WHALE OPTIMIZATION ALGORITHM FOR PRODUCTION SCHEDULING PROBLEMS IN SMALL BAKERY ENTERPRISES

Cecilia E. Nugraheni (1), Luciana Abednego (1), Maria Widyarini (2)*

(1) Informatics Dept., (2) Business Administration Dept.
Parahyangan Catholic University, Bandung
Indonesia

* Corresponding Author, e-mail: cheni@unpar.ac.id

Abstract: The no-wait flexible flowshop scheduling problem (NWFFSP) is a significant challenge in industries such as bakeries, where strict no-wait constraints and resource limitations, like oven availability, complicate scheduling. This research aims to evaluate the performance of the Whale Optimization Algorithm (WOA) in solving NWFFSP as a multi-objective problem, focusing on minimizing makespan and total oven idle time (ToIT). The study employs the Pareto principle to address trade-offs between these conflicting objectives. Experimental results demonstrate that WOA effectively balances both objectives, maintaining comparable efficiency to mono-objective optimization. The findings confirm WOA's potential as a robust tool for solving complex multi-objective scheduling problems in practical settings.

Key words: NWFFSP, WOA, multi-objective optimization, Pareto.

1. INTRODUCTION

In some country, like Indonesia, bread and other bakery products have been widely accepted as alternatives to staple foods like rice, reflecting their growing popularity and integration into local dietary habits. This situation has led to the proliferation of bread entrepreneurs, particularly in small-scale industries. Like other industries, the bread industry faces problems and challenges, including product innovation, availability of raw materials and equipment, production processes, marketing strategies, etc.

This study focuses on problems related to the production scheduling of the small-scale bakery industry. Small-scale bakeries often face unique production scheduling challenges. These businesses typically operate with limited resources, such as a small number of ovens, mixers, and workers, which must be allocated efficiently to meet customer demands. Additionally, certain constraints, such as no-wait conditions between consecutive production stages (e.g., mixing, proofing, and baking), complicate the scheduling process further.

Production scheduling in the bakery industry is usually modeled as a no-wait flow shop scheduling problem (NWFSP). NWFSP is a scheduling problem with the following characteristics: the execution of jobs follows a uniform sequence; one or more machines or operators are available at each sequence; a machine can only perform one job at a time, and there must be no delays between sequences. If more than one machine is available at a sequence, the problem is called a no-wait flexible flow shop scheduling problem (NWFFSP).

Scheduling problems are included in the NP-complete problems. Non-exact approaches are often used to solve complex scheduling problems, including metaheuristics. Metaheuristics have the advantage of finding solutions faster with acceptable optimality. Many bio-inspired metaheuristics have been used to solve various optimization problems in various fields, some of which are Genetic Algorithm (GA) and its variants, Particle Swarm Optimization (PSO) and its variants, Ant Colony Optimization (ACO), Dragonfly Algorithm, Whale Optimization Algorithm (WOA) and [9-16].

This study investigates the use of the Whale Optimization Algorithm (WOA) in solving the problem of scheduling small-scale bakery production. The scheduling is modeled as a two-stage NWFFSP. The performance of WOA will be measured based on multi-objective criteria, namely makespan (total time required to complete all jobs) and total idle time (total time the machine is not working). To address the trade-off between minimizing each objective, namely makespan and total oven idle time (ToIT), we use the Pareto principle. This study hypothesizes that the performance of WOA in a multi-objective scenario will not deteriorate compared to its performance in mono-objective optimization. Thus, this study aims to ensure that WOA maintains its efficiency and effectiveness while balancing multiple objectives.

2. RELATED WORK

Research on the bakery industry has been conducted in various aspects, including supply chain aspects [1, 2], distribution [3-5], and scheduling [6-11]. Trattner et al. explores the application of product wheels for scheduling in the baking industry through a case study, demonstrating their potential to improve production planning efficiency [6]. In their paper, Blamah et al. addresses a scheduling problem in the baking industry, proposing mathematical models and optimization techniques to improve production efficiency and meet operational constraints [7].

