www.czasopisma.pan.pl PAN POLSKA AKADEMIA NAUK www.journals.pan.pl

Management and Production Engineering Review
Volume 13 • Number 4 • December 2022 • pp. 98–106
DOI: 10.24425/mper.2022.142398

Scheduling of the Manufacturing Cell Work with the Use of a Genetic Algorithm on the Example of a Flexible Production System

Marcin Matuszny

University of Bielsko-Biala, Bielsko-Biała, Poland

Received: 27 February 2022
Accepted: 7 August 2022

Abstract
The article is to present the application of genetic algorithm in production scheduling in a production company. In the research work the assumptions of the methodology were described and the operation of the proposed genetic algorithm was presented in details. Genetic algorithms are useful in complex large scale combinatorial optimisation tasks and in the engineering tasks with numerous limitations in the production engineering. Moreover, they are more reliable than the existing direct search algorithms. The research is focused on the effectivity improvement and on the methodology of scheduling of a manufacturing cell work. The genetic algorithm used in the work appeared to be robust and fast in finding accurate solutions. It was shown by experiment that using this method enables obtaining schedules suitable for a model. It gives a group of solutions that are at least as good as those created by the heuristic rules.

Keywords
Production planning and control, Production scheduling, Genetic algorithm, Production engineering, Production process.

Introduction

Over the last decades researchers have started to observe the nature more carefully and to learn from it with humility. The result of such observations was creating, among others, procedures capable to imitate the nature, such as evolutionary algorithm or neuron networks. The incrementing interest in genetic algorithms has been observed since John Holland (Holland, 1992) published his work. Furthermore, the constant advancement in the computers development has influenced the practical application of genetic algorithms. The article presents the application of genetic algorithm in a productive company to production scheduling.

One of the basic task of the scheduling subsystem is setting the order of the realization of manufacturing operations. The most important task is to meet the clients' expectations through the guarantee of meeting the deadline of the production orders' realization with the simultaneous profit maximization gained through selling the productions orders received for realization (Jardzioch & Skobiej, 2013; Lee & Kim, 2001). The simultaneous goal of scheduling is maximization of utilization the machines, which is more widely described in the manuscripts (Jin et al., 2020; Wang et al., 2020), where the authors discuss the topic of improving resource utilization by timely fine-grained scheduling and detailed planning, as well as the problem of multi-objective production scheduling for balanced production of manufactures. The issue of scheduling the production tasks is a significantly complex area analyzed in many research centers. The analyzed problems concern the broad spectrum: from theoretical issues looking for optimal solutions to the works in which the authors suggest suboptimal solutions emphasizing the possibilities of practical application of the developed methods. The solutions that are suggested most often include static priority use, metaheuristic algorithms (ex. Tabu Search procedure), simulated annealing, immunological algorithms and genetic algorithms (Wang et al., 2013). In the production scheduling there are often used priority rules because of their simple structure and reduced

***Corresponding author**: Marcin Matuszny – Department of Production Engineering, Faculty of Mechanical Engineering and Computer Science, University of Bielsko-Biala, Willowa 2, 43-300 Bielsko-Biała, Poland, e-mail: marcinmatuszny@wp.pl*

computational complexity (Binchao & Matis, 2013; Jardzioch & Bulwan, 2013). Recently more and more research focus on the problem of robust scheduling to reduce the impact of uncertainties during the optimization process (Paprocka & Skołud, 2013; Cheref et al., 2016; Himmiche et al., 2018). The authors combine various aspects and fields of using robust scheduling in their works, for example in (Sobaszek et al., 2020) authors presents the framework of a predictive job scheduling technique for application in the job-shop environment under the machine failure constraint. Due to the fast development in the production engineering and the ongoing digitization trend, production companies handle with important challenges in market environments: a continuous tendency to shortening the production time and to optimization of the production processes (Waschneck et al., 2018).

