ISSN 2089-8673 (Print) | ISSN 2548-4265 (Online)
Volume 12, Issue 3, December 2023

IMPROVING THE PERFORMANCE OF THE GENERAL COURSE SCHEDULING SYSTEM AT UPN VETERAN JAWA TIMUR THROUGH THE APPLICATION OF THE IWCFPSO ALGORITHM

Made Hanindia Prami Swari[1], Chrystia Aji Putra[2], I Putu Susila Handika[3], Dwi Wahyuningtyas[4]

[1,2]Computer Science Department, UPN Veteran Jawa Timur
[3]Faculty of Technology and Informatics, Indonesian Institute of Business and Technology
[4]English for Business Communication and Professionals, Faculty of Social and Political Sciences, UPN Veteran Jawa Timur

e-mail:madehanindia.fik@upnjatim.ac.id[1], ajiputra@upnjatim.ac.id[2], susila.handika@instiki.ac.id[3], dwi.wahyuningtyas.ih@upnjatim.ac.id[4]

Abstract

Algorithm selection is the main key in producing a system, especially an artificial intelligence-based system, for example, a course scheduling system that involves many constraints in producing an optimal schedule. The use of inertia weight in the Modified Particle Swarm Optimization (MIPSO) algorithm which is used as a scheduling algorithm in other research can be optimized by using a combination of inertia weight and constriction factor that involved in Inertia Constriction Weight Factor Particle Swarm Optimization (ICWFPSO) algorithm. Based on that fact, the aim of this research is to develop a scheduling system using the ICWFPSO algorithm and measuring the performance of ICWFPSO compared to the scheduling system that was previously built using MIPSO in the terms of time consumption, CPU, and memory usage. Scheduling system development is carried out using the waterfall method with research steps namely problem and needs analysis, data collection and literature study, system design, system implementation, and ends with testing the performance of the algorithms that are applied to the scheduling system. Based on the tests carried out, it can be concluded that the ICWFPSO algorithm provides 200% of better performance compared to the MIPSO algorithm in terms of optimal schedule generation time. Meanwhile, tests carried out on CPU and RAM usage showed that there was no significant impact on these two parameters through the implementation of MIPSO and ICWFPSO in the course scheduling system created.

Keywords: best scheduling, ICWFPSO, number of iterations

Received:13-09-2023 | Revised: 28-11-2023 | Accepted: 06-12-2023
DOI: https://doi.org/10.23887/janapati.v12i3.68128

INTRODUCTION

Creating a course scheduling system is not an easy matter. Many previous studies have been carried out using various methods or algorithms to create a scheduling system that best meets the cases raised. Therefore the selection of the algorithm to be used in making general course schedules at UPN "Veteran" Jawa Timur is also a crucial thing that must be studied to produce a schedule that has minimal clashes and can meet the needs of managers.

In overcoming the weaknesses in the PSO algorithm, modifications are made to produce a better scheduling system. One of the studies that implemented the development of the PSO method was carried out by Kanata, et al. who compared the ability to modify the PSO algorithm, namely the Modified Particle Swarm Optimization (MPSO) algorithm with Modified Improved Particle Swarm Optimization (MIPSO) in the case of finding economic dispatch solutions on the Java-Bali 500kV electricity system. This study states that the MIPSO method produces method [1].This has been proven the most optimal solution compared to the MPSO by research conducted by researchers in the first year, where the results show that the MIPSO algorithm produces 5 times better performance or 500% compared to the genetic algorithm [2]. After the scheduling system was implemented at UPN "Veteran" Jawa Timur, a new problem emerged, many

lecturers were unable to teach according to the schedule produced by the scheduling system that had been created. This is because there is no basis in the form of information on the teaching hours and days desired by each lecturer, scheduling is purely made randomly without these parameters. Therefore, the addition of a questionnaire feature that functions to enter schedules by each lecturer in this study is needed to fulfilled. Based on research conducted by Kanata, et al [3], it was stated that in the case of Economic Generation in the IEEE 26 Bus Power System, The inertia weight in the MIPSO algorithm can be optimized using a combination of inertia weight and constriction factor (ICWFPSO). Therefore with the implementation of ICWFPSO in this research, it is necessary to develop a scheduling system with a better computational process in terms of time consumption in generating optimal scheduling results and with the additional ability to produce schedules according to the lecturer's desired time through the addition of a questionnaire feature.