This study focuses on problems related to the production scheduling of the bakery industry. Production scheduling in the bakery industry faces unique challenges due to the nature of processes that require strict adherence to non-delay schedules, as certain tasks cannot be interrupted or paused once started. Additionally, efficient use of ovens poses a critical issue, as they often serve as bottlenecks due to limited capacity and the need for precise timing in baking different products.

Babor et al. [8-10] have conducted several studies on solving bakery scheduling problems using metaheuristics. The study by Babor et al. focuses on optimizing no-wait flowshop scheduling in bakery production environments using a hybrid model that integrates Modified Particle Swarm Optimization (MPSO), Simulated Annealing (SA), Modified PSO, NSGA-II, SPEA2, GDE3, OMOPSO, SMPSO, Simulated Annealing, and Nawaz-Enscore-Ham (NEH) heuristic. The objective criteria studied by Babor et al. include mono-objective and multi-objective, such as makespan and ToIT

This study investigates the use of the Whale Optimization Algorithm (WOA) in solving the problem of scheduling small-scale bakery production. The scheduling is modeled as a two-stage NWFFSP. The performance of WOA will be measured based on multi-objective criteria, namely *makespan* (total time required to complete all jobs) and *total idle time of machines* (total time the machine is not working). We use the Pareto principle to address the trade-off between minimizing each objective, namely makespan and total oven idle time (ToIT). The very close work and our reference is from Babor et al. [8-10]. The difference between our work and that of Babor et al. is the use of machines that can vary in one stage (heterogeneous machines). This issue is not explicitly discussed in their papers.

This study is a continuation of our previous work on production scheduling optimization using WOA. In [15-16], we report on the results of our research on using WOA to solve optimization problems with multi-objectives. In [15], we used the original WOA, while in [16], we proposed hg-WOA, which is a combination of WOA and GA. In [16] we also utilized the Pareto principle for solving multi-objective scheduling problems in the textile industry.

3. METHODOLOGY

3.1. Bakery Scheduling Problem: A Case Study

Consider a small bakery that produces three types of baked goods: bread, cookies, and cakes. Each product requires three stages: mixing (Stage 1), proofing (Stage 2), and baking (Stage 3). The bakery has two mixing machines, a single proofing area, and two ovens. Both proofing and baking stages must follow a no-wait condition, meaning that once proofing for a product is complete, it must immediately proceed to baking. The processing times for each product on the available machines are shown in Table 1.

Table 1. Processing time.

Product	***Mixing***		***Proofing***	***Baking***	
	Mixer 1	***Mixer 2***		***Oven 1***	***Oven 2***
Bread	20	25	30	25	30
Cookies	20	15	20	20	25
Cakes	30	35	40	40	45

One schedule that meets the no-wait constraint is that Bread and Cookies are processed first, then Cakes. Since there are two mixing machines, Bread and Cookies can be processed in parallel, for example, Bread by Mixer 1 and Cookies by Mixer 2. After mixing, Cookies enter the proofing stage at the 15th minute and finish at the 35th minute. Next, Cookies enter the baking stage in Oven 1 at the 35th minute and finish at the 55th minute. Bread enters the proofing stage at the 20th minute and finishes at the 50th minute. Because Oven 1 is still working on Cookies, Bread is processed by Oven 2 for the baking stage and finishes at the 80th minute. Cakes begin to be processed at the 15th

minute by Mixer 2 after finishing Cookies. Next, Cakes enter proofing at the 50th minute and finish at the 90th minute. Finally, Cakes begin to be baked in Oven 1 at the 90th minute and finish at the 130th minute. This work process is visually represented by the Gantt chart in Figure 1. With this work sequence, the makespan is 130 minutes dan ToIT is 170 minutes. This ToIT is obtained by adding the oven idle time of Oven 1 and Oven 2 (the gray area in Figure 1). Oven 1 is idle for 65 minutes, and Oven 2 is idle for 105 minutes.