Genetic algorithms constitute a method for problems solving, especially for optimization tasks. They are characterized by versatility and simplicity of the best solutions searching through the stochastic methods. Genetic algorithms are search procedures based on the mechanisms of natural selection and inheritance. They use the evolutionary principle saying that the most adapted beings are able to survive. They differ from the traditional optimizing methods in some essential elements (Rutkowska et al., 1999). Because of the topic of this study, the figure below (Fig. 1) presents the fundamental characteristic of genetic algorithms. The research question in this work is the need of scheduling of the manufacturing cell work. Each manufacturing cell works requires scheduling, as shown by numerous researches using various tools (Serrano-Ruiz et al., 2021). Detailed issues, assumptions and the essence of the problem are developed more broadly and precisely in the further part of the manuscript. The result of the research is also to present in details the proposed algorithm operation

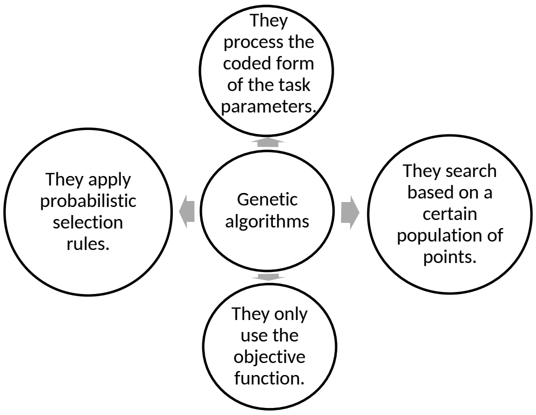

Fig. 1. Selected features of genetic algorithms due to the subject of the study

to the scheduling of the manufacturing centre work, basing on the example of a flexible production system. The proposed and presented algorithm at work solves the problem of scheduling of the manufacturing cell work in a flexible production system.

Literature review

Genetic algorithms are iterative optimization algorithms that simulate the process of natural evolution. Genetic algorithms consist of three main operators (Fig. 2) as shown in the figure below (Chen et al., 2016; 2018).

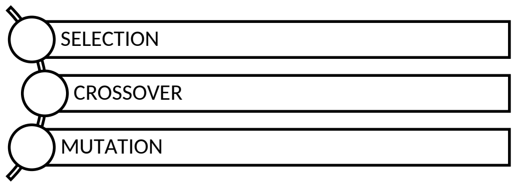

Fig. 2. Operators constituting the genetic algorithms

The essence and application of genetic algorithms

A group of candidate solutions are specified (the initial population). The fatnesses of these candidates are calculated and a selection operator selects a subset of the fitter candidates which are used to generate a new population. In crossover some pairs of these selected candidates are combined (using a variety of methods) to produce new candidate solutions. Some candidates are then subjected to mutation – random changes that will produce changes in fitness. After crossing and mutation, a new population was obtained. The process is repeated a number of times in order to generate fitter solutions than those in the initial population (Chen et al., 2018).

Genetic algorithm enables forecasting the demand for the particular products in order to set the production in a given planning period (Savsani et al., 2016). Genetic algorithms are a tool for searching optimum of the objective function that enables parallel solutions search. The function which is to search does not have to be smooth or continuous. It can be any function notated in any way. It is important that after substituting variables its value could be calculated. Below there is an example of the function generated in MATLAB (Fig. 3), which extremum can be searched through genetic algorithms.

Three main groups of genetic algorithms can be distinguished: search algorithms, optimisation algorithms and machine learning algorithms ((Kwaśnicka,

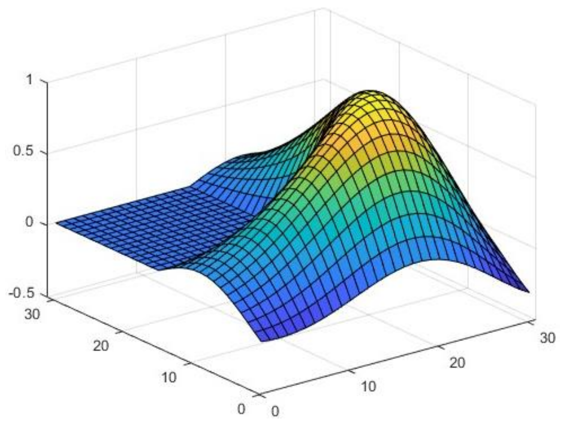

Fig. 3. Example of a function generated in the MATLAB environment, which extremum can be searched through genetic algorithms

1999). However, those groups are not separate and the boundary between them is smooth. The technical limitations are usually the same for the standard genetic algorithms and the genetic algorithms that were suggested and improved by the authors (Ancveire et al., 2019). Among those technical limitations there can be distinguished: population size limitations, objective function limitations (Stylianou & Andreou, 2012), the maximum number of iterations (Stylianou & Andreou, 2011; Li & Dong, 2018; Antoniol et al., 2004), data set size for model training (Samath et al., 2017), one selection method (Stylianou & Andreou, 2011), mutation rate and crossover rate limitations, (Stylianou & Andreou, 2011; Antoniol et al., 2004).

