Journal of Universal Computer Science, vol. 28, no. 3 (2022), 292-310
submitted: 1/11/2021, accepted: 22/12/2021, appeared: 28/3/2022

Solving the problem of scheduling the production process based on heuristic algorithms

Dagmara Łapczyńska
(Faculty of Mecanical Engineering, Wrocław University of Science and Technology, Wrocław, Poland
https://orcid.org/0000-0002-4745-0768, dagmara.lapczynska@pwr.edu.pl)

Konrad Łapczyński
(Faculty of Computer Science and Management, Wrocław University of Science and Technology, Wrocław, Poland
konrad.lapczynski@student.pwr.edu.pl)

Anna Burduk
(Faculty of Mecanical Engineering, Wrocław University of Science and Technology, Wrocław, Poland
https://orcid.org/0000-0003-2181-4380, ann.burduk@pwr.edu.pl)

Jose Machado
(Mechanical Engineering Department, School of Engineering, University of Minho, Guimarães, Portugal
https://orcid.org/0000-0002-4917-2474, jmachado@dem.uminho.pt)

Abstract: The paper deals with a production scheduling process, which is a problematic and it requires considering a lot of various factors while making the decision. Due to the specificity of the production system analysed in the practical example, the production scheduling problem was classified as a Job-shop Scheduling Problem (JSP). The production scheduling process, especially in the case of JSP, involves the analysis of a variety of data simultaneously and is well known as NP-hard problem. The research was performed in partnership with a company from the automotive industry. The production scheduling process is a task that is usually performed by process engineers. Thus, it can often be affected by mistakes of human nature e.g. habits, differences in experience and knowledge of engineers (their know-how), etc. The usage of heuristic algorithms was proposed as the solution. The chosen methods are genetic and greedy algorithms, as both of them are suitable to resolve a problem that requires analysing a lot of data. The paper presents both approaches: practical and theoretical aspects of the usefulness and effectiveness of genetic and greedy algorithms in a production scheduling process.

Keywords: production scheduling process, job-shop scheduling problem, heuristic methods, greedy algorithm, genetic algorithm.
Category: J.6, J.7, I.2, I.6, H.4
DOI: 10.3897/jucs.80750

1 Introduction

Nowadays customers need not only the products at the right price or quality, but also products delivered at the time required by them. Moreover, as the technology of communication develops, consumers are becoming more and more aware of their possibilities. Thus, the industry is rapidly developing to be able to fulfil the customers' requirements. Because of that, in order for the companies to stay competitive, they need to continuously improve their processes in the rapidly changing and developing industry. It results in the search of new solutions, which is more and more often connected to the use of computing and data-analysing methods [Sobaszek 2017; Zwolińska 2017; Burduk 2017; Jasiulewicz-Kaczmarek 2016; Olender 2018].

These methods allow to simulate and perform the analysis of various factors which need to be considered when solving scheduling problems. These factors in production process are e.g. availability of machines and operators, their possibilities, production lines supply, transport, etc. The method of production scheduling often used by companies is planning by production engineers who base their decisions on the past data and their know-how. This process does not only take a huge amount of work time but also can cause human-factor mistakes [Patalas-Maliszewska 2020; Antosz 2019; Więcek 2019; Stadnicka 2019].

The solution of this issue can be the usage of the heuristic methods that allows to propose the improvement solution based on the data in a relatively short time. The analysed problem in the work was classified as Job-shop Scheduling Problem (JSP), where a set of jobs must be processed on a set of machines in a sequence of consecutive technological operations for each job. Each operation requires exactly one machine, and the access to the machines is continuous. They can process one operation at a time without interruption. The consideration is about the way to arrange the operations on the machines in order to optimize the specified performance indicator. A typical performance indicator for JSP is the makespan, i.e., the time needed to complete all the jobs. In other words, the production scheduling problem allocates limited resources to tasks over time and determines the sequence of operations so that the system's constraints are met, and the performance criteria are optimized [Ahmadian 2021; Blum 2003; Çaliş 2015, Li 2014].

There were two aims of the research – practical one, which was to reschedule the production process and to decrease the total production time and the theoretical one to compare the heuristic algorithms in the area of their usefulness and effectiveness in a scheduling process. The practical goal was achieved by adopting the following research methodology:

1. Collecting company data on the production process and the current way of building production schedules.

2. Selection of production processes for which the production schedule will be built.

3. Adoption of the assumptions and objective functions (in this case, the total production time) for the tested algorithms.

3. Develop a genetic algorithm and a greedy algorithm for the analyzed practical case and conduct simulation experiments.

4. Comparison of the obtained results between the schedule using the loop method and the results obtained in simulation tests.

The structure of this work is as follows: in Section 1.1 and Section 1.2 the specificity of scheduling the production process and the application of selected heuristic methods are described. In Section 2, the manufacturing process for which optimization will be performed using genetic and greedy algorithms is described. Section 3 contains the solution of the optimization problem with the use of both heuristic algorithms and the comparison of the obtained results from the models with the traditional method of scheduling. Finally, in Section 4, the conclusions and direction of the future research are presented.

1.1 Scheduling process

In a scheduling process the decisions that concern the application and arrangement of production resources necessary for the execution of a production order are most often made by the process engineers. The task of a decision-making person is – in this case – to manage the resources in such a way to meet the demand for products at the best use of resources. Therefore, it is a process based on decision-making where it is necessary to plan the use of resources in production and distribution (including not only the volume but also the timing).

Production process scheduling is a very complex task, which complexity results from the necessity of taking into account many different factors simultaneously. Thus, in order to remain competitive, companies are forced to improve and find new methods of the process organization. Traditional methods are usually based on the knowledge and experience of process engineers, with the possible use of basic computer tools such as simple spreadsheets. These techniques are typical particularly for small and medium-sized enterprises in which employees' know-how is the basis for the proper functioning of processes.

