And., Intl.,J., Appl Sci., Eng., Tech, Vol 04, No 01, March 2024
e-ISSN: 2797-0442

Hybrid Model of Fuzzy Logic and Genetic Algorithm for Product Assembly Sequence Optimization

Novi Deswinda[1], Rika Ampuh Hadiguna[1,*)], Nikorn Sirivongpaisal[2]
[1]Department of Industrial Engineering, Faculty of Engineering, Universitas Andalas, Indonesia
[2]Faculty of Engineering, Prince Sonkla of University, Hatyai Campus, Songkla, Thailand
[*]Corresponding author, e-mail: hadiguna@eng.unand.ac.id

Abstract—The sequence of product assembly affects the efficiency and effectiveness of production because it reduces cycle time and production costs and reduces production errors. The application of artificial intelligence is growing to optimize the problem of assembly sequences of components or products, including genetic algorithms and fuzzy logic. These two models can complement each other to produce the best assembly sequence. This research consists of several stages: model formulation, model analysis, solution, and model verification and validation. A hybrid fuzzy and genetic algorithms model can optimize product assembly sequences more effectively and efficiently. Fuzzy logic can help determine the variables that must be optimized, while genetic algorithms can help find the optimal solution by combining these variables. Experiments using hybrid fuzzy logic and genetic algorithms to minimize assembly cycle time have resulted in product part assembly sequences that accommodate all geometric constraints, including assistive devices.

Keywords: Assembly sequence, Hybrid, Genetic algorithm, Fuzzy logic, Cycle time.

This article *is licensed under the CC–BY-SA license.*

1. Introduction

Hybrid fuzzy and genetic optimization is an exciting topic because combining these two techniques can provide better and more effective solutions for solving complex problems that are difficult to solve using conventional methods [1]. Fuzzy logic deals with problems involving uncertainty and ambiguity in data. In contrast, genetic algorithms optimize complex parameters or variables and can be used in complicated problems [2]. Combining these two techniques can create a more accurate mathematical model and provide a more optimal solution in a shorter time [1]. The results of hybrid fuzzy and genetic optimization are also easier to understand and apply in the real world [1], [3]–[12]. This hybrid optimization technique can help make better and more effective decisions in various fields.

Determining the sequence of product assembly is very important because it affects the efficiency and effectiveness of production [13]. The proper assembly sequence can reduce time, production costs, and errors [14]. Determining the correct assembly sequence makes it easier for operators to understand the steps of the assembly process. This can increase production efficiency and affect the quality of the final product. Several studies have produced models for determining assembly sequences [12], [15]–[18]. The application of artificial intelligence is growing to optimize the problem of component or product assembly sequences, including genetic algorithms [14], particle swarm optimization [12], neural networks [19], simulated annealing and genetic algorithm [20], genetic algorithm simulated annealing and ant colony optimization [13], fuzzy logic and genetic algorithm [21], knowledge-based approach [22].

Received: October 07, 2023, Revised: December 11, 2023, Accepted: March 29, 2024
https://doi.org/10.25077/aijaset.v4i1.104

This study pays more attention to applying hybrid Fuzzy and Genetic Algorithms because these two models can complement each other to produce the best assembly sequence. Combining these two models is appropriate because both can optimize the assembly order effectively and efficiently [21]. Fuzzy logic is a technique in intelligent systems that can handle uncertainty in data by using linguistic variables. In product assembly, fuzzy logic can assist in determining work priorities and grouping similar tasks based on their importance and complexity. A genetic algorithm is an evolutionary algorithm used to find optimal solutions to optimization problems. In the context of product assembly, genetic algorithms can assist in determining the optimal assembly sequence based on some criteria, such as production time and costs.

The primary motivation for building this integration is completing the crossover and mutation probability in the genetic algorithm process. These two probability values act as genetic operators, significantly affecting the level of exploration and exploitation experienced by the population of solutions. It is crucial to be monitored to obtain optimal results globally. A high crossover probability will allow the software to explore more chromosomes in the population. However, suppose the crossover probability is too large. In that case, it will result in the development of a premature optimal solution, meaning that the development of the optimal solution occurs too early because the crossover results in the exact chromosome change, and the population will soon be filled with homogeneous chromosomes so that the next genetic operation does not provide a significant increase.

