fusion (MJF), which uses a series of inkjet nozzles for ink deposition and a heat lamp for thermal fusing [8]. For (A) facility resourcing, a DES model of the factory is created and used to identify bottlenecks and evaluate worker and equipment utilization for each job schedule. The DES model captures additional information about the production to better represent real-world production issues than static heuristic models alone, such as worker shifts and travel and machine downtime for each scheduling evaluation. For (B) job scheduling problem is solved to minimize makespan using a genetic algorithm and test candidate solutions given each variation in the number of machines and workers.

The MJF workflow comprises five main stages: (1) slicing, (2) printing, (3) cooling, (4) unpacking, and (5) sandblasting. Slicing is the process of converting the 3D drawing into instructions that the printer understands. Printing is the actual process of depositing powder and forming the parts. Cooling refers to letting the printed parts cool down to maintain dimensional accuracy and surface finish, it occurs on the building unit which is the platform where the printed parts are constructed, which can be removed from the printer for further post-processing such as cooling and unpacking. Unpacking is the stage of extracting the printed parts from the powder in building units using a vacuum. Sandblasting refers to the Process of completely removing powder from printed parts.to achieve smooth and clean parts. This process is carried out by workers using a combination of

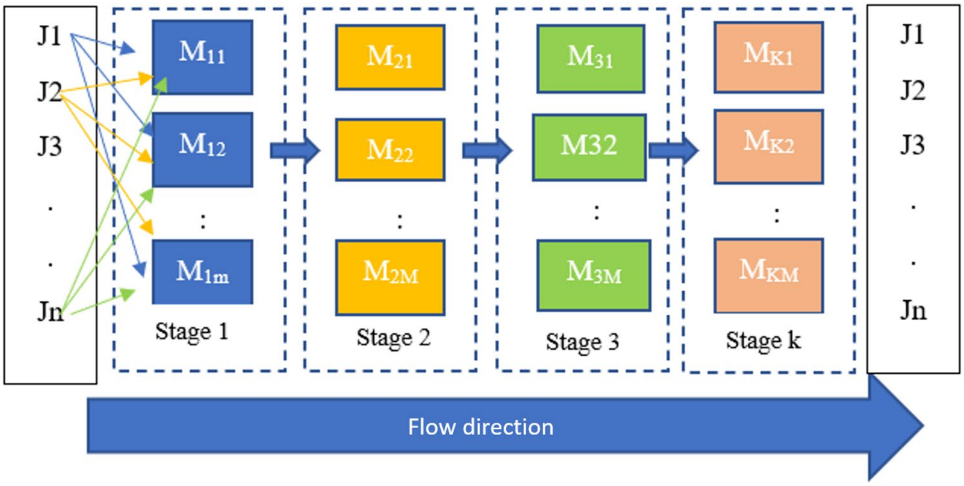


Fig. 1 Assignment structure of n jobs to m machines through k stages used to schedule production

airflow and sand. Other post-processing steps such as polishing and dying are possible but not included in this study. A digital model for production facilities was built to test the total production time of varying (a) parts, (b) workers, and (c) machines using Siemens Tecnomatix Plant Simulation [41] (Fig. 2). Among other simulation tools, this software was selected due to its advanced analysis and visualization capabilities, including bottleneck analysis, animation, and statistical analysis tools.

A test set of jobs was generated using historical data from a university additive manufacturing facility's HP Multi-Jet Fusion 4200 machine and randomly ordered. A range of factors influences the duration of each job stage. For instance, during unpacking and sandblasting, factors such as part geometry, packing density (number of parts), build height, material type, desired surface finish, and operator efficiency play a pivotal role in determining processing time. In this study, the number of parts was used as the primary

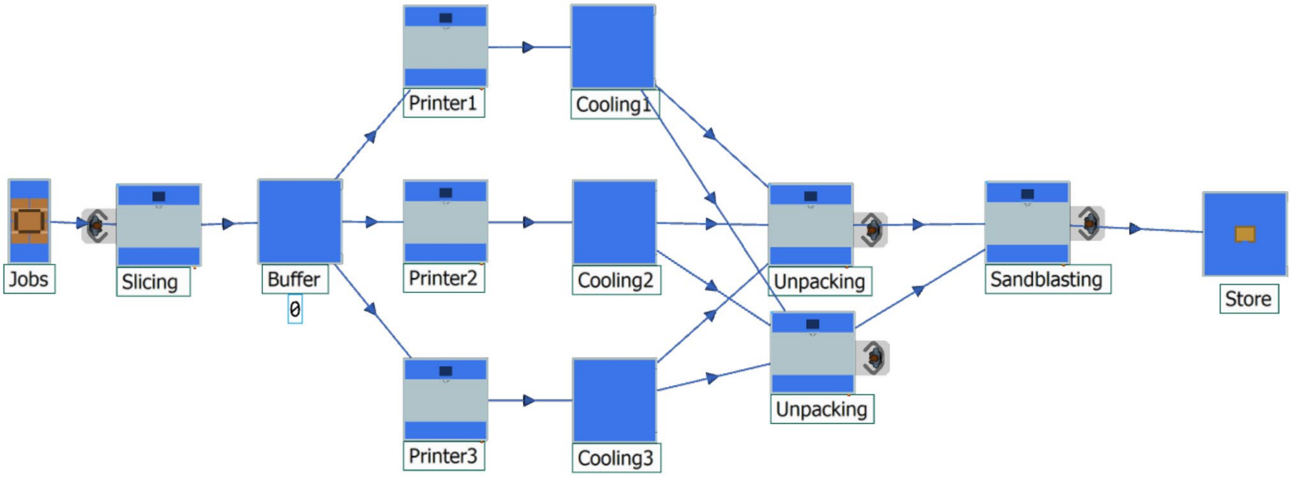


Fig. 2 The 3D digital model of the production line built in Tecnomatix Plant Simulation

Table 2 Default order of the generated list of 20 jobs for processing with processing times in each stage

Job	Number of parts (units)	Max Z height (inches)	P-Slicing (min)	P-Printing (min)	P-Unpacking (min)	P-Sand-blasting (min)
1	34	15	68	900	204	102
2	57	9	114	540	228	114
3	8	2	16	120	32	32
4	33	13	66	780	132	132
5	11	11	22	660	44	22
6	41	9	82	540	164	123
7	37	9	74	540	222	148
8	28	13	56	780	112	112
9	54	12	108	720	216	216
10	39	4	78	240	234	156
11	54	10	108	600	324	108
12	8	13	16	780	40	16
13	29	7	58	420	116	87
14	10	8	20	480	50	30
15	60	3	120	180	240	180
16	51	5	102	300	255	153
17	48	11	96	660	192	192
18	44	10	88	600	176	176
19	13	11	26	660	52	26
20	14	6	28	360	84	28

determinant of processing time assuming all the parts have the same geometry and complexity within the jobs and vary between jobs. To capture the variability in processing time for different parts' geometry, we have measured the processing time and created random values for (unpacking between 2 and 6 min and for sandblasting 2–4 min per part). These values are obtained from the university's facility(refer to the appendix to see the parts that have been used to compute the average time).