In more advanced and often larger enterprises, the ERP (Enterprise Resources Planning) software is used to support resource planning and management. This solution is often fully sufficient for the needs of a given enterprise. However, it requires the purchase of appropriate software and training in its use, as well as often technical support after the purchase which can be very expensive. Moreover, this kind of systems appears to be more and more versatile, which is often an advantage. However, in case of production processes which require individual solutions it can be an obstacle. Thus, with the science and technology development, more and more methods are based on intelligent solutions, i.e. heuristic algorithms [Rojek 2015; Dostatni 2018].

1.2 Heuristic methods in production process scheduling

Heuristic methods are usually algorithms that allow to find the solution that is a local optimum [Blum 2003]. These methods often enable to find a satisfying solution in a much faster way, based on the analysis made by the algorithm that involves multiple information and constrains simultaneously [Ahmadian 2021; Blum 2003] Kramer

2017; Kumanan 2006]. In addition, the use of heuristics allows decisions in real-time and, thus, can handle dynamic problems well. The scheduling heuristics can be implemented in real-world applications easily. The other advantage is that domain knowledge can be easily incorporated with priority-based scheduling heuristics [Zhang 2020]. Thus, in the area of production process scheduling, the heuristic methods allow to reduce or even eliminate the human-factor mistakes, what can have an impact on the effectiveness of a traditional production scheduling process. Various types of algorithms can be found in the literature review, which – among others – are:

- *Simulated Annealing* – algorithms based on the process of heating and slowing that occur during annealing process of ferromagnetics. It allows to reduce the stress inside. In these algorithms a single solution is chosen randomly from the neighbouring solutions, which is compared to the previous one [Franzin 2019; Chen 2019; Ku 2011].

- *Tabu Search* – algorithms that are based on searching for all the possibilities to solve the problem with a sequence of steps where some of them are "forbidden", called taboo [Cordeau 1997; Brandao 2004; Grabowski 2004]. Tabu Search is among the most cited and used metaheuristics for combinatorial optimization problems [Blum 2003].

- *Greedy Algorithms* – a method that determines potentially the best (called "greedy") solution at each step of the algorithm by making a locally optimal decision of the problem that is being considered [Hua 2018; DeVore 1996; Kahraman 2010].

- *Ant Colony Algorithms* – a method based on the nature of ants and their behaviour in colonies in the process of food acquiring. It is based on the pheromones that ants use to cooperate with each other in order to find the shortest possible path from the anthill to the food source [Dorigo 1996; Kulturel-Konak 2011; Oshin 2016; Kalinowski 2017].

- *Genetic Algorithms* – algorithms that are based on the mechanisms that occur in genetics, such as inheritance, crossover, mutation, etc. Then, the best – in the context of the selected criterion – descendants of the generations are analyzed [Damm 2016; Zegordi 2009; Dunker 2003; Paes 2017; Grznár 2021]. In the context of Genetic Algorithms, individuals are called genotypes, whereas the solutions that individuals encode are called phenotypes. This differentiates between the representation of solutions and solutions themselves [Blum 2003].

- *Particle Swarm Optimization* – an algorithm based on the laws of motion of a natural swarm (fish, birds etc.) consisting of particles, each of which has certain parameters such as speed and direction of motion defined in space [Cheng 2018; Kiranyaz 2014].

Theoretically, these techniques introduce slight modifications (perturbations) as well as they assess the schedules up to the point their objective purpose cannot be amended any further. With no more improvement possible, this operation is discontinued resulting in a first-class solution. All these techniques come together with their own perturbation methods, stopping rules and the ways to prevent the local optimum. In comparison to the global search operation, the local search doesn't assure an optimal solution. Generally, it tries to discover the solution better than the present one within its neighbour solution set. If one schedule is created by altering the other according to precise rules, then they are considered neighbours [Pindedo 1995]. The neighborhood search methods are becoming more and more significant in the scheduling problems resolving as the time required to find a solution is less important than in the mathematical programming techniques. One of the main drawbacks of these methods is the appearance of local optima [Gupta 2006].

The literature review, however, indicates the same conclusions regardless of the method used - a tendency to reduce the importance of a human factor (knowledge, experience, employee know-how, etc.) in scheduling in favour of computer solutions (CAD, ERP, artificial intelligence, algorithms, databases, etc.). Therefore, it can be concluded that the development of scheduling methods strives to reduce the influence of a human factor to the necessary minimum, and instead it is proposed to base on mathematical data.

The paper includes the review of using heuristic algorithms in a scheduling process, the characteristics of the studied production process, the implementation of a genetic algorithm and greedy algorithm to the examined case study as well as the results of performing this kind of improvement.

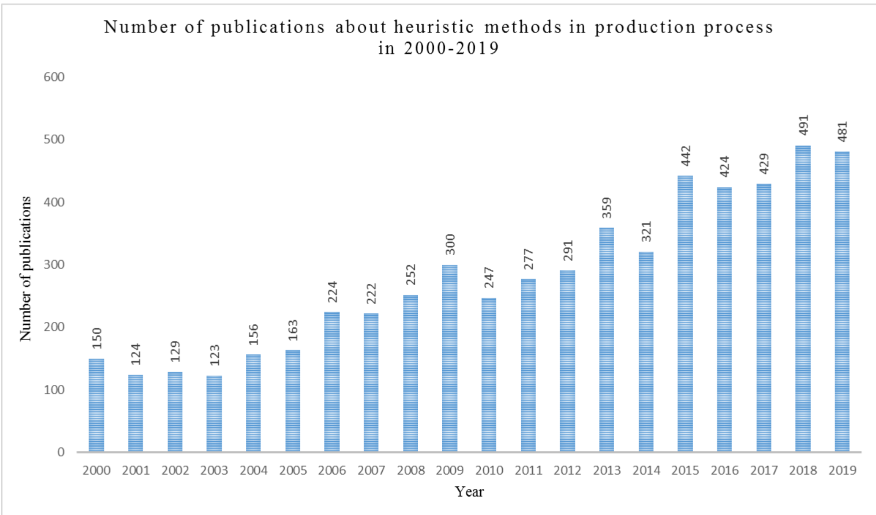

Figure 1: Number of publications about heuristic methods in production process in 2000-2019

The review of articles indicates that the use of heuristic methods in scheduling processes often allows for tangible benefits. These can be various algorithms [Bożejko 2019; Rojek 2019; Ociepka 2013]. The research shows that using this type of solutions is becoming more and more popular in the industry what can be seen in the Web of Science articles database (Fig. 1).