The MIPSO algorithm used in the development of the scheduling system in previous studies gave better results in terms of the time needed to make the schedule compared to the use of the Genetic algorithm. In this proposed research, the development of a scheduling system using the inertia weight algorithm with constriction factor (IWCFPSO) is also carried out, because the use of inertia weight in the MIPSO algorithm used as a scheduling algorithm in previous research can be optimized using a combination of inertia weight and constriction factor (IWCFPSO) [3]. Through the use of the IWCFPSO algorithm, it is hoped that a scheduling system that is better than the existing scheduling system will be produced especially in terms of the time required to produce an optimal schedule, RAM and CPU usage, and also may meets the wishes and needs of lecturers of general course in UPN “Veteran” Jawa Timur by adding the questionnaire feature in this research.

To validate the performance of the scheduling system which has been optimized using the ICWFPSO method, system performance testing was carried out using time speed testing in producing an optimal schedule, as well as testing to determine server resource usage from the two algorithms. Apart from performance testing, system functionality testing is also carried out using black box testing. Black box testing techniques are important to test the functionality of the system without knowing its inner detail which makes sure correct, consistent, complete and accurate behavior or function of a system [4].

METHODS

Particle swarm optimization (PSO) is a global heuristic optimization method originally proposed by J. Kennedy and Eberhart in 1995 [5]. This algorithm is based on animal intelligence or the movement behavior of flocks of birds or fish in search of food so that it can be applied to both scientific and engineering research methods [6] PSO improvement by applying inertia weight which functions to control the 4 speeds from time to time so that the speed decreases linearly which makes PSO look for a larger space at the beginning to get a position quickly, which is the most optimal solution. When the inertia weight decreases, the particle velocity also slows down to find a smoother partial solution so that this improvement method can provide a balance between global exploration and local exploitation [7]. Then improvements were made using a constriction factor to ensure convergence and the amplitude of the oscillations of the particles decreased from time to time without setting the maximum speed [8]. Research comparing inertia weights with constriction factors where the use of constriction factors has better convergence than using inertia weights [9]. Figure 1 is a flowchart for finding a solution.

The development of the IWCFPSO algorithm lies in epoch = k + 1, where particle velocity updates are carried out(V_{id}^{k+1}) and particle positions(x_{id}^{k+1}) by using computation as shown in Equation 1.

$$V_{id}^{k+1} = CF\ (V_{id}^{k} + C_1\gamma_1^k\left(Pbest_{id}^{k} - x_{id}^{k}\right) + C_2\gamma_2^k\left(Gbest_{id}^{k} - x_{id}^{k}\right) \quad (1)$$

In developing a system for scheduling courses for general course programs at UPN "Veteran" Jawa Timur, a structured approach was used using the software development method with the waterfall model as shown in Figure 2. This method is often used for in software development because of the systematic and sequential approach to creating a system [10] [11].

ISSN 2089-8673 (Print) | ISSN 2548-4265 (Online)
Volume 12, Issue 3, December 2023