In addition to minimizing the makespan for each facility layout and available resources, the trade-off of changing the number of machines and workers is explored to find an optimal facility design for a range of production scales.

This study uses the following assumptions:

- The processing time for each job is known a priori.
- Setup time is independent of the job sequence and is considered part of the processing time.
- It is not possible for a machine to process more than one job at a time.
- Each job contains a different number of parts, and the complexity (difficulty in extraction and cleaning) of parts is identical within and between the jobs.
- During the scheduling period, all machines are available at the beginning.
- To account for the possibility of failures, a 1% failure rate is incorporated into each printing machine.
- Printers processing time depends mainly on the maximum Z height in each job [11]. Derived from HP's specifications, the computation of printing time entails multiplying the job's maximum height by a rate of 1 inch per hour [42]. This computation is established on a layer thickness of 0.003 inches and a duration of 8.3 s for each layer. The parts' heights in each job are generated randomly from 2 to 15 inches (the maximum height can be made on this printer).

The type of production here is classified as flow shop production where all jobs follow the same route to completion. First, a DES model was built using Siemens Tecnomatix Plant Simulation to represent the real production process of printing parts using multi-jet fusion printers including travel distance between machines. Next, a GA finds optimal or near-optimal schedules that lead to a decrease in the makespan, which improves productivity and reduces production costs. GA starts with defining Valid inputs to establish the search space. These limits define the population of possible solutions. The GA procedure randomly samples from this population of potential solutions to a problem. A random sample selected by the GA represents a 'chromosome, or input string. Each selected chromosome is evaluated according to its fitness function. The fitness function is the cost function to be minimized or maximized. The chromosomes that are highly fit are retained for the next generation of possible solutions. In this next generation, retained chromosomes 'reproduce' through a crossover process by exchanging parts of their selected inputs with other chromosomes. Often this crossover is a simple average weighting of the inputs. This process produces new solutions or "offspring" to the optimization problem, leveraging the high-performance traits of the parents. The best fitting chromosomes are retained for each generation until the convergence criteria or computational search time limit is reached [29]. In this work, GA searches for the best job order for each DES setup (Fig. 3). Each DES setup is manually constructed by adding or removing machines. This type of scheduling problem is classified as an NP-hard problem, meaning finding guaranteed optimal solutions is not feasible. Finding an optimal solution to the flow shop scheduling problem requires exploring a large number of possible scheduling combinations, which becomes increasingly difficult as the number of jobs and machines increases. The jobs can be assigned to the machines in (n!) sequences. Which is in the first case is $20! = 2.4e^{18}$, and $100! = 9e^{157}$ in the second case. The best solution given a maximum search time using GA is kept. The jobs have been assigned to the printers as the original sequence (S): $s_0 = j_1, j_2, j_3, \ldots\ldots, j_{n-1}, j_n$. The GA creates a population of different sequences, each candidate sequence is generated randomly $S = [s_1, s_2, s_3 \ldots\ldots s_z]$. The jobs are assigned to the available machines according to their sequence with the priority rule of first in first out (FIFO) (Fig. 1). The GA parameters are set according to trial and error (number and size of generation). Crossover and mutation are applied to produce additional new solutions, after that the value of these solutions is evaluated by computing their fitness value which is in the makespan in this work. DES calculates the makespan of each sequence. Then, the best children (solutions) are used to generate a new population. This process is repeated through a number of generations until the best solution is found. A pseudo-code of the GA

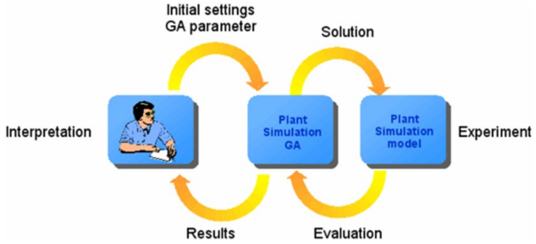


Fig. 3 Experimental flow for genetic algorithm schedule optimization in Tecnomatix Plant Simulation [41]

is provided Fig. 4(left) and in the flow chart of the GA; Fig. 4(right).