A genetic algorithm and greedy algorithm are types of heuristic methods that are more and more often used in solving the scheduling problem [Burduk 2017; Ahmadian 2021; Luo 2020; Kochańska 2019; Gola 2018; Bożejko 2018; Krenczyk 2020]. The analysis of a number of publications (Fig. 1) shows that the popularity of the use of heuristic methods in production process has been successively increasing for about 20 years. That is because these methods, of which most are various algorithms, turn out to be effective in decreasing the manufacturing time and increasing the profit, etc.

2 Production process scheduling – the research area

The production process scheduling that the research was based on concerns the case study of a company that is an automotive components producer. The research was performed in the company with the use of the information and documentation acquired from process engineers and by taking the measurements of production process parameters.

The products are car parts manufactured in three variants on three different production lines with work stations set according to the flow of the technological process. Each of the product variants is produced on a different production line. Correspondingly: Variant 1 is produced on production line 1, Variant 2 on production line 2, and variant 3 on production line 3. All production lines are equipped with components that obligatorily pass through an input inspection and a subassembly operation. The supply of production lines is carried out by transporting components from the input warehouse, while the finished products, after a prior output inspection, are transported to the output warehouse which also serves as the warehouse before loading and shipping to a customer (Fig. 2). The components can be transported individually (manually) or in collective packages.

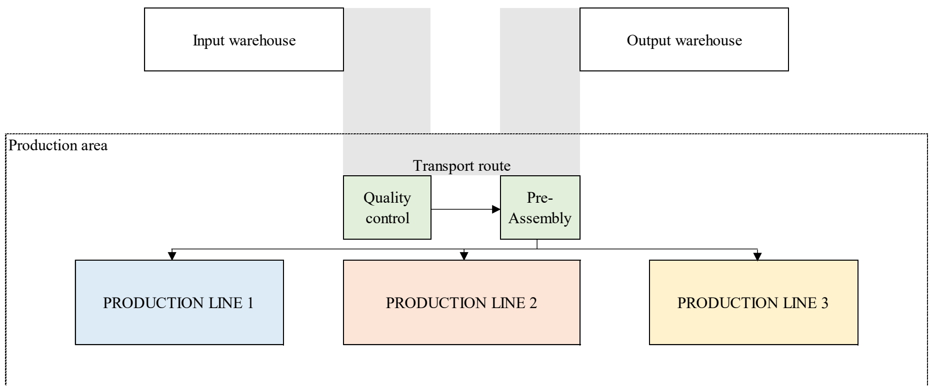

Figure 2: Production area scheme

The loop starts at the PreAssembly station (Fig. 2), where all the components of all variants need to be pre-assembled. The aim was to be able to perform the three production processes in parallel at the same time – as many products as possible. Thus, the company had to schedule the queue of this process. In this case, the production process scheduling includes the manufacturing of three different types (variants) of one product that is ordered in different amounts in the loop of batches. These are the first component (Variant 1, V1) ordered usually in about 150 pcs, divided into batches of 30 pcs; the second component (Variant 2, V2) – 140 pcs divided into 20 pcs batches and the third component (Variant 3, V3) of 120 pcs divided into 15 pcs batches. The company scheduled the production in the loop (Fig. 3), where they pre-assemble 30 pcs of V1, then setup the machine and pre-assemble 20 pcs of V2, and then setup the machine again and pre-assembly 15 pcs of V3. Then they continue to repeat these amounts of products as long as they get the right amount of variants done (150 pcs for V1, 140 pcs for V2, and 120 pcs of V3). After that, the order is finished. If they need to fulfill more orders, they repeat this loop repeatedly, which is the production schedule in the company.

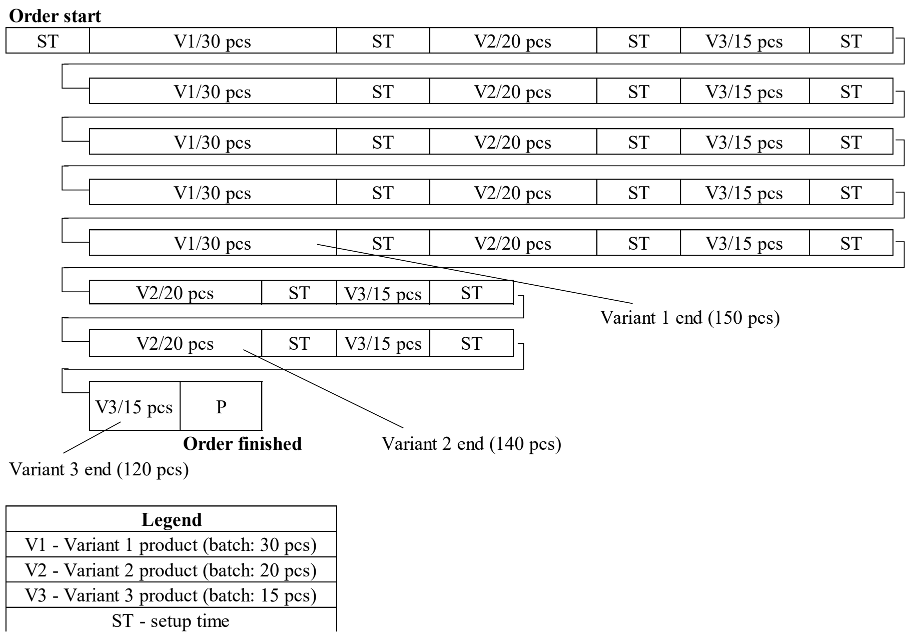

Figure 3: Production process schedule in loop

This type of production process scheduling is based on engineers' know-how and a simple analysis of the data on the manufacturing process. This way of manufacturing allows to fulfil the order of 410 pcs of all variants of products in total of 118 286 seconds (which is over 4 days of the worktime). In order to be able to measure the research results and combine the solutions, the total time of manufacturing the full order of 410 pcs of products was chosen as the improvement criterion.

