Multi-Site Aggregate Production Planning Using Particle Swarm Optimization

Aji Prasetya Wibawa[1], Wayan F. Mahmudy[2], Agung Mustika Rizki[2], Gusti Eka Yuliastuti[2], and Ishardita Pambudi Tama[3]

[1]Lecturer, Electrical Engineering Department, Universitas Negeri Malang, Malang, Indonesia. Email: aji.prasetya.ft@um.ac.id (corresponding author).
[2]Lecturer, Computer Science, Faculty of Computer Science, Brawijaya University, Malang, Indonesia.
[3]Lecturer, Industrial Engineering, Faculty of Engineering, Brawijaya University, Malang, Indonesia.

_____

Abstract: Aggregate planning is a crucial stage in the production process because it supports other processes. Careless production planning may cause production costs to spike sharply that hurts the company financially. This study explores the novel usage of particle swarm optimization (PSO) to discover a set of solutions among the objective of a multi-optimization problem in aggregate production planning. The study uses a small home textile industry with complex production processes of school uniforms as a case study. The results show that the production cost difference between actual data and the proposed method is IDR330,670,000. Thus, PSO can solve the multi-site aggregate planning by reducing the company production cost.

Keywords: Multi-site aggregate, particle swarm optimization (PSO), production planning, optimization

_____

1. Introduction

Managing the amount of production is essential in the production process. A slight mistake may disturb the supply to the customer and reduce the profits of the company. Some companies produce a product based on customer demand, while customer demand is not constant. The company has to determine the exact amount of production with minimal production costs. The problem will worsen if the company has several branches in some places that require proper production planning for each branch.

Production planning at the company requires a limited-time routine planning, considered aggregate production planning (Fahimnia et al., 2012). Aggregate planning is a medium-term capacity planning that typically covers 6 to 18 months (Chauhan et al., 2017). This planning is beneficial for companies with volatile demand. The purpose of aggregate planning is to create a production plan that effectively uses organizational resources to meet the expected demand. The plan adjust production rates, labor levels, inventory levels, overtime work, subcontracting rates, and other controllable variables to minimize cost over the planning period (Stevenson and Hojati, 2007). The planning process focuses on one or two individual products and will classify all products and handle them as a single product. In general, aggregate planning is related to the budget process. Therefore, this planning is crucial because it can help streamline flow along the supply chain (Pal et al., 2011). This planning affects cost, equipment usage, job level, and customer satisfaction. Aggregate planning involves the forecasting process therein (Chauhan et al., 2017). Forecasting is needed to get the results of demand predictions from consumers. This prediction will be used as a reference for the production process.

Several strategies for performing aggregate planning are capacity choice, demand choice, and mixing between the two previous options (Heizer and Render, 1993). A company uses the capacity choice by changing production levels, inventory levels, number of workers, subcontract workers, and overtime. Another option, the fundamental thing a company can do is to change the production rate based on the influence of demand, delaying orders during periods of high demand and also against seasonal trends. In this study, the authors use mixed choices so that the things to be considered include: consumer demand based on predicted results, production levels, inventory levels, number of labor, working time as well as recruitment and dismissal of workers. The number of production problems can be solved using various methods such as mathematical

and the heuristic method. Several methods can be applied in determining the number of production, among others: ant colony algorithm (Qamhan et al., 2020; Wu et al., 2012; Zhang et al., 2020) and PSO (Chakrabortty et al., 2015). Fuzzy linear programming (Vasant et al., 2004), Fuzzy TOPSIS and goal programming (Khemiri et al., 2017), variable neighborhood search (Almada-Lobo et al., 2008), simulated annealing (Baxendale et al., 2021; Tung et al., 2016), and genetic algorithm (Fahimnia et al., 2012; Ramezanian et al., 2012; Yuliastuti et al., 2019). Every method is efctive in solving such problems without considering the multi-site situation.

Multi-site production is the focus of this paper. We will apply the PSO method to solve aggregate production planning in companies with several branches. PSO algorithm is a simple concept that is easy to use, robust to control parameters, and computationally efficient compared with other mathematical methods (support vector machine and differential dynamic programming) and conventional evolutionary optimization algorithms such as simulated annealing, differential evolution, genetic algorithm, and shark algorithm (Jahandideh-Tehrani et al., 2020). The PSO implementation aims to reduce the production cost of a multi-site company.