The probability of mutation significantly influences the birth of an initial chromosome that is different from the parent chromosome. If the mutation probability is too large, the number of genes exchanged will increase. It can damage the stability of the optimization chromosome, which does not add value to the increase in fitness. However, suppose the probability value gives a mutation probability value that is too small. In that case, too little gene exchange is evaluated, so many genes have the potential to provide increased fitness but do not get the opportunity to exchange genes.

This study's problem is integrating fuzzy logic and genetic algorithms to optimize the assembly sequence. This study aims to determine the carburetor assembly sequence using a hybrid fuzzy logic and genetic algorithm to minimize assembly cycle time. This study limits the development of a hybrid algorithm for a single product on a laboratory scale. The benefit of this study is enriching the repertoire of developing hybrid fuzzy logic and genetic algorithms that apply to the assembly process on the production floor. This hybrid framework can be part of a model-based management system in a decision support system in an assembly system. Experimentation focuses on carburetor products, which are very complex because they have many assembly steps. Complexity is needed to test the reliability of an algorithm for determining product assembly sequences.

2. Method

This research consists of several stages: model formulation, model analysis, solution, and model verification and validation. The model formulation is the determination of performance measures from determining the sequence of assembly to be produced, the workings of genetic algorithms, and fuzzy logic in finding optimal solutions. The objective function is to minimize assembly time by finding the assembly sequence. The application of the genetic algorithm in determining the assembly sequence is to create a population of the initial assembly sequence. Two alternative initial assembly sequences are obtained from the assembly sequences applied to the last two periods. The assembly sequence will become Sub Population I and Sub Population II for further processing to obtain the final assembly sequence with the shortest cycle time. The Genetic Algorithm was chosen because this algorithm works with a set population of solutions and looks for the best solution from that population, not just one solution. The application of fuzzy logic is the determination of the genetic algorithm control parameters. The final value of this process is the existence of a control parameter value that can be used in solving genetic algorithm problems.

The second stage is the analysis and solution of the model, which is to build the logic of the model so that it works according to the assumptions set. Programming logic refers to the working principle of genetic algorithms and fuzzy logic. Programming uses the Java programming language. This stage formulates the algorithm by referring to the basic principles of genetic algorithms and fuzzy logic. This study started the development of models of genetic algorithms. Fuzzy logic is part of a genetic algorithm that supports accelerating the discovery of optimal process conditions.

The next stage is program verification, determining whether the program meets the set assumptions. Validation aims to test whether the model can be applied to solve real problems. This research uses carburetor product assembly as a validation process. The validation process carried out a one-means test with the initial hypothesis (H0) stating that the assembly cycle time with the optimal sequence resulting from program execution is the same as the average cycle time of the carburetor assembly being experimented with. Meanwhile, the counter hypothesis (H1) states that the assembly cycle time with the optimal sequence resulting from program execution is shorter than the average cycle time of the carburetor assembly that was experimented with.

3. Result and Discussion

This section describes the formulation of the model up to experimentation. This section is very important because it contains the substance of this study. Each stage is described in detail, accompanied by the study's results.

3.1 Model Formulation

This study formulates the objective function as a measure of the performance of the assembly sequence as the assembly cycle time, which is formulated as follows:

$$\text{Minimize f} = \sum_{i=0}^{n} WB_i + \sum_{j=0}^{m} WBO_j - \sum_{k=0}^{l} WBA_k \quad (1)$$

The decision variable is the assembly sequence of product parts based on the assembly cycle time. The steps for solving the problem are built using the genetic algorithm method and the control of genetic operators by fuzzy logic. The assembly sequence optimization steps using the genetic algorithm are as follows:

Step 1. Determine the initial assembly sequence and calculate the assembly cycle time
Step 2. A crossover process using the Order Crossover (OX) method. Compute the cycle times of each parent and child and compare parent to child. Choose which is better between parent and child based on the cycle time.
Step 3. The mutation process uses the inversion method to the parent chromosome, which is randomly selected and depends on the mutation probability. Calculate the cycle time for the new child and compare it with the parent to select the better one for the new population.
Step 4. A looping process by sorting the chromosomes using the shortest cycle time to select the best chromosome in that generation. Repeat crossover and mutation until the age reaches the optimal generation. A generation is concluded to be optimal if no better results are found after several repetitions.

