Jurnal Teknik Informatika (JUTIF)
Vol. 4, No. 5, October 2023, pp. 1151-1156

DOI: https://doi.org/10.52436/1.jutif.2023.4.5.1002
p-ISSN: 2723-3863
e-ISSN: 2723-3871

ANALYSIS OF IMPLEMENTATION OF PARTICLE SWARM OPTIMIZATION (PSO) METHOD ON LECTURERS ASSIGNMENTS TO STUDENTS

Dhiesky Chaerullah[1], Iqbal Chalid[2], Achmad Solichin[3]

[1,2,3]Faculty of Information Technology Budiluhur University, Indonesia
Email: [1]2111602195@student.budiluhur.ac.id, [2]2111602252@student.budiluhur.ac.id,
[3]achmad.solichin@budiluhur.ac.id

(Article received: April 15, 2023; Revision: May 30, 2023; published: October 15, 2023)

Abstract

When preparing a program for a conference, it is very important to divide teaching and learning tasks according to the areas in which you are involved for teaching and learning to be effective. At the University, the assignment process is still done manually which is very time consuming. Therefore, an appropriate optimization method is needed to handle this. This problem can be solved using a population-based heuristic approach, Particle Swarm Optimization (PSO) has been applied to various fields such as scheduling and assignment. The data used in this research is lecturer assignment data in the form of prioritizing lecturer interest in teaching certain subjects. Based on the calculation results, a test was carried out to determine the effect of the test parameters on the fitness value obtained. From the results of the PSO parameter test, the best number of particles is 100, the best number of repetitions is 100, and the speed combination parameters c1 and c2 are 1.5 and 1.5 with the appropriate value of 94878. The system results, the solution obtained gives good results, i.e. always within tolerance limits, the error scores obtained by placing teachers on subjects that suit their preferences are lower.

Keywords: *Particle Swarm Optimization, supporting lecturer, teaching interest.*

1. INTRODUCTION

In teaching and learning activities, one of the first things that must be considered is planning. The preparation of this schedule certainly cannot be done haphazardly, because the success of effective teaching and learning activities is determined by the optimal scheduling process [1], [2]. In preparing the schedule, it is very important to allocate teaching assignments to the workforce in accordance with the fields they are involved in and what they want, because jobs that are in accordance with the abilities of the workforce will require a lower requested cost than those that are not suitable [3].

Often, the assignment/assignment process is time-consuming due to complexity, given the large number of staff, courses and teaching spaces. One educational institution that has problems with this audience is the Faculty of Computer Science, an institution in Jakarta. At the beginning of the semester, undergraduates will distribute questionnaires to each FILKOM instructor or instructor to fill out in order to find out their interest in teaching the courses they will take during the semester. Data from the questionnaire filled out by the teacher will be scientifically processed to allocate teaching assignments for the course [4].

The division of teaching tasks leads to the selection of major and minor courses, as well as the priority order of the courses to be taught [5]. The division of teaching tasks is of course a must. Pay attention to the rules and limitations of each parameter. Such as the minimum and maximum number of credits and courses that can be taken. So far, scholars use Microsoft Excel software for preparation, so that if there is a discrepancy, corrections are also made manually [6], [7]. In addition to these aspects, there are ways to get the most out of the task assignment process. One way that can be used is to use the optimization process using Particle Swarm Optimization (PSO) [8]–[10].

In previous research conducted, the PSO method which is used as the basis for the automatic lesson plan model can be used as a schedule development tool to create a schedule that meets all the required criteria and can be adjusted to the teacher's preferences by the time of lesson. This can be seen from the results of fulfilling the hard constraints and minimizing the soft constraints that are owned by the position of each course with the solution set at the global optimum [11]. Meanwhile, the Genetic Algorithm Simulation Annealing (GA-SA) hybrid algorithm can be used to solve the problem of equalization of tasks, but the resulting cost is still relatively high due to the large amount of data used [12], [13].

PSO is a population-based heuristic method developed by Kennedy and Eberhart in 1995 that relies on the collective movement of organisms such as flocks of birds and schools of fish to simulate their foraging behavior[14], [15] One advantage of PSO over other optimization methods is that it achieves faster convergence. Another advantage is that PSO is easy to implement and has several functions and operational parameters to define [16]–[18]. In this PSO implementation, solution candidates represent course positions on the schedule board where each solution candidate has a cost value [19]. The population of elements is generated at the start of the first iteration, then these elements increase their position to the best position with each iteration, so that the optimal position of a course will be reached in the time table. This process is carried out for each course to get a complete schedule. With the PSO method, it is hoped that a short scheduling process with satisfactory accuracy and minimal error values will be obtained [20], [21].