2. Literature Reviews

There have been several studies associated with aggregate production planning. Almada-Lobo et al. (2008) solved production planning and scheduling problems in the glass container industry. The authors formulated a mathematical model of the problem to perform a computation process. They mentioned that the implementation of variable neighborhood search (VNS) successfully solved the case study problems. The combination of VNS with another technique (i.e., PSO) outperformed the single VNS (Zheng et al., 2020). Fahimnia et al. (2012) solved aggregate production planning and its distribution and optimized the result using a Genetic Algorithm. The authors constructed non-linear formulas to represent the production process and distribution and all the variables and constraints in this study. The Genetic Algorithm can be relied upon to solve the problem of mathematical model optimization and also problems with many variables. Wu et al. (2012) wrote that forecasting customer demand is significant and influential to the production process. Their study discussed the balance of demand and production and proposed a mathematical model with a dual purpose. Authors solved production scheduling problems using the Ant Colony Algorithm, which effectively solves the problem.

Qamhan et al. (2020) solved single-machine scheduling using the ant colony optimization (ACO) algorithm. The findings demonstrated that the suggested ACO was capable of obtaining precise answers in reasonable CPU time. The findings then revealed that the ACO beats Moore's method in all situations examined. It is possible to infer that the created ACO is exceptionally efficient and effective in tackling the problem addressed in this research. Chakrabortty et al. (2015) wrote that aggregate production planning is one of the demanding tasks. The task included the general but more complex production scheduling. Multi-period and multi-product

were used to formulate the mathematical model. The authors solved the problem using a possibilistic environment-based PSO.

Another research aims to look at production scheduling for rotomoulded plastics manufacturing in a multi-machine setting (Baxendale et al., 2021). The goal is to keep total delay to a minimum. The issue is similar to hybrid flow shop scheduling with batching in that additional limitations are required to govern which machines may be used at each step. The simulated annealing was evaluated using randomly generated issue instances. In general, the proposed algorithm produced the best outcomes in terms of solution quality. Pal et al. (2011) solved the problem of determining aggregate production by applying swarm intelligence. According to the authors, the application of swarm intelligence is perfect in scheduling issues, route planning, production planning, and other industrial decision-making. This swarm intelligence implementation technique has advantages that are easy in representing solutions and also few control parameters. Examples of popular swarm intelligence are artificial bee colony and PSO, that applied by authors. The experiment of swarm intelligence generated a meager production cost. The experiment showed that PSO is better in terms of quality and accuracy than fuzzy and genetic algorithms.

3. Method

In this study, the authors focus on the multi-site school uniform company with several regional branches of production in various cities in East Java, Indonesia. The problem faced by this company is that the fluctuating consumer demand and production capacity difference in each region. Fig. 1 illustrates the multi-site situation with Regional 1 as the central office and Regional 2 to 6 as the branch. Regional 1 may accept the order and distribute t to the branches. In other words, Regional 1 has to control all branches' production cycles. The problem is that the branch must follow the company standard and customer needs in quality and quantity. The production process requires a flexible manufacturing system with at least two neighboring working zones, the results of which include flexible non-linear process plans and assembly activities (Zhang et al., 2020). Thus, these non-trivial problems should be solved effectively.

The authors will predict consumer demand in the next period for 12 months using simple linear regression. The prediction results will be a reference in determining its production capacity using PSO. In addition, we consider various existing obstacles and applied rules. The rules in each region are different. Consequently, the standard production process in one region will not be the same as the other branches.