The next step is to formulate genetic operator control using fuzzy logic. Its purpose is to be input for crossover and mutation processes. The fuzzy logic built in this study uses the Tsukamoto method. There are five steps Step 1 is to formulate fuzzy rules. The level of exploitation and exploration that occurs in the population can be seen from the average variance of chromosomes and the average variance of gene values (VAC-Variance Average Chromosomes and AVA-Average Variance Alleles), which are determined by the following formula:

$$VAC = \frac{\sum_{i=1}^{N}(\bar{S}_i - \bar{S})^2}{N}, AVA = \frac{\sum_{j=1}^{n}\sum_{i=1}^{N}(S_{ij} - \bar{S}_j)^2}{n.N} \quad (2)$$

$$\bar{S}_i = \frac{\sum_{j=1}^{n} S_{ij}}{n}, \bar{S}_j = \frac{\sum_{i=1}^{N} S_{ij}}{N}, \bar{S} = \frac{\sum_{i=1}^{N}\sum_{j=1}^{n} S_{ij}}{n.N} \quad (3)$$

The process of determining pc and pm uses four fuzzy rules as follows:

[R1] IF VAC low, THEN pc must be raised
[R2] IF VAC is high, THEN pc must be lowered
[R3] IF AVA is low, THEN pm should be increased
[R4] IF AVA is high, THEN pm must be lowered

This study applies the triangular membership function, consisting of pc, pm, VAC, and AVA variables. All of these variables can be seen in Figures 1 to 4. The value 0 is the VACmin value, while VACmax is the VAC value in the initial generation, and the VACmax value can change for the next generation if a higher VAC value is found. It's the same with the VAC value. The value of 0 is the AVAmin value, AVAmax is the initial generation AVA value, and the largest AVA value is chosen for the next generation.

Step 2 is to apply function implication. The implication function is applied to the four fuzzy rules built before. Changes in the value on the left-hand side representation will impact the value on the right-hand side representation. Basic fuzzy logic rules are as follows:

[R1] IF VAC low, THEN pc must be raised
[R2] IF VAC is high, THEN pc must be lowered
[R3] IF AVA is low, THEN pm should be increased
[R4] IF AVA is high, THEN pm must be lowered

Step 3 is to find the crisp value of pc and pm. The output of the inference results from each rule is given strictly (crisp) based on the alpha predicate (α) so that the final result is obtained using a weighted average.