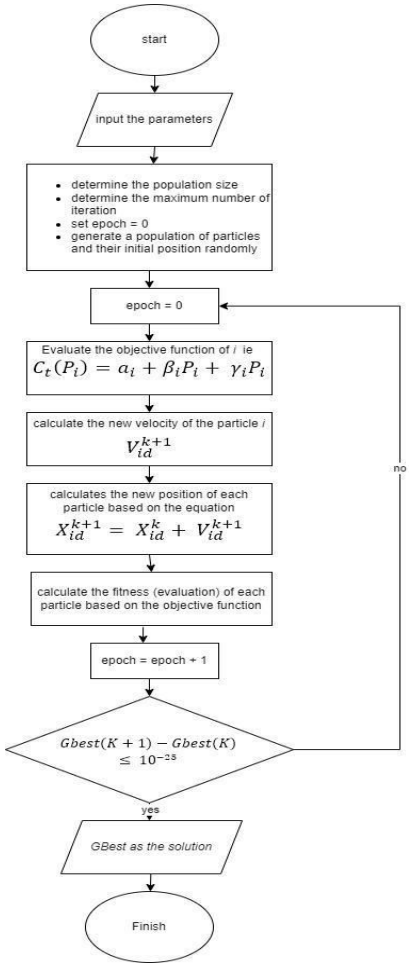

Figure 1. Solution Search Workflow

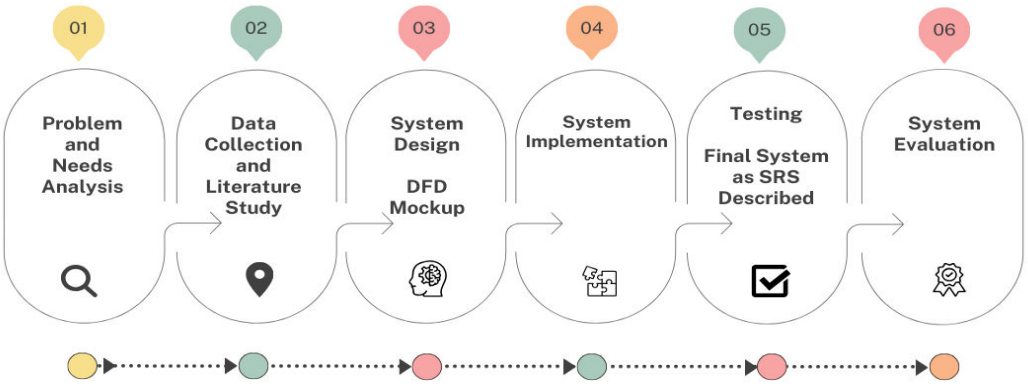

Figure 2. System Development Method

The manufacturing process follows the flow from analysis, design, code, testing, and maintenance [12]. Figure 2 is a scheduling system development method carried out in this study. Based on the research steps depicted in Figure 2, the activities carried out at each research step are:

1. Problem and Needs Analysis

Software requirements analysis is a part of the planning stage of software project [8]. Based on the results of an evaluation of the scheduling system that had been built in the previous year's research, it was found that there were still some developments that could be

ISSN 2089-8673 (Print) | ISSN 2548-4265 (Online)
Volume 12, Issue 3, December 2023

carried out to meet user needs, in this case, the needs of lecturers supporting general courses at UPN "Veteran" Jawa Timur. There are deficiencies in the scheduling system, namely, there are still many schedules that are not made according to the wishes and needs of prospective supervisors. This can happen because schedules are built automatically without considering the wishes of the supporting lecturers. For this reason, in this study, improvements were made to accommodate the schedule generated by the automatic scheduling system which was made as much as possible to meet the needs of prospective lecturers regarding the desired teaching schedule. The process of interviews and observations was carried out to find out additional features that will be implemented in the scheduling system that will be built in this study. After conducting interviews and observations, problems that will be solved through the development of this information system can then be mapped along with a list of the functional and non-functional requirements that must be accommodated by the system built in the form of a software recruitment specification (SRS) document [13]. SRS documents are usually compiled in natural langguage as we use in daily conversation [14].

2. Data Collection and Literature Study

To support the success of the system to be built, some data is needed which will become a source of knowledge for the system, namely a list of lecturers, classes, and students. In this step a literature study was also carried out related to the materials used in completing this research, for example, theories related to programming, databases, scheduling, and optimization algorithms such as the IWCFPSO method in particular and other journals related to similar research that had been carried out by previous researchers.

3. System Design

At this stage, a system design process is carried out that is useful for facilitating the implementation stage [15]. The results of system design activities are in the form of structured design diagrams including conceptual data models, physical data models, and data flow diagrams. The general system workflow design can be seen in Figure 3.

Based on Figure 3, it can be seen that there are 2 roles involved in the use of this scheduling system, namely the general course manager who is in charge of entering lecturer data and the number of classes and lecturers involved by entering the desired schedule. The scheduling system will provide schedule recommendations according to the limits that have been entered. If the resulting schedule is appropriate, each supporting lecturer can see the schedule that has been formed.