Genetic and greedy algorithms are heuristic methods that allow to find a local optimum solution, based on the data of production process, and which allows to avoid a human factor in the production process scheduling. Therefore, these two algorithms were chosen. In further research and the built models, minor disturbances appearing during the production process were not considered. The optimization concerned only the manufacturing process on the three analysed production lines. Therefore, the models do not consider the detected quality errors of the products, but the pre-assembly operations are included in the models.

3 Heuristic algorithm application

In order to be able to compare the actual schedule with the solutions proposed by heuristic algorithms, the production process simulation was implemented using Java. The basic element was to present the schedule as a vector:

[VARIANT_1, VARIANT_2, VARIANT_2, VARIANT_1, VARIANT_3, VARIANT_1]

It includes the information about the order and type of components that need to be delivered to the production area. Next, the four classes of abstraction were implemented, namely: PreAssembly, ProductionStep, ProductionLine and ProductionController.

This implementation allowed to develop production lines with different amount of operations (production steps) characterized by different duration times. It also allows to make quick changes in the future, based on the company needs (i.e. to add a new production line, etc.). The simulation includes a pre-assembly, setup time of a pre-assembly machine while changing the variant of the product that is being manufactured, transport between machines in the production line area and the time spent on waiting for the machine.

3.1 Genetic algorithm

The genetic algorithm (GA) is a technique that is widely used because of its flexibility and good performance [Homayouni 2020; Tang 2014; Bazzazi 2009; Du 2020]. These algorithms are based on the evolutionary mechanisms of a natural selection and inheritance. They are characterized by the search of an optimal solution by evolving the starting population. The parameters of a genetic algorithm are listed in Table 1.

In the prepared implementation of the algorithm, each individual has a chromosome (production schedule) and the result of the simulation of a production process for a given chromosome. The possibility of generating a random schedule for an individual based on the order quantity of individual types of products was also implemented. The first stage of the algorithm operation is to generate a population in accordance with the population size and order size specified as a parameter. For this purpose, the possibility of generating a random schedule for each individual was used. Then, the individuals underwent a schedule evaluation using a production process simulator.

No.	Parameter	Number	Unit
1	Population	500	individuals
2	Number of generations	150	generations
3	Mutation probability	10	%
4	Number of mutated genes	40	genes
5	Crossover probability	95	%
6	Number of crossovered genes	75	genes
7	Number of participants in the tournament during the selection	4	participants

Table 1: Genetic algorithm parameters

In this study the tournament method of selecting individuals was used. It consists of drawing from the population (without returning) a number of individuals equal in value to the appropriate simulation parameter. Then, the individual with the shortest order fulfilment time is selected and transferred to the next generation. All of the selected individuals are returned to the population, and the selection process is repeated until the new generation reaches the same population size as the previous ones.

Due to the coding of a chromosome (there are three different possible variants, their number is determined and order matters), order crossover was used. Before starting this process, it is checked for each individual whether it will undergo the crossover (according to the probability given as a parameter of the algorithm). Next, another individual is drawn and the crossover operation is performed.

In the first step, the index of the first crossover point is drawn (not greater than the difference between the length of the chromosome and the number of genes being crossed). Then, the second crossover point is determined (the sum of the index of the first crossover point and the number of genes crossed). Next, genes in the chromosome are determined on the basis of the sequence of genes in the other parent. If a gene with which the chromosome needs to be filled would exceed the number of available variants, then it is omitted. If it turns out that it is not possible to complete the construction of new chromosomes at the end of the crossover, the remaining genes are supplemented by randomizing them from the pool of so far unused products in the schedule.

After completing the crossover for each individual, the possibility of a mutation is checked (according to the probability given as a parameter of the algorithm). If an individual undergoes a mutation, a pool of gene indices in the chromosome is drawn. On this basis a set of genes that will be involved in the mutation process is created. Then, for each of the previously selected indices, a randomly selected gene from the set is assigned. This operation is repeated until the set of the randomly selected genes is empty. The following is an example of a crossover and mutation (Fig. 4).

<u>Simulation parameters:</u>
Number of crossover genes: 3
Number of Variant 1: 2
Number of Variant 2: 2
Number of Variant 3: 3
<u>Parents chromosomes:</u>
Parent 1: [VARIANT_1, VARIANT_2, VARIANT_2, VARIANT_3, VARIANT_3, VARIANT_3, VARIANT_1]
Parent 2: [VARIANT_3, VARIANT_2, VARIANT_1, VARIANT_1, VARIANT_3, VARIANT_3, VARIANT_2]
Random index of the first crossover spot: ***2***
Index of the second crossover spot: 2 + 3 = ***5***
<u>Parents chromosomes prepared to crossover:</u>
Parent 1: [VARIANT_1, VARIANT_2 | VARIANT_2, VARIANT_3, VARIANT_3 | VARIANT_3, VARIANT_1]
Parent 2: [VARIANT_3, VARIANT_2 | VARIANT_1, VARIANT_1, VARIANT_3 | VARIANT_3, VARIANT_2]
<u>Children chromosomes:</u>
Child 1: [-, - | VARIANT_2, VARIANT_3, VARIANT_3 | -, -]
Child 2: [-, - | VARIANT_1, VARIANT_1, VARIANT_3 | -, -]
<u>Children chromosomes:</u>
Child 1: [-, - | VARIANT_2, VARIANT_3, VARIANT_3 | VARIANT_3, VARIANT_2]
Child 2: [-, - | VARIANT_1, VARIANT_1, VARIANT_3 | VARIANT_3, VARIANT_2]
<u>Children chromosomes:</u>
Child 1: [VARIANT_1, VARIANT_1 | VARIANT_2, VARIANT_3, VARIANT_3 | VARIANT_3, VARIANT_2]
Child 2: [VARIANT_2, VARIANT_3 | VARIANT_1, VARIANT_1, VARIANT_3 | VARIANT_3, VARIANT_2]
<u>Simulation parameters:</u>
Number of mutable genes: 3
The indexes drawn: 0,2,3
Individual: [VARIANT_1, VARIANT_2, VARIANT_2, VARIANT_3, VARIANT_3, VARIANT_3, VARIANT_1]
Genes corresponding to the indices: VARIANT_1, VARIANT_2, VARIANT_3
<u>Individual after mutation:</u>
[VARIANT_1, VARIANT_2, VARIANT_3, VARIANT_2, VARIANT_3, VARIANT _3, VARIANT_1]

Figure 4: The example of a crossover and mutation

The genetic algorithm was run 8 times in order it is possible to analyse the average value of the production lead time. The results of the genetic algorithm implementation in all runs are shown in Fig. 5.

