SEAN INSTITUTE
Sharing Knowledge
http://infor.seaninstitute.org/index.php/infokum/index
JURNAL INFOKUM, Volume 9, No. 2,Juni 2021 ISSN : 2302-9706

COMBINATION OF ACO AND PSO TO MINIMIZE MAKESPAN IN ORDERED FLOWSHOP SCHEDULING PROBLEMS

[1]Sastra Wandi Nduru, [2]Ronsen Purba, [3]Andri
Master of Information Technology, STMIK Mikroskil, Medan, Sumatera Utara
e-mail: [1]sastrawandi2014@gmail.com, [2]ronsen@mikroskil.ac.id, [3]andri@mikroskil.ac.id

Article Info
Received 20 March 2021
Revised 20 April 2021
Accepted 10 June 2021

Abstract

The problem of scheduling flowshop production is one of the most versatile problems and is often encountered in many industries. Effective scheduling is important because it has a significant impact on reducing costs and increasing productivity. However, solving the ordered flowshop scheduling problem with the aim of minimizing makespan requires a difficult computation known as NP-hard. This research will contribute to the application of combination ACO and PSO to minimize makespan in the ordered flowshop scheduling problem. The performance of the proposed scheduling algorithm is evaluated by testing the data set of 600 ordered flowshop scheduling problems with various combinations of job and machine size combinations. The test results show that the ACO-PSO algorithm is able to provide a better scheduling solution for the scheduling group with small dimensions, namely 76 instances from a total of 600 inctances and is not good at obtaining makespan in the scheduling group with large dimensions. The ACO-PSO algorithm uses execution time which increases as the dimension size (multiple jobs and many machines) increases in a scheduled instance.

Keywords : *Ordered Flowshop Scheduling, Makepan Minimalization, ACO-PSO*

1. Introduction

The flowshop production scheduling problem is one of the most versatile and frequently encountered problems in many industries. In ordered flowshop scheduling, the problem is characterized by the following two conditions: (1) if the processing time of a job is less than that of another job on some machines, it must occur on all machines, and (2) if the processing time of a job on that machine is less than on other machines, then it should be the case for all jobs (Khatami et al., 2019). The problem in this scheduling in general is to minimize the total completion time of the entire job or called makespan (Fernandez-Viagas et al., 2017); (Allahverdi et al., 2018); (Assia et al., 2020).

Effective scheduling is important because it has a significant impact on reducing costs and increasing productivity (Gupta et al., 2020). If proper and effective scheduling is not carried out, it can cause idle time on the machine and hamper productivity so that which can cause an increase in product prices (Hossain et al., 2014). However, solving an ordered flowshop scheduling problem to minimize makespan requires computationally difficult and is known as NP-hard (Khatami et al., 2019).

Various heuristic algorithms have also been used by researchers, including Widyawati (2018) who has applied Ant Colony Optimization (ACO) intending to propose the best schedule that gives the smallest makespan in scheduling problems. Particle Swarm Optimization (PSO) is also a solution that is often chosen to solve scheduling problems (Muharni et al., 2019). However, in solving combinatorial problems, namely in the process of finding the best solution, the resulting solution may be trapped in local optimum conditions. To overcome this problem, this study proposes a makespan optimization solution approach to an ordered flowshop scheduling problem that combines the ACO and PSO algorithms. The superiority of the hybrid ACO and PSO has been demonstrated by Gao et al. (2019) who studied a hybrid method by combining ACO and PSO to solve the problem of efficient mobile path scheduling for mobile agent nodes in wireless sensor networks.

2. Method

Production scheduling is one of the most important activities of the company at the operational level in order to remain competitive in winning the consumer market while optimizing its supply chain (Habibi, 2017). Scheduling classification according to Pinedo (2016) consists of single machine scheduling and parallel scheduling. Parallel scheduling is divided into 6, namely:

a. Scheduling n jobs with identical parallel machines.
b. Scheduling n jobs with non-identical parallel machines where each machine has the same function with a different process.
c. Scheduling n jobs with unrelated parallel engine development from non-identical parallels. There are m parallel machines, machine i to process job j so that the machine speed becomes vij.
d. Scheduling flowshop and flexible flowshop there are m machines arranged in series where each job must be processed on each machine. After the job is done on the first machine, it will be continued on the next machine and so on.
e. Jobshop scheduling and flexible jobshops have m machines where each job has a production flow that must be followed.
f. Openshop scheduling where each job must be reprocessed for each machine.