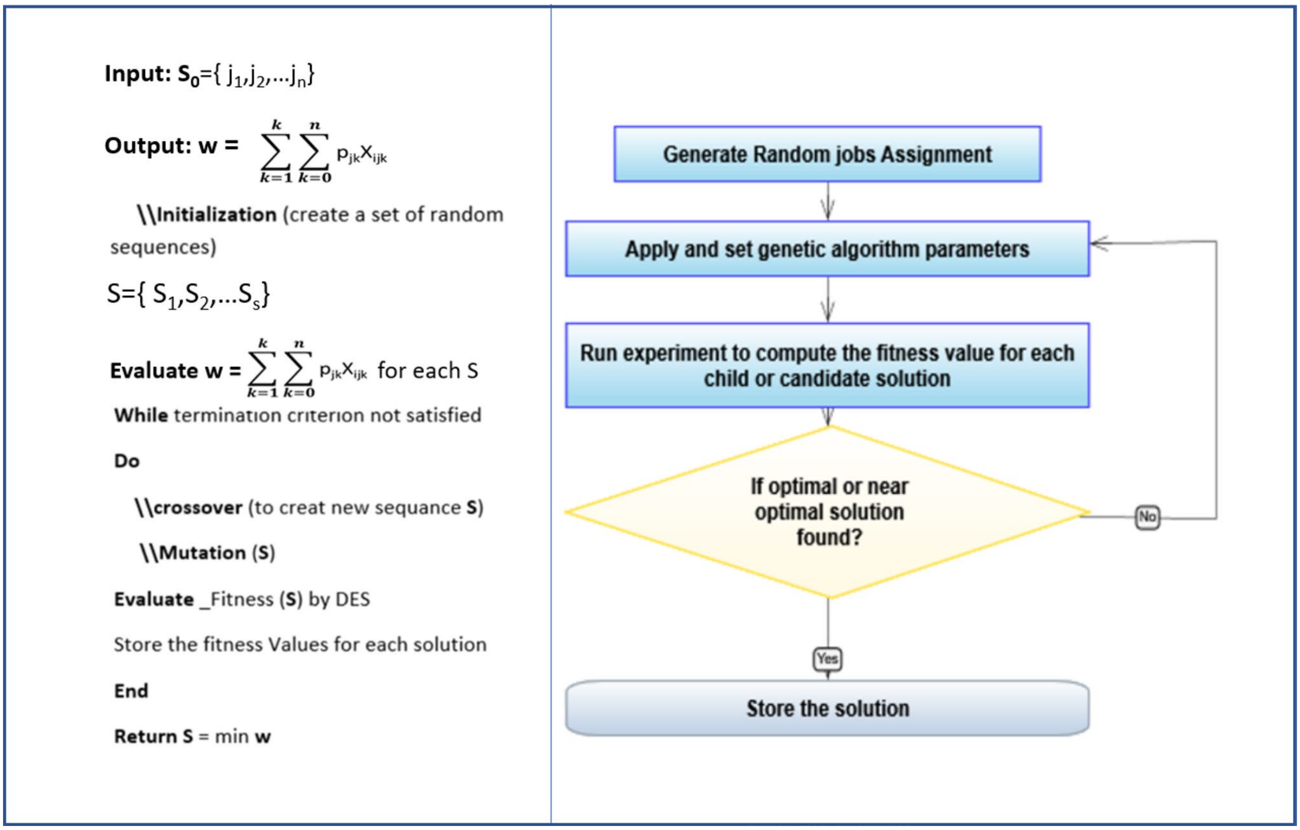
The objective is to minimize the total production time measured as the makespan, or total time to process all jobs in the schedule. The main two factors that affect makespan are job assignments and resource availability (workers, machines, etc.). The combination DES and GA method is applied to two example use cases of different sizes. The first smaller case study involves 20 jobs and the second is larger with 100 jobs. A digital model simulates the process and computes the makespan for several runs for two cases (small and large quantities). The makespan includes working time and non-working hours. There were two breaks (a short break of 15 min and a lunch break of 45 min) during the shift(8 hrs/day). The GA was run for 100 generations which was well after it converged.

4 Results

4.1 Case study 1: 20 jobs

For the first case study, 20 jobs were produced on 3 printing machines, with 1 building unit, 1 unpacking machine, 1 sandblasting, and 1 worker. Multiple experiments were conducted, starting with varying numbers of building units (the first highlighted column group in Table 3), followed by an increase in the number of workers (the second group). Then, the relation between workers and unpacking stations (third groups). Finally, the relation between the addition of extra sandblasting and various configurations of workers and the unpacking stations (fourth group).

According to Table 3, the makespan remains unaffected by the increase in the other resources such as building units, sandblasting, and unpacking stations when there is only one worker available. For example, adding an extra unpacking station kept the makespane constant(Exp1, Exp2). As these resources are not bottlenecking the work. However, with the addition of one extra worker, the makespan was significantly reduced by up to 30% (Exp1, Exp3), indicating that the bottleneck in this arrangement is the availability



```
Input: $S_0=\{ j_1,j_2,\ldots j_n\}$
Output: w = $\sum_{k=1}^{k}\sum_{k=0}^{n} p_{jk}X_{ijk}$
    \\Initialization (create a set of random
 sequences)
$S=\{ S_1,S_2,\ldots S_s\}$
Evaluate w = $\sum_{k=1}^{k}\sum_{k=0}^{n} p_{jk}X_{ijk}$ for each S
While termination criterion not satisfied
Do
    \\crossover (to creat new sequance S)
    \\Mutation (S)
Evaluate _Fitness (S) by DES
Store the fitness Values for each solution
End
Return S = min w
```

Generate Random jobs Assignment
Apply and set genetic algorithm parameters
Run experiment to compute the fitness value for each child or candidate solution
If optimal or near optimal solution found?
No
Yes
Store the solution

Fig. 4 The pseudo-code and the flow chart of the GA

Table 3 DOE

Exp (ID)	Workers (#)	Unpacking (#)	Sandblasting (#)	Makespan (dd:hh:mm:ss)	Makespan GA (dd:hh:mm:ss)	Reduction (%)
1	1	1	1	19:02:25:36	16:05:50:26	14
2	1	2	1	19:02:25:36	16:05:50:26	14
3	2	1	1	13:07:33:00	12:02:37:36	10
4	2	1	2	13:05:33:32	12:02:42:01	9
5	2	2	1	12:02:44:41	9:03:31:45	23
6	2	2	2	12:01:30:21	9:02:58:29	23
7	3	1	1	13:07:33:00	12:02:37:36	10
8	3	2	1	12:02:13:35	9:01:36:08	25
9	3	3	1	12:02:05:47	9:01:29:52	25
10	3	3	2	12:01:30:28	9:01:10:46	25

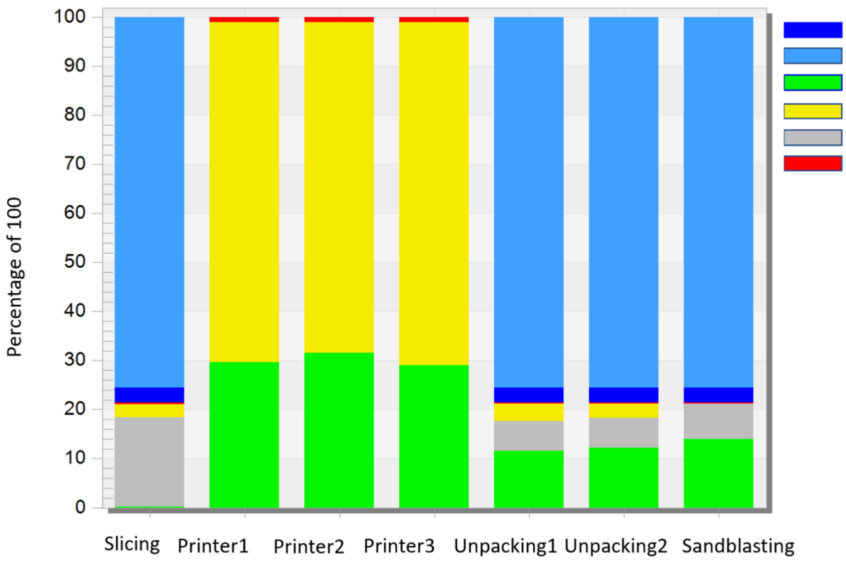
of workers. Additionally, introducing an unpacking station also dropped the makespan by 11% (Exp3, Exp5) further. While adding extra sandblasting improved the makespan by only 0.14%(Exp3, Exp4). In certain instances, the application of a genetic algorithm (GA) schedule leads to a more significant reduction in the makespan than adding extra workers. For instance, in Exp 5, the makespan was reduced to (9:03:31:45) after applying the GA, whereas in Exp 8, introducing an extra worker only reduced the makespan to (12:02:13:35). Figure 5 shows the resource statistic results after the addition of the unpacking station and one more worker for each workstation. It can be seen from the same figure that DES captures the dynamic nature of the system by taking into consideration the different operating phases of the workers and machines. For example, the worker break periods are deemed paused time, and ‘unplanned’ refers to time without a worker on shift. These variations in availability and the simulated job sequence give capture factory performance unique to the job sequence.