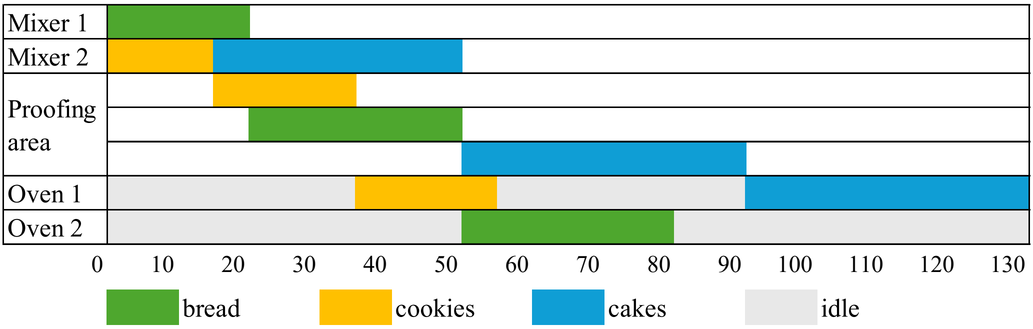

Figure 1. Gantt chart for the case study solution with the sequence of work on the jobs Bread, Cookies, and Cakes.

3.2. Whale Optimization Algorithm

The WOA's algorithm used in this work is given in Figure 2. Two parameters, namely population size (*nPop*) and maximum iteration (*maxIter*), are read at the beginning. A total of *nPop* whales are generated randomly and then the fitness of each whale is calculated. The whale with the best fitness is the temporary best solution. After that, several processes are carried out as many as *maxIter* times. In each iteration, the variables for updating the whale's position are recalculated; the fitnesses are recalculated and the best whale is determined accordingly. The output given at the end is the best whale, which is the solution to the given problem.

Algorithm *Whale Optimization Algorithm*

input *nPop, maxIter.*
Initial population generation.
Fitness calculation, best whale finding.
while *not terminating:*
    Initialize/update the algorithm's variables.
    Position update.
    Fitness calculation.
    Best whale update.
end while
output *best position.*

Figure 2. WOA's algorithm.

Following [15] we model a whale as a list of integers representing the *jobs*. As for the updating position methods, we also refer to the techniques described in [15].

3.3. Fitness

In WOA, a candidate solution is represented by a whale. A fitness function is defined to determine the quality of the solution. Since the focus of this experiment is to determine the effect of population size and maximum iteration on WOA performance based on makespan, ToIT, and the multi-objective makespan and ToIT, the fitness for makespan and ToIT are defined as follows:

$$f_1 = \frac{1}{makespan - 1} \tag{1}$$

$$f_2 = \frac{1}{ToIT - 1} \tag{2}$$

Given two solution candidates *S* with makespan and ToIT (*M*, *T*) and S′ with makespan and ToIT (*M*′, *T*′), the multi-criterion values for *S* and *S*′ can be calculated using the following formulas:

$$R = \frac{(M - min(M, M'))}{min(M, M')} + \frac{(T - min(T, T))}{min(T, T')} \tag{3}$$

$$R' = \frac{(M' - min(M, M'))}{min(M, M')} + \frac{(T' - min(T, T))}{min(T, T')} \tag{4}$$

Using *R* and *R*′ we can compare the quality of *S* and *S*′. If the value of *R*′ is smaller than that of *R*, then the more optimal scheduling sequence is *S*′, and vice versa.

3.4. Computational Experiments

3.4.1. Problem Modelling

For experimental purposes, we model the bakery scheduling problem into a 2-stage No-Wait Flexible Flow Shop Scheduling Problem. Assuming that the proofing area has unlimited capacity, the mixing and proofing stages can be combined into a single stage. This allows products to proceed directly from mixing to proofing without delay, effectively treating both as a continuous process. The no-wait constraint is then applied between the combined mixing-proofing stage and the baking stage. This simplification reduces the complexity of the problem while maintaining its relevance to real-world bakery operations and ensures a focus on optimizing critical transitions.