Ordered flowshop scheduling problems are a subcategory of classic flowshop scheduling problems, in that they have a structured property of processing time that is more common in real-world situations. In classic flowshop scheduling problems, job processing times are usually assumed to be independent of each other, and independent of machines. Whereas, job processing time in an industrial environment, however, may be related to the physical characteristics of the job and/or its machinery (Khatami et al., 2019).

2.1 Ant Colony Optimization (ACO)

Ant Colony Optimization (ACO) is a methodology invented in 1991 by Marco Dorigo. In 1997, Marco Dorigo and Gambardella introduced the Ant Colony System. Ant System has been widely implemented in combinatorial optimization problems, such as job scheduling, traveling salesman problem, quadratic assignment problem, vehicle routing, graph coloring, network routing. Based on the instincts of the ant colony, they can find the shortest route on the way from the nest to the places of food sources. Ants can cooperate with their colony and exchange information indirectly which is called stigmergy. When traveling on a route, ants release some information in the area they are passing, namely pheromones which are substances released by ants to detect and respond to the presence of ants. With this pheromone, the ant marks the area in its path. The next ant that follows the path will identify the pheromone marked by the previous ant and decide with a high probability to follow it and reinforce the chosen path by releasing and marking its pheromone. Ant Colony Optimization (ACO) is an algorithm that is often used in solving scheduling problems, such as research (Widyawati, 2018) that has implemented ACO intending to propose the best schedule that gives the smallest makespan in scheduling problems.

In solving scheduling problems using the ACO algorithm, 3 stages must be done, namely (Widyawati, 2018):

• Parameter Initialization
• Formation of Ant Paths

$$\boldsymbol{p_{ij}^k} = \frac{[\tau_{ij}]^{\alpha}.[\eta_{ij}]^{\beta}}{\sum\left([\tau_{i,j}]^{\alpha}.[\eta_{i,j}]^{\beta}\right)} \text{ untuk } \boldsymbol{j} \in \{\boldsymbol{N} - \boldsymbol{tabu_k}\} \quad (1)$$

$$\boldsymbol{p_{ij}^k} = \boldsymbol{0}, \quad \text{untuk } \boldsymbol{j} \text{ lainnya} \quad (2)$$

- Update Feromon

$$\Delta\boldsymbol{\tau_{ij}} = \sum_{\boldsymbol{k-1}}^{\boldsymbol{m}} \Delta\boldsymbol{\tau_{ij}^k} \quad (3)$$

$$\Delta\boldsymbol{\tau_{ij}^k} = \frac{\boldsymbol{Q}}{\boldsymbol{L_k}} \quad (4)$$

$$\Delta\boldsymbol{\tau_{ij}^k} = \boldsymbol{0} \quad (5)$$

$$\boldsymbol{\tau_{ij}} = \boldsymbol{\rho}.\boldsymbol{\tau_{ij}} + \Delta\boldsymbol{\tau_{ij}} \quad (6)$$

2.2 Particle Swarm Optimization (PSO)

Like ACO, PSO is classified as a Si-based metaheuristic optimization technique, both of which are adopted from socio-psychological principles that affect the social behavior of living things. This means that the environment has a big role in living things. Thus the interactions that occur between individuals and with their environment can optimize the way of thinking and the development of knowledge from each individual. Therefore, ACO and PSO are not just an optimization tool but also a tool that symbolizes the interaction of living things and their surrounding environment.

PSO is a population-based stochastic optimization technique (fish, bees, birds, etc.), proposed by Russell C. Eberhart and James Kennedy in 1995 which was inspired by the social behavior of the movement of birds or fish. PSO has many similarities with ACO, which is an algorithm adapted from the biology of ants. Both are inspired by social systems or biological systems and the results are obtained from the search for optimal values through random generation updates formed from random solutions. So that PSO can be applied where ACO can be applied. However, the two mechanisms have differences. PSO uses a one-way sharing method. In PSO, only gbest or pbest provide information to others to find the best solution quickly. After finding the two best values, update the velocity and position particles with the following equation (Tuegeh et al., 2009):

$$\boldsymbol{v_{ij}^{k+1}} = \boldsymbol{\omega_k} * \boldsymbol{v_{ij}^k} + \boldsymbol{c_1} * \boldsymbol{rand} * \left(\boldsymbol{pbest_{ij}^k} - \boldsymbol{x_{ij}^k}\right) + \boldsymbol{c_2} * \boldsymbol{rand} * \left(\boldsymbol{gbest_{ij}^k} - \boldsymbol{x_{ij}^k}\right) \quad (7)$$

$$\boldsymbol{x_{ij}^{k+1}} = \boldsymbol{x_{ij}^k} + \boldsymbol{v_{ij}^{k+1}} \quad (8)$$