Based on the description above, the authors propose a study on Optimizing the Distribution of Teaching Tasks for Teachers of the Faculty of Computer Information Technology (FILKOM), a campus in Jakarta. With the particle swarm optimization method. With this research it is hoped that it can solve the problem of dividing teaching assignments for FILKOM teachers, especially campuses in the city of Jakarta.

2. METHODS

This research methodology will explain the next steps in creating an optimal system for supervisor assignment using the Particle Swam Optimization method. The research methodology used in this study consisted of several steps, including literature review, data collection, system analysis and design, system implementation, system testing, and product development. The steps of this research are illustrated in Figure 1.

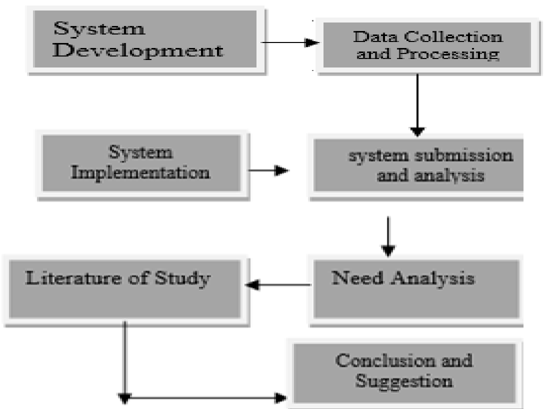

Figure 1. illustration of research steps

First: Literature studies from various sources are needed to realize the optimization of the assignment of supporting lecturers with the PSO method. Information and literature related to this research were obtained from journals, internet sites, supervisors, and fellow students

Second: Needs analysis in the development of the lecturer assignment optimization system using the Particle Swarm Optimization method is used to analyze what needs are needed in the development of this system so that it can achieve the desired goals. The requirements obtained at this stage are divided into two, namely Functional Requirements, and Data Requirements

Third: Data Collection and Processing From the analysis of predetermined needs, data is collected from informants, namely academics who are responsible for dividing teaching assignments for FILKOM lecturers each semester. However, due to the insufficient amount of data obtained, dummy data was used to determine the priority of lecturer courses based on the original data format. The dummy data used can be used as a benchmark in making the system, so that when the original data is applied in the future, the system can adapt more easily. The data used will be stored in the database to make it easier for the system to fetch data and store system output later.

Fourth: System Design System design is done by breaking down the system process from start to finish to simplify the implementation and testing process. From the solving process obtained the design of work steps from system as a whole which is divided into three, namely the input entered, the system process and the resulting output. The results of this system design are arranged into a model diagram that explains the work steps of the system in a structured manner.

Fifth: System Implementation. Interface implementation uses the Java programming language with NetBeans IDE 7.2 software. The research data used will be stored in the MySQL dataset using the XAMPP localhost.

Sixth: System Testing and Analysis. Tests were carried out to determine the relationship between fitness values and the parameters tested. A good parameter will be obtained if the fitness value obtained is high. The tests to be carried out are testing the number of particles, testing the number of iterations, and testing the speed parameters c1 and c2.

Seventh: Conclusions and Suggestions. Based on the results of the research steps carried out, conclusions and recommendations. The conclusion will be the answer to the problem formulation that has been done in the first part. While the recommendations are the author's hope for the purpose of correcting errors that have occurred and considering future research and development.

3. RESULTS AND DISCUSSION

3.1. Problem Formulation

The assignment problem is a problem regarding the arrangement of objects to carry out tasks, with the aim of minimizing costs, time, distance, and so on or maximizing profits. Meanwhile, an appropriate description of the assignment model is the best person to do a job. It is seen that workers who have abilities that match their jobs will require less cost than those who do not fit, so the purpose of this assignment model is to determine the minimum cost of assigning workers to a job. Assignment problems in certain circumstances will experience a count of assignments, where the number of m ≠ n. However, this problem can be solved by adding a dummy worker or dummy job. The division of course teaching tasks is the process of compiling a number of components consisting of lecturers, courses, and number of rooms with the aim of not causing a discontinuity that disrupts the teaching and learning process and taking into account certain limitations.