On the basis of (Rutkowska et al., 1999; Lee & Kim, 2001), block diagram of the classical genetic algorithm operation is presented in the Figure 4.

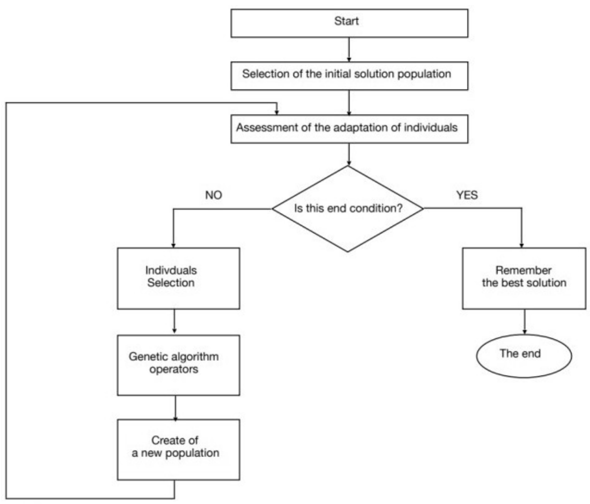

Fig. 4. Block diagram of the classical genetic algorithm

Materials and methods

Application of genetic algorithm in production scheduling

Assumptions of the methodology

The article presents the application of genetic algorithm in scheduling work centre operation. The algorithm used in conducting the research is shown below (Fig. 5).

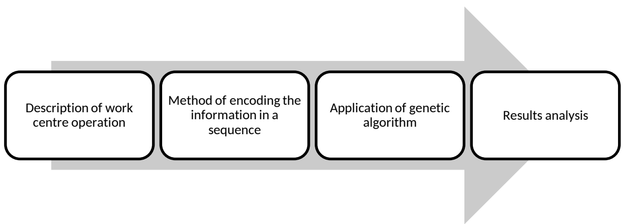

Fig. 5. Research methodology

The code sequence presented in the research has a simple structure, but reading the information on its basis is slightly more complex. Next, the terms used in the research part were described. "Element" indicates on an item or a group of identical items produced in the work centre. An example element is presented in the figure below (Fig. 6). In the company that is the subject of the research scheduling of the work centre operation is done in order to do several tasks. To produce a particular element a certain number of operations is to be realised. The operations are to be conducted in a particular order. Only one machine is occupied with a given operation in a specific time. None of the operations in the analysed company needs to be done through several machines at the same time.

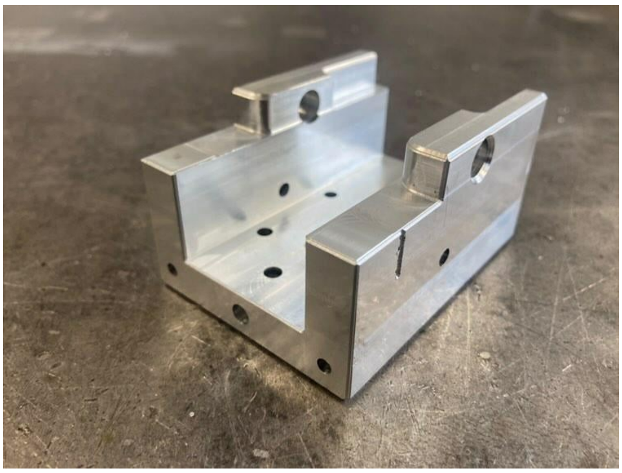

Fig. 6. Example of an element produced in the analyzed enterprise

Description of production system

In the analysed company there are 5 machines. The production system is an open system influenced by the environment in which it is realised. The environment can be divided into nearer and further and the factors affecting directly the production processes are distinguished there.