Where:

v_{ij}^{k+1}	: is the velocity of the particle
x_{ij}^k	: is the current position of the particle (solution)
$pbest_{ij}^k$	: the best value belongs to the individual around it
$gbest_{ij}^k$	: best value overall
$rand()$	: is a random number between (0,1)
c_1, c_2	: factors that affect the speed at which the particles move, Usually $c_1 = c_2 = 2$

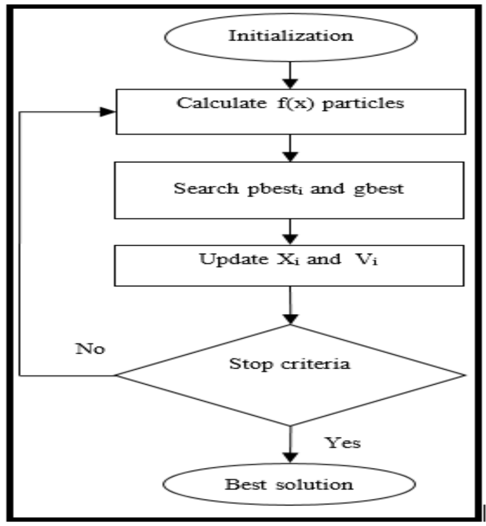

Figure 1. PSO Flowchart (Xiao et al., 2018)

2.3 Problem Solving Framework

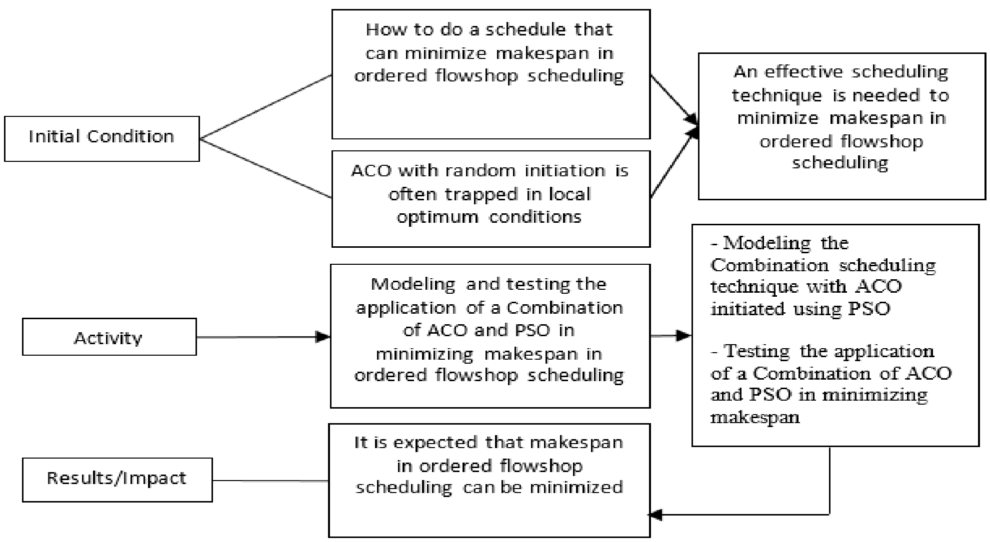

Figure 2. Problem Solving Framework

3. Results and Discussion

The S and L benchmark datasets are ordered flowshop datasets based on the flowshop permutation benchmark dataset of 240 Small instances (hereinafter referred to as Benchmark Small Vallada) and 240 Large instances (hereinafter referred to as Benchmark Large Vallada) compiled by Vallada et al (2015). Benchmark data While the T benchmark dataset is an ordered flowshop dataset that is based on the 120 instances flowshop permutation benchmark dataset (hereinafter referred to as Taillard's Benchmark) compiled by Taillard (1993).