The problem faced was that the division of teaching tasks for teachers in the direction of the lecture was opened by the academic community by reviewing data from a questionnaire filled in by each lecturer. In the questionnaire data there are several parameters such as the priority order of the subjects to be studied, from priority 1 to priority 5, information about priority 1 is the highest priority. The following parameters are major and minor which determine which subjects the teacher should prioritize. Superior

These major and minor courses are similar to course prioritization, where the priority of courses from the major category takes precedence. Each teacher can fill out a questionnaire according to the course he/she wants to teach taking into account existing limitations such as each teacher can only take the course he/she wants to teach provided that the minimum number of credits is 12 and a maximum of 25 and the number of courses is a minimum of 1 and a maximum of 4. Based on the data provided by the teachers, the academic will determine the courses that must be taken by each teacher. However, currently the process of dividing teaching assignments is still carried out using Microsoft Excel, so if there are irregularities it takes a long time to correct them because the workflow is still manual.

3.2. Troubleshooting cycles using Particle Swarm Optimization (PSO)

The data used in this study are dummy data i the form of lecturer teachingmapping even semester 2015/2016 S1 study program Informatics/Computer Science at FILKOM University Jakarta city. The following is an example of a mapping data snippet lecturers in Table 1.

Table 1. Lecturer Data Mapping Table

Lecture Name	Code	Priority Teach	Subject	
			Mayor	Minor
Lecture A	L1	1, 6, 5	1, 6	5
Lecture B	L2	1, 7, 6, 4	1, 7, 6	4
Lecture C	L3	7, 1, 5, 6	7, 1	5, 6
Lecture D	L4	2, 8, 1	2, 8	1
Lecture F	L6	1,9,6,3		3
Lecture G	L7	2,4,8,2	3,4	8,2
Lecture E	L 5	5,9,7	5,9/1,9,6	7

Table 2. Subject Table

No.	Subject	SKS	Class
1	System Analysis and Design	3	5
2	Computer network	4	11
3	Distributed Systems	3	5
4	Distributed Database	3	8
5	Desain dan Analisis Algoritma	3	6
6	Mixed Reality	3	4
7	Web Programming	4	9
8	Pattern recognition	3	5
9	Expert system	3	7

The problem solving cycle learned by PSO starts at:

1) Particle Initialization

Particle initialization is used to determine the initial position of the particle. Initialization of the beads is done by randomly generating an integer matrix from 1 to 20 (Mansur et al 2019) which depends on the number of teachers used, namely 20. The length of the particle chromosomes here is the same, according to the number of open layers, while the number of particles can be adjusted based on the input provided.

2) Generate a Cost Matrix

The cost matrix is used to determine one of the costs to find the appropriate value, namely interest expense. The interest fee is a violation of the particle chromosome for the data of the faculty of interest. In Table 1, there are priorities for teaching faculties along with majors and minors. Major courses are courses that take precedence over minor courses. From there, the weighted values for these priorities are set out in Table 3.

Table 3. Weighting Table

Priority Teach	Types of Courses	
	Mayor	Minor
PT-1	1	2
PT – 2	3	4
PT– 3	5	6
PT – 4	7	8
PT – 5	9	10
Non Priority	100	

From the weighting value that has been determined, a cost matrix is made. For example, for Lecturer 1, the cost of the System Analysis and Design (APS) course is 1, because the APS course is the first priority subject that he wants to teach as well as major courses. Then the cost for the Mixed Reality (MIXER) course is 3, because MIXER is the second priority subject and includes major types of courses, and so on. Whereas for courses that are not taken by the lecturer, they will be given a value of 100 to provide a fairly large cost value if the chromosome value contained in the particle gives a value that does not match the existing data. The cost matrix pieces can be seen in Table 4.