The problem of aggregate production planning uses in every company could be varied. The optimization in this study is limited in the use of the following parameters.
- Number of workers
- Average production amount per worker in 1 day on regular time (*rt*)
- Average production amount per worker in 1 day on overtime (*ot*)

- Average production amount per worker in 1 day on subcontract (st)
- Working hours in 1 day (regular time)
- Maximum number of overtime hours (overtime)
- Maximum number of subcontract workers
- Production costs on regular time (vrt)
- Production costs on overtime (vot)
- Production costs on subcontract (vst)
- The cost of hiring workers (vht)
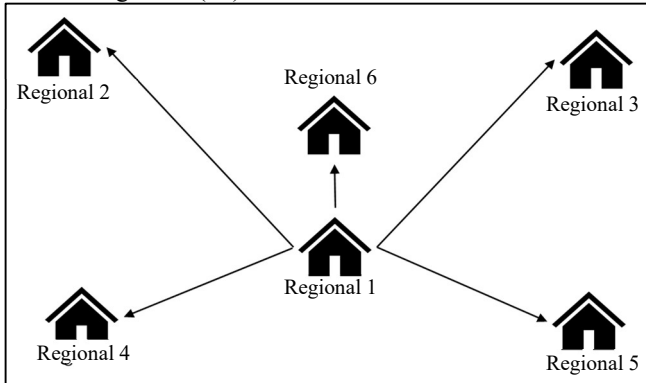- The cost of firing workers (vft)
- Storage cost (vit)

Fig. 1. Multi-site company illustration

3.1 Particle Swarm Optimization (PSO)

PSO is an optimization method based on swarm intelligence or more popularly called an evolution-based procedure. The population of the particle is known as an optimal solution for PSO (Alfarisy et al., 2018). There are two kinds of calculation in the iterative procedure: local optimum and global optimum. The local optimum is the best position the particle has ever achieved by comparing it with the preceding particles (Long et al., 2018), whereas the global optimum is one of the best positions of particles compared to the whole particle (Huang et al., 2021). There are several composing factors in the PSO algorithm (Alfarisy et al., 2018), detailed as follows.

- **Swarm** is the number of particles in the population on PSO. Swarm size depends on how complex the problem will be solved. In general, swarm sizes on PSOs tend to be smaller when compared to other evolutionary algorithms in searching for optimal solutions.
- **Particle** is an individual in a swarm that represents a problem-solving solution. Each particle has the position and velocity determined by the representation of the solution at that time.
- **Personal Best (pBest)** is the best position the particle has ever achieved by comparing the fitness to the present particle position with the previous one. Personal best prepared to get the best solution.
- **Global Best (gBest)** is the best position of the particles obtained by comparing the best fitness value of all the particles in the swarm.
- **Velocity (v)** is a vector that determines the direction of particle position displacement. Changes in velocity are performed each iteration to improve the position of the original particles.
- **Inertial weight (w)** is used to control the impact of changes in velocity given by particles.
- **Acceleration Coefficient** is the controlling factor to what extent the particles move in one iteration. In general, the acceleration coefficient values c_1 and c_2 are the same ie in the range 0 to 4. However, the value can be determined freely in the study.

Essential components in the PSO are particle position and particle velocity. The calculation of both components is shown in Eq. 1 and Eq. 2.

Velocity update:

$$v_{i,j}^{t+1} = w \cdot v_{i,j}^t + c_1 \cdot r_1 \left(pBest_{i,j}^t - x_{i,j}^t \right) + c_2 \cdot r_2 \left(gBest_{g,j}^t - x_{i,j}^t \right) \tag{1}$$

Position update:

$$x_{i,j}^{t+1} = x_{i,j}^t + x_{i,j}^{t+1} \tag{2}$$

Fig. 2 shows the procedure of PSO in the form of pseudocode for better understanding.

```
begin
        t = 0
        initialization of particle position (x_{i,j}^t), velocity (v_{i,j}^t), pBest_{i,j}^t, gBest_{g,j}^t
        calculate fitness value of each particle

        do
                t = t + 1
                update velocity v_{i,j}^t
                update particle position x_{i,j}^t
                calculate fitness value of each particle
                update pBest_{i,j}^t and gBest_{g,j}^t

        while (not a stop condition)
end
```

Fig. 2. Pseudocode of PSO

Table 1. Predicted data

Aug-16	Sep-16	Oct-16	Nov-16	Dec-16	Jan-17	Feb-17	Mar-17	Apr-17	May-17	Jun-17	Jul-17
1629	942	831	1028	826	1273	692	1124	2114	2483	2685	7482

Assessing the quality of a solution requires the existence of a calculation, or later will be called a fitness function (Alfarisy et al., 2018). The fitness function on each problem is different, as formulated in Eq. 3. Here, a penalty is a unit calculated on any error or rule violation. All sorts of things that can degrade the quality of the solution or have a negative impact are also included in the penalty category.

$$Cost = \sum rt * vrt + \sum ot * vot + \sum st * vst + \sum ht * vht + \sum ft * vft + \sum it + vit + Penalty \qquad (3)$$

In the following example, the authors solve the problem in one region where the production process is in other regions. In general, there is only a slight difference in the standard parameters in the production process of each region. Using simple linear regression, the authors get the data predicted consumer demand for the next period, as shown in Table 1. Then some parameters need to be determined according to the conditions at the company. Here is an example of initial parameter values, as shown in Table 2.