4. System Implementation

After the system design has been made, the next step is to implement all the results of the design into a programming language. The system implementation is carried out using the Codeigniter framework using the PHP programming language and the mariaDb database. The result of this stage is an automatic general course scheduling feature that can meet user needs as outlined in the Software Requirement Specification (SRS).

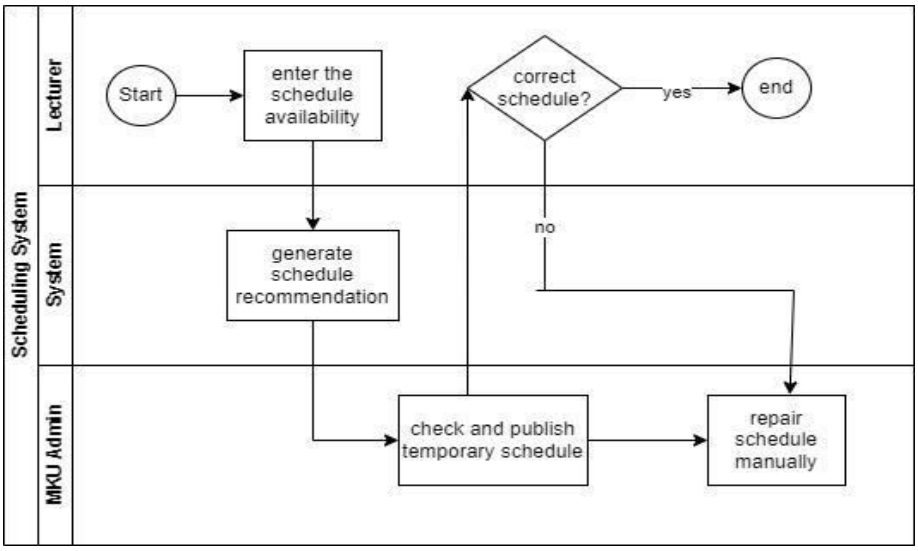

Figure 3. System Flow Design

ISSN 2089-8673 (Print) | ISSN 2548-4265 (Online)
Volume 12, Issue 3, December 2023

5. System Testing

The final step of the activity phase is to test the system. This step is important to do so that the system built is running well and meets all user needs as stated in the SRS document that was created in the early stages of research activities. Tests are carried out in the form of system functionality testing using a black box testing approach. Another test that is done is to check the schedule that is formed automatically and compare the better schedule between the two algorithms used in terms of the speed of system execution in producing the schedule and the effectiveness of the resulting schedule. Apart from that, testing of the use of server resources for the two algorithms used in the scheduling system was also carried out.

RESULTS AND DISCUSSION

The scheduling system is built on a web basis to make it easier for users to access the system. Figure 4 is the implementation of the scheduling system dashboard that has been built. At the beginning of each semester, lecturers can have a teaching schedule according to the courses taught and open for that semester. This can be seen in Figure 5.

Figure 4. Scheduling System Dashboard

Figure 5. Lecturer Questionnaire Menu

ISSN 2089-8673 (Print) | ISSN 2548-4265 (Online)
Volume 12, Issue 3, December 2023

After the lecturer has chosen the course and schedule chosen, the next process is to generate a schedule using the ICWFPSO algorithm which is carried out by employees. Figure 6 is the result of generating a schedule generated by the system according to the number of courses and questionnaires filled in by the lecturer.

After the implementation of the scheduling system has been carried out, the next step is to test the performance of the ICWFPSO algorithm in terms of the time required to generate a schedule compared to the time required to generate a schedule using the MIPSO algorithm. Table 1 is the result of this comparison. The performance of the time required to produce a schedule by the two algorithms is compared when it has produced an optimal schedule, where the optimal schedule in question is a schedule without conflicts.