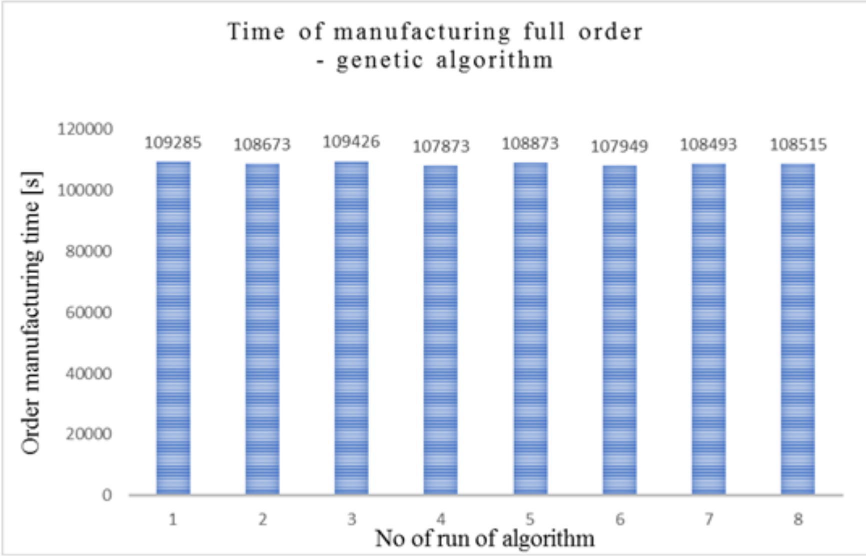

Figure 5: Time of manufacturing full order with the use of genetic algorithm

The lowest of all 8 runs equals 107 873 seconds (the fourth run). As observed, the genetic algorithm application in this case allowed to decrease the total manufacturing time of 410 pcs of products by 10 337 seconds (8.7%).

3.2 Greedy algorithm

The greedy algorithm was used as the second proposed solution. Likewise, also in this case an independent simulation of the production process and the same schedule presentation were used to calculate the order fulfilment time. Greedy algorithms are based on finding the solution by preparing it step-by-step, which means that the algorithm repeatedly finds the local optimum of the problem that is being analysed. Greedy algorithms are also used in various scheduling-type problems [Burduk 2019; Musiał 2019; Nagano 2020].

The only parameter that controls the operation of the algorithm is the step size. The step size is a number of elements for which the local optimization will be performed. In the implementation prepared for testing, the algorithm starts with an empty main production schedule. Then, depending on the step size, it generates all the possible schedule combinations according to the available variants (for example, when all Variants 1 are already manufactured, the combinations will be made from Variant 2 and Variant 3). Next, production times are calculated for all schedules. One of them, with the shortest order fulfilment time, is selected and entered into the main production schedule.

In the next step, the procedure of generating the combination is repeated. However, the evaluation of the production time takes into account the partial solutions previously added to the main schedule. The algorithm ends when there are no products left to be finished in the schedule. If the number of all products is not divisible by the step size, the schedule generation with the remaining products will be generated in the last loop of the algorithm. An example of the algorithm operation is presented below (Fig. 6).

Simulation parameters:
Step size: 1
Main schedule: []

1. *Generated combinations and production time*

[VARIANT_1] - 2134s
[VARIANT_2] - 2152s
[VARIANT_3] - 2107s

2. *Best result assignment to the main schedule*

Main schedule: [VARIANT_2]

3. *Next combinations generation*

[VARIANT_1] – 2134s
[VARIANT_2] - 2152s
[VARIANT_3] - 2107s

4. *Manufacturing time calculation with the main schedule*

[VARIANT_2, VARIANT_1] - 2671s
[VARIANT_2, VARIANT_2] - 2753s
[VARIANT_2, VARIANT_3] - 2594 s

5. *Best result assignment to the main schedule*

   Main schedule: [VARIANT_2, VARIANT_3]

Figure 6: The example of the greedy algorithm operation

As a result of scheduling the process with the greedy algorithm, eight results were obtained, depending on the previously discussed step sizes. This relationship is presented in Fig. 7. Due to a very long operation time of the algorithm that does not meet company requirements, for a step longer than 8, subsequent measurements were abandoned.

Figure 7: Time of manufacturing full order with the use of greedy algorithm

The greedy algorithm turned out to be a worse solution of fulfilling the order than the method that is currently being used (the loop schedule). The lowest manufacturing time equals 130 359 seconds, which is over 4.5 workdays. The solution proposed by a greedy algorithm resulted in the increase of the total manufacturing time of 410 pcs of products by 12 037 seconds, which is 10.2%. Thus, the further research on the use of a greedy algorithm in this case was abandoned.

3.3 Production lines idle time

The additional research of genetic algorithms in this case involved the comparison of the idle time of each line. The setup times of machines are made once per batch and are also added to idle time. Idle time means here "not in production". According to the calculations, the lines are idle for about 83 394 seconds in total (idle time depends on the line type). The implementation of a genetic algorithm allowed to decrease this time to 53 176 seconds in total (Tab. 2).

