



Analysis of metaheuristic optimisation techniques for simulated matrix production systems

Martin Benfer¹ · Valentin Heyer¹ · Oliver Brützel¹ · Christoph Liebrecht² · Sina Peukert¹ · Gisela Lanza¹

Received: 17 May 2023 / Accepted: 4 September 2023
© The Author(s) 2023

Abstract

Increasing demand for individualised products has led to the concept of mass customisation, combining high product variety with production efficiency coming along with mass production. Companies are moving to matrix production systems with complex product flows for mass customisation. One challenge in such systems is the determination of optimal system configurations to fulfil future demands while minimising production costs. An approach to determine the ideal configuration is to use metaheuristics like genetic algorithms or simulated annealing to optimise simulation models. However, it is unclear which methods are ideally suited to finding the best solutions. This contribution compares the performance of genetic algorithms and simulated annealing when optimising the configuration of a company-specific matrix production system using discrete event simulation. The methods are evaluated using different objective functions. For the genetic algorithm, different observation strategies are also tested. Overall, the simulated annealing approach delivers better results with shorter solution times. The contributing factors leading to the different results are discussed, and areas for future research are pointed out.

Keywords Matrix production system · Combinatorial optimisation problem · Genetic algorithm · Simulated annealing

1 Introduction

Global competition and technological progress have led to ever shorter product cycles in many industries in recent decades. Combined with the rising relevance of the individualisation of products, this results in higher levels of variance in manufacturing. New concepts were developed to increase part flexibility levels in production systems [1]. These concepts need to find an optimal tradeoff between cost efficiency and flexibility. One such concept is the so-called matrix production system, consisting of cycle-time-independent production cells with product-neutral equipment and flexible transportation that can be scheduled ad hoc [2, 3]. Although this system leads to a combined use of resources, shorter transport routes compared to workshop production, and short, intermediate storage, it requires a high

capital investment, a large spatial footprint, and increased coordination and control requirements [4]. One of the challenges in matrix production systems is investment planning [5]. Due to interdependencies between machines, conventional methods to optimise the production system, like mixed integer linear programming, are only possible with severe disadvantages. Alternatively, the production system can be modelled using discrete event simulation (DES) and agent-based simulation (ABS). Simulation can be used to predict the behaviour of system configurations given specific production programs. Metaheuristics can be used with simulation to create systems that predict behaviour and prescribe desirable configurations [6]. Metaheuristics like genetic algorithms (GA), simulated annealing (SA) and tabu search can optimise a broad range of models. However, the efficiency of different heuristics varies significantly depending on their specific application [7]. Furthermore, as heuristics cannot determine global optima, the quality of the solution may depend on the chosen method [7]. Thus, apt metaheuristics are crucial for successful simulation-based optimisation in matrix production systems.

This paper contributes a case study on using metaheuristics and simulation for configuring a matrix production system to the body of knowledge. Two different

✉ Martin Benfer
martin.benfer@kit.edu

¹ wbk Institute of Production Science, Karlsruhe Institute of Technology (KIT), Kaiserstr. 12, 76131 Karlsruhe, Germany

² Siemens Digital Industries, Siemens AG, Östliche Rheinbrückenstr. 50, 76187 Karlsruhe, Germany

metaheuristics, GA and SA, are tested and compared. Furthermore, the parametrisation of soft constraints and different observation strategies to cope with stochastic effects are examined. This work uses the simulation model of an industrial pc manufacturer implemented in Tecnomatix Plant Simulation to compare GA and SA regarding result quality and processing time.

The remainder of this contribution is structured as follows: Sect. 2 provides an overview of existing literature on the subject and its relationship with the present work. Section 3 presents the underlying use case, the simulation model and the metaheuristic algorithms used. The experimental results are described in Sect. 4 and discussed in Sect. 5. Finally, the work is summarised in Sect. 6.

2 Related work

2.1 Matrix production

The term matrix production or matrix manufacturing has only recently seen more attention. Matrix production systems address the challenges of high volume, high customisation production, which typical sequence-based or function-based production systems cannot meet [8]. In ideal matrix production systems, each station can process each product variant with minimal set-up times. The strict alignment of cycle times can be foregone in matrix production systems [2]. Orders are allocated to stations based on availability, optimising the capacity utilisation throughout the production system. Because of the required flexibility, matrix production systems have only become possible with the advent of Industry 4.0 and the accompanying computational capability to track and control the flow of products precisely. Hofmann et al. [8] investigate under which circumstances matrix production outperforms conventional production lines. Within matrix production systems, intelligent automatic scheduling has been a focus of research. Stricker et al. [9] propose a Monte Carlo tree search-based multi-objective optimisation of scheduling. May et al. [10] use an automated generation of orders that ideally utilise the ad-hoc available production capacities. Configuring or planning matrix production systems has also received considerable research interest. Filz et al. [11] use a data analysis framework and discrete event simulation to evaluate different configuration options for matrix production systems, specifically focusing on intra-logistics with automated ground vehicles (AGV). Trierweiler and Bauernhansl [5] propose an approach for the dynamic reconfiguration of production systems based on constant requirements monitoring. They propose the use of optimisation techniques for the reconfiguration.

2.2 Discrete event simulation

DES is commonly used to replicate the behaviour of production systems as they are characterised by discrete state changes and stochastic nature [12]. A large body of research examines the use of simulation for configuration and operation or control, though more recently, the focus has shifted towards the latter [13]. Mourtzis [14] examines the state of the art of simulation in the design and operation of manufacturing systems. He points out that DES has become increasingly sophisticated and can accurately depict real production systems. Schönemann et al. [15] create a multiscale framework for the simulation of production systems and use it for different challenges concerning production systems like configuration and control.