The makespan was notably improved after adding an extra unpacking station but more improvement was observed by adjusting the jobs schedule through the GA (Table 3). The GA ran for 100 generations, and each generation contained five candidate solutions. The GA largely converged after 16 generations, with only a small improvement found after 60 generations, as shown in Fig. 6.

The best performing assignment order for 3 workers, 3 printers, 2 unpacking stations, and 1 building unit(Exp 8) saves approximately 3 working days (a 25% reduction in manufacturing time),

The Sankey diagram was utilized to monitor each job, as demonstrated in Fig. 7, which illustrates the material flow through the shop. The diagram uses colors to indicate the route taken by each job. No obvious trends were identified from viewing these job flows.

Fig. 5 Continuous processing statistics of each station by activity in the presence of two workers



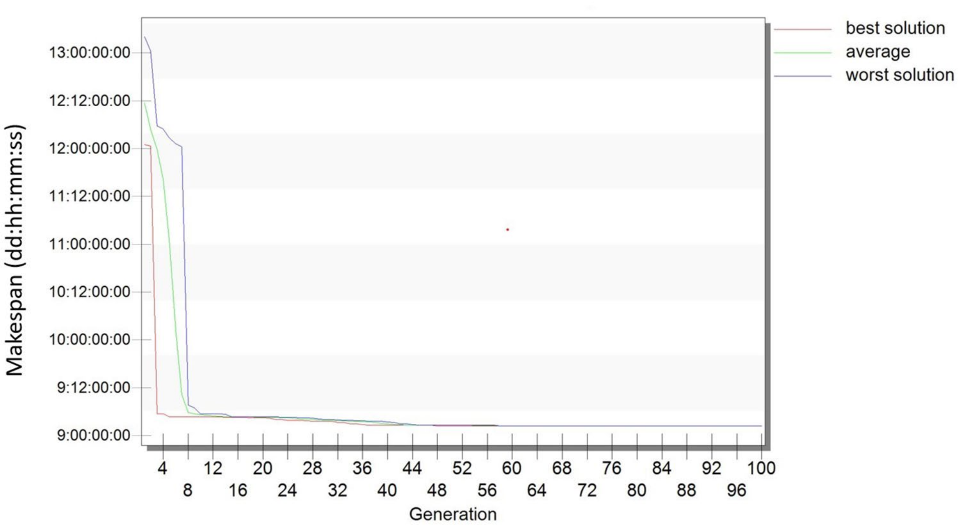


Fig. 6 Example performance curve of genetic algorithm in reducing the makespan with increasing generations

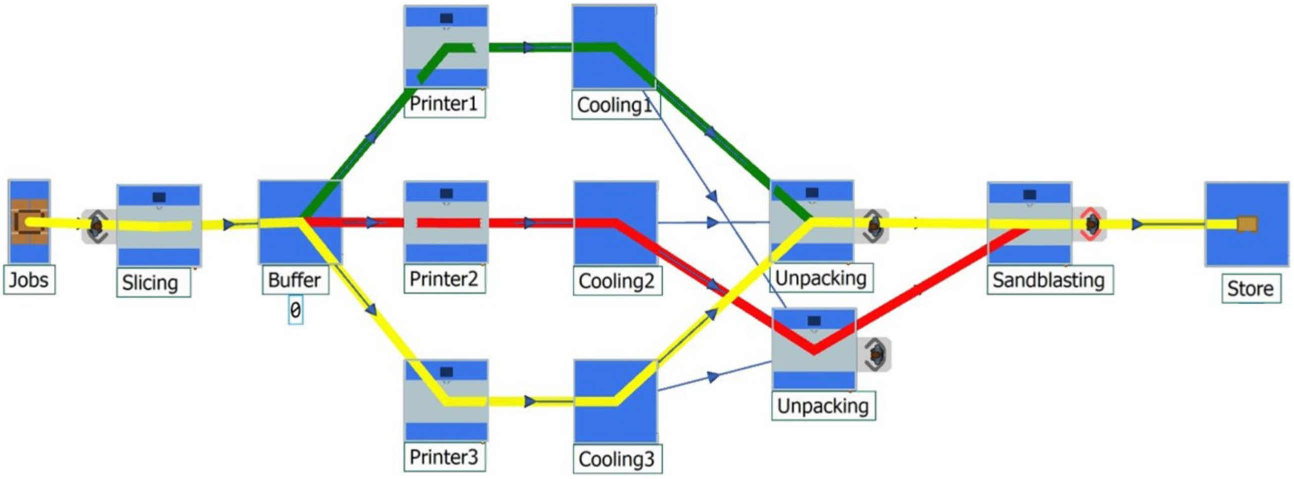


Fig. 7 Monitoring jobs' assignments using Sankey diagram

4.2 Case study 2: 100 jobs

The second case study had a larger number of resources to produce 100 jobs. The experiment was set up with 3 printers, 3 unpacking, and 3 sandblasting stations. Similar to the first case study, several runs were executed to test the system performance (makespan) with varying resource (number of workers and building units) availability (Tables 4 and 5). Which shows the makespan results at different setups before (Table 4) and after (Table 5) applying the genetic algorithm. The results of experiments that were conducted to determine the optimal number of building units for each printer in the case of the existence of one worker revealed that two building units per printer resulted in a 4-day saving in production. The addition of extra building units is observed to be useful in cases where there are three or more workers, with a maximum addition of four units (Fig. 8). This shows an interconnected relationship between the worker and building unit quantities. With regards to the number of workers, the results show that 1 worker is not enough to handle this amount of production leading to a high makespan. Adding just 1 worker dropped the makespan significantly, to almost half 45%. In the 20 job case, three workers are sufficient, and adding more workers does not significantly improve the performance. This finding can be utilized to determine the number of employees required based on the number of unpacking stations and the number of jobs. These results are summarized in Fig. 8. Which shows makespan values