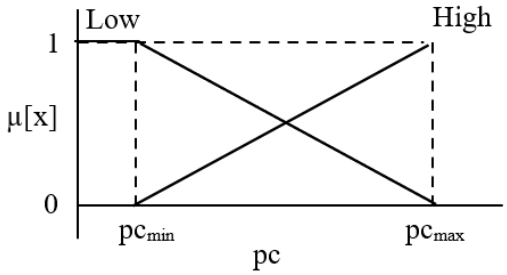


Figure 1. Membership Function Variable pc

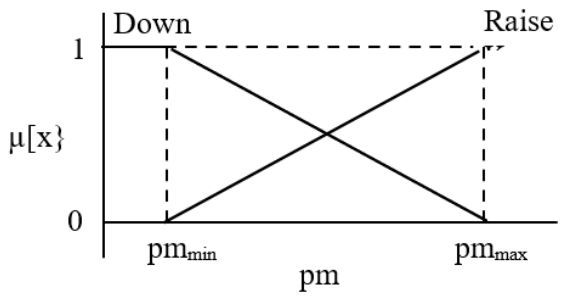


Figure 2. Membership Function Variable pm

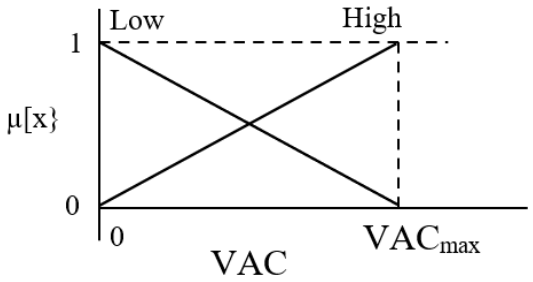


Figure 3. Membership Function of The VAC Variable

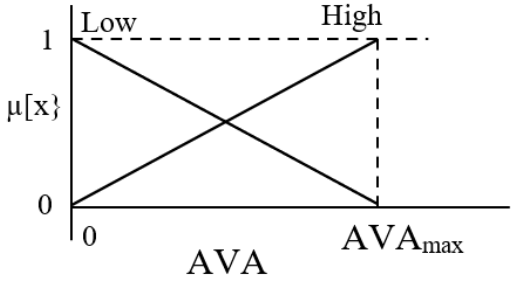


Figure 4. AVA Variable Membership Function

3.2 Model Analysis and Solutions

The first step in defining fuzzy logic and genetic algorithms in a programming language is data collection and processing. The data collected relates to the information needed to determine the carburetor assembly sequence. The following information collected is associated with the orientation of the part installation and the tools needed by the part during assembly. The list of product parts and assembly orientation can be seen in Table 1. The coordinate formulation of the product can be seen in Figure 5. Creating programming logic is the next step. Programming logic is based on algorithmic and fuzzy logic and information from the data collection results. Finally, the program listing was made using the Java programming language.

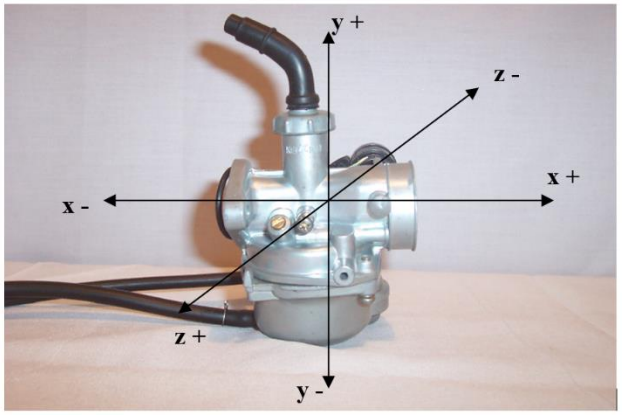


Figure 5. Carburetor Part Assembly Coordinate Axis

The current carburetor assembly sequence is the initial data that becomes input in developing fuzzy logic and genetic algorithms for determining the optimal carburetor assembly sequence. The assembly sequence in the genetic algorithm will act as the initial assembly sequence, forming the initial solution population.

3.3 Verification and Validation

This stage checks the calculation results obtained from the computer program output. The examination is carried out by manually checking the calculation results of the computer program. If the results obtained are not under manual calculations, then an iteration is carried out to the initial stage, namely program design, through reviewing the program. If the program is valid, then the program can be used to obtain solutions for determining product assembly sequences.

Table 1. Product Part and Assembly Orientation

Parts Code	Parts Name	Unit Quantity	Assembly Orientation
1	Bottom chamber	1	y -
2	Top chamber	1	y +
3	Valve component float	1	y -
4	Cap fuel strainer	1	y -
5	Cock set fuel	1	z -
6	Top set	1	y +
7	Cap choke dust	1	z -
8	Screw set D	1	z +
9	Screw set C	1	z +
10	Jet slow & jet main	1	y -
11	Holder needle jet	1	y -
12	Screw & ring	1	z -
13	Handle	1	z -
14	Float set	1	y -
15	Valve set	1	y +
16	Throttle	1	y +
17	Tube Airvent	1	x -
18	Gasket set D	1	z -
19	Cup filter	1	y -
20	Gasket set B1	1	y -
21	Gasket set A	1	x -
22	Gasket set C	1	y +
23	Screw set B	2	y -
24	Screw	2	z -
25	Tube Airflow	1	x -
26	Gasket set B2	1	z -

The data processing is related to the objective function of optimizing the carburetor assembly sequence. The objective function to be achieved is to minimize assembly cycle time. Because the data collected is in the form of an assembly work sequence, it is necessary to convert the assembly work into time. Conversion is done by measuring the time of assembly work. Time measurement is done time measurement indirectly. The indirect time measurement method used is the Maynard Operation Sequence Technique (MOST), given the involvement of manual tools in the assembly process. The procedure is done by describing the assembly work into the elements of the assembly work, describing the sequence of work, and finally determining the time based on the MOST table.

Step 1 is to apply the initial assembly sequence (current state) and create an initial population by generating a random chromosome sequence. The current assembly sequence is 1-26(-5-24)-18-22-17-20(-19-4)-2-21-10-7-3(-14)-11-13(-12) -23-25-9-8-15(-16-6). This study generated 30 populations. Each population's assembly cycle time is calculated based on each chromosome sequence. The calculation process begins with defining the assembly time for each part, the time for changing the part's orientation through the identification of the coordinates of the part assembly, and the need for auxiliary

tools, through the identification of the assembly of parts that require assistance. Step 2 is to calculate the value of VAC(0) and AVA(0) using equations (2) and (3).