The processes are multi-stage socket type with the parallel machine (CNC milling machines). Each of the operations implemented into the system can be done by one of several machines available. Necessary condition is that each of the machines should enable doing the particular operation completely. In other words, it is impossible to divide an operation into a few sub-operations done with the use of different CNC milling machines. If more than one machine that can do an operation is available, then it is possible to choose one of many potential ways to pass the elements through the system. The element will be done in 6 operations. In the table below the possibility of doing a particular operation on a particular machine is presented (Table 1).

Table 1
Machine assignment to particular operations

Operation number	Machine number				
	M_1	M_2	M_3	M_4	M_5
E_1	YES	YES	YES	NO	NO
E_2	YES	YES	YES	NO	YES
E_3	YES	YES	YES	YES	YES
E_4	YES	YES	YES	YES	NO
E_5	YES	NO	YES	YES	NO
E_6	YES	NO	NO	YES	NO

Example: the E6 operation can be done by the machine M1 or M4, the machine M1 can do every operation. The operations E2 and E3 can be done by the machine M5, the operation E3 can be done by every machine, etc.

It is worth to emphasise that the time of doing an operation may differ depending on the machine it will be done by. Example: the operation E6 will last longer when done by the machine M4 than by the machine M1.

According to the assumptions mentioned above, 1440 routes of passing the material through the system were created. In the table below (Table 2) there are presented 20 examples. In the columns there are numbers of the machines that should do the operations. Example: operation E1 by the machine M3 → the operation E2 by the machine M5 → the operation E3 by the machine M2 → the operation E4 by the machine M3 → the operation E5 by the machine M4 → operation E6 by the machine M4.

Moreover, time of doing an operation by a particular machine means time needed to do this operation for all the elements that are on the pallet with time of the manipulating with these items. It is assumed that pallet is the smallest production unit moving in the production system. It is not relevant how many elements are on the pallet. To determine the size of the production batch the amount of the elements on the pallet is taken into consideration. Then, the amount of the elements in the batch are divided by the size (the amount of elements on the pallet) and this way the size of the production batch is presented by the amount of the pallets that are to be passed through the system.

Input includes: time of doing the operation for five machines introduced into the production system; moreover, distance between production sites and the velocity of the pallets movement are also introduced, what enables determining the delays related to the transport of those elements. The time needed for changeover is assigned to the particular machines. This time is assumed to be constant and the changeover occurs always between two different operations and before the machine is to do the first operation. In the case being discussed the deadlines of the specific tasks are also important. Exceeding the

Table 2
Example routes for the element production in the system with five machines and six operations

Operation number	Machine number																			
E_1	M_3	M_1	M_3	M_2	M_3	M_3	M_1	M_1	M_2	M_3	M_1	M_3	M_2	M_3	M_2	M_1	M_3	M_1	M_1	M_3
E_2	M_5	M_1	M_2	M_2	M_5	M_3	M_1	M_2	M_2	M_5	M_3	M_1	M_2	M_3	M_5	M_1	M_3	M_3	M_5	M_1
E_3	M_2	M_1	M_4	M_2	M_5	M_3	M_5	M_1	M_5	M_3	M_3	M_5	M_4	M_3	M_2	M_4	M_1	M_3	M_5	M_1
E_4	M_3	M_1	M_3	M_2	M_1	M_3	M_1	M_1	M_3	M_3	M_1	M_1	M_4	M_2	M_3	M_4	M_3	M_2	M_1	M_1
E_5	M_4	M_1	M_4	M_1	M_3	M_3	M_4	M_1	M_3	M_3	M_1	M_3	M_4	M_4	M_1	M_4	M_1	M_1	M_1	M_1
E_6	M_4	M_1	M_4	M_1	M_4	M_1	M_1	M_1	M_4	M_4	M_1	M_4	M_1	M_1	M_4	M_4	M_4	M_4	M_1	M_4

deadline of any task causes imposing additional penalties on a code sequence representing the schedule. Before the schedule for the input given is established, it is not necessary that the machines are free in the beginning. It can be declared when, after the schedule was started to be created, the particular machines are free. In the tables below (Table 3–8) the necessary input for which genetic algorithm was searching the solution is presented. The task is to be understood as an item or a group of items produced in the manufacturing cell, so from this the difference between the operations' time for particular tasks on the proper machines in a flexible production system results. Various tasks are realized there. The aim of scheduling is to create such a schedule in which – with the established number of tasks to be done – work time of the system, total time of the pallets transport and total time of changeover will be the shortest, the delays in performing a tasks will be the slightest and the number of the delayed tasks (which deadline was exceeded) will be limited to minimum. The time of the deadlines is given in minutes. The velocity of the pallets is 0.5 [m/s].