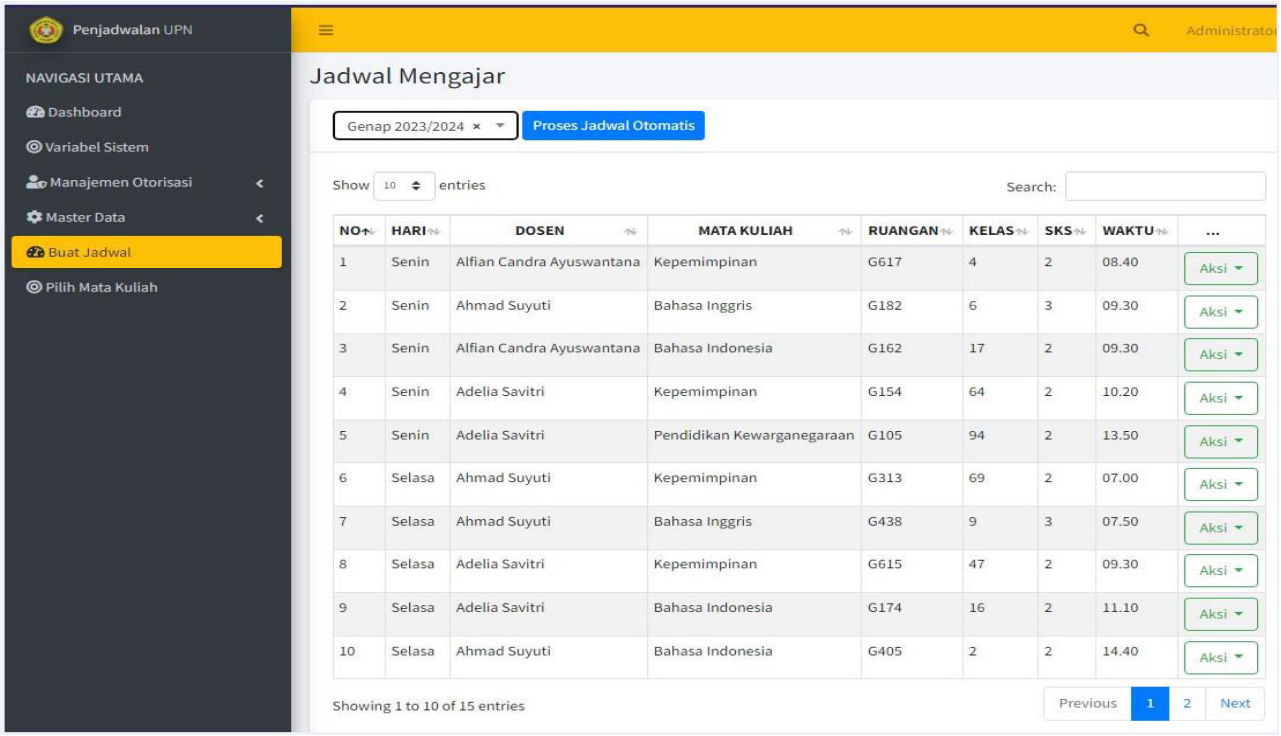

Figure 6. Scheduling Generate Results

Table 1. Comparison Scheduling System with ICWFPSO and MIPSO

No of test	ICWFPSO		MIPSO	
	Time(s)	Iteration	Time(s)	Iteration
1	57.1542	1	188.1179	4
2	77.5165	2	178.0561	4
3	94.6568	2	134.0227	3
4	97.0031	2	165.1586	4
5	67.1797	2	190.3659	5
Average	**78.70206**		**171.1442**	

Based on the test results presented in Table 1, it can be concluded that the ICWFPSO algorithm produces better performance in terms of time consumption to produced optimal schedule compared to the MIPSO algorithm. Optimal schedule means that there are no conflictiong schedule. The ICWFPSO algorithm produces schedules 2 times faster than the MIPSO algorithm, and produces the best schedule on the first trial and only requires 1-time iteration.

Meanwhile, the results of testing the use of server resources used by each algorithm in producing an optimal schedule can be seen in Table 2. Table 2 is a test related to RAM usage required by the PSO algorithm which uses constriction factor to It ensures convergence of particle swarm optimization algorithm to obtain optimal power flow solutions [15], compared to the use of the particle swarm optimization method using the inertia weight approach.

Table 2. RAM Usage Testing

No of Test	Inertia (%)	Constriction Factor (%)
1	6.3	6.4
2	6.4	6.4
3	6.4	6.4
4	6.4	6.3
5	6.4	6.4

In table 2 it can be seen that neither the use of inertia weight nor constriction factor in the particle swarm optimization algorithm has an impact on RAM usage in the course scheduling process. Where each approach gives an average value of 6.4%.

