Proceedings of the 3rd World Congress on Mechanical, Chemical, and Material Engineering (MCM'17)
Rome, Italy – June 8 – 10, 2017
Paper No. ICMIE 118
ISSN: 2369-8136
DOI: 10.11159/icmie17.118

Scheduling Customized Orders: A Case Study at BEST Transformers Company

M.A. Beyazit Ocaktan[1], Ibrahim Kucukkoc[1], Aslan Deniz Karaoglan[1], Abdullah Cicibas[2], Kadir Buyukozkan[3]
[1]Balikesir University, Department of Industrial Engineering
10145 Cagis Campus, Balikesir, Turkey
ocaktan@balikesir.edu.tr; ikucukkoc@balikesir.edu.tr, deniz@balikesir.edu.tr
[2]BEST Transformers, Research & Development Department
Organized Industrial Zone 7th Street, Balikesir, Turkey
abdullah.cicibas@besttransformer.com
[3]Karadeniz Technical University, Department of Industrial Engineering
Kanuni Campus, Trabzon, Turkey
kbuyukozkan@ktu.edu.tr

Abstract **-** Stochastic operation times make job-shop scheduling harder at companies which work based on project type labor-intensive production in a dynamic environment. The operation times are not known before production and change based on the orders' technical specifications. In performing required operations with the aim of producing a final product, scheduling is required for different purposes such as minimizing makespan, maximizing resource utilization, etc. This is important as it enables companies to meet customer demands by due date and reduce the labor cost on the finalized product. In this study, an order scheduling algorithm is proposed for nearly optimizing average makespan for several waiting orders in a transformer company's core production workshop considering dynamical production environment. The proposed algorithm adopts the technical order specifications, computes the stochastic operation times making use of simulation, and schedule orders using one of the widely used meta-heuristics, namely genetic algorithm. The objective is to determine the entry sequence of the waiting orders to the core production workshop for minimizing their average makespan which directly influences the resource utilization, efficiency, and labor costs.

Keywords**:** job-shop scheduling, makespan, genetic algorithm, simulation, Arena

1. Introduction

Transformers are electro-mechanic devices used to transmit the electrical power to consumers from its source [1] and produced in a labor-intensive project type production system. Except a few similar transformers produced frequently; there are unlimited types of different transformer orders varying based on the technical specifications demanded by the customers for those to be produced for the first time. This variation causes considerable amount of differences in processing times, makespan and due date for each type of order which are unknown and must be predicted before starting the production [2-4]. For this purpose, an effective scheduling is required for minimizing the labor cost and due date.

Because of the project-type labor intensive work, the state of the shop floor is changed continuously. Therefore, the dynamic shop conditions must be considered for better estimations and effective schedules to minimize costs. Offline deterministic scheduling is not an effective tool for this type of production. In deterministic scheduling, processing times for each job are known before starting the production and there are no system disturbances, except few production changes. This type of systems may be widely computer-controlled. Stochastic scheduling is more suitable for scheduling in dynamic shop floors. In stochastic scheduling, the required operations are known but not the processing times. These two types of scheduling strategies may be performed offline or online. In online scheduling, worker assignments to the tasks and allocation of the machines to the jobs are random and vary based on the order properties, status of the machines, machining capabilities, etc. [5].

In the electro-mechanic industry, the manufacturing is performed by project type production. For this reason, the conventional scheduling approach that only considers the general job and shop characteristics (the number of number of jobs in the queue, etc.) are insufficient. In such a production system, each of the received transformer order be considered as a new product. The same transformers' designs with equal power and voltage numbers may be completely different because of their technical specifications demanded by the customer [3]. This makes considering technical specifications of the transformer design a vital factor in obtaining an effective schedule.