Table 3
Input for the task 1 – time of the operation, deadline: 700

Machine number	Operation number					
	E_1	E_2	E_3	E_4	E_5	E_6
M_1	12			6		
M_2		20	34			
M_3	16			17		
M_4					31	
M_5		36	58			

Table 4
Input for the task 2 – time of the operation, deadline: 300

Machine number	Operation number					
	E_1	E_2	E_3	E_4	E_5	E_6
M_1			39		42	15
M_2	41	16				
M_3	56		51		64	
M_4				16		19
M_5		24				

Moreover, the moment in which the machine can start working was provided, respectively: M1 = 0 [s], M2 = 58 [s], M3 = 30 [s], M4 = 0 [s] and M5 = 5 [s].

Table 5
Input for the task 3 – time of the operation, deadline: 360

Machine number	Operation number					
	E_1	E_2	E_3	E_4	E_5	E_6
M_1		9	5			
M_2	15					
M_3	32	15	14			
M_4				19		
M_5						

Table 6
Input for the task 4 – time of the operation, deadline: 700

Machine number	Operation number					
	E_1	E_2	E_3	E_4	E_5	E_6
M_1	16	28			19	
M_2			14	23		
M_3	23	49			26	
M_4				58		21
M_5			29			

Table 7
Input data – distances between machines [m]

	M_1	M_2	M_3	M_4	M_5
M_1		12	50	5	16
M_2	12		52	17	20
M_3	50	52		50	34
M_4	5	17	50		17
M_5	16	20	34	17	

Table 8
Time for machine retooling [s]

M_1	M_2	M_3	M_4	M_5
2	3	1	5	0.17

Any required number of the pallets subject to transport from one production site to another can be in the system at any moment. In the analysed production system there can be any required number of the pallets subject to transport from one production site to another. In the assumptions the time of transport from the input warehouse and to the initial warehouse has not been included.

Description of the genetic algorithm operation

The way of data coding shown in the example lets the pallets of one production run have different routes. In the table below (Table 9) the structure of the code sequence is presented.

Table 9
Example fragment of the code sequence structure

[task number, operation number, machine number], [task number, operation number, machine number]...
[2, 5, 1], [1, 3 , 5], [4, 6, 4], [1, 1, 3], [3, 1, 2], [3, 2, 1], [4, 5, 3], [3, 3, 3], [3, 2, 3], [1, 1, 1]...

In the table above it can be seen that in the subsequent positions there are coded: task number, operation number and the number of the machine which will do an operation. Code sequence includes all the permissible configurations of the operations and machines. The order of machines and operations appearing in a sequence is very important. After choosing the sequence positions intended for further processing the fitness function for the given sequence may be calculated. The objective function is presented below (1) and the limitations of the analysed production system are described in detail in the earlier part of the work. Each of the summands can be multiplied by the appropriate weight what may allow to decide which of the summands of the objective function is the most important and which influences its value the most, depending on the type of optimization problem to which this function is to be applied. The objective function (1) assumes minimization of the total time of the task execution.

$$\min F_A = T_w + D \cdot N + T_s + \sum_{m=1}^{M} T_{rm}, \qquad (1)$$

where: F_A – objective function, T_w – system working time, D – delays of the delayed tasks in total, N – number of delayed tasks, T_s – transporting time of the pallets with their elements between production sites, T_{rm} – sum of the machine changeover time.

The mathematical model is presented by describing the parameters, variables, assumptions, and constraints in an earlier chapter. The main assumptions taken into account for this mathematical model are: the operating time of all pieces on any type of machine is definite, the number of operation is definite and fixed, the number of machines in the system is definite and fixed, not included a breakdown time for the machines, and the start-up times for each machine are specific and depend on the cell sequence. The values of the input parameters that should be given at the beginning of solving the mathematical model are included in the tables (Table 1, Tables 3–8).