Meanwhile, table 3 shows the CPU resource usage required by each scheduling algorithm used in the scheduling system to produce an optimal schedule.

Table 3. CPU Usage Testing

No of Test	Inertia (%)	Constriction Factor (%)
1	60.8	60.5
2	61	60.9
3	60.4	60.4
4	60.5	60.8
5	60.4	60.8

As can be seen in Table 3, in terms of PU resources there is also no significant difference between the use of inertia weight and constriction factor in the particle swarm optimization algorithm to generate the schedule according the limitation that specify by the user, where the average CPU resource usage required is 60.8%.

The final test carried out in this research was to test the functionality of the scheduling system using the ICWFPSO method. Blackbox testing used in this research to ensure all system requirements have been met in the course scheduling system that has been built in the previous research step [17].

This testing step is carried out in three steps, namely Input Requirement and functional specification of the system are examined, followed by tester constructs test cases with the selected input and execute them, after all these testing, tester gets desired output and prepares final report [18]. The results of testing the functionality of the scheduling system using black box testing were carried out using test scenarios and test results as shown in Table 4.

Table 4. Functionality Testing Result

No of Test	Test Case	Scenario	Expected Results	Test Result
1	Login	Entered the wrong username and/or password	An error message appears, unable to access the dashboard	Valid
		Entered the valid username and/or password	Log in to the dashboard according to user access rights	Valid
2	Master Data Management	Entered lecturer, classes, and course	Lecturer, classes, and courses were added to the database	Valid
		Modify lecturer, classes, and courses data	Lecturer, classes, and courses had changed according to the new data entered by user	Valid
		Delete lecturer, classes, and courses data	Lecturer, classes, and course had deleted from the database	Valid
3	Geberate the Schedule	Course schedule generating as lecturer questionnaire filled out using ICWFPSO algorithm	Courses schedule is formed and can be accessed according to the limitation that had been entered by user	Valid
4	Choose the Courses	Display the choice of courses that available on that semester	The courses taught in the semester appears in the scheduling system	Valid
		Choose the courses as desires	The course schedule selected by users is stored on database	Valid
5	Authorization Management	Add users	The new users data entered is stored on database	Valid
		Add user types	The new user types according to the selected user access rights is saved	Valid

There are 5 test scenarios carried out in this research according to the main features of the scheduling system being built. These features include the login feature which functions to ensure that only registered users can use this scheduling system, so that through testing the login functionality it can be ensured that each user can access the features in the scheduling system according to the access rights granted. Apart from the login feature, the master data management feature is a feature that must be checked for functionality. The master data management feature can give admins access to manage master data which will be a limitation or input to the scheduling process by the ICWFPSO algorithm. As for managing master data, the functions checked include the functions of entering new data, deleting old data, and updating data according to admin input. The next feature is the main feature of the scheduling system, where after going through a functionality test this feature can guarantee that the scheduling system created in this research can produce general course schedules according to the constraints

that have been determined in the master data management feature.

CONCLUSION

Based on the tests that have been carried out, it can be concluded that the ICWFPSO algorithm provides better performance than using the MIPSO algorithm, this is because PSO performance is affected by how long a particle moves to the optimal location and how fast the particle moves to the optimal location. Inertia controls how far the particle moves to the optimal location, while the application of the Construction Factor in the ICWFPSO algorithm combines inertia with particle speed so that through the workings of this algorithm it can control how far it has to move and its moving speed, or in other words the role of CF is to prevent the particles do not pass through the optimal solution. The difference in the number of iterations needed to generate the optimal schedule is because the schedule search process begins by generating a schedule randomly so that if the initial error value generated is close to the optimal value, the number of iterations will be fewer. In the future, research can be developed to look for techniques to generate near-optimal schedules so that the performance of the ICWFSO algorithm can be improved again. Meanwhile, in terms of CPU and RAM resource usage, there is no significant difference in the number of resource usage between the implementation of the MIPSO and ICWFPSO algorithms in the scheduling system built in this research.

ACKNOWLEDGMENTS

The author is grateful to LPPM UPN "Veteran" Jawa Timur for funding assistance so that this research activity can be completed properly.