2.3 Metaheuristics

Concerning decision-making processes, simulation models are classified as predictive models, able to project the behaviour of a system under given circumstances [16]. However, in many decision situations, prescriptive decision support is required, which identifies suitable solutions for problems. For this purpose, either specifically developed exact solution methods can be used, or generalistic, usually heuristic methods that cannot guarantee to find optimal solutions but are more tolerant towards the complexity of the modelled system [17, 18]. Many metaheuristics are inspired by biology and nature, coping strategies used to find suitable solutions for a long time [19]. Metaheuristics can be classified as population-based or single-point-based [20]. GA is likely the most often used population-based method, while SA and tabu search are among the most common single-point-based methods [20]. GA is a direct, stochastic search method that effectively evaluates large search areas. The search begins with a population of random solutions and develops this population over several generations by applying probability techniques and reproduction operators to each member of the population [19]. SA imitates the gradual cooling of metals called annealing, leading to low-energy crystal structure formation [20]. In SA, the initially high temperature allows for a broad-ranging search of the solution space, whereas the low temperature promotes a targeted search for optimal solutions [20]. Several works have compared different metaheuristics in different combinatorial problems [6]. Mohan et al. [21] investigate the performance of hill-climbing, guided local search, tabu search, and SA for a vehicle routing problem, finding that SA was prone to get stuck in local optima. Halim and Ismail [7] compare six different metaheuristics to solve the travelling

salesman problem, finding that especially for large problems, GA, tree physiology optimisation (TPO), and SA are suited. Zolfaghari and Liang [22] compares GA, SA, and Tabu Search (TS) in machine grouping problems. Also, several existing works have combined metaheuristics with DES. Hatami-Marbini et al. [23] develop a simulation-optimisation framework using DES and SA to find optimal control policies for producing perishable goods. de Sousa Junior et al. [24] propose the use of machine learning to improve metaheuristics used for the optimisation of shop floor simulations.

2.4 Metaheuristics for the configuration of production systems

As described above, the configuration of production systems is one application area for prescriptive models. The use of DES and metaheuristics has been examined in several previous works. Petroodi et al. [25] combine DES and SA to optimise the configuration of an automotive production system. Rabe et al. [26] propose a simheuristic framework that dynamically adapts the granularity and number of simulation experiments to examine the results of heuristically determined solutions. They validate the framework using a configuration problem in a job-shop manufacturing system using DES and GA. In addition, an SA procedure is also proposed for the dynamic layout problem by [27] and a GA [28]. Optimising the spatial arrangement of system resources is a subset of the configuration problem. Kia et al. [29] and Zhang et al. [30] deal with multiple-level warehouse layout problems. Kia et al. [29] minimise the total transport, machining and other costs, whereas [30] focus solely on transportation costs. Both use GA to generate satisfactory solutions. Arostegui et al. [31] compare GA, SA, and TS methods in different facility layout problems (FLP): the capacitive FLP, the periodical FLP, and the multi-commodity FLP. Zupan et al. [32] combine a GA with a digital twin to optimise the layout of a production cell. Furthermore, Tubaileh [33] propose an SA method to optimise the machine layout in a flexible manufacturing system. Only a few applications of metaheuristics are directly concerned with matrix production systems. For example, Völker and Verbeet [34] discuss a simulation-based model of a matrix production system. The model is optimised using an SA method. Bányai et al. [35] model a matrix production system and optimise the logistics using a sequential black hole-floral pollination algorithm.

The existing work on simulation-based heuristics indicates the interest in this topic. However, most existing works focus on developing specific new methods for configurational problems, and only a few examine different metaheuristics in a specific context. Furthermore, many works that compare different algorithms use synthetic use

cases. Thus, the contribution of this paper is to compare two of the most commonly used metaheuristics, GA and SA, in a real-world configuration problem. Thereby, insights into the particular challenges the application of these methods faces are provided.

3 Experimental setting and methodology

3.1 Use case

The considered use case is located in the electrical testing of circuit boards. The production of boards includes 100% testing using a product-specific protocol with tactile resistance measurements. The *testing time* of each board varies depending on the protocol, the testing machine generation, and the necessity to repeat tests. If a test is repeatedly unsuccessful, the board is checked by an operator and placed on a “defect” magazine sent to a repair station when all products of an order have been tested, or the magazine is full. The production system consists of two rows of testing machines with twelve available spaces. In the system, a wide variety of boards is processed. Each board can be processed at any of the testing stations, given that it fits the testing machine loaders, of which there are two sizes. Based on the flexibility of testing machines regarding work assignments and the transportation system, this system can be characterised as a matrix production system. Since many boards need to be tested from both sides, two machines are sometimes *linked* together so that the first can test side A and the second can test side B. Such a linking of two machines consists of an entry and an exit loader, which allow magazines of products to be entered and taken out, a turning station, and the testing machines themselves. As some products only need testing from one side, machines can also be operated as *standalone* with two loaders. The boards are transported in magazines which can be adjusted to fit the specific board size. The products are assembled on eight assembly lines, which produce order-specific batches. The assembly lines are only included in the simulation model as sources. One batch can consist of one or multiple magazines, each of which can hold about 20 boards, depending on the board height. Some boards require specific, larger loader sizes to be tested and can thus only be allocated to machines equipped with those loaders. The orders are assigned to testing machines based on a first-in-first-out principle, though products can be delayed due to loader incompatibility and prioritised if a machine is already set up for the board type. The system is operated by four employees, three of which are assigned to four machines each, while one “jumper” can freely support.