3.4.2. Dataset

A dataset, a collection of problem instances, is needed in an experiment. Table 2 lists the case designs used. There are 18 cases distinguished by the number of jobs (*nJob*), the number of machines in stage 1 (*nM1*), and the number of machines in stage 2 (*nM2*).

Table 2. The design of cases.

Case	*nJob*	*nM1*	*nM2*
S1	8	2	2
S2	8	3	2
S3	8	3	4
S4	10	2	2
S5	10	3	2
S6	10	3	4
S7	14	2	2
S8	14	3	2
S9	14	3	4

Case	*nJob*	*nM1*	*nM2*
S10	16	2	2
S11	16	3	2
S12	16	3	4
S13	20	2	2
S14	20	3	2
S15	20	3	4
S16	24	2	2
S17	24	3	2
S18	24	3	4

Based on the above design, the dataset is constructed by randomly generating processing times for each job by each machine at each stage following a $U(4, 40)$ distribution.

3.4.3. Parameter Setting

The focus of this experiment is to determine the effect of population size and maximum iteration on WOA performance based on makespan, ToIT, and the combination of makespan and ToIT. For population size, we choose 20, 30, 40, and 50; whereas for maximum iteration, we choose 50, 100, 150, and 200. Thus, for each case, there are 16 combinations of parameter value pairs as shown in Table 2.

Table 3. Parameter combination.

CombID	***C1***	***C2***	***C3***	***C4***	***C5***	***C6***	***C7***	***C8***	***C9***	***C10***	***C11***	***C12***	***C13***	***C14***	***C15***	***C16***
popSize	20	30	40	50	20	30	40	50	20	30	40	50	20	30	40	50
maxIter	50	50	50	50	100	100	100	100	150	150	150	150	200	200	200	200

3.4.4. Scenarios

For each combination, we run the algorithm 10 times for each case. We calculate the average, minimum, and maximum values from the resulting makespan and ToIT. The effect of population size on WOA performance is done by calculating the average makespan and/or ToIT for cases grouped by population size and iteration, as shown in Table 4 and Table 5, respectively.

Table 4. Group by population size.

Size	***Combination Id***
20	C1, C5, C9, C13
30	C2, C6, C10, C14
40	C3, C7, C11, C15
50	C4, C8, C12, C16

Table 5. Group by iteration.

Iteration	***Combination***
50	C1, C2, C3, C4
100	C5, C6, C7, C8
150	C9, C10, C11, C12
200	C13, C14, C15, C16

4. RESULTS AND DISCUSSIONS

We have analysed the following aspects to compare the mono-objective and multi-objective results:

1. Trend analysis for both makespan and ToIT across population sizes and iterations.
2. Performance differences between mono-objective and multi-objective scenarios.

The performance comparison of WOA for mono-objective and multi-objective optimization can be visualized and is given in Figure 3 and Figure 4, respectively.