REFERENCES

[1] S. Kanata, Sarjiya and S. P. Hadi, "Modified Improved Particle Swarm Optimization (MIPSO) sebagai Solusi Economic Dispatch pada Sistem Kelistrikan 500 kV Jawa-Bali," *Transmisi: Jurnal Ilmiah Teknik Elektro,* vol. 15, no. 2, pp. 66-72, 2013.

[2] M. H. P. Swari, C. A. Putra, Handika and I. P. S, "Analisis Perbandingan Algoritma Genetika dan Modified Improved Particle Swarim Optimization dalam Penjadwalan Mata Kuliah," *Jurnal Janapati,* vol. 11, no. 2, 2022.

[3] S. Kanata, "Kombinasi Inertia Weight dan Constriction Factor pada Particle Swarm Optimization sebagai Solusi Pembangkitan Ekonomi pada Sistem Tenaga IEEE 26 Bus," *Jurnal Electrans,* vol. 12, no. 2, 2013.

[4] U. Saeed and A. M. Amjad, "ISTQB: Black Box testing Strategies used in Financial Industry for Functional testing," School of Engineering Blekinge Institute of Technology, Sweden, 2009.

[5] A. S. Ashour and Y. G. In, Optimization-Based Neutrosophic Set in Computer-Aided Diagnosis, Elsevier Inc. All, 2020.

[6] M. Ram and J. P. Davim, Mathematics Applied to Engineering, Elsevier, 2017.

[7] Y. Shi and R. Eberhart, "A Modified Particle Swarm Optimizer," in *IEEE International Conference Evolution Computation*, 1998.

[8] M. Clerc, "The Swarm and The Queen : Towards a Deterministic and Adaptive Particle Swarm Optimization," in *Congress on Evalutiory Computation*, Washington, 1999.

[9] R. Eberhart and Y. Shi, "Comparing Inertia Weight and Constriction Factors in Particle Swarim Optimizartion," in *Proceeding of the 2000 Congress on Evalutionary Computation*, 2000.

[10] S. Herawati, Y. D. P. Negara, H. F. Febriyansyah and D. A. Fatah, "Application of the Waterfall Method on a Web-Based Job Training Management Information System at Trunojoyo University Madura," in *E3S Web of Conferences*, 2021.

[11] I. G. I. Sudipa, Metode Penelitian Bidang Ilmu Informatika (Teori & Referensi Berbasis Studi Kasus), Jambi: PT. Sonpedia Publishing Indonesia, 2023.

[12] R. Pressman, Rekayasa Perangkat Lunak (Pendekatan Praktisi), Yogyakarta: ANDI, 2012.

[13] F. P. Putra and D. Enda, "Model Design for Grammatical Error Identification in Software Requirements Specification using Statistics and Rule-Based Techniques," in *J. Phys.: Conf. Ser*, 2020.

[14] D. Enda and D. Siahaan, "Rekomendasi Perbaikan Pernyataan Kebutuhan yang Rancu dalam Spesifikasi Kebutuhan Perangkat Lunak Menggunakan Teknik Berbasis Aturan," *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK),* vol. 5, no. 2, pp. 207-215, 2018.

[15] J. W. Satzinger, R. B. Jackson and S. D. Burd, Systems Analysis and Design in a Changing World, 7/E., New York: Cengage

Learning, 2016.

[16] b. B. Naik, P. C. Raju and S. S. Rao, "A Constriction Factor Based Particle Swarm Optimization for Congestion Management in Transmission Systems," *International Journal on Electrical Engineering and Informatics,* vol. 10, no. 2, pp. 232-241, 2018.

[17] L. C. Hermawan, M. R. Mubarok, H. Mairudin and A. Mahdiyan, "Pengujian Black Box pada Aplikasi Verifikasi Data Nasabah dengan Menggunakan Metode Boundary Value Analysis," *Jurnal Teknologi Sistem Informasi dan Aplikasi,* vol. 3, no. 3, 2020.

[18] M. E. Khan, "Different Approaches to Black Box Testing Technique for Finding Errors," *International Journal of Software Engineering & Applications (IJSEA),* vol. 2, no. 4, pp. 31-40, 2011.