The configuration of the simulated system can be changed in terms of the number of machines, their generation and linkage configuration, the loader configuration, and the

number and occupation of workers. The number of machines is limited to twelve but can be reduced if necessary. For the machines, four generations are available. *Generation 1* needs a manual set-up process and is generally slower than *Generation 2*. *Generation 3* machines do not require relevant set-up times to switch between variants, and *Generation 4* machines can even test both sides of the board while eliminating the need for linkages and turning machines. Machines can be linked to reducing necessary operator interactions. However, this decreases the efficiency of the linked pair, as the testing time of sides A and B typically deviates and subsequently, one of the two needs to wait for the other. This problem can be mitigated by combining slower machines of generations 1 and 2 with generation 3. Machines can also be operated in an *alternating linkage*, where each machine alternates between testing sides A and B and testing times are harmonised. This linkage, however, also requires two additional turning machines. For each machine or each linkage, different loaders can be chosen. The larger loaders are slightly more expensive. Finally, the number and occupation of employees can be changed to include between three and five employees and different numbers of jumpers. The available parameters and a graphical representation are shown in Fig. 1.

3.2 DES modeling and fitness function

The simulation model used for the experimental comparison between GA and SA represents the matrix production system of an industrial PC manufacturer as discussed in the previous section. The model has been implemented in Tecnomatix Plant Simulation (TPS). The model contains several processing stations, multiple employees with specific assigned tasks, transport units, and product instances. The modelling used an agile development process adapted from VDI3633 [12] described in [36]. Information regarding the order and machine-specific processing times were defined based on actual processing times enhanced with master data whenever no recordings were available. Layout information and working times were based on master data.

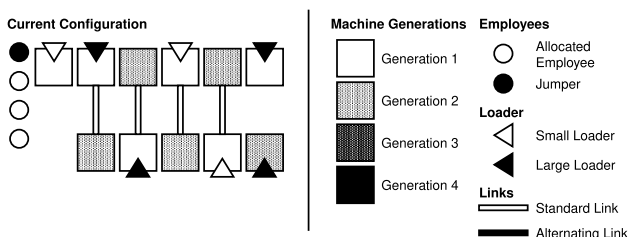


Fig. 1 Configuration parameters of the matrix production system and graphical representation of the current configuration (linked machines can only have the same loader type; thus, only one is shown)

The number of transport units was assumed to be infinite, though transport units were only allowed to spawn and despawn at specific source and sink points, ensuring realistic logistic processes within the modelled production segment. The dynamic failure rates and machine downtimes thereby required specific model adaptations. The machines are used to test finished products using a product-specific testing protocol electrically. The operator needs to confirm the result should any test be unsuccessful. As operators are assigned to multiple machines at a time, this can lead to waiting times if no operator is available. Additionally, the failure rate at a machine dynamically changed as an order was processed due to necessary protocol adaptations by the operator to avoid parts falsely flagged as defective. This mechanism was represented in the model using a stochastic learning process, decreasing the likelihood of these type II errors after each occurrence using the following formula,

$$\psi_{II,j+1} = \alpha_{teach} p(s) \psi_{II,j} \tag{1}$$

where $\psi_{II,j}$ is the likelihood of a type II error at the j th tested board, $p(s)$ is a triangular density function with a maximum at $s = 0$, and α_{teach} reflects the teaching parameter that was fitted to the distribution of errors found in the use case. The function parameters were fitted using half a year of process recordings. The time to rectify a problem at a machine by adapting the protocol or manual product inspection was not distinguished from waiting times in the recordings. Thus, these times needed to be calibrated in an iterative procedure using the model to determine the waiting times with the real production system. The model was then validated using extensive testing and comparing it to multiple production periods.

The fitness of a particular configuration was determined as a combination of the achieved production capacity and the required investments and costs for employees. The capacity was evaluated using the simulation model, capturing the time necessary to fulfil the production orders of a week. As users of such optimisation models are typically only concerned with fulfilling the required capacity, the capacity fitness function $fit_{cap,i,o,s}$ of a particular configuration $i \in I$, for a given observation $o \in O$ and chosen capacity increase scenario $s \in S$ was defined as

$$fit_{cap,i,o,s} = \begin{cases} 0, & T_{i,o} \leq T_{ref,s} \\ \frac{T_{i,o} - T_{ref,s}}{T_{ref,s}}, & T_{i,o} > T_{ref,s} \end{cases} \tag{2}$$

where $T_{i,o}$ denotes the time elapsed for the production in the model and $T_{ref,s}$ refers to the reference time set for a specific capacity expansion scenario. The capacity expansion reference time is calculated as the reference time $T_{ref,0}$ divided by the desired capacity increase. The yearly cost C_i was

calculated based on the necessary investment for machines and loaders as the average yearly depreciation as well as the employee costs:

$$C_i = \sum_{u \in U_i} \left(C_{inv,u} \left(\frac{1}{T_{use}} + \frac{ir}{2} \right) \right) + n_{emp,i} C_{emp} \quad (3)$$

where $u \in U_i$ denote the necessary changes to the starting configuration, $C_{inv,u}$ the overall investment for each new asset, T_{use} the useful life of the asset, and ir the interest rate. $n_{emp,i}$ describes the number of employees in configuration i and C_{emp} the average yearly cost of an employee. The cost fitness function $fit_{cost,i}$ is then defined as

$$fit_{cost,i} = \frac{C_i}{C_{max}} \quad (4)$$

where C_{max} are the maximum possible investment costs. The overall fitness function $fit_{i,o,w,s}$ of a configuration i for an observation o , with temporal weight w and capacity threshold scenario s is defined as

$$fit_{i,o,w,s} = fit_{cost,i} + g_w fit_{cap,i,o,s} \quad (5)$$

where g_w is the weight factor. Using different weights, the algorithm can be influenced in the tradeoff between higher capacities and lower costs.