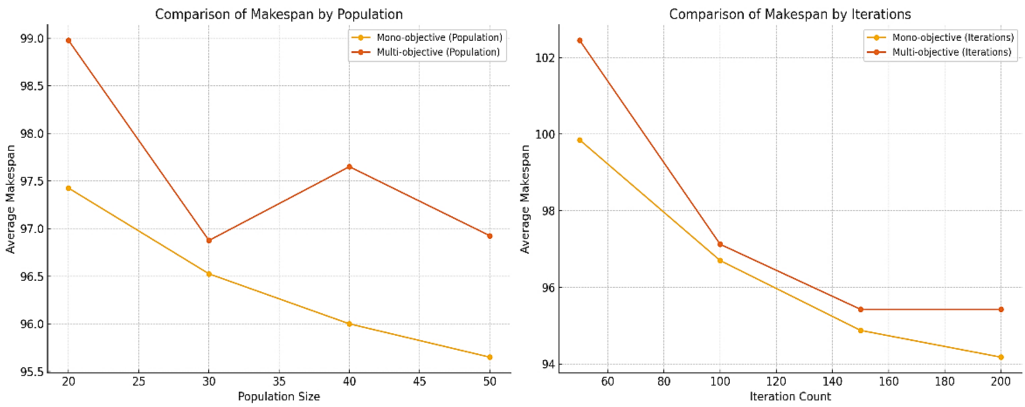

Figure 3. Comparison of makespan by population and by iteration.

Figure 3 illustrates the impact of population size and iteration count on the average makespan for mono-objective and multi-objective optimization strategies. The left graph shows that for both mono-objective and multi-objective optimization, the average makespan decreases as the population size increases. This trend highlights the benefit of larger populations, which provide greater diversity in candidate solutions and enable the algorithm to explore the solution space more effectively. However, the improvement in makespan diminishes as the population size grows beyond 40, suggesting that increasing the population size further yields limited additional benefits. Mono-objective optimization consistently achieves lower makespan values compared to multi-objective optimization, as it focuses solely on makespan minimization without balancing other objectives.

The right graph examines the effect of iteration count on average makespan. Increasing the number of iterations improves performance for both optimization strategies, as evidenced by the reduction in makespan. This improvement is most pronounced at lower iteration counts. However, after 150 iterations, the reduction in makespan slows down, particularly for the multi-objective approach, indicating diminishing returns with additional iterations. Despite this, multi-objective optimization remains competitive while balancing multiple objectives, achieving results close to those of mono-objective optimization at higher iteration counts.

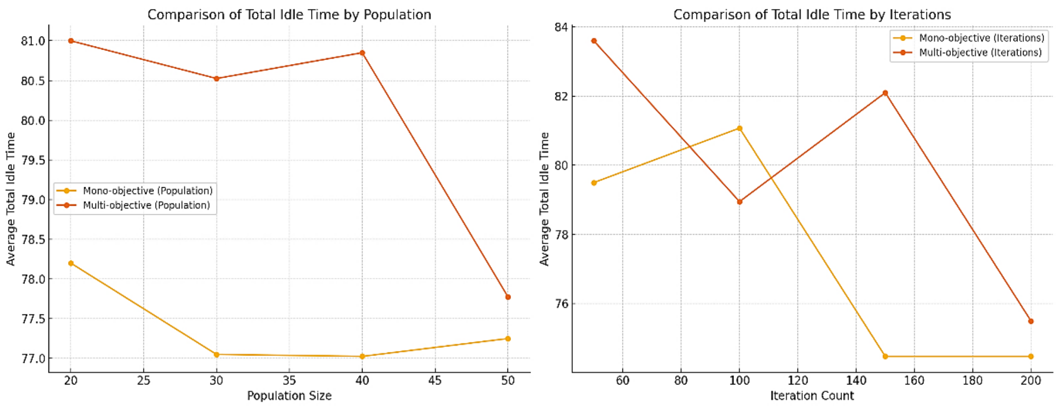

Figure 4. Comparison of total oven idle time by population and by iteration.

Figure 4 illustrates the impact of population size and iteration count on the average ToIT for mono-objective and multi-objective optimization strategies. The left graph shows that for mono-objective optimization, ToIT remains relatively stable across population sizes, with only slight fluctuations. However, for multi-objective optimization, the ToIT values vary significantly. Regarding smaller population sizes, the ToIT is relatively high, which suggests that the limited diversity in candidate solutions hinders WOA's ability to explore the solution space effectively. As the population size increases to 40 and 50, ToIT decreases notably, indicating that a larger population size enhances the algorithm ability to balance the trade-offs between multiple objectives.