The production operations of a transformer consist of four main steps, (i) magnetic core production, (ii) winding production, (iii) active part assembly, and (iv) final assembly [2]. This study is focused on order scheduling for the magnetic core production unit. Genetic algorithm (GA) is used for scheduling orders under comprehensive design constraints. Arena simulation is used to calculate processing times and makespan of operations based on order sequences generated by GA. The simulation model includes mathematical models and probability distributions – which use technical design parameters of orders as input variables – to calculate processing times. Although various GA approaches have been implemented for solving job-shop scheduling problem in the literature [6-11], the technical specifications of orders are considered in none of them. The following section gives a brief description of the proposed GA based scheduling approach and the simulation technique integrated into it. The case study and obtained results are presented in Section 3, followed by some concluding remarks presented in the Section 4.

2. Proposed Method

2.1. Scheduling

GA is used in this research to find the best sequence of customer orders, which minimizes average makespan. Average makespan here corresponds to the completion of latest order divided by the total number of orders, customized based on customer demands. GA belongs to the family of evolutionary algorithms and is inspired by the natural selection principle. GAs have been originally developed by Holland [12] and been extensively used in combinatorial optimization problems (e.g., travelling salesman [13, 14], job-shop scheduling [15, 16], assembly line balancing [17, 18], lot-sizing [19, 20], etc.) where exact solution methods fail to investigate the optimal solution in a reasonable amount of time. Please refer to Goldberg [21] for a detailed discussion on the structure and components of GA.

GAs are based on a population of individuals (called chromosome), each of which represents a complete solution. Each chromosome is composed of several genes, and the algorithm iterates by evolving the population via genetic operators, i.e. selection, crossover, and mutation. The representation of a solution (encoding) plays vital importance in the performance of the algorithm. Therefore, an appropriate encoding/decoding mechanism should be determined before applying those operators. In permutation encoding, every chromosome is a string of numbers and each number shows only once. Each gene (or digit) usually corresponds to a real value, e.g. job number, task number and city number. The fitness value of each chromosome is computed and those having better fitness values (in accordance with the objective – minimization or maximization) are favored to be reproduced and transferred to the next generation. After applying genetic operators for several iterations, the solution which has the best fitness value is reported as the final solution.

2.2. Simulation

Simulation is "*a numerical technique for conducting experiments on a digital computer which involves logical and mathematical relationships that interact to describe the behavior of a system over time*" [22]. Nowadays, computational power and storage capacity of contemporary computers and software are better than ever, and because of these improvements, simulation is often used in various fields and industries. There are many competing languages to conduct a simulation work in a computer. Some of these languages are specially created for simulation, and use drag and drop modules like Arena and Promodel. Arena is one of the most popular simulation software, and exploits ActiveX Automation and Visual Basic for Applications (VBA) for integrating directly with other programs. In this paper, Arena 14 simulation software is used for the simulation [3].

2.3. Progress of the Proposed Algorithm

As Arena is fully compatible with Microsoft technologies, the proposed GA is coded in VBA environment which makes connection to Arena simulation software hassle-free. All GA specific components, including genetic operators and iterating the algorithm until the maximum number of iterations (*MaxNbIt)*, are implemented in VBA. Arena is used for computing the fitness value (FV) of each chromosome via simulating the complex system. The system here corresponds to the dynamic labor-intensive project-type core production system including various operations, queues and capacity constraints of which the details will be provided in the next section.

Figure 1 depicts the flowchart of the proposed GA-based optimization approach to minimize average makespan, namely, the total completion time of all customer orders. GA is used here for tuning the sequence of waiting customized orders for importing them to the system. The algorithm starts with randomly generating the initial population, which involves *PopSize* number of chromosomes. Each chromosome is made up of a permuted sequence of orders from 1 to the total number of orders, where each gene corresponds to an order number. Figure 2 presents a schematic view of a sample chromosome consisting of 14 orders. As seen from the figure, each order is shown only once on the chromosome. This is a sample complete solution for our problem. Orders are transferred to the production system in which they are shown on the chromosome, i.e. 4, 10, 8, 13, and so on.