Idle time [sec]	**Schedule type**	**Line 1**	**Line 2**	**Line 3**	**Sum**
	Loop 30/20/15	21 620	27 577	34 197	83 394
	Genetic algorithm	7 565	19 764	25 847	53 176

Table 2: Idle time of each line

As observed, a genetic algorithm proposed the solution that allowed not only to save the time of fulfilling the order, but also to decrease the idle time of all lines by 30 218 seconds in total, which is over 36%. It means that the company is able to use the machines in a more effective way only by performing a genetic algorithm-based production scheduling process. The machines utilization, which is connected among others with their idle time, is also considered as a scheduling factor [Lee 2020; Liou 2020; Nazar 2018].

3.4 Results

In order to perform the analysis of effectiveness of the proposed solutions, the comparison of manufacturing time of a total order was performed (Fig. 8).

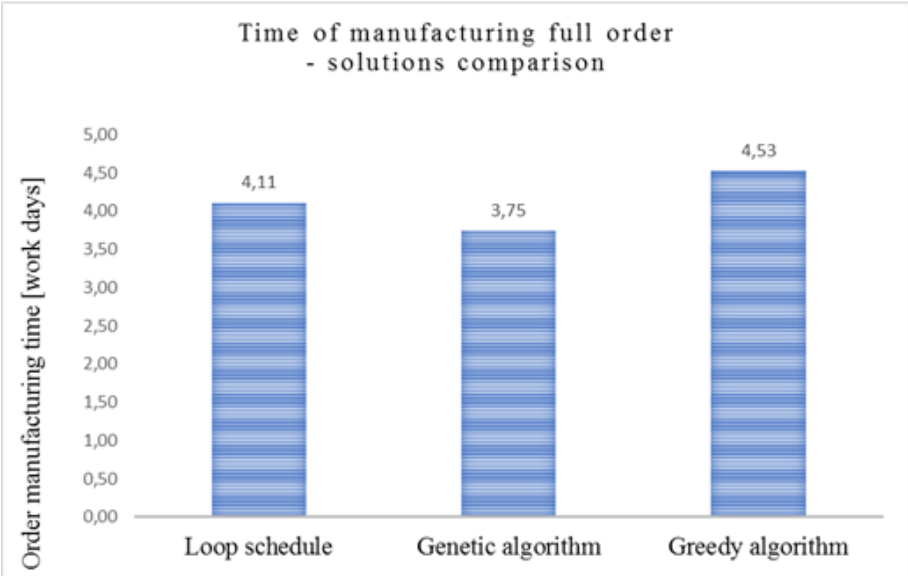

Figure 8: Manufacturing time – schedules comparison

In case of the studied company, the schedule used by process engineers (loop) allows the implementation of 410 pcs of products in 4.11 work days. If a genetic algorithm is used, the production of a full order can be done in 3.75 work days, which means that this time can be shortened and it allows a total savings of 8.7% of manufacturing time for one order. However, the greedy algorithm allowed to schedule the production of an order to 4.53 work days, which means it is worse than the current scheduling method.

4 Conclusions

There were two aims of the research – a practical one, which was to reschedule the production process and to decrease the total production time, and a theoretical one – to compare the heuristic algorithms in the area of their usefulness and effectiveness in a scheduling process. The first aim was achieved and described in the previous part of the paper. The second one was achieved too. There is a possibility to use both of the proposed algorithms in the production process scheduling by adjusting them to the examined case. However, only one of them yielded a better result than the loop approach.

Furthermore, what is important is that the proposed optimization methods require the company only to change the organization of production without incurring any costs, only by using the existing resources. It is a really desirable factor, allowing not only to save time but also to use resources and financial savings more efficiently.

The conclusion in this case is that the greedy algorithm performed a worse production lead time than the method that is currently used by engineers (a loop method), but the genetic algorithm allowed to achieve the production lead time reduction, which makes it a better solution in this case.

However, both algorithms are possible to be used in scheduling problems in other manufacturing processes. These kinds of heuristic methods are especially usable in the processes that require to consider various factors and resources which are used in a process and where a human factor can result in increasing the production lead time.

In the near future it seems to be reasonable to conduct research on the use of heuristic algorithms in scheduling processes primarily in the context of two aspects: the continuation of the research on the effectiveness of the use of algorithms in a production scheduling process and the study of specific types of algorithms in the field of the characteristics of industries (i.e. which algorithm allows you to achieve the best results in which type of a process).

References

[Ahmadian 2021] Ahmadian, M. M., Salehipour, A., Cheng, T. C. E., A meta-heuristic to solve the just-in-time job-shop scheduling problem; European Journal Of Operational Research, vol. 288(1), pp. 14-29, 2021.

[Antosz 2019] Antosz, K., Ratnayake, R. C., Spare parts' criticality assessment and prioritization for enhancing manufacturing systems' availability and reliability; Journal of Manufacturing Systems, vol. 50, pp. 212-225, Oxon, 2019.

[Bazzazi 2009] Bazzazi, M., Safaei, N., Javadian, N., A genetic algorithm to solve the storage space allocation problem in a container terminal; Computers & Industrial Engineering, vol. 56(1), pp. 44-52, 2009.

[Blum 2003] Blum, C., Roli, A. Metaheuristics in combinatorial optimization: Overview and conceptual comparison; ACM computing surveys (CSUR), vol. 35(3), pp. 268-308, 2003.

[Bożejko 2018] Bożejko, W., Pempera, J., Wodecki, M., Minimal cycle time determination and golf neighborhood generation for the cyclic flexible job shop problem; Bulletin Of The Polish Academy Of Sciences-Technical Sciences, vol. 66, pp. 333-344, Philadelphia, 2018.

[Bożejko 2019] Bożejko, W., Pempera, J., Wodecki, M., Minimization of the number of employees in manufacturing cells; In: Graña, M., et al. (eds.) SOCO'18-CISIS'18-ICEUTE'18 2018. AISC, Springer, Cham, vol. 771, pp. 241–248, 2019.

[Brandao 2004] Brandão J., A tabu search algorithm for the open vehicle routing problem; European Journal of Operational Research, vol. 157(3), pp. 552-564, 2004.

[Burduk 2017] Burduk, A., Musiał, K., Genetic algorithm adoption to transport task optimization; In: Graña, M., López-Guede, J., Etxaniz, O., Herrero, Á., Quintián, H., Corchado, E. (eds.) International Joint Conference ICEUTE 2016, SOCO 2016, CISIS 2016. Advances in Intelligent Systems and Computing, Springer, Cham, vol. 527, pp. 366-375, 2017.