3.3 Metaheuristics

For both investigated metaheuristics, the system’s configuration was parametrised using 31 parameters resulting in a total of $3.02E+11$ possible configurations. TPS provides a built-in tool for optimisation using GA, the *GAAssistant*. This tool allows the selection of parameters to be optimised and their permissible value ranges. Furthermore, some hyperparameters of the GA can also be changed, such as the population size and the number of observations per generation. A proprietary SA algorithm was developed and implemented because TPS does not provide alternative metaheuristics. The algorithm uses an exponentially decreasing temperature which is defined as the temperature Θ_n at step $n \in [1, N]$

$$\Theta_n = \Theta_0 \left(\frac{\Theta_N}{\Theta_0} \right)^{\frac{n}{N}} \quad (6)$$

where Θ_0 denotes the initial temperature and Θ_N the final temperature. The temperature Θ_n defines the likelihood ϕ_n to accept an inferior configuration i instead of the current configuration \hat{i} for the succeeding proximity search at step n .

$$\phi_n = e^{-\frac{fit_i - fit_{\hat{i}}}{\Theta_n}} \quad (7)$$

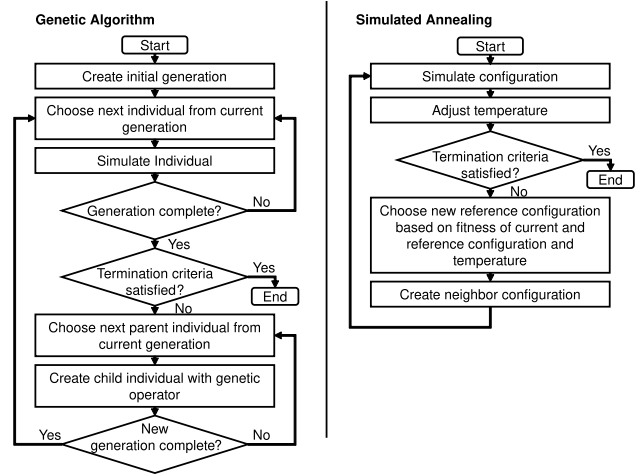


Fig. 2 Flow chart of optimisation routines for GA & SA

Table 1 Examined experimental values

Variables	Values		
N	1500	2950	4500
$T_{ref,s}$ [s]	389,520	339,840	297,360
g_w	3	4	5

The creation of neighbour configurations in SA was facilitated by the operations “flip”, “insert”, and “change”. Moreover, these operations were optimised for the given parameter space. Figure 2 shows the GA and SA algorithms as a flowchart.

3.4 Experimental design

To compare SA and GA, several experiments were conducted. Since the number of required simulation runs is the major determinant of the required computing time, GA and SA were compared for three sets of allowed numbers of simulation runs N , 1500, 2950, and 4500. Two aspects of the fitness function $fit_{i,o,w,s}$ were also varied, namely the reference time $T_{ref,s}$, simulating different requirements towards the production system, and the weight factor for the capacity fitness function g_w to distinguish the severity with which too low capacity is punished. The variation of the reference Time is equivalent to enforcing a higher throughput of the system, as the same number of orders needs to be processes in less time. The goal of the meta heuristic optimisation is thus the increase of throughput at minimal investment and operative costs. An overview of the experiments is given in Table 1. To limit the number of experiments, the number of simulation runs was only varied for $g_w = 4$, for 3 and 5 $N = 2950$ was fixed.

The starting configuration for all experiments was the current configuration, which required an average production time of $\bar{T}_0 = 0.9895 T_{ref,0}$. It results in an average capacity fitness of $\bar{f}_{it}_{cap,0,s=1} = 0.0994$. The fitness of the starting configuration for each value of $T_{ref,s}$ and g_w is shown in Table 2.

One important factor to consider in simulation-based optimisation is the stochastic nature of the simulation and, thus, the necessity to perform multiple simulation runs to evaluate a given configuration i [26]. The simulation model shows an average relative standard deviation of the production time $T_{i,o}$ of 1.67%, estimated using ten different configurations and 100 observations each. This deviation is considerable, as the desired improvements of the capacity are only one order of magnitude bigger. In the above-described experiments, each configuration is tested with a random seed. This method poses the danger of reducing the algorithm's relevance for the stochastic solution space due to seed-optimised configurations. Thus, the effect of this seed value dependence was tested by comparing GA runs based on only one *fixed seed* value for all runs, with GA runs with *ten observations* with different seeds per configuration and runs with *alternating seeds* between each configuration. This test showed only a slight improvement in the fitness value for ten observations compared to alternating and fixed seeds, as shown in Fig. 3, even though only a tenth of the computation time was necessary for the latter. Furthermore, the fixed seeds' results seemed to slightly outperform the alternating seeds, even when comparing the final results for multiple different seeds. This may be due to decreased selection efficiency when the simulation seed changes. Therefore, fixed seeds were chosen for the following experiments.