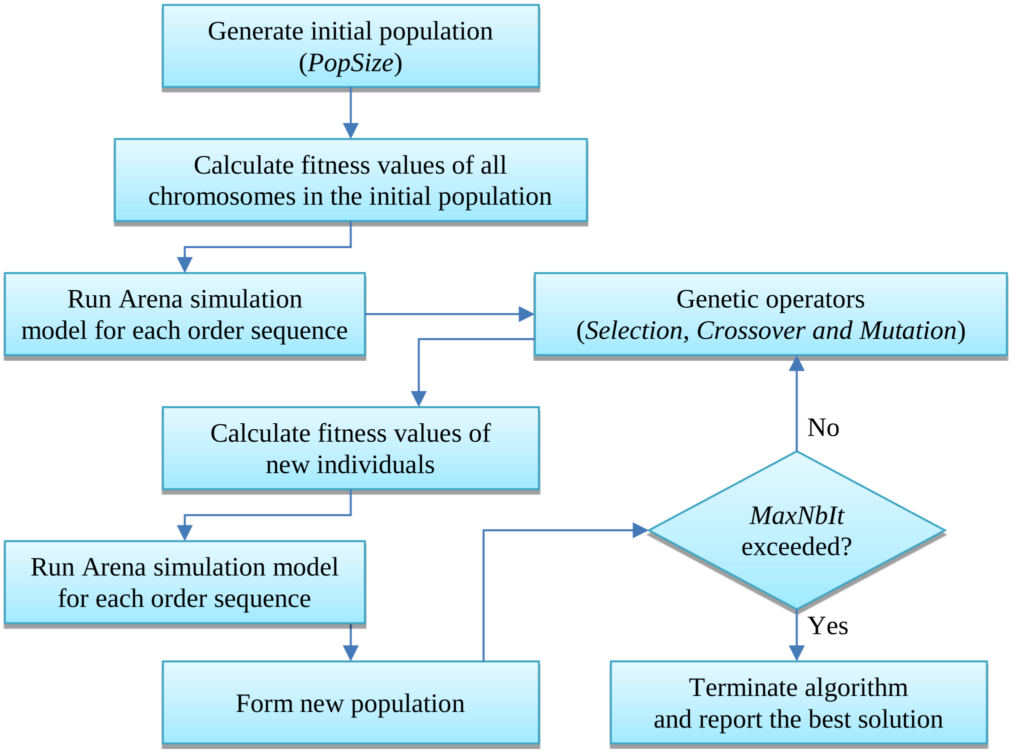
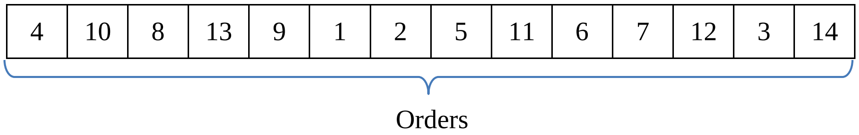

Generate initial population
(*PopSize*)
Calculate fitness values of all chromosomes in the initial population
Run Arena simulation model for each order sequence
Genetic operators
(*Selection, Crossover and Mutation*)
Calculate fitness values of new individuals
No
MaxNbIt exceeded?
Run Arena simulation model for each order sequence
Yes
Form new population
Terminate algorithm and report the best solution

Fig. 1: The flowchart of the proposed approach.