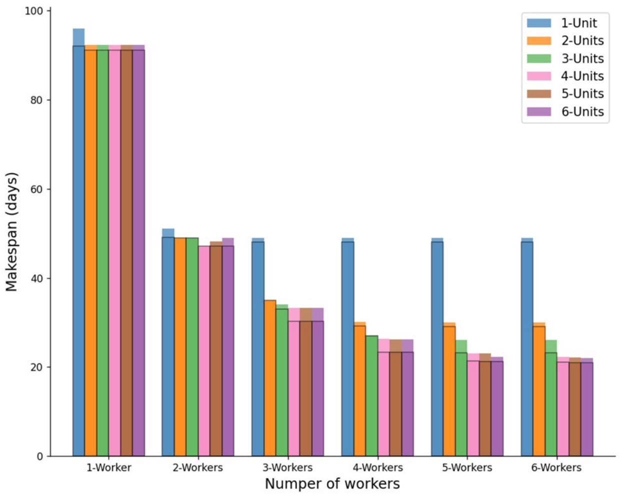


Fig. 8 Makespan output according to workers(W) and unit(U) numbers before(without outlines) and after (with outlines)

according to the number of resources and job sequences. From the same figure, it can be seen that in most cases there is an extra improvement in makespan when the parts are rescheduled using GA (the outlined bars).

For a fixed number of workers, the maximum improvement from the combined DES and GA approach is a 78% (Table 6). This plateaus for 5 workers and 4 units for the 100 job case. In order to assess how enlarging job sizes affects the performance of GA in minimizing makespan, we conducted a series of experiments using a consistent configuration (3 workers, 3 printers, 1 building unit, 3 unpacking stations, and 3 sandblasting stations) while varying the job sizes (20, 40, 60, 80, and 100 jobs). Figure 9 shows that increasing the problem size leads to greater enhancements in GA performance, as indicated by the increased disparity between the best and worst solutions.

Table 4 Makespan values using the default job ordering in Table 7

Workers	Units					
	1	2	3	4	5	6
1	96:01:52:08	92:07:29:54	92:07:29:54	92:07:29:54	92:07:29:54	92:07:29:54
2	51:03:47:34	49:00:38:49	49:00:26:06	47:05:02:11	48:05:43:03	49:00:34:46
3	49:01:52:08	35:00:37:02	34:01:52:08	33:05:32:15	33:05:32:14	33:05:32:14
4	49:01:52:08	30:02:49:22	27:02:29:48	26:07:25:38	26:05:46:52	26:05:51:51
5	49:01:52:08	30:01:52:00	26:02:07:11	23:02:50:12	23:00:04:34	22:05:45:04
6	49:01:52:08	30:00:37:00	26:01:52:00	22:05:57:16	22:02:29:52	22:00:55:28

Table 5 Makespan values after applying the genetic algorithm updating the results from Table 4

Workers	Units					
	1	2	3	4	5	6
1	92:02:28:27	91:04:55:01	91:04:55:01	91:04:55:01	91:04:55:01	91:04:55:01
2	49:03:48:53	49:00:38:49	49:00:26:06	47:05:02:11	47:04:54:23	47:04:44:45
3	48:01:04:00	35:00:37:00	33:00:06:35	30:07:13:21	30:07:33:13	30:07:33:55
4	48:01:04:00	29:04:12:17	27:00:07:03	23:07:20:06	23:07:22:57	23:07:26:00
5	48:01:04:00	29:01:28:06	23:04:47:00	21:07:31:58	21:05:47:07	21:04:31:27
6	48:01:04:00	29:01:28:00	23:04:47:00	21:01:44:00	21:00:34:36	21:00:38:14

Table 6 Makespan improvement from both DES and GA combined as a percentage (%)

Workers	Units					
	1	2	3	4	5	6
1	4	5	5	5	5	5
2	49	49	49	51	51	51
3	50	64	66	68	68	68
4	50	70	72	76	76	76
5	50	70	76	78	78	78
6	50	70	76	78	78	78

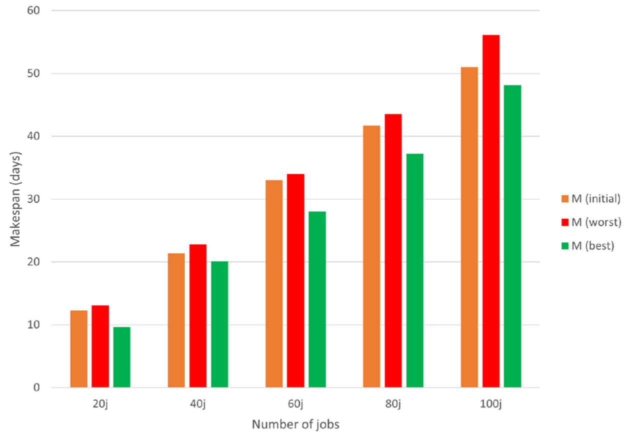


Fig. 9 The value of makespan at different sizes of jobs showing the initial value of makespan (M initial) and the best and worst solutions obtained by GA (M best and M worst)

5 Discussion

A simulation model was developed to explore the design of a powder bed fusion factory using a combined DES and GA method. The developed method of combining DES and GA heuristic optimization strategy showed improvement opportunities from both industrial resource changes and scheduling improvements. This highlights the multiple dimensions of production system design addressed by the combined approach and not provided by either DES or GA in isolation. The two case studies illustrate these opportunities and give guidance on scaling up PBF to factory-level production. While the first case study considered a small set of jobs (20), the second addressed a larger set of jobs (100). The objective was to minimize total production time (makespan). This approach to production planning and scheduling in additive manufacturing offers a path for the scale-up of production volume. The introduction of this approach overcomes problems associated with other scheduling works for AM factory planning, which rely on assumptions of static systems and modeling simplifications that are often too restrictive to capture real-world scenarios. The experiments show the necessary number of workers is contingent upon the quantity of unpacking stations and building units present in the facility. This is due to the fact that the unpacking stations necessitate a manual unpacking process, which can obstruct the flow that comes from the printers. Moreover, if the number of workers falls short of the desired number, adding extra resources may prove to be redundant and lead to increased costs without improving the makespan. For instance, when a single worker is already present and adding more building units (2–6) will not lead to any improvement (Table 4, first row). Because the building units in this case are not the bottleneck. However, the addition of extra units could be more useful in cases where there are three or more workers, with a maximum addition of four units. This is because the workers are responsible for transporting the produced parts to the subsequent stations and processing them. If there are insufficient workers, adding more building units will not add value.