4 Results

Figure 4 shows the best configuration found by the metaheuristics for different values of s . The metaheuristics improved the fitness value of the original configuration in every case at least by a factor of 2. In practice, each of the final results was able to process the specified number of orders in reduced time as specified by $T_{ref,s}$ in Table 1. Thereby, the potential throughput of the system was increased by 11.11%, 27.35%, and 45.54% respectively. For

Table 2 Average fitness value of current configuration for different values of $T_{ref,s}$ and g_w

		$T_{ref,s} [s]$		
		389, 520	339, 840	297, 360
g_w	3	0.2982	0.7104	1.2405
	4	0.3976	0.9473	1.6540
	5	0.4970	1.1841	2.0675

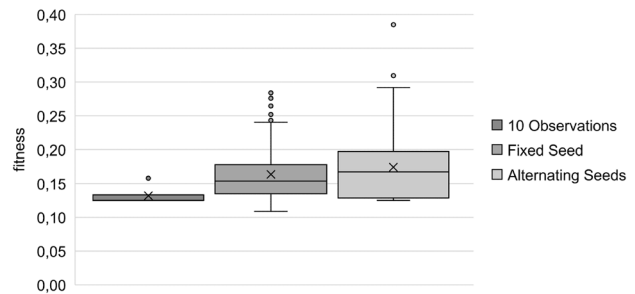


Fig. 3 Resulting distributions of observation strategies with fixed seeds, ten observations per individual and alternating seeds using GA)

the best SA configurations investment costs of 196, 944, and 1710 currency units would be necessary. The best solutions consistently used the same number of employees, but made all of them jumpers. Figure 5 shows the fitness achieved by the final chosen configuration of GA framed in black and SA framed in grey using ten observations in the simulation model. On average, the chosen SA configuration shows a relative improvement of 0.0967 compared to the GA result, using a paired t-test on all observations; this difference is significant with $p = 0.00035$ [6]. As expected, the experiments also show a fitness improvement with increasing N , though this improvement is not as substantial as expected. GA showed an average relative improvement of 0.1582 from $N = 1500 \rightarrow 2950$ and a decline of 0.0519 for $N = 2950 \rightarrow 4500$. SA improved by 0.1216 and 0.0383, respectively.

As shown in Fig. 6, the final configurations are a trade-off between minimising the capacity fitness function and the cost fitness function. This figure shows an overview of all tested configurations for $N = 2950$ and of the final best

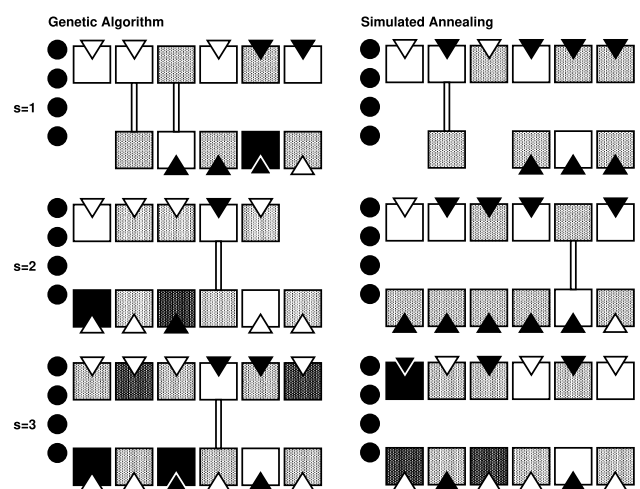


Fig. 4 Resulting best configurations for GA and SA using $N = 2950, g_w = 4$ for different s

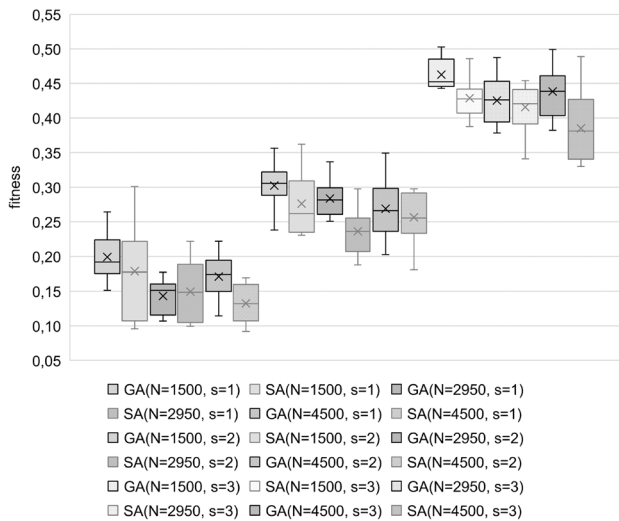


Fig. 5 Fitness of best configurations for GA and SA using $g_w = 4$ for ten different observations each, lower is better

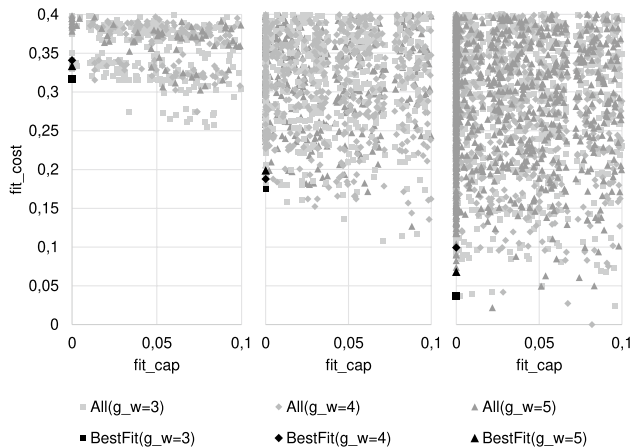


Fig. 6 SA configuration performance for $s = 3$ (left), 2 (center), and 1 (right) and chosen best solution for $g_w = (3, 4, 5)$ using $N = 2950$

configurations the SA algorithm was able to find with different values of g_w and s . All chosen configurations show a value of fit_{cap} of or close to zero.

Interestingly, the best configurations in all three s were found for $g_w = 3$, indicating that a less strict capacity threshold could be beneficial for finding a solution. This becomes apparent in Fig. 7, which shows a broad view of the performances of different configurations as well as isovalue lines, i.e. lines with equal values of $fit_{i,o,w,s}$, for different values of g_w . The figure indicates that the slope of the Pareto optimal set of configurations, effectively the line limiting the space of found solutions from the bottom, is significantly shallower than all isovalue lines, and the algorithm is forced towards values of $fit_{cap,i,o,s}$ close to 0.