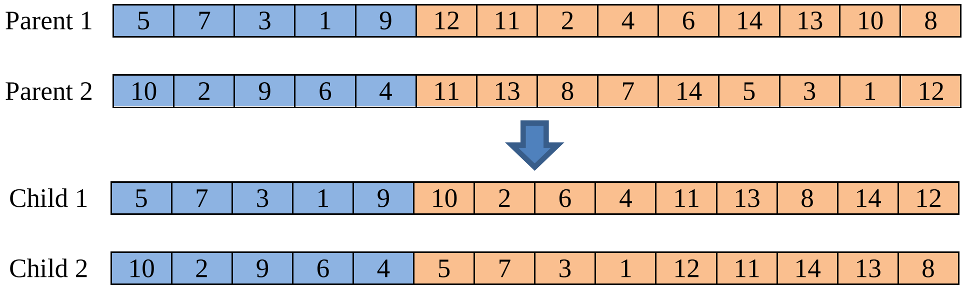
After generating the initial population, the fitness value of each chromosome is computed by running Arena for each of these chromosomes. For computing fitness value of a specific chromosome, the orders existing in the input file of Arena are sequenced based on their sequence on the chromosome and Arena is run. When Arena completes the simulation, FV (average makespan) is exported to an Excel file, read by the GA and integrated with the corresponding chromosome. Arena is run again every time the fitness value of a chromosome to be computed.



4	10	8	13	9	1	2	5	11	6	7	12	3	14

Fig. 2: The representation of a sample chromosome.

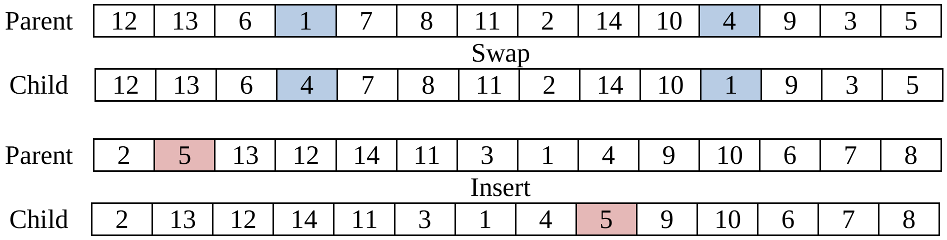
When the FVs of all chromosomes in the initial population are computed, roulette wheel selection (or fitness-proportional selection) is applied to select chromosomes for crossover and mutation. Thus, reproductive trials are allocated to chromosomes in accordance with their fitness values. The probability *P(i)* of selecting the *i*-th individual is $P(i) = \frac{FV(i)}{\sum_{k=1}^{PopSize} FV(k)}$. In this method, the selection probability of an individual is proportional to its *FV*. The number of chromosomes to apply crossover is determined by the multiplication of *PopSize* and crossover rate (*cr*). For example, if *PopSize = 20 and cr = 0.5*, four chromosomes are selected for crossover, in binary pairs. A random number, between 1 and *PopSize-1* is determined as the cutting point (*cp*) and one-point crossover is applied. Figure 3 shows the crossover operator (*cp=5*). As seen, the head of the first child is directly taken from Parent 1 and the tail is completed by adding missing genes based on their sequence in Parent 2. Similarly, Child 2 is generated from the head of Parent 2 and the sequence of missing genes in which they appear in Parent 1.



Parent 1	5	7	3	1	9	12	11	2	4	6	14	13	10	8
Parent 2	10	2	9	6	4	11	13	8	7	14	5	3	1	12
Child 1	5	7	3	1	9	10	2	6	4	11	13	8	14	12
Child 2	10	2	9	6	4	5	7	3	1	12	11	14	13	8

Fig. 3: The crossover operator.

Mutation operator is applied to selected chromosomes in two ways: swap and insert. The number of chromosomes to apply mutation is determined by the multiplication of *PopSize* and mutation rate (*mr*). For each selected chromosome, a random number (*rnd*) is determined between 0 and 1 and if *rnd < 0.5*, swap is applied; otherwise, insert is applied. This stochastic mechanism aims to increase the diversity in the population. Figure 4 shows the schematic representation of the mutation. As seen, swap is applied by swapping genes at two randomly determined points (*sp1*, and *sp2*). In this example, genes 1 and 4 are swapped (swapping points are determined as *sp1=4* and *sp2=11*). In the insert method, two random points, *ins1* and *ins2 (ins1< ins2)* are determined and the gene located at *ins1* is moved to the location *ins2*. In the example given in Figure 4, insert points are determined as *ins1=2* and *ins2=9*. After obtaining children, their *FVs* are computed and the once which have better *FVs* are included in the population replacing worse ones. This cycle continues until *MaxNbIt* is exceeded and the algorithm is terminated reporting the best solution investigated.