The example facility used HP Multi-Jet Fusion 4200 machines with all major production tasks included from file preparation through to cleaning printed parts. This example facility can be used as a representative of a polymer factory with many parallels to metal machines, The biggest improvements from a single resource addition are observed when adding the second worker in all production unit arrangements with up to 46% reduction in makespan. This shows the power of DES in bottleneck identification. From there, additional improvement scenarios were tested. The performance improvements peak at a 78% makespan reduction for 5 workers and 4 building units, where the addition of more workers or building units has no impact. This is caused by the building units acting as a buffer supplying the workers with parts that are ready to be processed at the next stations. The extra worker over the number of building units allows all building units to be unpacked while some other operations are simultaneously being performed. Thus, further addition of workers or building units is not advantageous for the tested number of jobs suggesting a decreasing marginal benefit with the addition of more capital resources as the factory scales up.

A failure rate is included which adds unplanned downtime that is not captured in static analyses but offers the genetic algorithm some additional search space. In a continuous job arrival mode using the same job list on loop, the resource statistical results show that the printers have a low working time due to blockages (Fig. 5). The main reason for this blockage is the cooling time is about 24 hrs which is done on the building unit so the printer can't be run until the parts are unpacked and removed or using a new building unit. The printers do not have unplanned or paused time because they could be run without a worker present, i.e. 24 h per day. However, printer starts are limited by the availability of a worker to commence a job and downstream processing of printed parts. This suggests that multiple shifts could increase production and reduce or eliminate the high percentage of the blockage.

The GA schedule perturbation showed improved makespan for most of the experiments, up to 11%. However, in three cases it did not find a better job order, such as when there were 2 workers with 2, 3, or 4 units. For these scenarios, the default job order may be a sufficiently good order given the worker availability constraints. The initial job order was randomly determined and is not the worst possible job ordering, so it may also be that the random order is a pretty good solution. It is possible that a much longer search time may find some additional improvement in the job order. Processing time and complexity of the produced parts were assumed to not vary with individual part geometry, therefore the only factors that affected the processing time at any station were the number of parts and their maximum build height in the building unit. This provided some variation in job processing time sufficient for this level of simulation. This assumption is easily adjusted for specific use cases by future practitioners. The effectiveness of combining discrete event simulation with a genetic algorithm may vary in magnitude for other equipment or companies.

The combined DES and GA method allows “what-if" scenarios to be evaluated, in addition to identifying bottlenecks in an existing line. Based on the results, integrating scheduling with bottleneck identification gives decision makers the capability to identify the weak points or vulnerabilities in their production line which can aid in deciding which type of resource investment will have the biggest impact. As in some cases adding extra resources of one kind leads to adding additional cost without improving system productivity. This framework will help decision-makers design and maximize production in current and future factories for the AM industry. Existing factories can use this approach to understand their system and needs in a virtual environment without interrupting the production line and to test different scenarios before implementation.

6 Conclusion

This work introduces a simulation optimization-based analysis to support additive manufacturing planning and scheduling for factory production based on combining discrete event simulation (DES) and a genetic algorithm (GA). DES was used first as an evaluation tool to analyze the performance of different resource configurations and to identify bottlenecks. Subsequently, GA was used to further improve the makespan by reordering the available jobs. The case studies revealed that the bottleneck was often the worker availability. A maximum reduction in makespan of 78% was possible with added workers and building units. In both case studies, adding an extra worker led to a significant reduction in makespan by 40-45%. The DES experiments identified the makespan reduction with either the addition of more workers or more building units, or a combination. Integrating GA into the DES model for scheduling introduced a further reduction in the makespan by up to 11%. However, it also showed adding extra resources may prove to be redundant and lead to increased costs without improving the makespan if the number of workers falls short of the desired number. Instead, it helps to find the right balance between the number of workers and the number of different equipment for a given job set. This shows the effectiveness of this hybrid approach as a production planning tool for dynamic production systems that can help decision-makers to determine the proper number of resources to deliver parts in time and how to allocate limited resources in the most efficient and flexible way. From an academic perspective, this paper’s contribution is the successful transfer of a combined DES-GA method to AM. Aside from enhancing the understanding of manufacturing flow related AM challenges, it also builds a better understanding of how these could be overcome. As this is the first application of DES and GA to factory level production in powder bed fusion additive manufacturing, further research is needed. Future studies could include more primary data and additional factors that may impact processing time at each station or increase the variability of arrival rates of jobs. To validate the new method and its results, it is also important to physically implement and test original and improved setups.

Appendix A. Extended data

See Table 7.

Table 7 Processing times for the 100 jobs in each stage in minutes (min)

Job	Number of parts (units)	Max Z height (inches)	P-Slicing (min)	P-Printing (min)	P-Unpacking (min)	P-Sand-blasting (min)
1	39	4	78	240	195	117
2	60	3	120	180	300	180
3	33	13	66	780	165	99
4	60	3	120	180	300	180
5	57	9	114	540	285	171
6	51	5	102	300	255	153
7	48	11	96	660	240	144
8	33	13	66	780	165	99
9	39	4	78	240	195	117
10	8	2	16	120	40	24
11	41	9	82	540	205	123
12	57	9	114	540	285	171
13	10	8	20	480	50	30
14	10	8	20	480	50	30
15	41	9	82	540	205	123
16	8	2	16	120	40	24
17	8	13	16	780	40	24
18	14	6	28	360	70	42
19	14	6	28	360	70	42
20	51	5	102	300	255	153
21	48	11	96	660	240	144
22	37	9	74	540	185	111
23	10	8	20	480	50	30
24	33	13	66	780	165	99
25	29	7	58	420	145	87
26	60	3	120	180	300	180
27	44	10	88	600	220	132
28	48	11	96	660	240	144
29	54	12	108	720	270	162
30	41	9	82	540	205	123
31	10	8	20	480	50	30
32	14	6	28	360	70	42
33	41	9	82	540	205	123
34	28	13	56	780	140	84
35	28	13	56	780	140	84
36	28	13	56	780	140	84
37	11	11	22	660	55	33
38	41	9	82	540	205	123
39	39	4	78	240	195	117
40	33	13	66	780	165	99
41	37	9	74	540	185	111
42	34	15	68	900	170	102
43	44	10	88	600	220	132
44	57	9	114	540	285	171
45	29	7	58	420	145	87

Table 7 (continued)