The right graph examines the effect of iteration count on ToIT. For mono-objective optimization, ToIT decreases consistently with more iterations, reflecting WOA's ability to refine solutions through continued exploitation of the search space. For multi-objective optimization, the pattern is less consistent. There is a temporary increase in ToIT at 100 iterations, likely due to premature convergence to suboptimal solutions. However, beyond 150 iterations, ToIT decreases significantly, demonstrating that WOA can overcome initial stagnation and achieve improved results with sufficient iterations. At 200 iterations, the ToIT for multi-objective optimization approaches values comparable to those achieved by mono-objective optimization.

Based on both figures' observations, overall, mono-objective optimization demonstrates superior performance in minimizing makespan and consistently achieving lower ToIT due to its singular focus. While showing slightly higher makespan and ToIT values, multi-objective optimization provides a balanced approach that effectively incorporates additional objectives, particularly when configured with larger population sizes and sufficient iteration counts. These results underscore the importance of carefully tuning WOA parameters, such as population size and iteration count, to optimize performance for practical applications that require trade-offs between conflicting objectives.

Additionally, Figure 5 provides a comparative analysis of Makespan (left graph) and Total Idle Time (ToIT) (right graph) for mono-objective and multi-objective optimization strategies across different population sizes.

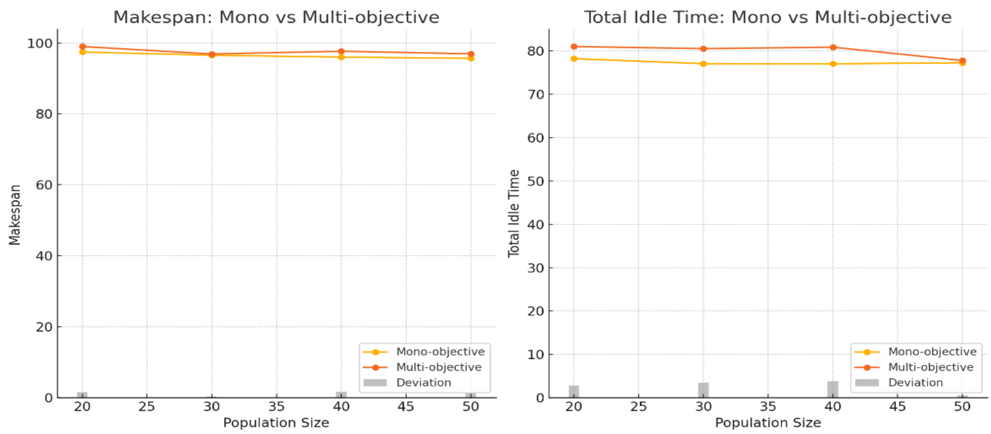

Figure 5. Comparison of the WOA's performance for makespan and total oven idle time between mono-objective and multi-objective settings across different population sizes.

The left graph in Figure 5 shows that the average makespan values for both mono-objective and multi-objective optimization remain relatively stable across all population sizes, with minimal variability as indicated by the deviation bars. Mono-objective optimization achieves slightly lower makespan values due to its singular focus, while multi-objective optimization compromises minimally to balance additional objectives like reducing ToIT. Similarly, the right graph demonstrates that Total Idle Time (ToIT) is consistently lower for mono-objective optimization, whereas multi-objective optimization shows slightly higher values, particularly at smaller population sizes. However, this difference diminishes as the population size increases, with both approaches exhibiting stable performance across runs, as indicated by the low deviation bars.

Figure 6. the Pareto trends for makespan versus ToIT across different iterations.

Lastly, the scatter plot in Figure 6 visualizes the Pareto trends for makespan versus ToIT across different iteration counts (50, 100, 150, 200). Each point represents an optimized solution using the Whale Optimization Algorithm (WOA). The graph highlights how solutions move closer to the lower left corner (representing lower makespan and ToIT) as the iteration count increases. This demonstrates the algorithm's ability to balance conflicting objectives more effectively with additional iterations.