Swap

Parent	12	13	6	1	7	8	11	2	14	10	4	9	3	5
Child	12	13	6	4	7	8	11	2	14	10	1	9	3	5

Insert

Parent	2	5	13	12	14	11	3	1	4	9	10	6	7	8
Child	2	13	12	14	11	3	1	4	5	9	10	6	7	8

Fig. 4: The mutation operator.

3. Case Study

3.1. Production System of BEST Transformers Company

BEST Transformers is a transformer producer located in Balikesir, Turkey. This study is focused on order scheduling for the magnetic core production process of BEST transformers. The manufacturing line is designed as a labor-intensive project-type production, composed of sequential processes. A summarized production flowchart of the magnetic core production is given in Figure 5. In magnetic core production, cold-rolled grain-oriented silicon-steel is used. This steel has high conductivity and low power losses which is necessary for core production. Core material is sliced and cut into dimensions which are planned as per the needs of the specifications prepared by the engineering department. After the cutting operation, all the lamination sheets are stacked in manual or semi-automatic machines [2, 3].

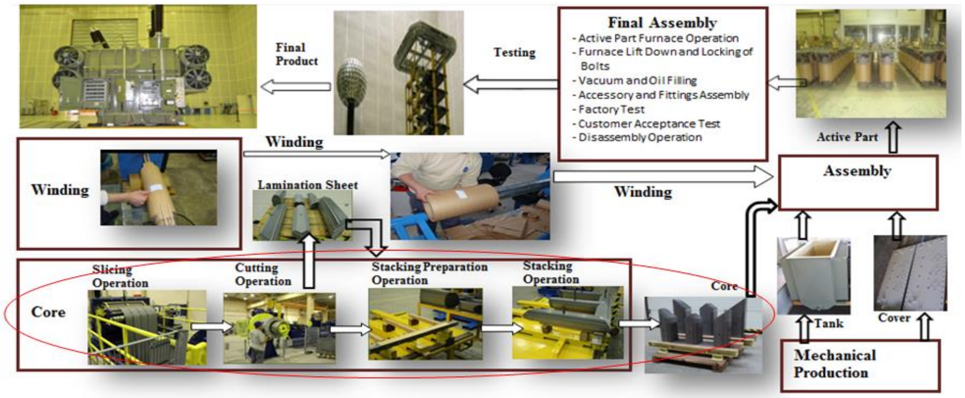


Fig. 5: Summarized production flow chart for magnetic core production.

In slicing operation, steel slice stocks are primarily checked and those not existing in the stock are sliced in slice cutting machine from the roll in width set design. In this study, it is assumed that the steel slice stocks are ready before starting the cutting operation and there is no tardiness resulted from this operation. In cutting operation, sliced rolls are cut in measurements determined in design phase. The processing time of cutting is affected from the weight of the core and the number of sheets - which are also determined in the design phase. Two cutting machines exist in the system. In the case that the number of transformers in an order is more than 1, cutting operation should be performed primarily in cutting machine 1 (MC1). Cutting process has capacity limitations for MC1 based on the maximum width of the sheet, as given in Table 1. For example, if the width of the sliced rolls is between 160 mm and 250 mm, then cutting machine can cut 5 orders at the same time. Similarly, for the width ranges from 250 mm to 310 mm, up to 4 transformer orders can be cut at the same time. MC2 is preferred for orders including one transformer as it is able to cut sheets belonging to one transformer up to 540 mm, at a time.

Table 1: Grouping constraint related to cutting machine 1 (MC1).