Step 3 is a crossover using the Order Crossover method. For example, the selected chromosome pairs are: 6-14, then the first step is to identify the parent pairs that will participate in the crossover.

Parents 6
1 22 26 2 15 18 21 11 7 10 3 9 20 8 23 17 25 13
Parents 14
1 22 20 17 2 8 21 15 3 10 11 13 9 26 18 25 7 23

After that generate two random numbers, for example obtained 7 and 14

Child 1
X X X X X X X 11 7 10 3 9 20 8 X X X X
Child 2
X X X X X X X 15 3 10 11 13 9 26 X X X X

The next step is to generate the protochild by copying the substring to the appropriate position.

Child 1
23 17 25 13 1 22 26 2 15 18 21 11 7 10 3 9 20 8
Child 2
18 25 7 23 1 22 20 17 2 8 21 15 3 10 11 13 9 26

Remove values that are already on the protochild of the second parent.

23-17-25-1-22-2-18-21-7-20-8
18-25-23-1-22-17-2-21-15-13-26

Placement of the remaining values of the second protochild's unfilled parent from left to right.

Child 1
22 17 2 21 15 13 26 11 7 10 3 9 20 8 18 25 23 1
Child 2
22 2 18 21 7 20 8 15 3 10 11 13 9 26 23 17 25 1

A summary of the child chromosomes obtained from the crossover results of chromosome pairs 11-12 and 16-23 can be seen in Table 2. This new formation calculates the cycle time and compares parent to child. It turned out that no child resulted in a shorter cycle time than the parent's cycle time, so the population composition remained constant.

Table 2. Child Chromosome Crossover Results

Child	Chromosome
1	22 17 2 21 15 13 26 11 7 10 3 9 20 8 18 25 23 1
2	22 2 18 21 7 20 8 15 3 10 11 13 9 26 23 17 25 1
3	2 15 11 17 20 21 8 9 10 7 25 3 18 13 26 23 1 22
4	17 2 26 11 20 7 25 9 3 21 18 13 8 10 22 23 15 1

Step 4 is to mutate the crossover population by randomly selecting chromosomes. Suppose the selected chromosomes are chromosomes: 23 and 27. Then, the random number generation obtained for chromosome 23 is r1 = 6 and r2 = 16. Perform mutations using the inversion method to obtain the following:
26

Parents 1
1 26 2 10 20 22 21 11 8 13 17 18 15 3 9 7 23 25
Child 2
1 26 2 10 20 22 7 9 3 15 18 17 13 8 11 21 23 25

The results of mutations on chromosomes 23 and 27 are as follows:

Child 1
1 26 2 10 20 22 7 9 3 15 18 17 13 8 11 21 23 25
Child 2
1 20 18 2 9 15 23 25 22 17 11 7 21 26 10 3 13 8

Compare the parent cycle time with the child cycle time resulting from the mutation. The comparison results show that no child has a shorter cycle time, so no child replaces his parent's position in the population. This population will become the population of the next generation. Sort the population based on the best order priority, starting from the shortest cycle time. The sequencing results obtained one chromosome with the best cycle time for that generation: 1-18-22-2-10-9-8-11-7-26-13-21-15-25-3-20-23-17 with a cycle time of 178,560 seconds.

In replacing a parent with a child, both the results of crossover and mutation of existing physical constraints, the population resulting from this crossover and mutation will undergo the same operations as the previous population. Only in this case, what changes are pc and pm according to the results of fuzzy logic control.

Step 5 calculates pc and pm, which will be used for genetic processing in the next generation. The procedure for controlling fuzzy logic is to identify the variable values obtained by VAC (0) = 24,636, AVA (0) = 44,883, VAC (1) = 11,093 and AVA (1) = 27,434. Because VAC(1)<VAC(0) and AVA(1)<AVA(0), then VACmax=VAC(0) and AVAmax=AVA(0). The result is VAC max = 24.636, AVA max = 44.883, pc min = 0.10, pm min = 0.10, pc max = 1.00, and pm max = 1.00.

The next is to determine the value of VAC membership obtained μVAC LOW[a] = 0.550 and μVAC HIGH [a] = 0.450, For AVA obtained μAVA LOW[b] = 0.389 and μAVA HIGH [b] = 0.611, The process is continued by using the implication function. The next calculation calculates the crisp value using the weighted average method.