Table 2. Example of parameters value

Parameters	Value	Units
Number of workers	6	persons
rt	14	pcs/day
ot	5	pcs/day
st	14	pcs/day
Working hours in 1 day (regular time)	8	h/day
Max. number of overtime hours	3	h/day
Max. number of subcontracts	3	h/day
vrt	23,000	IDR
vot	25,000	IDR
vst	24,000	IDR
vht	2,700,000	IDR
vft	500,000	IDR
vit	500 ,-	IDR

The data sample then proceeds to the aggregate production planning process with the production value of random value-generating results, as shown in Table 3. There is a minus value on inventory that means that demand is great, but has less supply, so more production is needed. Minus value is later calculated as a penalty as it can harm the company for each minus value multiplied by IDR30,000 as a reference in calculating the amount of loss of the company. The result is the fitness function (Eq. 3) that is equal to IDR 593,196,000.

In this study, integer coding is used as a representation of the initial solution of particles. The choice of representation type of solution is chosen because it can better represent the true solution value of production capacity in aggregate production planning. Some other factors are also considered in aggregate production planning, such as regular time production costs (*rt*),

overtime production costs (*ot*), subcontracting production costs (*st*), labor recruitment costs (*ht*), cost of dismissal (*ft*), and the amount of inventory (*it*). An illustration of the representation of a solution is shown in Fig 3. In this study, there are six regions. Each region has 12 particles that show the production amount in the next 12 months. In other words, 72 particles act as the representation of the overall solution.

3.2 Testing Scenario

There are many tests to obtain the best parameter values in the PSO, including particle size testing, iterative value testing, inertial weight testing, and acceleration coefficient testing. Testing the number of particles aims to obtain the optimal particle value values to be used on the hybridization mechanism. With the corresponding number of particles, the computation time can be reduced, but the results are also satisfactory. While testing, the iteration value is done to determine the best iteration value so that computation time can be suppressed but does not degrade the quality of the result given. Inertial weight testing needs to be done so that particle velocity changes can be controlled. For the last test, the acceleration coefficient is done to find the best combination coefficient acceleration value. It is intended to set the rate of displacement of particles in 1 iteration.

The PSO testing will run five times to get the average value for more fitting and representative results. PSO is a stochastic algorithm which in each execution, may produce different values. For testing purposes, the authors determine the initial PSO's parameter as follows.

- Number of iterations = 200
- Initial velocity value (v) = 0
- Weight value of inertia (w) = 0.5
- Acceleration coefficient value c_1 = 1 and c_2 = 1

4. Experimental Results

4.1 Particle Size Testing

The more particles used will be directly proportional to the resulting solution, or the results will get better. However, it is indeed inversely proportional to the computation time. The more particles used, the longer the time it takes to execute.

Therefore, the determination of the number of these particles must be exact. Fig. 4 shows particle size testing results. The patterns produced at the time of particle size testing are very volatile but tend to decrease. The pattern reaches the highest cost, where the number of particles is 5 to 15. It is the worse condition. The results get better when the number of particles is 20 and 25. In the number of particles 30 to 50, the graph pattern moves but tends to be stable. The best result is at the time of particles of 25, considering the good results and the computation time.