Maximum width of the sheet, x (mm)	$x \leq 160$	$160 < x \leq 250$	$250 < x \leq 310$	$310 < x \leq 440$
Maximum number of orders to be cut at once	7	5	4	3

Stacking is a labor-intensive operation. Sheets cut are manually stacked in stacking tools by workers. In this operation, the constraints are related to the power, core weight, precedence relationships, and labor. For example, orders which have power higher than 3000 kVA, are stacked in automatic tools primarily. However, automatic tools support transformers to a certain weight while this is not the case for manual tools. In addition, there are totally 10 workers in stacking operation; while there are no labor resource limitations in other processes. In stacking area, one worker is allocated to each stacking tool. In case that the core is heavier than 1000 kg, two workers are allocated to this operation. So, one stacking tool becomes unavailable due to the lack of operator (who is helping to his mate in another tool) for order allocation in such a case. If the weight of the transformer is between 0 and 2000 kg, it is operated at automatic stacking tools, DM1-DM4. Also, it is primarily allocated to automatic tools DM3-DM4, if the weight is between 2001 kg and 3000 kg. Manual tools are identical and there is no preference in between.

3.2. Numerical data and results

The data for a total of 14 orders have been gathered from the company's current waiting orders list, as listed in Table 2. These 14 orders correspond to the production of a total of 40 transformers. Two cutting machines exist in the system, MC1 and MC2. When there is more than one transformer in an order, the cutting operation constraints given in Section 3.1 are considered in determining the machine type to use and lot size to cut. Four automatic stacking tools (DM1, DM2, DM3 and DM4) and six manual stacking tools (FM1, FM2, FM3, FM4, FM5 and FM6) serve in the stacking area. Stacking constraints given in Section 3.1 are considered in determining the machine type to use for stacking. The aim here is to minimize the average of makespan of orders listed in Table 2 using the proposed GA-based approach.

Table 2: Technical specifications of orders.

Order	Number of transformer in order	Max width (mm)	Cutting machines	Power (kVA)	Total weight (kg)	Stacking machines (priority)	Stacking machines
1	1	300	MC2, MC1	4000	2.142	DM3,DM4	FM1,FM2,FM3, FM4,FM5,FM6
2	1	230	MC2, MC1	3000	1.696	DM1,DM2, DM3,DM4	FM1,FM2,FM3, FM4,FM5,FM6
3	1	300	MC2, MC1	4000	2.100	DM3,DM4	FM1,FM2,FM3, FM4,FM5,FM6
4	4	230	MC1	3000	1.765	DM1,DM2, DM3,DM4	FM1,FM2,FM3, FM4,FM5,FM6
5	10	200	MC1	2000	1.457	FM1,FM2,FM3, FM4,FM5,FM6	-
6	8	190	MC1	2000	1.286	FM1,FM2,FM3, FM4, FM5,FM6	-
7	4	280	MC1	3400	1.704	DM1,DM2, DM3,DM4	FM1,FM2,FM3, FM4,FM5,FM6
8	2	130	MC1	150	133	FM1,FM2,FM3, FM4,FM5, FM6	-
9	2	210	MC1	3000	1.621	FM1,FM2,FM3, FM4,FM5,FM6	-
10	1	180	MC1	1600	1.099	FM1,FM2,FM3, FM4,FM5,FM6	-
11	1	170	MC1	1250	929	FM1,FM2,FM3, FM4,FM5,FM6	-
12	2	210	MC1	3000	1.621	FM1,FM2,FM3, FM4,FM5,FM6	-
13	1	170	MC1	160	412	FM1,FM2,FM3, FM4,FM5,FM6	-
14	2	200	MC1	2500	1.420	FM1,FM2,FM3, FM4,FM5,FM6	-

To see the difference that the proposed methodology makes in the average makespan, the company's current order sequence, 1-2-3-4-5-6-7-8-9-10-11-12-13-14, is simulated using Arena when the replication number is set to 30. The average makespan is found about 225 hours. Later, the GA is run with parameters *PopSize = 25*, *MaxNbIt = 40*, *cr = 0.5*, and *mr = 0.1*. A total of 625 chromosomes have been generated during the algorithm run and the chromosomes of which the completion time exceeds the delivery time to the next process are eliminated from the population. Figure 6 shows the average makespan of chromosomes generated during the iterations. The individual which gives the best FV (≅187 hours) is 3-9-14-11-4-13-10-12-1-2-6-8-7-5. As seen from the figure, the average makespan is reduced from 225 to 187 hours, with a 16.8% reduction.

