Permutation Flowshop Scheduling in ED Aluminium Using Metaheuristic Approaches

Haposan Vincentius Manalu[1], Fatiha Widyanti[2], Nur Mayke Eka Normasari[3], Andiny Trie Oktavia[4], Achmad Pratama Rifai[5]
[1,2,3,4,5] Departement of Mechanical and Industrial Engineering, Universitas Gadjah Mada, Jl. Grafika No. 2, Yogyakarta 55281, Indonesia
Email: achmad.p.rifai@ugm.ac.id[5]

Abstract

This study introduces novel metaheuristics aimed at addressing the permutation flowshop scheduling problem within the context of ED Aluminium, a company specializing in kitchen utensil production. The primary goal is to determine the optimal sequence for processing products, ultimately minimizing both makespan and total flowtime. To achieve this, three distinct metaheuristics—Simulated Annealing (SA), Large Neighborhood Search (LNS), and Ant Colony Optimization (ACO)—have been developed and investigated. Through various experiments conducted within this research, the performance of these algorithms was rigorously evaluated. The outcomes indicate that the SA algorithm stands out as the most effective, showcasing a notably shorter makespan of only 20.8 days as compared to the 42.09 days for LNS, and the 45.24 days for ACO. This study's key contribution lies in the advancement of Simulated Annealing, Large Neighborhood Search, and Ant Colony Optimization techniques tailored to tackle this specific scheduling challenge.

Keywords : Simulated Annealing, Large Neighbourhood Search, Ant Colony Optimization, Flowshop scheduling

INTRODUCTION

Planning production is critical for small and medium-sized enterprises (SMEs) as they often grapple with limitations in materials, machinery, and human resources. Effective production planning systems are essential to boost SMEs' efficiency and effectiveness. By implementing better systems, SMEs can slash potential losses, like overstock and stockouts. Overstock, referring to excess inventory, can incur increased storage costs, product deterioration, and hindered cash flow due to excessive investment. Conversely, stockouts—insufficient or no inventory—can cause production delays, unmet demand, and costs due to inventory shortages (Rachmawati and Mutiara, 2022).

Take ED Aluminium, a company specializing in casting household utensils using aluminum as raw material, for example. With 47 product variations, the company still relies on intuition to predict demand, resulting in substantial finished and semi-finished goods inventory. Lacking a structured monthly production schedule makes it challenging for ED Aluminium to control its production process. Their reliance on intuition, forecasting annual sales by adding 7% to the previous month's sales and determining production based on available warehouse stock and daily sales requirements, leads to an accumulation of semi-finished goods inventory, aimed at anticipating demand spikes and expediting production.

Like many SMEs, ED Aluminium's production planning remains relatively simple, leading to significant inefficiencies in their production line and considerable waste. One critical aspect needing improvement is their production scheduling, plagued by long makespan and low utilization. Thus, this study aims to enhance ED Aluminium's production scheduling, addressing it as a permutation flowshop scheduling problem. Three metaheuristics—Simulated Annealing, Large Neighborhood Search, and Ant Colony Optimization—have been developed to tackle this issue.

LITERATURE REVIEW

In an industrial setting, production scheduling seeks to minimize both time and expenses while striving to enhance operational efficiency and decrease production costs (Patricia and Hadi, 2011). Within this scheduling process, a critical parameter often considered is the makespan, denoting the duration needed to complete all tasks across the utilized machines (Muharni et al., 2019).

Prior research on flow shop scheduling has been extensive, exemplified by the study conducted by Widodo et al. (2014). This particular research centered on a skewer machine company facing challenges in scheduling planning, specifically grappling with high demand for gear spare parts that surpassed its production capacity, resulting in unmet customer demands. One potential solution to this scheduling predicament lies in employing a metaheuristic approach. In their study, a hybrid algorithm combining cross entropy and genetic algorithm (CEGA) was utilized to tackle the problem. The study compared computations using enumeration techniques, the CEGA algorithm, and the company's existing method. In another instance, Widyaningsih et al. (2017) applied the Earliest Due Date method to optimize starter pack activation scheduling and minimize delays.

Another notable study by Nurdiansyah (2011) focused on flow shop scheduling with dual objectives: minimizing makespan and total flowtime. The minimization of makespan aims to enhance production utilization, while reducing flowtime ensures stable resource consumption, faster job turnover, and diminished work-in-process inventory. This research employed the Differential Evolution (DE) algorithm, integrating adaptive parameters and a local search strategy to tackle the issue. The outcomes demonstrated the DE algorithm's success in significantly minimizing both makespan and total flowtime, consistently achieving an average relative percentage below 1%.

Drawing from this background, this study aims to advance production scheduling methods by specifically targeting the reduction of the makespan parameter. Three distinct approaches are utilized: simulated annealing, large neighborhood search, and ant colony optimization. Simulated Annealing (SA) has found application in reentrant permutation flow-shop scheduling, facility layout planning, and home health care supply chain optimization. Meanwhile, Large Neighborhood Search (LNS) and Adaptive Large Neighborhood Search (ALNS) have been utilized in distributed reentrant permutation flow shop scheduling, complex traveling salesman problems, and sequence-dependent job sequencing and tool switching problems. Additionally, Ant Colony Optimization (ACO) has been effectively employed in traveling salesman problems, scheduling dilemmas, and open shop scheduling challenges.

Motivated by the success of these metaheuristic approaches in prior studies, this paper tackles the permutation flow shop scheduling problem within ED Aluminium, employing these algorithms to enhance scheduling efficiency and productivity.