Table 3. Aggregate production planning

| Period	Demand (pcs)	Working Day	Production (pcs)	rt (pcs/day)	ot (pcs/day)	st (pcs/day)	vht (IDR)	vft (IDR)	vit (IDR)
Aug-16	1629	21	1200	1200					-429
Sep-16	942	20	1250	1250					308
Oct-16	831	21	1500	1500					977
Nov-16	1028	21	1900	1764	136				1849
Dec-16	826	21	2380	1764	600	16			3043
Jan-17	1273	20	1640	1640					3410
Feb-17	692	22	2800	1848	600	352			5518
Mar-17	1124	20	2250	1680	570				6644
Apr-17	2114	20	2500	1680	600	220			6644
May-17	2483	17	1780	1428	352				5941
Jun-17	2685	21	1830	1764	66				5086
Jul-17	7482	22	1450	1450					-956
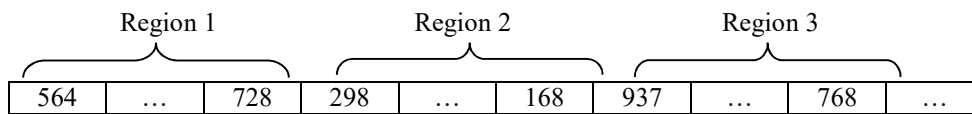
Fig. 1. Representation of the solution
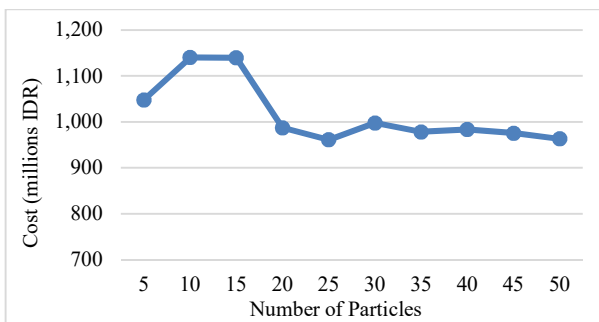
Fig. 2. The result of particles size testing

Iterative Value Testing

After doing particle size testing, the best value obtained is 25. Then the value is used to be a parameter for iterative value testing. The purpose of this test is to obtain an optimal iteration value. The results obtained are shown in Fig. 5.
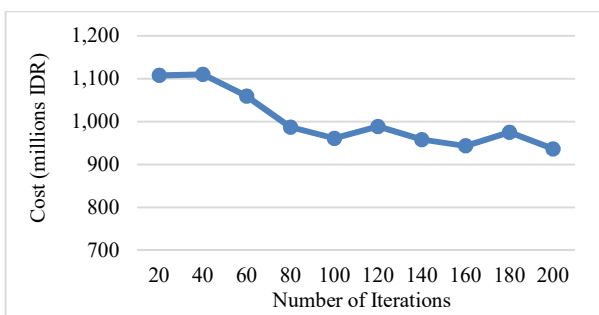

Fig. 3. The result of iterative value testing

Based on the graph, the results obtained are pretty volatile. However, there is a tendency for more and more iterations done, and then the results are also getting better. The authors decided that the best result is at the 160 iterations even though the 200 is slightly better. It is 160 iterations consumes less computational time than the 200 iterations

Inertial Weight Testing

The weight of this inertia test on how well its value is in controlling the change in velocity because a change of uncontrolled speed can cause the particles to move irregularly. If the displacement of particles is irregular, it will have difficulty reaching the optimal point. The inertial weight allows the particles to move along the path to reach the target point more accurately and efficiently. The high value of inertial weight will focus on global search (global exploration), whereas the low value is more directed to local search. For a less focused search on one part and exploring new search areas within a specific dimension space, it is necessary to obtain an optimal inertia weight value to maintain global and local exploration levels. The inertial weight testing results obtained are shown in Fig. 6.
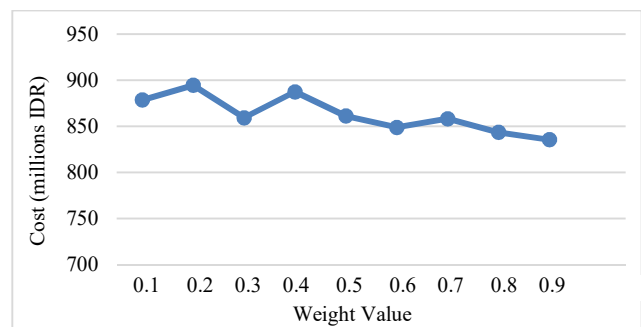

Fig. 4. Inertial weight testing result

Inertial weight test results show that the higher the value of weighting, the better results obtained. Although a value of 0.9 shows the best results though, it is not directly used as a reference to decide that value is suitable as the weight of inertia. Considering the level of exploration as well as the balance of local and global searches feeds the 0.6 value chosen by the author as the most optimal value as the weight of inertia. A value of 0.6 is considered capable of exploring new search areas while maintaining quality on local searches.