[Burduk 2018] Burduk, A., Kochańska, J., Musiał, K., Rationalization of Production Order Execution with Use of the Greedy and Tabu Search Algorithms; In: Advances in Intelligent Systems and Computing, Springer, Cham ,vol. 771, pp. p. 251-259, 2018.

[Çaliş 2015] Çaliş, B., Bulkan, S., A research survey: review of AI solution strategies of job shop scheduling problem; Journal of Intelligent Manufacturing, vol. 26(5), pp. 961-973, 2015.

[Chen 2019] Chen, C. and Tiong, L. K., Using queuing theory and simulated annealing to design the facility layout in an agv-based modular manufacturing system; International Journal of Production Research, vol. 57(17), pp. 5538-5555, 2019.

[Cheng 2014] Cheng, S., Lu, H., Lei, X., Shi, Y., A quarter century of particle swarm optimization; Complex & Intelligent Systems, vol. 4(3), pp. 227-239, 2018.

[Cordeau 2004] Cordeau, J. F., Gendreau, M., Laporte G., A tabu search heuristic for periodic and multi-depot vehicle routing problems; Networks, vol. 30(2), pp. 105-119, 1997.

[Damm 2016] Damm, R. B., Resende, M. G. Ronconi, D. P., A biased random key genetic algorithm for the field technician scheduling problem; Computers & Operations Research, vol. 75, pp. 49-63, 2016.

[Dorigo 1996] Dorigo, M., Caro, G. D. I Gambardella, L. M., Ant system: optimization by a colony of cooperating agents; In: IEEE Transactions on Systems, Man and Cybernetics, vol. 26(1), pp. 29-41, 1996.

[Dostatni 2018] Dostatni, E., Rojek, I., Hamrol, A., The Use of Machine Learning Method in Concurrent Ecodesign of Products and Technological Processes; Advances in Manufacturing, 5th International Scientific - Technical Conference (MANUFACTURING), Poznań, pp. 321-330, 2018.

[Du 2020] Du, J., Dong, P., Sugumaran, V., Castro-Lacouture, D., Dynamic decision support framework for production scheduling using a combined genetic algorithm and multiagent model; Expert Systems, vol. 38(1), e12533, 2020.

[Dunker 2003] Dunker, T., Radons, G., Westkamper, E., A coevolutionary algo-rithm for a facility layout problem; International Journal of Production Research, vol. 41(15), pp. 3479-3500, 2003.

[Franzin 2019] Franzin, A., Stutzle, T., Revisiting simulated annealing: A component-based analysis; Computers & Operations Research, vol. 104, pp. 191-206, 2019.

[Gola 2018] Gola, A., Kłosowski, G., Application of fuzzy logic and genetic algorithms in automated works transport organization; Omatu, S., Rodríguez, S., Villarrubia, G., Faria, P., Sitek, P., Prieto, J. (eds.) DCAI 2017, AISC, Springer, Cham, vol. 620, pp. 29–36, 2018.

[Grabowski 2004] Grabowski J. i Wodecki M., A very fast tabu search algorithm for the permutation flow shop problem with makespan criterion; Computers & Operations Research, vol. 31(11), 2004, pp. 1891-1909

[Grznár 2021] Grznár, P., et al. et al., The Use of a Genetic Algorithm for Sorting Warehouse Optimisation; Processes, vol. 7(9), pp. 1-1, 2021.

[Gupta 2006] Gupta, A. K., Appa I. S., Job shop scheduling techniques in semiconductor manufacturing; The International Journal of Advanced Manufacturing Technology, 27(11), pp. 1163-1169, 2006.

[Homyouni 2020] Homayouni, S. M., Fontesa D. B. M. M., Gonçalves, J. F., A multistart biased random key genetic algorithm for the flexible job shop scheduling problem with transportation; International Transactions in Operational Research, 2020.

[Hua 1996] Hua M., A greedy algorithm for interval greedoids; Central European Journal of Mathematics, vol. 18(1), 2018.

[DeVore 1996] DeVore R. A. i Temlyakov V. N., Some remarks on greedy algorithms; Advances in Computational Mathematics, vol. 5(1), pp. 173-187, 1996.

[Jasiulewicz-Kaczmarek 2016] Jasiulewicz-Kaczmarek, M., Bartkowiak, T., Improving the performance of a filling line based on simulation; ModTech International Conference - Modern Technologies in Industrial Engineering IV, IOP Conf. Series: Materials Science and Engineering 145, Iasi, Romania, Vol. 145(4), 2016.

[Kahraman 2010] Kahraman, C., Engin, O., Kaya, I., & Öztürk, R. E., Multiprocessor task scheduling in multistage hybrid flow shops: A parallel greedy algorithm approach; Applied Soft Computing, vol. 10(4), pp. 1293-1300, 2010.

[Kalinowski 2017] Kalinowski, K., Krenczyk, D., Paprocka, I., Kempa, W. M., Grabowik, C., Ant colony optimisation for scheduling of flexible job shop with multi-resources requirements; 21st Innovative Manufacturing Engineering & Energy International Conference: IManE&E, Vol. 112, 2017.

[Kiranyaz 2014] Kiranyaz S., Ince T. and Gabbouj M., Multidimensional particle swarm optimization for machine learning and pattern recognition, Springer, pp. 625-641, 2014

[Kochańska 2019] Kochańska, J., Musiał, K., Burduk, A., Rationalization of decision-making process in selection of suppliers with use of the greedy and Tabu Search algorithms; Intelligent Systems in Production Engineering And Maintenance, Springer, Cham, pp. 275-284, 2019.

[Kramer 2021] Kramer O., Genetic algorithm essentials, Springer International Publishing, pp. 11-19, 2017.

[Krenczyk 2020] Krenczyk, D., Dziki, K.: Heuristic and backtracking algorithm for multimanned assembly line balancing problem with location constraints; Cybernetics And Systems, vol. 51(7), pp. 698-713, 2020.