An assembly sequence determination model was built with assembly cycle time as the performance criterion. The assembly cycle time for the current state is 161,280 seconds. The next step is to test the results of applying the hybrid algorithm in this study. The experiment was conducted 30 times by running a computer program and applying a one-mean hypothesis test to the average cycle time. This average test is carried out so that the cycle time obtained from the model solution can be used as a reference for the standard time for assembling a carburetor. This experiment performed normality, adequacy, and normal time calculation tests. The experimental results obtained an assembly cycle time of 166,052 seconds. Based on Ho is $\mu = 161.28$ seconds and Hi is $\mu > 161.28$ seconds with $\alpha = 0.05$, it is concluded that accepting Ho that the average assembly cycle time is the same as the current conditions.

Based on the results of the hypothesis testing above, it can be seen that the minimum assembly cycle time obtained by the genetic algorithm and fuzzy logic approach can be used as a reference for a carburetor assembly line. In other words, if a line's carburetor assembly cycle time is above 161.28 seconds, this time can likely be minimized. Based on the verification and validation processes that have been carried out, it can be concluded that the program built is valid. Therefore, the program can find solutions according to the objective function.

3.4 Assembly Sequence Optimization

This study has successfully formulated a hybrid fuzzy algorithm and a genetic algorithm. This model should find a better assembly sequence. The initial step in running the program is to determine the input parameters of the program in the form of population size, number of generations, and number of replications. In this study, the population size used was 30, and the number of generations selected was 1000. The experiment's results using several generations of 1000 with ten replications resulted in an assembly sequence with a minimum cycle time of 161.28 seconds. The best order based on part code is 1-20-19-4-2-22-15-16-6-26-5-24-7-18-13-12-8-9-10-11-3-14- 23-17-25-21. The current state sequence is 1-26-5-24-18-22-17-20-19-4-2-21-10-7-3-14-11-13-12-23-25-9-8 -15-16-6.

When compared with the assembly sequence in the current conditions, the recommendation from the hybrid fuzzy logic and genetic algorithm model has several advantages. Sequence recommendations from the model result in a smaller number of reorientations and a smaller number of tool changes. This advantage indicates that the assembly sequence makes distributing parts to workstations easy. Parts requiring the same tools have been positioned in adjacent facilities. The same conditions can be enjoyed by identical parts even though the different assembly functions will be nearby. The benefit obtained is to reduce the occurrence of the spread of the same part or tool to several different workstations.

4 Discussion

4.1 Methodological Innovation

Combining fuzzy logic and genetic algorithms is an innovation in research methodology. This approach contributes to the research in designing solutions to optimization problems, especially in the context of product assembly sequences. Fuzzy logic allows the representation of variables that are not firm or definite. In manufacturing assembly, many variables may not have exact values. Integration with fuzzy logic enables the model to consider these uncertainties, resulting in more robust and applicable solutions in the real world. Fuzzy logic can adapt to environmental changes or dynamic production conditions. In assembly processes, where multiple variables or parameters can change rapidly, fuzzy logic's ability to handle uncertainty and environmental changes provides significant adaptation advantages. Fuzzy logic can help determine the most relevant variables to optimize. In assembly manufacturing, where many factors influence efficiency, fuzzy logic can help identify critical variables that need to be optimized.

Genetic algorithms can explore the solution space efficiently with their explorative and exploitative properties. By integrating genetic algorithms, this research can explore various possible solutions by exploiting the exploratory advantages of genetic algorithms. Genetic algorithms are known for their ability to reach optimal global solutions. By integrating fuzzy logic, which helps detail and identify the most promising solution areas, genetic algorithms can focus more on finding more efficient global solutions. Integrating fuzzy logic and genetic algorithms can produce efficient solutions (reducing assembly cycle time) and effectively overcome various production constraints, such as geometric constraints and auxiliary devices. Combining both advantages, integrating fuzzy logic and genetic algorithms creates a holistic and practical approach to addressing complexity and uncertainty in manufacturing assembly problems.