Acceleration Coefficient Testing

The previous test has obtained some best parameter values: the number of particles value of 25, the iteration value of 160, and the inertia weight value of 0.6. The last parameter tested is a combination of two acceleration coefficients values (c_1 and c_2) that will affect the displacement of particles at each iteration. The c_1 value will affect the local search, while the c_2 value affects the global search—the results of the acceleration coefficient test as shown in Fig. 7.
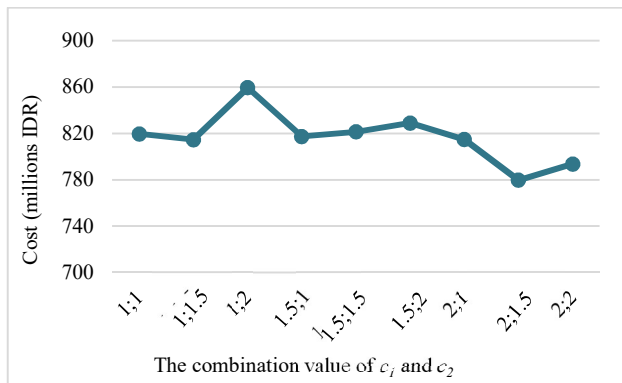

Fig. 5. Acceleration coefficient testing

Based on the graphs generated, the test coefficient values show varying results. This is due to the acceleration coefficient values that affect the displacement of each particle. The particles move freely, which allows the desired result to be missed at a point. Based on the graph, the combination of the best acceleration coefficient value is c_1 of 2 and c_2 of 1.5. Thus all the best parameter values have been obtained to be then implemented into the system to be applied. The authors compared the results of the PSO method implementation with the consumer demand and the actual production capacity to find out whether the method is effective enough in solving the aggregate production planning problem, as shown in Fig. 8.

Fig. 8 shows that consumer demand is very volatile, and the company is always trying to determine the production capacity far exceeds the demand of consumers in each month. This indeed leads to a waste of production costs for the company. While the results with the application of PSO, it appears that more stable, not always exceed consumer demand every month but overall fulfilled with fewer production costs. The cost incurred for the 12 months amounted to 1495374500, while the cost that needs to be spent using the PSO only amounted to 1164704500. The resulting difference significantly affects the profits earned by the company. The results of this trial show that PSO effectively solves the problems of aggregate production planning to reduce production costs and optimize corporate profits.

6. Conclusions and Future Work

The implementation of PSO to solve the problem of aggregate production planning by using all the best parameter values proved optimal. Costs generated by implementing PSO are much lower than the original data on aggregate production planning results on a trial-and-error basis. It is undoubtedly very crucial for the company. Because with the implementation of PSO, production costs are to be spent far less than before, so the company can maximize profits. PSO is a metaheuristic, can investigate a wide range of viable solutions. Furthermore, PSO has no gradient in the optimized issue, which means, as with conventional optimization methods such as gradient descent and quasi-newton methods, that PSO needs no differentiation in the optimization problem. On the other hand, metaheuristics like PSO do not provide the ideal answer, indicates uncertainty.