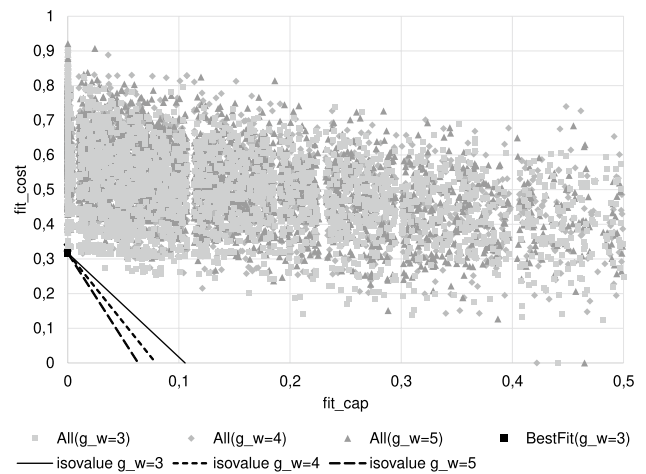


Fig. 7 Distribution of SA results for $s = 3$ with isovalue lines for different g_w

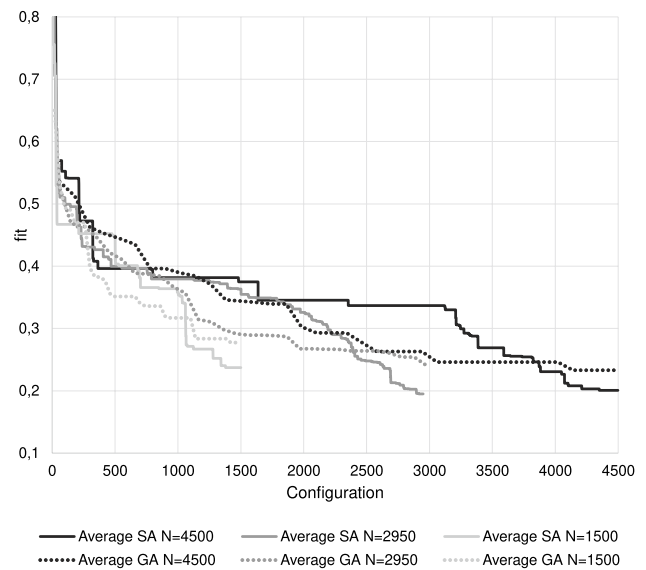


Fig. 8 Average best fitness throughout different runs of SA and GA. (Note: For $N = 2950$, 9 runs are aggregated for SA and GA, respectively, for $N = 1500, 4500$ only 3 runs were performed)

Another interesting aspect of these experiments is each heuristic's progress throughout the run. Figure 8 shows the fitness of the average best configuration at each point of the algorithm. The progress of the GA algorithm shows a continuous approximation towards the optimum, where the average rate of progress slows down with each generation. SA, on the other hand, shows an interesting behaviour. The algorithm underperforms compared to GA until the last third of the experiment. Then, when Θ is low enough, SA seems to be much more efficient at refining the solution and dominates GA in every case. For both SA and GA, $N = 2950$

seems to be the saturation point, as no significant average improvement is found for $N = 4500$.

5 Discussion

The results show that the combination of metaheuristics and simulation is valuable in complex decision-making in production system planning. However, the chosen metaheuristic and its parametrisation have a considered impact on the quality of the generated results. The results above clearly indicate that the SA algorithm used is superior to the GA algorithm provided by TPS, as SA dominates GA in every experiment. This is likely due to SA being more efficient in selecting configurations to be tested. It is particularly noteworthy that the GA cannot reduce the gap to SA even in the case of larger numbers of simulation runs, where an efficiency gap would be expected to diminish. Other studies have also observed similar comparative behaviour between GA and SA [7].

Interestingly, the most significant increases in result quality were found by SA in the last third of the process, where Θ was already relatively low. For the problem presented, it might have been more efficient to start at a lower value of Θ to limit the amount of inefficient searching. Thus, the selection of an apt cooling schedule should be considered crucial. In addition, numerous different cooling behaviour functions are applied in the literature. It would be interesting to investigate the influence of different functions on the configuration solution in this specific problem. Since these functions are the compromise between exploration and exploitation, the convergence of the method would also be affected.

To further enhance the present evaluation of the methods for such optimisation problems, it would be desirable to carry out even more different experiments with further implementation of the methods in the simulation model. This way, the degrees of freedom of discontinuation of the SA procedure and the GA could be further investigated. In this work, the GA is set with the standard parameters of TPS. It would also be interesting to observe the results of this procedure in other settings. One degree of freedom of the GA is, for example, the population size that influences the convergence of the GA [37]. In addition, crossover operators other than those used for this optimisation could be introduced and their impact investigated. Some crossover operators are better suited for certain problems [38]. The same applies to mutation operators and rates; other crossover operators can also be used for the SA method.

The parametrisation of the fitness function also seems to considerably influence the quality of the solutions. In the presented case, the capacity part of the fitness function was implemented as an increasing linear term for values

exceeding a threshold time. The resulting isovalue lines of capacity and cost fitness were significantly steeper than the local gradient of the optimal Pareto set of solutions. Therefore, the resulting best solutions were always located on the $fit_{cap,ios} = 0$ line. Thus, the evaluation of the final solution was not directly dependent on the weighting of the capacity fitness. However, lower weightings of the capacity fitness and, thus, shallower isovalue lines seemed to allow the algorithms to explore solutions close to the threshold more efficiently and thus improve the solution quality. To optimise the algorithms, it may be prudent to choose weightings for such soft constraints that result in isovalue lines close to the expected gradient of the Pareto optimal set. This idea should be investigated further in subsequent research.