There are several criteria to consider in the overall planning approach. All the criteria for cell production planning must be combined in a way that functions as a objective function. But because of the complexity of the problem and the computing time required problem. It is not possible to include all of them. Therefore, in this paper, only the total construction time completion of tasks. The aim of the mathematical model is to minimize the total production time and transporting time of the pallets with their elements between production sites.

It is worth noticing that during scheduling for such large number of tasks, operations and pallets the code sequence has an adequate number of positions. Reading information from the sequences occurs as presented below (Fig. 7). By the example of one code sequence the order of inserting the specific operations to the Gantt chart is presented. Decoding of the information provided in the code sequence is done as follows: the number of task represented by the first position of the code sequence is checked. Later, all the positions with operations needed for this task are searched. In each of the positions it is indicated which machine can do the particular operation. After setting the operations in the correct order in a specific task, first fragment of the Gantt chart can be drawn. Positions representing the operations have already been inserted into the Gantt chart (in a task given as first in the chart they are in bold) and they are to be removed from the code sequence. From the rest of the positions again the first will be chosen and again it will be checked which task's operation this position

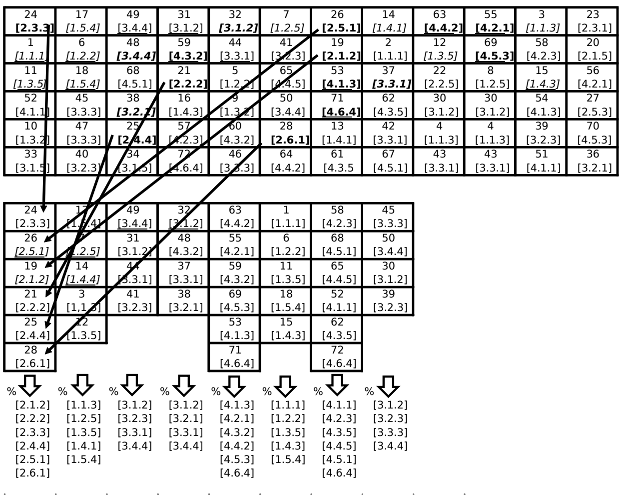

Fig. 7. Reading information from the sequences

represents. After finding all the positions representing the other operations in a given task, they are ordered and inserted into the Gantt chart and the operations are ascribed to suitable machines. The found positions are removed from the code sequence (in a task which is given as the second they are in italics). The procedure lasts until all the needed positions are removed from the sequence and all the operations are inserted in the Gantt chart.

As it can be observed, operations from the task 1 including two pallets are ascribed to the machines in such a way that each of the pallets has different route in a production system. In order to ensure the maximum machines utilisation, the time breaks between some of the operations are taken into account. The breaks are usually completed with the first operations, however, it is not a rule. During the genetic algorithm operations in sequences the operation of crossover with the 0.9 probability and the mutation with the 0.01 probability were done. The genetic algorithm processed 200 generations. During intermediate generation formation the elitist selection was used. The pairs of sequences were chosen at random and then, the sequence with a lower function value in the pair was inserted into the intermediate generation. Sampling was done until the intermediate generation obtained the required number of beings.

Results

Genetic algorithm is one of the most important metaheuristic algorithms that is used to optimize various functions. A genetic algorithm is a search algorithm that searches for an answer in an area and mimics biological evaluation processes. The genetic algorithm according to the characteristics described was used to solve the problem in this paper.

Changes in the objective function value during the genetic algorithm operation is presented below (Fig. 8). The chart presented below makes it possible

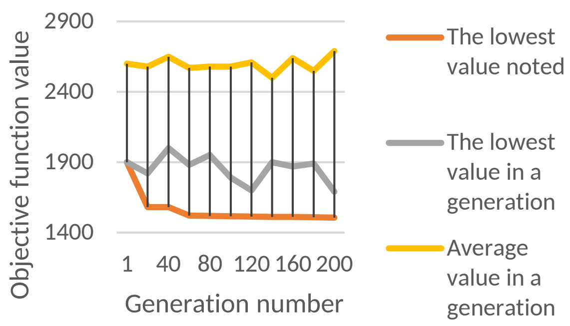

Fig. 8. Changes in objective function value

to assess the effectiveness of the proposed approach to scheduling of the manufacturing cell in a flexible production system. According to the research, this problem is of the indefinite polynomial type, which can be solved by increasing the number of tasks and machines through optimization software. Approaches such as branched and limiting and dynamic programming have a limited computation time and limited memory on the computer as the number of tasks increases. Therefore, the use of innovative and metaheuristic algorithms can be effective.