3.1 Initialize ACO using PSO algorithm

The schedule preparation process is carried out in 3 stages; first, initialize the ACO-PSO parameters and build a random initial schedule; then second, the schedule is arranged using the PSO algorithm with a certain number of iterations and third; The best schedule generated by PSO becomes the initialization of ACO which is then processed by the ACO algorithm to obtain the minimum makespan for a certain number of iterations.

To run the ACO and PSO algorithms in this study, the parameters determined at the beginning were determined, namely:

Number of particles or Ants : 50 particles (ants)
PSO inertia weight (p_0) : 0.3
PSO cognitive weight (p_1) : 0.3
PSO social weight (p_2) : 0.4
Probability of pheromone evaporation : 1 (pheromone does not evaporate)
Maximum PSO Iteration : 100 iterations

Inertia weight (p_0) is used as a parameter of the PSO algorithm to control the effect of the previous particle velocity. If the inertia value is too large, the velocity will continue to increase so that the particle will diverge. The particle distance to its optimum value will continue to increase with each iteration. Another parameter in the PSO algorithm is the cognitive weight (p_1) and social weight (p_2), which is the acceleration constant that influences the speed of convergence. The value of p_0, p_1, and p_2 is determined at the beginning with the provisions of $p_0 + p_1 + p_2 = 1$. Xue et al (2014) recommend the use of a combination of values $p_0 = 0.3$, $p_1 = 0.3$ and $p_2 = 0.4$, state that gbest is more influential than pbest. The number of particles and the number of ants used in each test of this research is fixed, namely 50 particles or ants. The number of particles used for PSO is the same as the number of ants used for the ACO algorithm because later the initialization of ACO will use the best solution generated from the PSO algorithm.

Table 1. Random Value of Initial Particle Representation on Data S_10_5_1

Works	*Machine* 1	2	3	4	5	6	7	8	9	10
1	0.184	0.971	0.865	0.821	0.666	0.860	0.127	0.359	0.011	0.228
2	0.572	0.350	0.376	0.726	0.573	0.410	0.981	0.643	0.840	0.258
3	0.593	0.112	0.783	0.031	0.997	0.656	0.414	0.591	0.307	0.359
4	0.744	0.567	0.717	0.219	0.935	0.506	0.855	0.166	0.911	0.015
5	0.138	0.858	0.330	0.948	0.736	0.061	0.884	0.001	0.200	0.790

Table 2. Representation of schedule sequence on particle-1 of Table 4.1

Works	*Machine* 1	2	3	4	5	6	7	8	9	10
1	4	1	1	2	4	4	1	3	5	5
2	3	4	4	5	2	2	4	2	1	2
3	2	5	2	1	3	1	5	5	2	4
4	1	3	3	4	1	3	2	1	3	3
5	5	2	5	3	5	5	3	4	4	1

3.2 Application of ACO Algorithm with PSO initialization (ACO-PSO Combination)

3.2.1 Calculating Ordered Flowshop Scheduling Makespan

The schedule arrangement for each particle is then evaluated by calculating the makespan. For example, the makespan will be calculated from the schedule arrangement in Table 2, as follows:

Machine 1:

On machine 1, 5 jobs have been sorted, it can be seen that the order of jobs on machine 1 is: 4, 3, 2, 1, 5.

Assignment i=1, job j=4, on machine r=1:

Start time : $s_{1,1} = 0$

Assignment completion time : $C_{1,1} = s_{1,1} + p_{1,4} = 0 + 19 = 19$

Job completion time : $z_{1,4} = s_{1,1} + p_{1,4} = 0 + 19 = 19$

Assignment i=2, job j=3, on machine r=1:

Start time : $s_{1,2} = C_{1,1} = 19$

Completion time : $C_{1,2} = s_{1,2} + p_{1,3} = 19 + 7 = 26$

Assignment i=3, job j=2, on machine r=1:

Start time : $s_{1,3} = C_{1,3} = 26$

Completion time : $C_{1,2} = s_{1,3} + p_{1,2} = 26 + 19 = 45$

Assignment i=4, job j=1, on machine r=1:

Start time : $s_{1,4} = C_{1,2} = 45$

Completion time : $C_{1,1} = s_{1,4} + p_{1,1} = 45 + 10 = 55$

Assignment i=5, job j=5, on machine r=1:

Start time : $s_{1,5} = C_{1,1} = 55$

Completion time : $C_{1,5} = s_{1,5} + p_{1,5} = 55 + 5 = 60$

After all the work is done, the makespan value is obtained.