The experiments also show that, at least for the experimentation, the stochastic nature of simulations can be ignored to drastically improve the computational effort required. However, this may not necessarily be the case for different problems. For example, fixed seed observation methods will likely perform worse in systems with higher levels of stochastic variation. Thus, finding thresholds for using different observation methods would be interesting, though very laborious. Also, dynamically adjusted observation methods, as proposed by [26], could be used.

6 Conclusion

This contribution investigated genetic algorithms and simulated annealing to optimise a material flow simulation in a matrix production system. The discrete event simulation model was developed and validated to resemble the production system closely. Subsequently, both GA and SA algorithms were tested using different parametrisations. The results show that SA is superior to GA in the tested use case and meta-heuristic configuration, likely due to the increased efficiency in the final phase of the optimisation. In conclusion, it can thus be said that using such approaches is a valuable addition to the planning of matrix production systems.

Acknowledgements Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – project number 454608456.

Funding Open Access funding enabled and organized by Projekt DEAL.

Data availability The datasets generated during the study are available from the corresponding author on reasonable request.

Declarations

Conflict of interest The authors declare that they have no conflict of interest

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Blecker T, Kaluza B (2004) Heterarchische Hierarchie: Ein Organisationsprinzip flexibler Produktionssysteme, Diskussionsbeiträge / Universität Klagenfurt, Institut für Wirtschaftswissenschaften, vol 2004/01. Inst. für Wirtschaftswiss, Klagenfurt
- Greschke P, Schönemann M, Thiede S et al (2014) Matrix structures for high volumes and flexibility in production systems. *Procedia CIRP* 17:160–165. <https://doi.org/10.1016/j.procir.2014.02.040>
- Schönemann M, Herrmann C, Greschke P et al (2015) Simulation of matrix-structured manufacturing systems. *J Manuf Syst* 37:104–112. <https://doi.org/10.1016/j.jmsy.2015.09.002>
- Schmidtke N, Rettmann A, Behrendt F (2021) Hawaii international conference on system sciences 2021. University of Hawai'i at Manoa Hamilton Library, Honolulu, HI. <https://scholarspace.manoa.hawaii.edu/handle/10125/72112>
- Trierweiler M, Bauernhansl T (2021) Reconfiguration of production equipment of matrix manufacturing systems. In: Weißgraeber P, Heieck F, Ackermann C (eds) *Advances in automotive production technology—theory and application*. ARENA2036. Springer, Berlin, pp 20–27. https://doi.org/10.1007/978-3-662-62962-8_3
- Halim AH, Ismail I, Das S (2021) Performance assessment of the metaheuristic optimization algorithms: an exhaustive review. *Artif Intell Rev* 54(3):2323–2409. <https://doi.org/10.1007/s10462-020-09906-6>
- Halim AH, Ismail I (2019) Combinatorial optimization: comparison of heuristic algorithms in travelling salesman problem. *Arch Comput Methods Eng* 26(2):367–380. <https://doi.org/10.1007/s11831-017-9247-y>
- Hofmann C, Brakemeier N, Krahe C et al (2019) The impact of routing and operation flexibility on the performance of matrix production compared to a production line. In: Schmitt R, Schuh G (eds) *Advances in production research*. Springer, Cham, pp 155–165. https://doi.org/10.1007/978-3-030-03451-1_16
- Stricker N, Kuhnle A, Hofmann C et al (2021) Self-adjusting multi-objective scheduling based on Monte Carlo tree search for matrix production assembly systems. *CIRP Ann* 70(1):381–384. <https://doi.org/10.1016/j.cirp.2021.04.036>
- May MC, Schmidt S, Kuhnle A et al (2021) Product generation module: automated production planning for optimized workload and increased efficiency in matrix production systems. *Procedia CIRP* 96:45–50. <https://doi.org/10.1016/j.procir.2021.01.050>
- Filz MA, Herrmann C, Thiede S (2020) Simulation-based data analysis to support the planning of flexible manufacturing systems. *SNE Simul Notes Europe* 30(4):131–137. <https://doi.org/10.11128/sne.30.tn.10531>
- Gutenschwager K, Rabe M, Spieckermann S et al (2017) *Simulation in Produktion und Logistik*. Springer, Berlin. <https://doi.org/10.1007/978-3-662-55745-7>
- Negahban A, Smith JS (2014) Simulation for manufacturing system design and operation: literature review and analysis. *J Manuf Syst* 33(2):241–261. <https://doi.org/10.1016/j.jmsy.2013.12.007>
- Mourtzis D (2020) Simulation in the design and operation of manufacturing systems: state of the art and new trends. *Int J Prod Res* 58(7):1927–1949. <https://doi.org/10.1080/00207543.2019.1636321>
- Schönemann M, Bockholt H, Thiede S et al (2019) Multiscale simulation approach for production systems. *Int J Adv Manuf Technol* 102(5–8):1373–1390. <https://doi.org/10.1007/s00170-018-3054-y>
- Medina FG, Umpierrez AW, Martinez V et al (2021) A maturity model for digital twin implementations in the commercial aerospace oem industry. In: 2021 10th international conference on industrial technology and management (ICITM). IEEE, pp 149–156. <https://doi.org/10.1109/ICITM52822.2021.00034>
- Fu MC (2002) Feature article: optimization for simulation: theory vs. practice. *INFORMS J Comput* 14(3):192–215. <https://doi.org/10.1287/ijoc.14.3.192.113>
- Meinecke (2017) Ein lösungsverfahren für die integrierte planung der produktion in der werkstattfertigung und den überbetrieblichen transport. Doctoral dissertation, Universität Bremen
- Sattarvand J, Niemann-Delius C (2013) Past, present and future of metaheuristic optimization methods in long-term production planning of open pits. *BHM Berg-und Huettenmaenn Monatsh* 158(4):146–154. <https://doi.org/10.1007/s00501-013-0127-y>
- Gogna A, Tayal A (2013) Metaheuristics: review and application. *J Exp Theor Artif Intell* 25(4):503–526. <https://doi.org/10.1080/0952813X.2013.782347>
- Mohan A, Dileep A, Ajayan S et al (2020) Comparison of metaheuristics for a vehicle routing problem in a farming community. In: Thampi SM, Trajkovic L, Li KC et al (eds) *Machine learning and metaheuristics algorithms, and applications, communications in computer and information science*, vol 1203. Springer, Singapore, pp 49–63. https://doi.org/10.1007/978-981-15-4301-2_5
- Zolfaghari S, Liang M (2002) Comparative study of simulated annealing, genetic algorithms and tabu search for solving binary and comprehensive machine-grouping problems. *Int J Prod Res* 40(9):2141–2158. <https://doi.org/10.1080/00207540210131851>
- Hatami-Marbini A, Sajadi SM, Malekpour H (2020) Optimal control and simulation for production planning of network failure-prone manufacturing systems with perishable goods. *Comput Ind Eng* 146(106):614. <https://doi.org/10.1016/j.cie.2020.106614>
- de Sousa Junior WT, Montevechi JAB, Miranda RC et al (2020) Shop floor simulation optimization using machine learning to improve parallel metaheuristics. *Expert Syst Appl* 150(113):272. <https://doi.org/10.1016/j.eswa.2020.113272>
- Petroodi SEH, Eynaud ABD, Klement N et al (2019) Simulation-based optimization approach with scenario-based product sequence in a reconfigurable manufacturing system (rms): a case study. *IFAC-PapersOnLine* 52(13):2638–2643. <https://doi.org/10.1016/j.ifacol.2019.11.605>
- Rabe M, Deininger M, Juan AA (2020) Speeding up computational times in simheuristics combining genetic algorithms with discrete-event simulation. *Simul Model Pract Theory* 103(102):089. <https://doi.org/10.1016/j.simpat.2020.102089>
- Baykasoğlu A, Gindy NN (2001) A simulated annealing algorithm for dynamic layout problem. *Comput Oper Res* 28(14):1403–1426. [https://doi.org/10.1016/S0305-0548\(00\)00049-6](https://doi.org/10.1016/S0305-0548(00)00049-6)
- Balakrishnan J, Cheng CH (2000) Genetic search and the dynamic layout problem. *Comput Oper Res* 27(6):587–593. [https://doi.org/10.1016/S0305-0548\(99\)00052-0](https://doi.org/10.1016/S0305-0548(99)00052-0)
- Kia R, Khaksar-Haghani F, Javadian N et al (2014) Solving a multi-floor layout design model of a dynamic cellular