As it can be observed, from the beginning of the calculations the genetic algorithm has not found any solutions that could be definitely better than the solutions obtained in the previous generations. Long breaks in machines working time are caused by the rather slow movement of the pallets between the production sites. Work centre operates 449 seconds, only one of the tasks was delayed (its deadline was exceeded by 89 seconds). The results of the proposed method have good results, as shown in the chart above. It can therefore be concluded that the use of a genetic algorithm to scheduling the work of the manufacturing cell on the example of a flexible production system is justified and helpful in solving this problem.

Conclusions

Genetic algorithms constitute a very good optimisation tool which can be successfully used in problems which production engineering deals with. Production companies are equipped with numerous processing lines to produce elements that differ from each other. Machines and devices forming components of the production lines require continued optimisation. Optimisation may be concerned about parameters of the technological processes. Constructions and parameters of the machines and devices are significantly diversified and it is caused by the variety of the processed primary products, the produced products and the changeability of their features. Genetic algorithms may be useful in optimisation of the systems controlling technological-transport processes. The systems may be supported by the procedures searching better solutions with the artificial intelligence methods.

The work occupies with resolving the scheduling of manufacturing cell work problem in a production company, where the production is done in a flexible production system. The aim of the work was to elaborate such a method of scheduling that would permit to do this task effectively for the conditions of variable and dynamic environment with less input and time needed. The conducted research confirm that using genetic algorithms is justified and effective for the production scheduling problem. Genetic algorithm enables fast generating of acceptable solutions determining the order deadlines and balancing the production line.

In comparison to the classical static optimisation method, genetic algorithm requires more calculations, because every present population is composed of the particular number of beings for which all parameters needed are calculated and it causes that the analysed process is more time-consuming (Rutczyńska-Wdowiak & Stefański, 2010). However, it cannot be expected that a solution found by genetic algorithm will be an optimal and the best solution. Although it is not excluded, the obtained solution is very close to the optimal solution and in majority of the problems it is satisfying as there is no necessity of further searching of the optimal solution (Lee & Kim, 2001). According to the author (Michalewicz, 1999), genetic algorithms may be useful in complex combinatorial optimisation tasks on a large scale and in engineering task with many limitations. They differ from the random algorithms as they connect the elements of direct and stochastic search; this is why genetic algorithms are more reliable than direct search algorithms. Another important feature of the methods based on the genetic search is the fact that they preserve entire populations of the potential solutions, while the other methods process only one point of space search.

Acknowledgments

The author greatly appreciates the anonymous referees for their very valuable and helpful suggestions on an earlier version of this paper.

References

Ancveire I. and Poďaka I. (2019), Aplication of Genetic Algorithms for Decision-Making in Project Management: A Literature Review, *Information Technology and Management Science*, No. 22, pp. 22–31.

Antoniol G., Di Penta M., and Harman M. (2004), Search-Based Techniques for Optimizing Software Project Resource Allocation, *Genetic and Evolutionary Computation Conference*, pp. 1425–1426.

Binchao C. and Matis T.I. (2013), A flexible dispatching rule for minimizing tardiness in job shop scheduling, *International Journal of Production Economics*, No. 121, Vol. 1, pp. 360–365.

Chen Y., Elliot M., and Sakshaug J. (2016), *A genetic algorithm approach to synthetic data production*, in: PrAISe 2016 Proceedings of the 1st International Workshop on AI for Privacy and Security, Hague, Netherlands.

Chen Y., Elliot M., and Smith D. (2018), *The Application of Genetic Algorithms to Data Synthesis: A Comparison of Three Crossover Methods*, International Conference on Privacy in Statistical Databases, pp. 160–171.

Cheref A., Artigues C., and Billaut J.C. (2016), *A new robust approach for a production scheduling and delivery routing problem*, IFAC-PapersOnLine, 4 pp. 886–891.

Himmiche S., Marange P., Aubry A., and Petin J.F. (2018), *Robust production scheduling under machine failures – A DES based evaluation approach*, IFAC-PapersOnLine, No. 7, Vol. 51, pp. 271–276.