It is related to the fact that production planning systems may not detect or acknowledge uncertainty. Typically, plans are developed because the forecast is faultless and production and supply procedures perfect. However, in uncertain environments, such systems are being deployed. Therefore coping methods must be developed by the planning organization. The authors believe that the modifications to the PSO will produce better solutions for such problems. In the future, the authors will conduct further research related to aggregate production planning by applying modified PSO or combining PSO with other methods, the hybridization method.
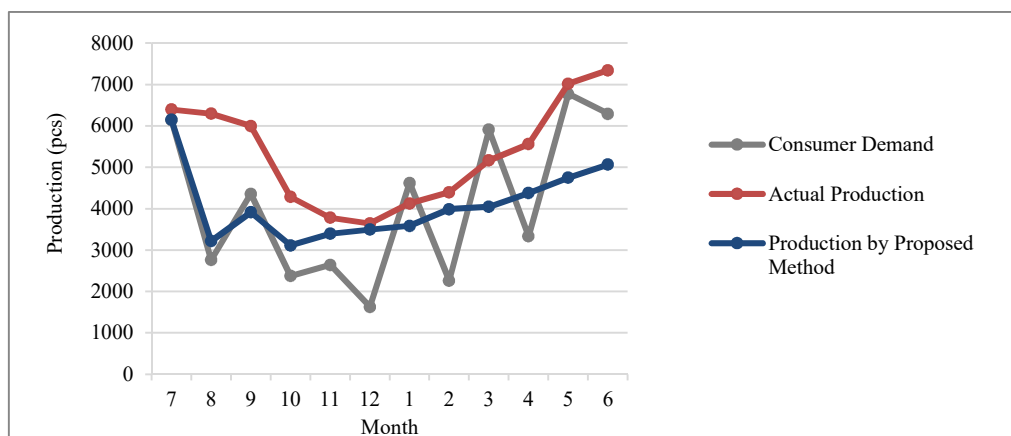

Fig. 6. Comparison of results

References

Alfarisy, G. A. F., Mahmudy, W. F., and Natsir, M. H. (2018). Optimizing laying hen diet using particle swarm optimization with two swarms. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, 10(1–6), 113–119.

Almada-Lobo, B., Oliveira, J. F., and Carravilla, M. A. (2008). Production planning and scheduling in the glass container industry: A VNS approach. *International Journal of Production Economics*, 114(1), 363–375. https://doi.org/https://doi.org/10.1016/j.ijpe.2007.02.052

Baxendale, M., McGree, J. M., Bellette, A., and Corry, P. (2021). Machine-based production scheduling for rotomoulded plastics manufacturing. *International Journal of Production Research*, 59(5), 1301–1318. https://doi.org/10.1080/00207543.2020.1727046

Chakrabortty, R. K., Akhtar Hasin, M. A., Sarker, R. A., and Essam, D. L. (2015). A possibilistic environment based particle swarm optimization for aggregate production planning. *Computers and Industrial Engineering*, 88, 366–377. https://doi.org/10.1016/j.cie.2015.07.021

Chauhan, Y., Aggarwal, V., and Kumar, P. (2017). Application of FMOMILP for aggregate production planning: A case of multi-product and multi-period production model. *2017 International Conference on Advances in Mechanical, Industrial, Automation and Management Systems (AMIAMS)*, 266–271. https://doi.org/10.1109/AMIAMS.2017.8069222

Fahimnia, B., Luong, L., and Marian, R. (2012). Genetic algorithm optimisation of an integrated aggregate production–distribution plan in supply chains. *International Journal of Production Research*, 50(1), 81–96. https://doi.org/10.1080/00207543.2011.571447

Heizer, J. and Render, B. (1993). Production and Operation Management: Strategic and Tactic Decision. *Practice Hall*. New Jersey.

Huang, W., Yang, K., Zhu, Y., and Lu, S. (2021). Data-driven parameter tuning for rational feedforward controller: Achieving optimal estimation via instrumental variable. *IET Control Theory and Applications*, 15(7), 937–948. https://doi.org/10.1049/cth2.12093

Jahandideh-Tehrani, M., Bozorg-Haddad, O., and Loáiciga, H. A. (2020). Application of particle swarm optimization to water management: an introduction and overview. *Environmental Monitoring and Assessment*, 192(5), 281. https://doi.org/10.1007/s10661-020-8228-z

Khemiri, R., Elbedoui-Maktouf, K., Grabot, B., and Zouari, B. (2017). Integrating fuzzy TOPSIS and goal programming for multiple objective integrated procurement-production planning. *The 2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*.