[Ku 2011] Ku, M. Y., Hu, M. H, Wang, M. J., Simulated annealing based parallel genetic algorithm for facility layout problem; International Journal of Production Research, vol. 49(6), pp. 1801-1812, 2011.

[Kulturel-Konak 2011] Kulturel-Konak, S. Konak, A., Unequal area flexible bayfacility layout using ant colony optimisation; International Journal of Production Research, vol. 49, pp. 1877-1902, 2011.

[Kumann 2006] Kumanan, S., Jegan, J. G., Raja K., Multi-project scheduling using an heuristic and a genetic algorithm; The International Journal of Advanced Manufacturing Technology, vol. 31, pp. 360-366, 2006.

[Lee 2020] Lee, S., Issabakhsh, M., Jeon, H.W., Idle time and capacity control for a single machine scheduling problem with dynamic electricity pricing; Operations Management Research, vol. 13, pp. 197–217, 2020.

[Liou 2020] Liou, J.P., Dominance conditions determination based on machine idle times for the permutation flowshop scheduling problem; Computers & Operations Research, vol. 122, 2020.

[Li 2014] Li, X., Xiaoyu W., Liang G., An effective genetic algorithm for multi-objective integrated process planning and scheduling with various flexibilities in process planning. J. Univers. Comput. Sci. vol. 20(14), pp. 1926-1950, 2014.

[Luo 2020] Luo, Q., Deng, QW. Gong, GL., Zhang, LK. Han, WW., Li, KX. An efficient memetic algorithm for distributed flexible job shop scheduling problem with transfers; Expert Systems With Applications, vol. 160, 2020.

[Musiał 2019] Musiał, K., Burduk, A., Łampika, Ł., Case study of production planning optimization with use of the greedy and tabu search algorithms; Advances in Intelligent Systems and Computing, Springer, Cham, vol. 835, pp. 66-75, 2019.

[Nagano 2020] Nagano, M.S., de Almeida, F.S., Miyata, H.H., An iterated greedy algorithm for the no-wait flowshop scheduling problem to minimize makespan subject to total completion time; Engineering Optimization, 53(8), pp. 1431-1449, 2020.

[Nazar 2018] Nazar, K.P.A., Pillai, V.M., Mixed-model sequencing problem under capacity and machine idle time constraints in JIT production systems; Computers & Industrial Engineering, vol. 118, pp. 226-236, 2018.

[Ociepka 2013] Ociepka, P., Gwiazda, A., Concept of hybrid system for computer aided of machines design process; Selected Engineering Problems. vol. 4, 2013.

[Olender 2018] Olender, M., Kalinowski, K., Grabowik, C., Practical approach of flexible job shop scheduling using costs and finishing times of operations; Advances in Intelligent Systems and Computing, Springer, Cham, vol 835, pp. 391-400, 2018.

[Oshin 2016] Oshin C. A., Job scheduling using ant colony optimization in grid environment; International Conference on Electrical, Electronics and Optimization Techniques (ICEEOT), pp. 2845-2850, 2016.

[Paes 2017] Paes F. G., Pessoa A. A. and Vidal T., A hybrid genetic algorithm with decomposition phases for the unequal area facility layout problem; European Journal of Operational Research, vol. 256(3), pp. 742-756, 2017.

[Patalas-Maliszewska 2020] Patalas-Maliszewska J., Topczak M., Kłos, S., The level of the additive manufacturing technology use in polish metal and automotive manufacturing enterprises; Applied Sciences-Basel, vol. 10(3), Switzerland, 2020.

[Pindedo 1995] Pinedo, M., Scheduling, theory, algorithms and systems PrenticeHall, Englewood Cliffs, New York, 1995.

[Rojek 2015] Rojek I., Tooling selection in technological processes using neural networks; Archives of Mechanical Technology and Materials, vol. 35, 2015.

[Rojek 2019] Rojek, I., Dostatni, E., Hamrol, A., Automation and digitization of the material selection process for eco-design; In: Burduk, A., Chlebus, E., Nowakowski, T., Tubis, A. (eds.) ISPEM 2018. AISC, Springer, Cham, vol. 835, pp. 523–532, 2019.

[Sobaszek 2017] Sobaszek, Ł., Gola, A., Kozłowski, E., Application of survival function in robust scheduling of production jobs; In: Ganzha, M., Maciaszek, M., Paprzycki, M. (eds.) Proceedings of the Federated Conference on Computer Science and Information Systems (FEDCSIS), New York, 2017.

[Stadnicka 2019] Stadnicka, D., Antonelli, D., Human-robot collaborative work cell implementation through lean thinking; International Journal of Computer Integrated Manufacturing, vol. 32(6), pp. 580-595, 2019.

[Tang 2014] Tang, S. H., Homayouni, S. M., Motlagh, O., A genetic algorithm for optimization of integrated scheduling of cranes, vehicles, and storage platforms at automated container terminals; Journal of Computational And Applied Mathematics, vol. 270, pp. 545-556, 2014.

[Więcek 2019] Więcek, D., Więcek, D., Kuric, I., Bučková, M., Krajčovič M., Evaluation of the effectiveness of implementing production logistics automation systems supported by computer simulation tools; Manufacturing (Mtem 2019), MATEC Web of Conferences, vol. 299, Romania 2019.

[Zegordi 2009] Zegordi S. and Nia M. B., A multi-population genetic algorithm for transportation scheduling; Transportation Research Part E: Logistics and Transportation Review, vol. 45(6), pp. 946-959, 2009.

[Zhang 2020] Zhang, F., Mei, Y., Nguyen, S., Zhang, M., Evolving scheduling heuristics via genetic programming with feature selection in dynamic flexible job-shop scheduling; IEEE Transactions on Cybernetics, 51(4), pp. 1797-1811, 2020.

[Zwolińska 2017] Zwolinska, B., Grzybowska, K., Kubica, L., Shaping production change variability in relation to the utilized technology; DEStech Transactions on Engineering and Technology Research (ICPR 2017), Destech Publications Inc., 2017.