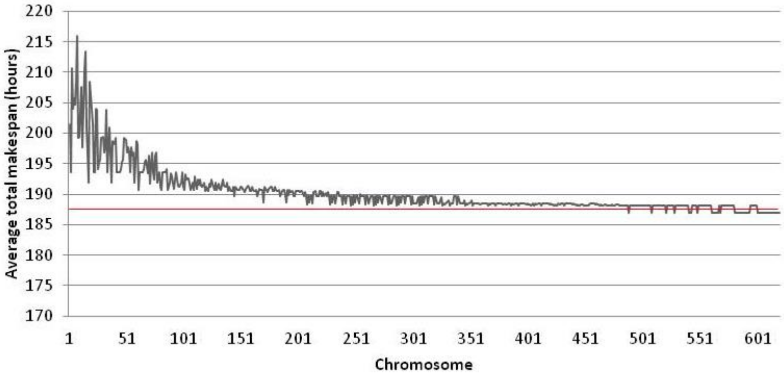


Fig. 6: The change in FVs of chromosomes generated.

4. Conclusions

This study focused on determining the best possible processing sequence of the orders to minimize the average makespan under the dynamic job-shop conditions. For this purpose, Arena simulation is used for modeling the dynamic conditions of the labor-intensive project type manufacturing environment of BEST Transformers and GA is used for determining the optimal sequence of customized orders waiting for processing. Mathematical models and probability distributions are used in Arena simulation for the calculation of stochastic times. This study makes significant contributions to knowledge. First, existing researches in the literature have not directly considered the technical design specifications of the orders in scheduling. Because of project-type manufacturing environment, the majority of the transformers are produced only once and the processing times for these orders are unknown. Previous studies on this subject considered general job and shop characteristics (such as the number of resources, labors, jobs in queue, etc.) and the processing times are thought to be known before starting the production. However, in this study, in addition to the job-shop characteristics of the production environment, the technical specifications (such as power, weight, number of slices, etc.) of the orders demanded by the customers are directly considered in scheduling orders. Also, GA and simulation are integrated to obtain a feasible as well as powerful solution in sequencing customer orders. The result of the case study made it clear that the average makespan is reduced thanks to the proposed methodology. Using the proposed methodology in similar project-type production systems may provide significant advantages.

Acknowledgements

This research is supported by Republic of Turkey Ministry of Science, Industry and Technology [grant number: SANTEZ-0682.STZ.2014]. The authors would gratefully like to thank BEST Transformer Research & Development Department for its valuable supports lead to reveal this paper.

References

[1] J. H. Harlow, *Electric Power Transformer Engineering*, 3rdEdn. USA, CRS Press, 2012.

[2] A. D. Karaoglan and O. Karademir, "Flow time and product cost estimation by using an artificial neural network (ANN): A case study for transformer orders," *Engineering Economist*, 2016, DOI: 10.1080/0013791X. 2016.1185808.

[3] A. D. Karaoglan, M. A. B. Ocaktan and A. Cicibas, "Flow Time and Due Date Estimation for Customer Orders According to Design Criteria: A Case Study of BEST Transformers Company," *4th Annual International Conference on Industrial, Systems and Design Engineering*, Athens, Greece, 2016, vol. IND 2016-1974.

[4] A. D. Karaoglan, "Residual based flow time estimation algorithm for labor intensive project type production systems," *Dokuz Eylul University Faculty of Engineering Journal of Science and Engineering*, vol. 18, no. 54, pp. 580-595, 2016.