Table 4. Cost Matrix Table

Code Subject	Lecture						
	1	2	3	4 5	5	6	7
APS_1	1	1	3	5	1 00	1	0
APS_2	1	1	3	5	100	1	10
APS_3	1	1	4	5	100	1	100
DAA_1	6	100	6	100	1	100	100
DAA_2	6	100	6	100	1	100	100
DAA_3	6	100	6	100	1	100	100
DAA_4	6	100	6	100	1	100	100
DAA_5	6	100	6	100	1	100	100
MIXER_1	3	5	8	100	100	5	100
MIXER_2	3	5	8	100	100	5	100
MIXER_3	3	5	8	100	100	5	100

3.3. Calculating Fitness Value

Each particle has a fitness value which is obtained by calculating the total of the three costs, namely the interest cost, the SKS cost, and the class cost. The cost of interest is obtained from the cost matrix process. The cost of credits is obtained by looking at the total credits taken by the lecturer. If the total credits taken are less than 12, then the cost is to subtract 12 from the total credits. But if the total cost is more than 25, then the cost is subtracting the total credits by 25. As for the class cost, it is obtained by looking at the total number of courses taken by the lecturer. If the total courses taken are less than 1, then the cost is 1. If the total courses taken are more than 4, then the cost is obtained by subtracting the total courses by 4. The constant used here is worth 100000, because this number is easily reduced by the total cost value obtained from each particle which ranges up to tens of thousands. In addition, the result of the reduction of the constant can be used as a fitness value, which generally views a higher value as a better value. 4. Determining pbest and gbest particles Pbest particles are the best particles, in other words they have the greatest fitness value compared to the surrounding particles. Determining the pbest particle is done by comparing the fitness value of each particle in an iteration. The gbest particle is the best particle of the whole iteration.

Determining the gbest particles is done by comparing the pbest particles so that the particle with the highest fitness value is obtained. 5. Calculating Velocity Value Velocity is a function of PSO to determine the direction of the new particle position. Each particle chromosome is searched for its velocity value using the formula in Equation 4. The value of w used here is 0.75 based on experiments in Eberhart & Shi (1998), while the values of c1 and c2 can be determined through the given input. 6. Calculating the New Position Value Calculating the new position value or updating the particle position is used to determine the particle position in the next iteration by adding the velocity value that has been carried out with the particle chromosome value in the previous position. The particle in this new position will be searched for its fitness value and compared with the fitness value of the particle in the previous position to find the pbest and gbest particles.

3.4. Testing and Analysis

3.4.1. Testing the number of particles

Testing the number of particles is carried out to see the effect of the number of particles used on the fitness value. The number of particles used was 10, 20, 30, 40, 50, 60, 70, 80, 90, 100 which were carried out 10 times each to find the average fitness value in each test. The number of iterations used is 10 with parameter values c1 = 1 and c2 = 1. From the results of testing the number of particles, the lowest fitness value is in the smallest number of particles, namely 10 with an average fitness value of 87187.7 and the highest is in particles with a total of 100 with an average value -average fitness 89248.8. From Figure 2 it can be concluded that the more the number of particles used, the higher the fitness value obtained. The increase in the fitness value occurs because the more particles, the greater the scope for finding solutions and the variation in the fitness value obtained. The large variation of fitness values makes early convergence avoidable.

3.4.2. Testing the Number of Iterations

Testing the number of iterations is done to see the effect of the number of iterations on the fitness value. The number of iterations used here is 10, 20, 30, 40, 50, 60, 70, 80, 90, 100 with 10 trials each. Whereas the number of particles used is 100 which is obtained from the best results from testing the number of particles and for parameter values c1 = 1 and c2 = 1. The lowest fitness is at the smallest number of iterations, namely 10 with an average fitness value of 89248.8 and the highest is at the number 100 iterations with an average fitness value of 93924.5. From Figure 3 above, it can be seen that the average value of fitness experiencing stability. This is caused by the more frequent process of updating the fitness value and the more frequent movement of particles to a position, whether it is optimum or not, but as the iterations increase, the probability of the optimum position obtained will tend to be high so that the fitness value obtained will be better.