Job	Number of parts (units)	Max Z height (inches)	P-Slicing (min)	P-Printing (min)	P-Unpacking (min)	P-Sand-blasting (min)
46	28	13	56	780	140	84
47	54	12	108	720	270	162
48	28	13	56	780	140	84
49	48	11	96	660	240	144
50	34	15	68	900	170	102
51	54	10	108	600	270	162
52	29	7	58	420	145	87
53	13	11	26	660	65	39
54	33	13	66	780	165	99
55	44	10	88	600	220	132
56	54	12	108	720	270	162
57	11	11	22	660	55	33
58	60	3	120	180	300	180
59	8	13	16	780	40	24
60	8	2	16	120	40	24
61	54	12	108	720	270	162
62	8	2	16	120	40	24
63	8	13	16	780	40	24
64	11	11	22	660	55	33
65	44	10	88	600	220	132
66	13	11	26	660	65	39
67	57	9	114	540	285	171
68	8	13	16	780	40	24
69	13	11	26	660	65	39
70	51	5	102	300	255	153
71	51	5	102	300	255	153
72	11	11	22	660	55	33
73	29	7	58	420	145	87
74	39	4	78	240	195	117
75	37	9	74	540	185	111
76	54	10	108	600	270	162
77	54	10	108	600	270	162
78	29	7	58	420	145	87
79	54	12	108	720	270	162
80	14	6	28	360	70	42
81	48	11	96	660	240	144
82	10	8	20	480	50	30
83	34	15	68	900	170	102
84	54	10	108	600	270	162
85	8	13	16	780	40	24
86	14	6	28	360	70	42
87	51	5	102	300	255	153
88	13	11	26	660	65	39
89	57	9	114	540	285	171
90	44	10	88	600	220	132

Table 7 (continued)

Job	Number of parts (units)	Max Z height (inches)	P-Slicing (min)	P-Printing (min)	P-Unpacking (min)	P-Sand-blasting (min)
91	8	2	16	120	40	24
92	13	11	26	660	65	39
93	34	15	68	900	170	102
94	54	10	108	600	270	162
95	37	9	74	540	185	111
96	11	11	22	660	55	33
97	34	15	68	900	170	102
98	39	4	78	240	195	117
99	60	3	120	180	300	180
100	37	9	74	540	185	111

Appendix B

Unpacking and sandblasting processing time for different parts that used. Example of these parts illustrated in Fig. 10 and the processing time in Table 8.

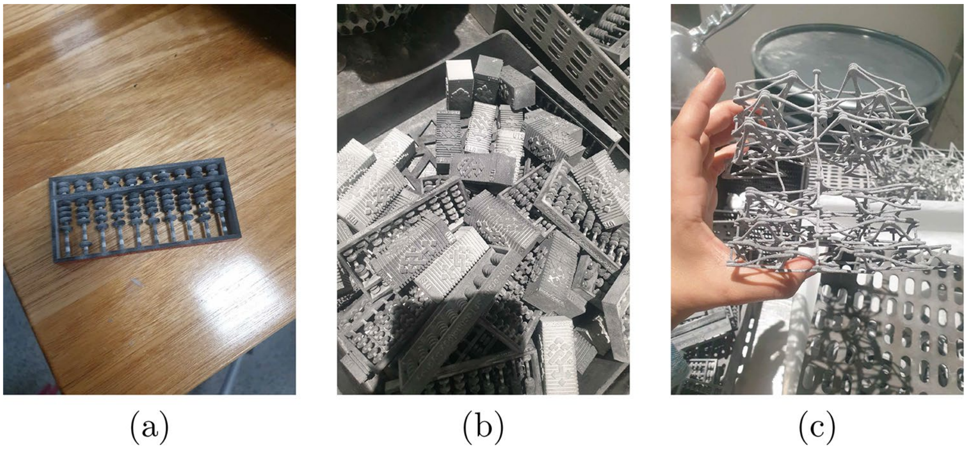


Fig. 10 Example parts used to calibrate processing time in production steps

Table 8 Processing time (min) Reading for unpacking and sandblasting process for three different parts in Fig. 10

Part ID	Process	1	2	3	4	5	6	Average (min)
(a)	unpacking	1.8	1.9	2	2.3	2.1	2.1	2.03
	sandblasting	2.1	2	2.2	2.1	2.3	2	2.11
(b)	unpacking	3.4	3.2	3	3	2.1	3.2	2.98
	sandblasting	3.6	3.3	3.2	3.5	3.1	3.1	3.3
(c)	unpacking	5	5.5	5.4	6	6.1	6.3	5.71
	sandblasting	4	4.5	5	5.2	4.7	3.9	4.5

Author Contributions Ruba Al-zqebah: Conceptualization, Methodology, Software, Investigation, Validation, Formal analysis, Data curation, Writing original draft. Matthias Guertler: Conceptual revision, Writing-review & editing, Supervision. Lee Clemon: Conceptualization, Methodology, Resources, Writing-review & editing, Supervision. AI tools use-No AI or AI-assisted technologies prepared any portion of this work. The authors take full responsibility for the content in the publication.

Funding Open Access funding enabled and organized by CAUL and its Member Institutions.

Data availability All data is available within the manuscript or supplementary materials.

Materials availability Not applicable.

Code availability Not applicable.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Ethics approval and consent to participate Not applicable.