5. CONCLUSION

This study has successfully demonstrated that the Whale Optimization Algorithm (WOA) is an effective approach for solving multi-objective optimization problems in small bakery production scheduling. The hypothesis that the performance of multi-objective WOA does not significantly differ from mono-objective WOA has been validated. The results show that multi-objective WOA achieves competitive makespan values while balancing additional objectives, such as reducing Total Oven Idle Time (ToIT). Stability across various parameter settings further reinforces its robustness.

The performance of WOA highlights its flexibility in handling trade-offs between conflicting objectives. Larger population sizes and sufficient iterations enhance its ability to explore and exploit the solution space effectively. These findings emphasize the importance of careful parameter tuning to achieve optimal performance.

While this study does not include a direct quantitative comparison of WOA with other metaheuristic algorithms, this limitation arises from the lack of existing studies addressing identical scheduling problems under comparable conditions. Future work could address this gap by identifying representative case studies and applying other metaheuristics, such as PSO or GA, to the same problem framework. This would allow a more comprehensive evaluation of WOA's relative performance and robustness.

As a practical extension, this research can be applied to real-world small bakery scheduling problems in Bandung and surrounding areas. By collecting production data from local bakeries, it is possible to build realistic models reflecting their operational conditions. Using these models, WOA can be tested to assess potential time savings and efficiency improvements in daily operations. This approach can provide actionable and context-specific solutions for small businesses in the region.

ACKNOWLEDGEMENT

This research is funded by the Indonesian Ministry of Education, Culture, Research and Technology with the Regular Fundamental Research scheme in 2024 with contract numbers 006/SP2H/RT-MONO/LL4/2024, III/R/2024-06/1591-A-E and III/LPPM/2024-06/105-PE.

REFERENCES

[1] A.P. Utami, A. Kusumawardhani. Application of Supply Chain Management Process on Business Performance (Case Study on Roti-Qu bakery in Batang). *International Journal of Economics, Business and Accounting Research (IJEBAR)*, 2021, Vol-5, Issue-4. DOI: 10.29040/ijebar.v5i4.4185.

[2] N. Xu, Y.-F. Huang, M.-W. Weng, M.-H. Do. New Retailing Problem for an Integrated Food Supply Chain in the Baking Industry. *Applied Sciences*, 2021, 11(3), 946. DOI: https://doi.org/10.3390/app11030946.

[3] T. Keskinturk, MB. Yildirim. A genetic algorithm metaheuristic for bakery distribution vehicle routing problem with load balancing. *International Symposium on Innovations in Intelligent Systems and Applications*, 2021, pp. 287-291. DOI: https://api.semanticscholar.org/CorpusID:9810877.

[4] K. Aljohani. Optimizing the Distribution Network of a Bakery Facility: A Reduced Travelled Distance and Food-Waste Minimization Perspective. *Sustainability*. 2023; 15(4):3654. DOI: https://doi.org/10.3390/su15043654.

[5] A.D. Pop, D.R. Florentin. Analysis of the Distribution of Finished Products in the Bakery Industry in Romania. J*ournal on Emerging Topics in Industrial Engineering*, ISSN: 2960-5407 (Online), 2022. DOI: https://doi.org/10.21428/92f19a8b.6139f655.

[6] A. Trattner, Z. Hansen, L. Hvam. Product Wheels for Scheduling in the Baking Industry: A Case Study. *International Journal of Production Management and Engineering*. 2018. DOI: https://doi.org/10.4995/ijpme.2018.8662.

[7] Blamah, N. et al. A Scheduling Problem in the Baking Industry. *Journal of Applied Mathematics,* Hindawi Publishing Corporation. 2019. https://doi.org/10.1155/2014/964120.