3.4.3. Discussion on the Analysis of Optimization Results with the Particle Swarm Optimization (PSO) Method

From testing the number of particles, testing the number of iterations, testing the speed parameters c1 and c2 it was found that the optimal parameter values were the number of particles of 100, the number of particles of 100, and the number of combinations c1 and c2 of 1.5 and 1.5. From the parameter values as mentioned above, the fitness value with the highest number is 94878 from the range 83170 to 99835 or around 0.703 with the range 0 to 1. The minimum number of violations results in a fitness value of 99835, which is obtained from reducing the value of the constant to the violation value of 165 The value of this minimum violation is obtained by looking at the value of the minimum interest cost, which is 1 for each of the 165 chromosomes, and 0 for the value of the cost of credits and class costs. While the maximum number of violations results in a fitness value of 83170. This maximum violation value is obtained from reducing the constant value of the violation value of 16830, namely the interest cost of 100 for each 165 chromosomes, 1 costsks value and 1 class cost value for each 165 chromosome. The resulting fitness value can still be improved by seeing that the convergence of values has not been seen so that the addition of the maximum number of parameters can be done. However, adding the maximum number of parameters will also lead to longer calculation times and larger memory allocations, so that a machine specification with larger RAM memory and better memory allocation is required. The large number of test parameter values used here is the author's consideration by looking at the capabilities of the devices used. As for the output results from the system, the resulting error value decreases as the number of parameters used increases. From the resulting fitness value, there were errors in lecturer placement for courses with an average of 52 out of 165 courses. This can be reduced by increasing the parameter values. Increasing parameter values such as the number of particles causes a wider scope of search so that the optimum position is achieved more quickly and the variation in the fitness value is greater. This causes a reduction in the occurrence of early convergence on the fitness value so that the search for solutions can be continued. Increasing the number of iterations causes more fitness value updating processes to be carried out because the movement of particles to a position is more frequent. The direction of the fitness value graph resulting in an increase in the number of iterations tends to be higher. This is influenced by the movement of particles towards a more optimum position. Meanwhile, a better combination of speed parameters c1 and c2 is obtained from the same combination value and a larger number. This is due to the direction of the particle search which is getting closer to the position of the pbest and gbest particles so that generally the optimal solution will be reached more quickly as the fitness value is getting higher.

4. CONCLUSION

The application of the Particle Swarm Optimization method to optimize assignments for teachers in this study gave quite good results when it was observed that the number of errors continued to decrease as the number of particle installations increased. Even though the results obtained in this study are still quite weak, if you look further, the fit value increases with each additional number of parameters and there is no convergence of values. This can be overcome by adding the maximum number of test parameters even though it requires long computations and a larger device memory allocation.

1) By conducting tests to determine the effect of parameter values on the fitness values obtained, namely the number of particles test, the number of repetitions test and the speed parameters test c1 and c2. Based on the tests carried out, the best results for the PSO parameter using the number of particles are 100, the number of iterations is 100, and the parameter values c1 and c2 are 1.5 and 1.5 respectively with a matching value of 94878. The matching value can be increased further by increasing the maximum amount. Although the experimental setup requires longer computation time and more device memory allocation, it is better to use higher spec hardware for better memory allocation.
2) The value of the cost obtained by the PSO method is still relatively high, so for assignment problems this PSO method can be combined with other optimization methods.
3) Interesting additional data, causing an error in the course placement above.
4) Calculations can be minimized with more and more data.

REFERENCES

[1] P. Chakraborty, P. Mittal, M. S. Gupta, S. Yadav, dan A. Arora, “Opinion of students on online education during the COVID-19 pandemic,” *Hum. Behav. Emerg. Technol.*, vol. 3, no. 3, 2021, doi: 10.1002/hbe2.240.

[2] Ellyzabeth Sukmawati, Iwan Adhicandra, dan Nur Sucahyo, “Information System Design of Online-Based Technology News Forum,” *Int. J. Artif. Intell. Res.*, vol. 1.2, Des 2022, doi: https://doi.org/10.29099/ijair.v6i1.2.593.

[3] B. Agnesiana, D. Mahendika, S. Rumfot, dan E. Sukmawati, “The Analysis of Teacher Efforts to Developing Students Interpersonal and Intrapersonal Intelligence in Learning Activities,” *J. Educ.*, vol. 05, no. 04, hal. 11661–11666, 2023.

[4] Y. Duan, N. Chen, L. Chang, Y. Ni, S. V. N. S. Kumar, dan P. Zhang, "CAPSO: Chaos Adaptive Particle Swarm Optimization Algorithm," *IEEE Access*, vol. 10, 2022, doi: 10.1109/ACCESS.2022.3158666.

[5] H. Liu, B. Wu, A. Maleki, dan F. Pourfayaz, "An improved particle swarm optimization for optimal configuration of standalone photovoltaic scheme components," *Energy Sci. Eng.*, vol. 10, no. 3, 2022, doi: 10.1002/ese3.1052.