Long, K., Wang, X., and Gu, X. (2018). Local optimum in multi-material topology optimization and solution by reciprocal variables. *Structural and Multidisciplinary Optimization*, 57(3), 1283–1295. https://doi.org/10.1007/s00158-017-1811-4

Pal, A., Chan, F. T. S., Mahanty, B., and Tiwari, M. K. (2011). Aggregate procurement, production, and shipment planning decision problem for a three-echelon supply chain using swarm-based heuristics. *International Journal of Production Research*, 49(10), 2873–2905. https://doi.org/10.1080/00207541003730847

Qamhan, A. A., Ahmed, A., Al-Harkan, I. M., Badwelan, A., Al-Samhan, A. M., and Hidri, L. (2020). An exact method and ant colony optimization for single machine scheduling problem with time window periodic maintenance. *IEEE Access*, 8, 44836–44845. https://doi.org/10.1109/ACCESS.2020.2977234

Ramezanian, R., Rahmani, D., and Barzinpour, F. (2012). An aggregate production planning model for two phase production systems: Solving with genetic algorithm and tabu search. *Expert Systems with Applications*, 39(1), 1256–1263.

Stevenson, W. J. and Hojati, M. (2007). Operations Management. *McMcGraw-Hill Irwin*. Boston.

Tung, K., Chen, C., and Hung, Y. (2016). Solving cutting scheduling problem by simulated annealing search method. *2016 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*.

Vasant, P., Nagarajan, R., and Yaacob, S. (2004). Decision making in industrial production planning using fuzzy linear programming. *IMA Journal of Management Mathematics*, 15(1), 53–65.

Wu, Z., Zhang, C., and Zhu, X. (2012). An ant colony algorithm for master production scheduling optimization. *Proceedings of the 2012 IEEE 16th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, 775–779. https://doi.org/10.1109/CSCWD.2012.6221908

Yuliastuti, G. E., Rizki, A. M., Mahmudy, W. F., and Tama, I. P. (2019). Optimization of multi-product aggregate production planning using hybrid simulated annealing and adaptive genetic algorithm. *International Journal of Advanced Computer Science and Applications*, 10(11), 484–489. https://doi.org/10.14569/IJACSA.2019.0101167

Zhang, S., Li, X., Zhang, B., and Wang, S. (2020). Multi-objective optimisation in flexible assembly job shop scheduling using a distributed ant colony system. *European Journal of Operational Research*, 283(2), 441–460. https://doi.org/10.1016/j.ejor.2019.11.016

Zheng, Q., Wang, W., and Gao, B. (2020). A dynamic scheduling algorithm for flexible production workshop based on fitness landscape theory. *Academic Journal of Manufacturing Engineering*, 18(4), 9–14.

Aji Prasetya Wibawa is a head of electrical engineering department in Universitas Negeri Malang (UM), Indonesia. He has a Ph.D. in electrical and information engineering from the University of South Australia. He leads a research group focused on knowledge engineering and data science. His research interest is AI, data mining, machine translation, and social informatics.

Wayan Firdaus Mahmudy obtained a bachelor degree in mathematics from Universitas Brawijaya, Indonesia, a master degree in information technology from Institut Teknologi Sepuluh November (ITS), Indonesia, and completed his Ph.D. in Manufacturing Engineering at the University of South Australia. He is a Lecturer at the Department of Computer Science, Universitas Brawijaya. His research interests include optimization of combinatorial problems and machine learning.

Agung Mustika Rizki is a lecturer in the Informatics Engineering at Universitas Pembangunan Nasional "Veteran" Jawa Timur Surabaya. He obtained a bachelor's degree in Informatics Engineering in 2016 and a master's degree in Computer Science in 2018 from Universitas Brawijaya, Malang, Indonesia. He has a research interest in Evolutionary Algorithm, Applied Artificial, Decision Support Sytems.

Gusti Eka Yuliastuti is a lecturer in the Informatics Engineering at Institut Teknologi Adhi Tama Surabaya. She obtained a bachelor degree in Informatics Engineering in 2016 and a master degree in Computer Science in 2018 from Universitas Brawijaya, Malang, Indonesia. Her research interest on Data Mining, Evolutionary Algorithm, Applied Artificial.

Ishardita Pambudi Tama is a lecturer in the Faculty of Engineering Universitas Brawijaya, Malang, Indonesia. He holds a Bachelor of Mechanical Engineering, Master of Industrial Technology, and Ph.D. of Manufacturing Management. He is an experienced lecturer and researcher. His research interests include Manufacturing System, SimulationProduct Development, Ergonomics Industrial Psychology.