3.2.1 Update Schedule Using the ACO-PSO Algorithm

Table 3. The initial position of machine-1 particle-1, compared to Pbest and Gbest Particle positions

Particle position	**Process Order**				
	1	2	3	4	5
Starting position	Job 4	Job 3	Job 2	Job 1	Job 5
Pbest position	Job 4	Job 3	Job 2	Job 1	Job 5
Gbest position	Job 5	Job 1	Job 3	Job 2	Job 4

In Table 3, row 1 shows the initial position of a particle. Next, the matching is done with the Pbest position in line 2 and Gbest in line 3. The initial position has the same job processing order as the Pbest position, so $p_{p,1} = p_1 = 0{,}3$. The initial position also has a different job processing order from the Gbest position, so $p_{g,1} = 0$. So we get : $p_{1,1} = p_0 + p_{p,1} + p_{g,1} = 0.3 + 0.3 + 0 = 0.6$.

Then a random value of r is built, for example, r = 0.7635 compared to the value of. Because of the value of, the starting position of machine 1's schedule sequence is preserved. And so on until the last machine (r=10). The position update in iteration 1 is continued for particle i=2 and so on until particle i=50. After the schedule preparation using the PSO algorithm reaches the iteration limit, the next step is to process the best solution from PSO as ACO initialization.

JURNAL INFOKUM, Volume 9, No. 2,Juni 2021 ISSN : 2302-9706

3.3 Discussion

To measure the performance of the ACO-PSO approach in finding a schedule arrangement that produces the minimum makespan, the test scenario uses benchmark data of 600 schedules with variations in dimension sizes. The makespan value obtained by the ACO-PSO algorithm is compared with the makespan generated by Taillard (1993) for the Taillard benchmark data set and the makespan generated by Vallada et al. (2017) for the Small Vallada and Large Vallada benchmark data sets. The evaluation technique used consists of two criteria, namely the number of best solutions obtained (NBest) and the average relative percentage deviation (ARPD) to evaluate the performance of the algorithm used in a simple system that has been built using MatLab 2016a. The complete ACO-PSO test results in minimizing makespan on the Small Vallada benchmark data set can be seen in Appendix 1. The makespan value generated by the ACO-PSO scheduling solution in Appendix 1 is then compared with the makespan value generated by the NEH algorithm scheduling solution from previous studies (Vallada et al. 2017) which is in appendix 4, to get a comparison of the ARPD and NBest values for each algorithm.

The quality of the algorithm performance results indicated by the relative deviation percentage (RPD), was obtained using the following formula (Fernandez-Viagas et al., 2017):

$$\boldsymbol{RPD = \frac{z - z'}{z'} x100} \qquad (9)$$

Where:
z = the value of the objective function (makespan) of an algorithm used
z' = the best value of the objective function (makespan) of all the algorithms used

In this study, each dimension of the benchmark data consists of 10 instances, so in summary the number of ARPDs is obtained using the formula (Fernandez-Viagas et al., 2017):

$$\boldsymbol{ARPD = \frac{total\ RPD\ 10\ instance\ (same\ size)}{10}} \qquad (10)$$

. Table 4. Summary of NBest and ARPD on 3 Benchmark Data Sets

	Benchmark	**ACO PSO**	**NEH**
NBest	Taillard	30	90
	Small Vallada	46	194
	Large Vallada	0	240
	Total	76	524
ARPD	Taillard	0.31	0.01
	Small Vallada	0.28	0.02
	Large Vallada	0.42	0.00
	Average	0.34	0.01

Table 4 shows a summary of the achievements of the ACO-PSO algorithm in minimizing the makespan of order flowshop scheduling in the three benchmark data sets, which is compared with the achievements of the NEH algorithm from previous studies (Vallada et al, 2017 and Taillard, 1993).

4. Conclusions

The experimental results show that the ACO-PSO algorithm can provide a better scheduling solution in the scheduling group with small dimensions of 76 instances out of a total of 600 instances and is not good at obtaining makespan in the scheduling group with large dimensions.