Consent for publication The authors consent to this publication being published

References

1. ISO, ASTM (2021) Additive manufacturing-general principles-terminology. Standard, International Organization for Standardization, Geneva
2. Schäfer L, Klenk F, Maier T, Zehner M, Peukert S, Linzbach R, Treiber T, Lanza G (2024) A systematic approach for simulation-based dimensioning of production systems during the concept phase of factory planning. Prod Eng 1-13
3. Cai C, Tey WS, Chen J, Zhu W, Liu X, Liu T, Zhao L, Zhou K (2021) Comparative study on 3d printing of polyamide 12 by selective laser sintering and multi jet fusion. J Mater Process Technol 288:116882. https://doi.org/10.1016/j.jmatprotec.2020.116882
4. Fast Radius: MJF vs. FDM: what you need to know. (March 30, 2022). https://www.fastradius.com/resources/mjf-vs-fdm/. Accessed 10 Aug 2023
5. HP: HP multi jet fusion 3D printing technology for end of arm tooling. (November 8, 2021). https://www.roboticstomorrow.com/article/2021/07/hp-multi-jet-fusion-3d-printing-technology-for-end-of-arm-tooling/17230. Accessed 10 Aug 2023
6. Cimquest: HP Multi Jet Fusion 3D Printing Streamlines Machinery and Production Lines. (2023). https://cimquest-inc.com/hp-multi-jet-fusion-3d-printing-streamlines-machinery-and-production-lines/. Accessed 10 Aug 2023
7. iOrthotics: case study: how HP's 3D printing technology helps iOrthotics develop products that benefit people everywhere. (July 24, 2018). https://www.iorthotics.com.au/case-study-how-hps-3d-printing-technology-helps-iorthotics-develop-products-that-benefit-people-everywhere/. Accessed 10 Aug 2023
8. HP: HP Industrial 3D printers-leading the commercial 3D printing revolution. https://www.hp.com/us-en/printers/3d-printers.html Accessed 20 Mar 2023
9. Simons M (2018) Additive manufacturing-a revolution in progress? Insights from a multiple case study. Int J Adv Manuf Technol 96(1):735–749. https://doi.org/10.1007/s00170-018-1601-1. (**Accessed 2023-03-23**)
10. Ngo TD, Kashani A, Imbalzano G, Nguyen KT, Hui D (2018) Additive manufacturing (3d printing): a review of materials, methods, applications and challenges. Compos B Eng 143:172–196
11. Aloui A, Hadj-Hamou K (2021) A heuristic approach for a scheduling problem in additive manufacturing under technological constraints. Comput Ind Eng 154:107115
12. Ransikarbum K, Ha S, Ma J, Kim N (2017) Multi-objective optimization analysis for part-to-printer assignment in a network of 3d fused deposition modeling. J Manuf Syst 43:35–46
13. Li Q, Kucukkoc I, Zhang DZ (2017) Production planning in additive manufacturing and 3d printing. Comput Oper Res 83:157–172
14. Kucukkoc I, Li Q, He N, Zhang D (2018) Scheduling of multiple additive manufacturing and 3d printing machines to minimise maximum lateness. Twent Int Work Semin Prod Econ 1:237–247
15. Zhang Y, Bernard A, Harik R, Karunakaran K (2017) Build orientation optimization for multi-part production in additive manufacturing. J Intell Manuf 28(6):1393–1407
16. Kucukkoc I, Li Q, Zhang D (2016) Increasing the utilization of additive manufacturing and 3d printing machines considering order delivery times. In: 19th International Working Seminar on production economics, pp 195–201
17. Kucukkoc I (2019) Milp models to minimise makespan in additive manufacturing machine scheduling problems. Comput Oper Res 105:58–67
18. Dvorak F, Micali M, Mathieug M (2018) Planning and scheduling in additive manufacturing. Intel Artif 21(62):40–52
19. Zhang J, Yao X, Li Y (2020) Improved evolutionary algorithm for parallel batch processing machine scheduling in additive manufacturing. Int J Prod Res 58(8):2263–2282
20. Oh Y, Zhou C, Behdad S (2018) Production planning for mass customization in additive manufacturing: build orientation determination, 2d packing and scheduling. In: International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, vol. 51753, pp. 02–03033. American Society of Mechanical Engineers
21. Chergui A, Hadj-Hamou K, Vignat F (2018) Production scheduling and nesting in additive manufacturing. Comput Ind Eng 126:292–301
22. Stittgen T, Schleifenbaum JH (2021) Simulation of utilization for LPBF manufacturing systems. Prod Eng 15:45–56
23. Hodoň R, Kovalskỳ M, Gregor M, Grznár P (2018) New approaches in production scheduling using dynamic simulation. IOP Conf Ser Mater Sci Eng 393:012023
24. Birgersson E, Eriksson D (2022) Production improvement using discrete event simulation: case study of Volvo Penta engine production line, p 120

25. Budde L, Liao S, Haenggi R, Friedli T (2022) Use of des to develop a decision support system for lot size decision-making in manufacturing companies. Prod Manuf Res 10(1):494–518
26. Che Y, Hu K, Zhang Z, Lim A (2021) Machine scheduling with orientation selection and two-dimensional packing for additive manufacturing. Comput Oper Res 130:105245
27. Liu W, Chen L, Mai G, Song L (2020) Toolpath planning for additive manufacturing using sliced model decomposition and metaheuristic algorithms. Adv Eng Softw 149:102906
28. Fera M, Fruggiero F, Lambiase A, Macchiaroli R, Todisco V (2018) A modified genetic algorithm for time and cost optimization of an additive manufacturing single-machine scheduling. Int J Ind Eng Comput 9(4):423–438
29. Chaudhry S, Luo W (2005) Application of genetic algorithms in production and operations management: a review. Int J Prod Res 43(19):4083–4101
30. Sivanandam S, Deepa S, Sivanandam S, Deepa S (2008) Genetic algorithms. Springer
31. Adnyana AAP, Widiartha IM, Muliantara A, Astuti LG, Raharja MA, Darmawan IDMBA (2022) Implementasi metode hybrid particle swarm optimization dan genetic algorithm pada penjadwalan job shop scheduling. Jurnal Elektronik Ilmu Komputer Udayana 11(3):539. https://doi.org/10.24843/JLK.2023.v11.i03.p09
32. Saleh NFB, Hussain ARB (2008) Genetic algorithms for optimizing manufacturing facility layout, p 1–6
33. Lee S, Khoo L, Yin X (2000) Optimising an assembly line through simulation augmented by genetic algorithms. Int J Adv Manuf Technol 16:220–228
34. Renna P (2017) A decision investment model to design manufacturing systems based on a genetic algorithm and monte-carlo simulation. Int J Comput Integr Manuf 30(6):590–605
35. Shi L, Guo G, Song X (2021) Multi-agent based dynamic scheduling optimisation of the sustainable hybrid flow shop in a ubiquitous environment. Int J Prod Res 59(2):576–597
36. Nili MH, Taghaddos H, Zahraie B (2021) Integrating discrete event simulation and genetic algorithm optimization for bridge maintenance planning. Autom Constr 122:103513
37. Rashid K, Louis J, Swanson C (2020) Optimizing labor allocation in modular construction factory using discrete event simulation and genetic algorithm. In: 2020 Winter Simulation Conference (WSC), pp. 2569–2576. IEEE
38. Fumagalli L, Negri E, Sottoriva E, Polenghi A, Macchi M (2018) A novel scheduling framework: Integrating genetic algorithms and discrete event simulation. Int J Manag Decis Mak 17(4):371–395
39. Sun C (2020) Research on investment decision-making model from the perspective of “internet of things+ big data’’. Futur Gener Comput Syst 107:286–292
40. Rudy J (2021) Parallel makespan calculation for flow shop scheduling problem with minimal and maximal idle time. Appl Sci 11(17):8204
41. Software SDI (2022) Tecnomatix Plant Simulation. https://ngo.sw.siemens.com/en-US/product/tecno-plant-sim/. Accessed 13 Feb 2023
42. Merkezi DP (2023) HP Multi Jet Fusion 4200 3D Printer. https://3dprintmerkezi.com/EN/p/3/hp-multi-jet-fusion-4200-3d-printer. Accessed 4 Aug 2023

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.