[6] A. Iqbal, D. Ying, A. Saleem, M. A. Hayat, dan M. Mateen, "Proposed particle swarm optimization technique for the wind turbine control system," *Meas. Control (United Kingdom)*, vol. 53, no. 5–6, 2020, doi: 10.1177/0020294020902785.

[7] L. Zhang, H. Zheng, Q. Hu, B. Su, dan L. Lyu, "An Adaptive Droop Control Strategy for Islanded Microgrid Based on Improved Particle Swarm Optimization," *IEEE Access*, vol. 8, 2020, doi: 10.1109/ACCESS.2019.2960871.

[8] T. M. Shami, A. A. El-Saleh, M. Alswaitti, Q. Al-Tashi, M. A. Summakieh, dan S. Mirjalili, "Particle Swarm Optimization: A Comprehensive Survey," *IEEE Access*, vol. 10, 2022, doi: 10.1109/ACCESS.2022.3142859.

[9] L. Vanneschi dan S. Silva, "Particle Swarm Optimization," in *Natural Computing Series*, 2023. doi: 10.1007/978-3-031-17922-8_4.

[10] A. P. Piotrowski, J. J. Napiorkowski, dan A. E. Piotrowska, "Population size in Particle Swarm Optimization," *Swarm Evol. Comput.*, vol. 58, 2020, doi: 10.1016/j.swevo.2020.100718.

[11] P. Asaah, L. Hao, dan J. Ji, "Optimal Placement of Wind Turbines in Wind Farm Layout Using Particle Swarm Optimization," *J. Mod. Power Syst. Clean Energy*, vol. 9, no. 2, 2021, doi: 10.35833/MPCE.2019.000087.

[12] D. Wang, D. Tan, dan L. Liu, "Particle swarm optimization algorithm: an overview," *Soft Comput.*, vol. 22, no. 2, 2018, doi: 10.1007/s00500-016-2474-6.

[13] J. Peng, Y. Li, H. Kang, Y. Shen, X. Sun, dan Q. Chen, "Impact of population topology on particle swarm optimization and its variants: An information propagation perspective," *Swarm Evol. Comput.*, vol. 69, 2022, doi: 10.1016/j.swevo.2021.100990.

[14] A. P. Piotrowski dan A. E. Piotrowska, "Differential evolution and particle swarm optimization against COVID-19," *Artif. Intell. Rev.*, vol. 55, no. 3, 2022, doi: 10.1007/s10462-021-10052-w.

[15] D. Sedighizadeh, E. Masehian, M. Sedighizadeh, dan H. Akbaripour, "GEPSO: A new generalized particle swarm optimization algorithm," *Math. Comput. Simul.*, vol. 179, 2021, doi: 10.1016/j.matcom.2020.08.013.

[16] J. C. Mba dan M. M. Mai, "A Particle Swarm Optimization Copula-Based Approach with Application to Cryptocurrency Portfolio Optimisation," *J. Risk Financ. Manag.*, vol. 15, no. 7, 2022, doi: 10.3390/jrfm15070285.

[17] F. Pace, A. Santilano, dan A. Godio, "A Review of Geophysical Modeling Based on Particle Swarm Optimization," *Surveys in Geophysics*, vol. 42, no. 3. 2021. doi: 10.1007/s10712-021-09638-4.

[18] C. Chen dan C. Li, "Process Synthesis and Design Problems Based on a Global Particle Swarm Optimization Algorithm," *IEEE Access*, vol. 9, 2021, doi: 10.1109/ACCESS.2021.3049175.

[19] F. Wang, H. Zhang, dan A. Zhou, "A particle swarm optimization algorithm for mixed-variable optimization problems," *Swarm Evol. Comput.*, vol. 60, 2021, doi: 10.1016/j.swevo.2020.100808.

[20] M. Sharif *et al.*, "Recognition of different types of leukocytes using YOLoV2 and optimized bag-of-features," *IEEE Access*, vol. 8, 2020, doi: 10.1109/ACCESS.2020.3021660.

[21] F. Buonamici, M. Carfagni, R. Furferi, L. Governi, A. Lapini, dan Y. Volpe, "Reverse engineering of mechanical parts: A template-based approach," *J. Comput. Des. Eng.*, vol. 5, no. 2, 2018, doi: 10.1016/j.jcde.2017.11.009.