4.2 Practical Industrial Implications

This research focuses on product assembly sequence optimization, a critical aspect of the manufacturing process. Its contribution is to provide new insights and solutions to increase production efficiency and effectiveness by reducing cycle times, production costs, and assembly errors. This research offers solutions relevant to the needs of the manufacturing industry, especially in the context of product assembly. By optimizing assembly sequences, production efficiency can be improved, reducing costs and improving product quality. Reductions in cycle time, production costs, and assembly errors can be achieved through various interrelated and mutually supporting factors. Several key factors

contributing to production efficiency and effectiveness include intelligently arranging the assembly task sequence, which can reduce cycle time significantly. Identifying tasks that can be performed simultaneously or in parallel and prioritizing critical steps can speed up the assembly process.

The integration of genetic algorithms and fuzzy logic as artificial intelligence methods shows the practical application of these technologies in real-world contexts. This kind of research helps drive the application of artificial intelligence in manufacturing. Manufacturing environments are often full of variability and uncertainty. Fuzzy logic allows the model to handle ill-defined and changeable variables, while genetic algorithms provide adaptability to changing conditions and dynamic production requirements. This research offers practical applications of artificial intelligence in manufacturing environments, resulting in solutions that can be implemented directly at the production scale. This encourages the application of artificial intelligence technology in the manufacturing industry to increase efficiency and effectiveness. Manufacturing companies can increase competitiveness by optimizing production processes, creating efficient assembly sequences, and reducing errors. The application of artificial intelligence can be a critical factor in responding to demands for more innovative and more adaptive production in the industrial era 4.0.

5 Conclusion

Fuzzy hybrid models and genetic algorithms can optimize product assembly sequences more effectively and efficiently. Fuzzy logic can help determine the variables that must be optimized, while genetic algorithms can help find the optimal solution by combining these variables. In addition, hybrid fuzzy-genetic algorithms can also deal with uncertainty in data and explore the solution space more effectively. This can help achieve a more optimal solution under production needs. Thus, hybrid fuzzy and genetic algorithms are very appropriate for optimizing product assembly sequences because they can overcome the complexity of problems and find optimal solutions effectively and efficiently. Experiments using hybrid fuzzy logic and genetic algorithms to minimize assembly cycle time have resulted in product part assembly sequences that accommodate all geometric constraints, including assistive devices.

Acknowledgment

The authors appreciate the support of the Andalas University Production Systems Laboratory for the experimentation of assembling and using various equipment.

References

[1] C. T. Utari, M. Zarlis, and Sutarman, “Optimization of production scheduling and actual stock using fuzzy genetic algorithm,” in *Journal of Physics: Conference Series*, Institute of Physics Publishing, Jul. 2019. doi: 10.1088/1742-6596/1235/1/012035.

[2] S. Marrouchi and S. Ben Saber, “A comparative study of fuzzy logic, genetic algorithm, and gradient-genetic algorithm optimization methods for solving the unit commitment problem,” *Math Probl Eng*, vol. 2014, 2014, doi: 10.1155/2014/708275.

[3] K. Biswas, P. Shivakumara, U. Pal, T. Chakraborti, T. Lu, and M. N. Bin Ayub, “Fuzzy and genetic algorithm based approach for classification of personality traits oriented social media images,” *Knowl Based Syst*, vol. 241, Apr. 2022, doi: 10.1016/j.knosys.2021.108024.

[4] G. Human, G. van Schoor, and K. R. Uren, “Genetic fuzzy rule extraction for optimised sizing and control of hybrid renewable energy hydrogen systems,” *Int J Hydrogen Energy*, vol. 46, no. 5, pp. 3576–3594, Jan. 2021, doi: 10.1016/j.ijhydene.2020.10.238.

[5] M. Jalali Varnamkhasti and L. S. Lee, “A fuzzy genetic algorithm based on binary encoding for solving multidimensional knapsack problems,” *J Appl Math*, vol. 2012, 2012, doi: 10.1155/2012/703601.