JURNAL INFOKUM, Volume 9, No. 2,Juni 2021 ISSN : 2302-9706

Reference

[1] Allahverdi, A., Aydilek, H. & Aydilek, A., (2018). No-Wait Flowshop Scheduling Problem With Two Criteria; Total Tardiness And Makespan. European Journal of Operational Research, Vol. 269(2), pp. 590–601.
[2] Arora, D. & Agarwal, G., (2016). Meta-Heuristic Approaches For Flowshop Scheduling Problems: A Review. International Journal of Advanced Operations Management, Vol. 8(1), pp. 1–16.
[3] Assia, S., El-Abbassi, I., El-Barkany, A., Darcherif, M., & El-Biyaali, A., (2020). Green Scheduling of Jobs and Flexible Periods of Maintenance in a Two-Machine Flowshop to Minimize Makespan, a Measure of Service Level and Total Energy Consumption. Advances in Operations Research, pp. 1–9.
[4] Baker, K. R., & Trietsch, D. (2009). Three Heuristic Procedures for the Stochastic Flow Shop Problem. In Proceedings, Multidisciplinary International Conference on Scheduling. Theory and Applications (MISTA 2009), 10-12 August 2009, Dublin, Ireland.
[5] Choi, B. C., & Park, M. J. (2016). An Ordered Flow Shop With Two Agents. Asia-Pacific Journal of Operational Research, Vol. 33, No. 5 pp. 1650037 (24 pages).
[6] Davendra, D., Zelinka, I., Bialic-Davendra, M., Senkerik, R., & Jasek, R., (2013). Discrete Self-Organising Migrating Algorithm For Flow-Shop Scheduling With No-Wait Makespan. Mathematical and Computer Modelling, Vol. 57, pp. 100–110.
[7] Enxing, Z. & Ranran, L. (2017). Routing Technology in Wireless Sensor Network Based on Ant Colony Optimization Algorithm. Wireless Personal Communications, Vol. 95(3), pp. 1911–1925.
[8] Fernandez-Viagas, V., Ruiz, R. & Framinan, J. M., (2017). A new vision of approximate methods for the permutation flowshop to minimise makespan: State-of-the-art and computational evaluation. European Journal of Operational Research, Vol. 257(3), pp. 707–721.
[9] Fuchigami, H. Y., & Rangel, S. (2018). A Survey Of Case Studies In Production Scheduling: Analysis And Perspectives. Journal of Computational Science.
[10] Gao, Y., Wang, J., Wu, W., Sangaiah, A.K., & Lim, S., (2019). A Hybrid Method For Mobile Agent Moving Trajectory Scheduling Using ACO and PSO in WSNs. Sensors (Switzerland), Vol. 19(3), pp. 1-19.
[11] Gupta, J. N. D., Majumder, A. & Laha, D., (2020). Flowshop Scheduling With Artificial Neural Networks. Journal of the Operational Research Society, Vol. 71(10), pp. 1619–1637.
[12] Habibi, R. (2017). Teknik Linierisasi Untuk Menyelesaikan Persoalan Rantai Suplai Lokasi-Inventori. Jurnal As-Salam, Vol. 1(1), pp. 44–55.
[13] Hornig, E. J. S. & Zimmermann, H. J. (2013). Scheduling Multi-Stage Batch Production Systems With Continuity Constraints: The Steelmaking And Continuous Casting System. publications.rwth-aachen.de. Available at:
[14] http://publications.rwth-aachen.de/record/229769.
[15] Javier, E., Hornig, S. and Dyckhoff, H. (2013). Scheduling Multi-Stage Batch Production Systems With Continuity Constraints. The Steelmaking and Continuous Casting System.
[16] Hossain, M. S. , Asadujjaman, M. & Bhattacharya, P., (2014). Minimization of Makespan in Flow Shop Scheduling Using Heuristics. ICMIEE.
[17] Ilić, A. (2015). On The Variable Common Due Date, Minimal Tardy Jobs Bicriteria Two-Machine Flow Shop Problem With Ordered Machines. Theoretical Computer Science, Vol, 582, pp. 70–73.
[18] Kao, Y., Chen, M. H. & Huang, Y. T. (2012). A Hybrid Algorithm Based On Aco And Pso For Capacitated Vehicle Routing Problems. Mathematical Problems in Engineering. hindawi.com. Available at: https://www.hindawi.com/journals/mpe/2012/726564/abs/.