- manufacturing system by an efficient genetic algorithm. *J Manuf Syst* 33(1):218–232. <https://doi.org/10.1016/j.jmsy.2013.12.005>
30. Zhang GQ, Xue J, Lai KK (2002) A class of genetic algorithms for multiple-level warehouse layout problems. *Int J Prod Res* 40(3):731–744. <https://doi.org/10.1080/00207540110093909>
 31. Arostegui MA, Kadipasaoglu SN, Khumawala BM (2006) An empirical comparison of tabu search, simulated annealing, and genetic algorithms for facilities location problems. *Int J Prod Econ* 103(2):742–754. <https://doi.org/10.1016/j.ijpe.2005.08.010>
 32. Zupan H, Herakovic N, Zerovnik J et al (2016) Layout optimization of a production cell. *Int J Simul Model* 16(4):603–616. [https://doi.org/10.2507/IJSIMM16\(4\)4.396](https://doi.org/10.2507/IJSIMM16(4)4.396)
 33. Tubaileh AS (2014) Layout of flexible manufacturing systems based on kinematic constraints of the autonomous material handling system. *Int J Adv Manuf Technol* 74(9–12):1521–1537. <https://doi.org/10.1007/s00170-014-6063-5>
 34. Völker S, Verbeet R (2021) Simulationsbasierte konfiguration der stationen von matrix-produktionssystemen. In: Jörg Franke PS (ed) *Simulation in Produktion und Logistik 2021*, pp 285–294
 35. Bányai Á, Illés B, Glistau E et al (2019) Smart cyber-physical manufacturing: extended and real-time optimization of logistics resources in matrix production. *Appl Sci* 9(7):1287. <https://doi.org/10.3390/app9071287>
 36. Benfer M, Autenrieth M, Brützel O et al (2022) Agile erstellung von materialflusssimulationen. *Zeitschrift für wirtschaftlichen Fabrikbetrieb* 117(12):867–871. <https://doi.org/10.1515/zwf-2022-1158>
 37. Butteltmann M, Lohmann B (2004) Optimierung mit genetischen algorithmen und eine anwendung zur modellreduktion (optimization with genetic algorithms and an application for model reduction). *at-Automatisierungstechnik* 52(4):151–163. <https://doi.org/10.1524/auto.52.4.151.29416>
 38. Kora P, Yadlapalli P (2017) Crossover operators in genetic algorithms: a review. *Int J Comput Appl* 162(10):34–36. <https://doi.org/10.5120/ijca2017913370>

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