[5] D. Krige, J. Bekker and C. Schutte, "Simulation-based online scheduling in a make-to-order job shop," *South African Journal of Industrial Engineering*, vol. 20, no. 2, pp. 205-217, 2009.

[6] R. Tavakkoli-Moghaddam and M. Daneshmand-Mehr, "A computer simulation model for job shop scheduling problems minimizing makespan," *Computers & Industrial Engineering*, vol. 48, no. 4, pp. 811-823, 2005.

[7] B. S. de Ugarte, R. Pellerin and A. Artiba, "An improved genetic algorithm approach for on-line optimisation problems," *Production Planning & Control*, vol. 22, no. 8, pp. 742-753, 2011.

[8] Z. Jia, X. Lu, J. Yang and D. Jia, "Research on job-shop scheduling problem based on genetic algorithm," *International Journal of Production Research*, vol. 49, no. 12, pp. 3585-3604, 2011.

[9] R. Qing-dao-er-ji and Y. Wang, "A new hybrid genetic algorithm for job shop scheduling problem," *Computers & Operations Research*, vol. 39, no. 10, pp. 2291-2299, 2012.

[10] X. W. Huang, X. Y. Zhao and X. L. Ma, "An improved genetic algorithm for job-shop scheduling problem with process sequence flexibility," *International Journal of Simulation Modelling*, vol. 13, no. 4, pp. 510-522, 2014.

[11] H. Zupan, N. Herakovic, M. Starbek and J. Kusar, "Hybrid algorithm based on priority rules for simulation of workshop production," *International Journal of Simulation Modelling*, vol. 15, no. 1, pp. 29-41, 2016.

[12] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Cambridge, MIT Press, 1975.

[13] S. Chatterjee, C. Carrera and L. A. Lynch, "Genetic algorithms and traveling salesman problems," *European Journal of Operational Research*, vol. 93, no. 3, pp. 490-510, 1996.

[14] J. Y. Potvin, “Genetic algorithms for the traveling salesman problem,” *Annals of Operations Research*, vol. 63, no. 3, pp. 337-70, 1996.

[15] J. E. Biegel and J. J. Davern, “Genetic algorithms and job shop scheduling,” *Computers & Industrial Engineering*, vol. 19, no. 1-4, pp. 81-91, 1990.

[16] J. F. Goncalves, J. J. de Magalhães Mendes and M. G. C. Resende, “A hybrid genetic algorithm for the job shop scheduling problem,” *European Journal of Operational Research*, vol. 167, no. 1, pp. 77-95, 2005.

[17] I. Kucukkoc and D. Z. Zhang, “Integrating ant colony and genetic algorithms in the balancing and scheduling of complex assembly lines,” *The International Journal of Advanced Manufacturing Technology*, vol. 82, no. 1, pp. 265–85, 2016.

[18] M. Chica, O. Cordon and S. Damas, “An advanced multiobjective genetic algorithm design for the time and space assembly line balancing problem,” *Computers & Industrial Engineering*, vol. 61, no. 1, pp. 103-117, 2011.

[19] J. Xie and J. Dong, “Heuristic genetic algorithms for general capacitated lot-sizing problems,” *Computers & Mathematics with Applications*, vol. 44, no. 1-2, pp. 263-276, 2002.

[20] D. Wang, O. Grunder and A.E.L. Moudni, “Using genetic algorithm for lot sizing and scheduling problem with arbitrary job volumes and distinct job due date considerations,” *International Journal of Systems Science*, vol. 45, no. 8, pp. 1694-707, 2013.

[21] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Boston, MA, USA: Addison-Wesley Longman Publishing Co. Inc., 1989.

[22] M. D. Rosetti, *Simulation Modeling and Arena*, 2nd Ed, New Jersey: John Wiley&Sons Inc., 2010.