[19] Khatami, M., Salehipour, A. & Hwang, F. J. (2019). Makespan Minimization For The M-Machine Ordered Flow Shop Scheduling Problem. Computers and Operations Research, Vol. 111, pp. 400–414.
[20] Kim, B. G., Choi, B. C. & Park, M. J. (2017). Two-Machine And Two-Agent Flow Shop With Special Processing Times Structures. Asia-Pacific Journal of Operational. ….
[21] Available at: https://www.worldscientific.com/doi/abs/10.1142/S0217595917500178.
[22] Kiran, S., Guru, J. & Sharma, M., (2018) . Credit Card Fraud Detection Using Naïve Bayes Model Based And KNN Classifier. International Journal of Advance Research, Ideas and Innovations in Technology, Vol. 4, pp. 44-47.
[23] Koulamas, C. & Panwalkar, S. S., (2015). Job Selection In Two-Stage Shops With Ordered Machines. Computers and Industrial Engineering, Vol. 88, pp. 350–353.
[24] Lee, K., Zheng, F. & Pinedo, M. L., (2019). Online Scheduling Of Ordered Flow Shops. European Journal of Operational Research, Vol. 272(1), pp. 50–60.
[25] Muharni, Y., Febianti, E. & Sofa, N. N., (2019). Minimasi Makespan Pada Penjadwalan Flow Shop Mesin Paralel Produk Steel Bridge B-60 Menggunakan Metode Longest Processing Time Dan Particle Swarm Optimization. Journal Industrial Servicess, Vol. 4(2).
[26] Panwalkar, S. S. & Koulamas, C., (2012). An O (N2) Algorithm For The Variable Common Due Date, Minimal Tardy Jobs Bicriteria Two-Machine Flow Shop Problem With Ordered Machines. European Journal of Operational Research, Vol. 221, Issue 1, pp. 7-13.
[27] Panwalkar, S. S. & Koulamas, C. (2017). On The Dominance Of Permutation Schedules For Some Ordered And Proportionate Flow Shop Problems. Computers and Industrial Engineering, Vol. 107, pp. 105–108.
[28] Panwalkar, S. S., Smith, M. L. & Koulamas, C., (2013). Review Of The Ordered And Proportionate Flow Shop Scheduling Research. Naval Research Logistics (NRL).
[29] Panwalkar, S.S. & Smith, Milton & Koulamas, Christos. (2013). Review of the Ordered and Proportionate Flow Shop Scheduling Research. Naval Research Logistics (NRL), Vol. 60(1).
[30] Pinedo M.L., (2016). Flow Shops, Job Shops and Open Shops (Stochastic). In: Scheduling. Springer, Cham. https://doi.org/10.1007/978-3-319-26580-3_13
[31] Rahman, H. F., Janardhanan, M. N. & Nielsen, I. E., (2019). Real-Time Order Acceptance and Scheduling Problems in a Flow Shop Environment Using Hybrid GA-PSO Algorithm. IEEE Access. Available at: https://ieeexplore.ieee.org/abstract/document/8798615/.
[32] Riahi, V. & Kazemi, M., (2018). A New Hybrid Ant Colony Algorithm For Scheduling Of No-Wait Flowshop. Operational Research, Vol. 18(1), pp. 55–74.
[33] Sun, Y., Dong, W. & Chen, Y., (2017). An Improved Routing Algorithm Based On Ant Colony Optimization In Wireless Sensor Networks. IEEE communications Letters.
[34] Tuegeh, M., Soeprijanto & Purnomo, M. H., (2009). Modified Improved Particle Swarm Optimization For Optimal. Seminar Nasional Aplikasi Teknologi Informassi 2009 (SNATI 2009), pp. 85–90.
[35] Widyawati, W. (2018). Penerapan Algoritma Ant Colony Optimization (ACO) Pada Job Shop SCHEDULING PROBLEM (JSSP) di PT. Siemens Indonesia (Cilegon Factory). Jurnal Sistem Informasi Dan Informatika (SIMIKA), Vol. 1(01), pp. 35-51.
[36] Xiao, Y., Wang, Y. & Sun, Y. (2018). Reactive Power Optimal Control Of A Wind Farm For Minimizing Collector System Losses. Energies.