[8] M. Babor, O. Paquet-Durand, R. Kohlus, R. et al. Modeling and optimization of bakery production scheduling to minimize makespan and oven idle time. *Scientific Reports,* 13, article number 235, 2023. DOI: https://doi.org/10.1038/s41598-022-26866-9.

[9] M. Babor, J. Senge, C.M. Rosell, D. Rodrigo, B. Hitzmann. Optimization of No-Wait Flowshop Scheduling Problem in Bakery Production with Modified PSO, NEH and SA. *Processes*, 2021, 9(11), 2044. DOI: https://doi.org/10.3390/pr9112044.

[10] M. Babor, L. Pedersen, U. Kidmose, O. Paquet-Durand, B. Hitzmann. Application of Non-Dominated Sorting Genetic Algorithm (NSGA-II) to Increase the Efficiency of Bakery Production: A Case Study. *Processes*, 2022; 10(8), 1623. DOI: https://doi.org/10.3390/pr10081623.

[11] M. Babor, B. Hitzmann. Application of Nature-Inspired Multi-Objective Optimization Algorithms to Improve the Bakery Production Efficiency. *Engineering Proceedings*. 2022; 19(1):31. DOI: https://doi.org/10.3390/ECP2022-12630.

[12] R. Singh, R.L. Ujjwal. Feature selection using bio-inspired optimization for IoT intrusion detection and prevention system. *International Journal on Information Technologies and Security*, vol. 15, no. 3, 2023, pp. 87-96. DOI: https://doi.org/10.59035/OEXI6498.

[13] A. Nelli, R. Jogdand. Dragonfly soft-computing approach for workload scheduling resource utilization maximization using multi-cloud platform. *International Journal on Information Technologies and Security*, vol. 15, no. 1, 2023, pp. 27-38. DOI: https://doi.org/10.59035/MBUL3714.

[14] S. Mirjalili, A. Lewis. The whale optimization algorithm. *Advances in engineering software*, vol.95, p.51–67, 2016. DOI:https://doi.org/10.1016/j.advengsoft.2016.01.008.

[15] C.E. Nugrahenı, L. Abednego, C.S. Saputra. Whale Optimization Algorithms for Multi-Objective Flowshop Scheduling Problems. *The Eurasia Proceedings of Science Technology Engineering and Mathematics,* 21, 2022, PP. 441-451. DOI: https://doi.org/10.55549/epstem.1226675.

[16] C.E. Nugrahenı, L. Abednego, C.S. Saputra. Hybridization of Whale Optimization Algorithm for Multi-objective Flow Shop Scheduling Problems. 2023 Eighth International Conference on Informatics and Computing (ICIC), Manado, Indonesia, 2023, pp. 1-6. DOI: https://10.1109/ICIC60109.2023.10381910.

Information about the authors:

Cecilia E. Nugrahenı – Cecilia E. Nugraheni received her bachelor’s degree (1993) and master degree (1995) from Dept. of Informatic Engineering, Bandung Institute of Technology (ITB),Bandung, Indonesia. She has received PhD Degree (2004) from Dept. of Informatics, Ludwig Maximilians Universität, Munich, Germany. Her research interest includes formal methods, intelligent systems, machine learning, meta-heuristic and hyperheuristic techniques.

Luciana Abednego – received her bachelor’s degree from Dept. of Informatics, Parahyangan Catholic University, Bandung, Indonesia. She has done her Master in Informatics from Bandung Institut of Technology, Bandung, Indonesia. Currenty, she is working as a lecturer at the Dept. of Informatics, Parahyangan Catholic University. Her research interest includes machine learning and intelligent systems.

Maria Widyarini – received her PhD from School of Business Management, Bandung Institute of Technology (ITB). She was involved in SME particularly as trainer and researcher in microfinancing issues. Her specializing is in Microfinance for SME. She was Director of Center of Excellence – Small Medium Enterprise and Development (COE SMED) and Head of BA and MBA Dept in Parahyangan Catholic University (2018 - 2022).

Manuscript received on 27 December 2024