Holland J.H. (1992), *Adaptation in Natural and Artifical Systems: An Introductory Analysis with Applications to Biology, Control, and Artifical Intelligence*, 4th printing 1995, Massachusetts Institute of Technology.

Jardzioch A. and Skobiej B. (2013), Analysis of Variable Changeover Times Impact on the Revenue in Manufacturing Process, *Academic Journal of Manufacturing Engineering*, No. 11, Vol. 4, pp. 114–117.

Jardzioch A. and Bulwan K. (2013), The Prioritisation of Production Orders Under the Bee Colony Algorithm, *Advances in Manufacturing Science and Technology*, No. 37, Vol. 4, pp. 49–59.

Jin T., Cai Z., Li B., Zheng C., Jiang G., and Cheng J. (2020), *Improving resource utilization by timely fine-grained scheduling*, EuroSys '20: Proceedings of the Fifteenth European Conference on Computer Systems, pp. 1–16.

Kwaśnicka H. (1999), *Evolutionary computing in artificial intelligence*, Publishing House of the Wrocław University of Technology.

Lee H. and Kim S. (2001), Integration of process planning and scheduling using simulation based genetic algorithms, *The International Journal of Advanced Manufacturing Technology*, No. 18, pp. 586–590.

Li H. and Dong X. (2018), Multi-mode resource levelling in projects with mode dependent generalized precedence relations, *Expert Systems with Applications,* No. 97, pp. 193–204.

Michalewicz Z. (1999), *Genetic algorithms + Data structure = Evolutionary programs*, Scientific and Technical Publishing House.

Paprocka I. and Skołud B. (2013), Robust Scheduling, a Production Scheduling Model of Failures, *Applied Mechanics and Materials*, No. 307, pp. 443–446.

Rutkowska D., Piliński M., and Rutkowski L. (1999), *Neural networks, genetic algorithms and fuzzy systems,* Polish Scientific Publishers PWN.

Rutczyńska-Wdowiak K. and Stefański T. (2010), Comparative analysis of classical optimization methods and genetic algorithm on the example of filter design, *Measurements. Automation. Control*, No. 65, Vol. 6, pp. 624–627.

Samath S., Udalagama D., Kurukulasooriya H., Premarathne D., and Thelijjagoda S. (2017), *Collabcrew-An intelligent tool for dynamic task allocation within a software development team*, International Conference on Software, Knowledge Information, Industrial Management and Applications, Malabe, pp. 1–9.

Savsani P., Banthia G., Gupta J., and Vyas R. (2016), *Optimal Aggregate Production Planning by Using Genetic Algorithm*, Proceedings of the 2016 International Conference on Industrial Engineering and Operations Management, Kuala Lumpur, Malaysia, pp. 863–874.

Serrano-Ruiz J.C., Mula J., and Poler R. (2021), Smart manufacturing scheduling: A literature review, *Journal of Manufacturing Systems*, Vol. 61, pp. 265–287.

Sobaszek Ł., Gola A., and Kozłowski E. (2020), Predictive Scheduling with Markov Chains and ARIMA Models, *Applied Sciences*, No. 17, Vol. 10.

Stylianou C. and Andreou A.S. (2012), A multi-objective genetic algorithm for software development team staffing based on personality types, *IFIP Advances in Information and Communication Technology*, No. 381, Vol. 1, pp. 37–47.

Stylianou C. and Andreou A.S. (2011), Intelligent software project scheduling and team staffing with genetic algorithms, *IFIP Advances in Information and Communication Technology*, No. 364, Vol. 2, pp. 169–178.

Wang X., Xie X., and Cheng T.C.E. (2013), A modified artificial bee colony algorithm for order acceptance in two-machine flow shops, *International Journal Production Economics*, No. 141, Vol. 1, pp. 14–23.

Waschneck B., Reichstaller A., Belzner L., Altenmuller T., Bauernhansl T., Knapp A., and Kyek A. (2018), *Optimization of global production scheduling with deep reinforcement learning*, 51st CIRP Conference on Manufacturing Systems, Procedia CIRP 72, pp. 1264–1269.

Wang C., Yang B., and Wang H.Q. (2020), Multi-objective master production schedule for balanced production of manufactures, *International Journal of Simulation Modelling*, No. 4, Vol. 19, pp. 678–688.