[6] H. C. W. Lau, D. Nakandala, and L. Zhao, "Development of a hybrid fuzzy genetic algorithm model for solving transportation scheduling problem," *Journal of Information Systems and Technology Management*, vol. 12, no. 3, Jan. 2016, doi: 10.4301/s1807-17752015000300001.
[7] A. Mahrooghi and E. Lakzian, "Optimization of Wells turbine performance using a hybrid artificial neural fuzzy inference system (ANFIS) - Genetic algorithm (GA)," *Ocean Engineering*, vol. 226, Apr. 2021, doi: 10.1016/j.oceaneng.2021.108861.
[8] R. Muruganandham, J. Dinesh, A. Mohammed, R. Basha, and B. Devamaindhan, "Genetic algorithm and fuzzy logic based decision making technique for shop floor control in operations management," *International Journal of Recent Technology and Engineering*, pp. 2277–3878, 2019.
[9] B. Sena, B. Dewantara, and J. Miura, "Optimizing fuzzy rule base for illumination compensation in face recognition using genetic algorithms," *EMITTER International Journal of Engineering Technology*, vol. 2, no. 2, 2014.
[10] Y. Shi, T. Boudouh, and O. Grunder, "A hybrid genetic algorithm for a home health care routing problem with time window and fuzzy demand," *Expert Syst Appl*, vol. 72, pp. 160–176, Apr. 2017, doi: 10.1016/j.eswa.2016.12.013.
[11] A. Tabadkani, A. Nikkhah Dehnavi, F. Mostafavi, and H. G. Naeini, "Targeting modular adaptive façade personalization in a shared office space using fuzzy logic and genetic optimization," *Journal of Building Engineering*, vol. 69, p. 106118, Jun. 2023, doi: 10.1016/j.jobe.2023.106118.
[12] Y. Wang and J. H. Liu, "Chaotic particle swarm optimization for assembly sequence planning," *Robot Comput Integr Manuf*, vol. 26, no. 2, pp. 212–222, Apr. 2010, doi: 10.1016/j.rcim.2009.05.003.
[13] H. Shan, S. Zhou, and Z. Sun, "Research on assembly sequence planning based on genetic simulated annealing algorithm and ant colony optimization algorithm," *Assembly Automation*, vol. 29, no. 3, pp. 249–256, Jul. 2009, doi: 10.1108/01445150910972921.
[14] Y.-J. Tseng, H.-T. Kao, and F.-Y. Huang, "Integrated assembly and disassembly sequence planning using a GA approach," *Int J Prod Res*, vol. 48, no. 20, pp. 5991–6013, Oct. 2010, doi: 10.1080/00207540903229173.
[15] Alfadhlani, T. M. A. A. Samadhi, A. Ma'Ruf, and I. S. Toha, "A Framework for the development of automatic DFA method to minimize the number of components and assembly reorientations," in *IOP Conference Series: Materials Science and Engineering*, Institute of Physics Publishing, Mar. 2018. doi: 10.1088/1757-899X/319/1/012083.
[16] R. R. Iwankowicz, "An efficient evolutionary method of assembly sequence planning for shipbuilding industry," *Assembly Automation*, vol. 36, no. 1, pp. 60–71, Feb. 2016, doi: 10.1108/AA-02-2015-013.
[17] A. E. Kurnosenko and D. I. Arabov, "Optimization of electronic components mounting sequence for 3D MID assembly process," *KnE Engineering*, vol. 3, no. 6, p. 311, Oct. 2018, doi: 10.18502/keg.v3i6.3009.
[18] M. Taraska, R. Iwańkowicz, T. Urbański, and T. Graczyk, "Review of assembly sequence planning methods in terms of their applicability in shipbuilding processes," *Polish Maritime Research*, vol. 25, no. s1, pp. 124–133, May 2018, doi: 10.2478/pomr-2018-0033.
[19] M. Suszyński and K. Peta, "Assembly sequence planning using artificial neural networks for mechanical parts based on selected criteria," *Applied Sciences (Switzerland)*, vol. 11, no. 21, Nov. 2021, doi: 10.3390/app112110414.
[20] C. Liu, F. Zhang, H. Zhang, Z. Shi, and H. Zhu, "Optimization of assembly sequence of building components based on simulated annealing genetic algorithm," *Alexandria Engineering Journal*, vol. 62, pp. 257–268, Jan. 2023, doi: 10.1016/j.aej.2022.07.025.
[21] L. M. Galantucci, G. Percoco, and R. Spina, "Assembly and disassembly planning by using Fuzzy Logic & Genetic algorithms," *Int J Adv Robot Syst*, vol. 1, no. 1, pp. 67–74, Jun. 2004, doi: 10.5772/5622.
[22] M. Wu, Y. Zhao, and C. Wang, "Knowledge-based approach to assembly sequence planning for wind-driven generator," *Math Probl Eng*, vol. 2013, pp. 1–7, 2013, doi: 10.1155/2013/908316.