METHODS

This research devises three metaheuristic approaches (SA, LNS, and ACO) aimed at resolving the permutation flow shop scheduling challenge encountered in ED Aluminium. The visual representation of this study's process is illustrated in Figure 1, showcasing the research flowchart.

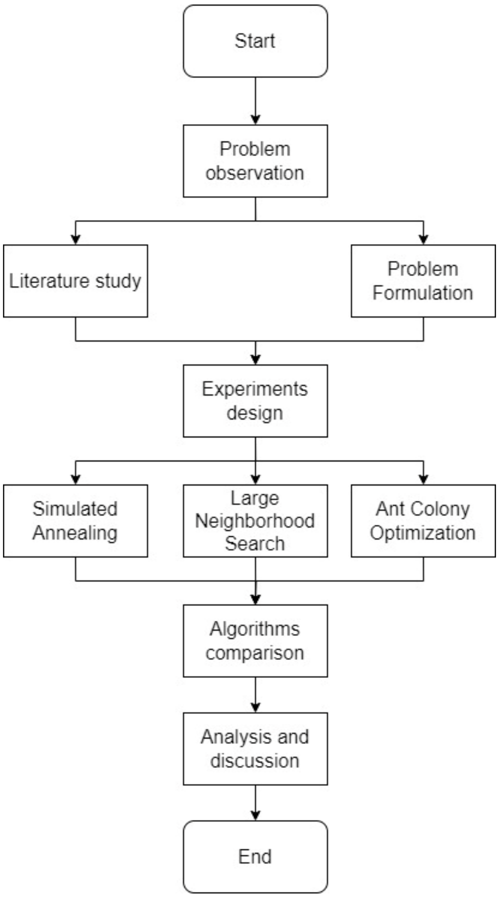

Figure 1. Flowchart of the study

Research Framework

This study aims to leverage the research findings previously established by Isnaini (2016). The gathered data will undergo analysis through three distinct metaheuristic algorithms, all geared toward the primary goal of minimizing makespan. Within this endeavor, the decision variables encompass the sequence in which products are slated for processing. The array of products, comprising eighteen distinct items, each requiring an operation sequence determination, serves as the focal point of this task, as detailed in Table 1.

Table 1. Product Types of ED Aluminium

Product Code	Product Type
1	Ordinary Pan 12
2	Ordinary Pan 13
3	Ordinary Pan 14
4	Ordinary Pan 15
5	Ordinary Pan 16
6	Ordinary Pan 18
7	Ordinary Pan 20
8	Ordinary Pan 26
9	Super Dinar Pan 15
10	Super Dinar Pan 16
11	Super Dinar Pan 18
12	Super Dinar Pan 20
13	Super Dinar Pan 22
14	Super Dinar Pan 24
15	Cauldron 26
16	Cauldron 28
17	Cauldron 45
18	Pot 40

The outcomes derived from the SA, LNS, and ACO algorithms will undergo a comparative analysis against Isnaini (2016) findings. The most optimal method among these algorithms will be identified based on their performance, aiming to select the approach that yields the most favorable results.

Simulated Annealing

Introduced by Kirkpatrick in 1983, this method draws inspiration from the annealing process in metal manufacturing and serves as an evolution of optimization techniques. The fundamental principle of simulated annealing revolves around managing the system's temperature across a specific duration. This temperature gradually declines from a higher level to a lower one. Simulated Annealing has found application in solving combinatorial optimization problems, such as location and vehicle routing challenges. Within the SA method, five key parameters play a crucial role: T0 (initial temperature), Tmax (final temperature), iterations (the number of search iterations for a solution), non-improving criteria, and α (the coefficient determining the temperature reduction rate) (Winarno, 2020).

Simulated annealing involves three primary procedures: swap, reverse, and insert. These procedures are randomly selected during each iteration to generate a new solution (Redi, 2019). The swap operation entails randomly exchanging the positions of two jobs within the sequence, while the insert operation relocates a job from one position to another randomly in the sequence. On the other hand, the reverse operation randomly alters the order of several jobs on the machine. These operations are iteratively performed until the best solution is achieved.

Large Neighborhood Search

LNS operates by segmenting the problem into smaller entities known as fragments. These chosen fragments are extracted from the initial solution and substituted with a freshly generated random solution. Subsequently, this replacement solution undergoes refinement through a local search algorithm, aiming to elevate its quality. Within the realm of LNS, the dimensions and characteristics of these fragments can be tailored to suit the specific problem at hand. The primary objective is to expedite and optimize the process of seeking a solution. Moreover, by substituting several fragments within the original solution, LNS effectively circumvents the entrapment within local solutions, facilitating the acquisition of superior global solutions.

Ant Colony Optimization

ACO operates as a metaheuristic method, drawing inspiration from the intricate behaviors of ants. This model serves as a framework for devising novel algorithms tailored to optimization solutions and distributed control dilemmas. Diverse facets of ant colony behavior, including foraging, task allocation, parent sorting, and cooperative transportation, have served as sources of inspiration, fueling the creation of various "Ant Algorithms" in recent times. ACO particularly mimics the foraging instincts observed in ant colonies, specifically addressing discrete optimization problems (Mohan, 2012).

In the natural world, ants utilize specific chemical compounds known as pheromones to mark paths connecting food sources to their colonies. These pheromone trails serve as guidance for subsequent ants in locating food, as the pheromone presence amplifies the likelihood of a path being selected (Liantoni, 2015). The operational stages within ACO encompass several key steps: initialization, ant movement, ant tour, pheromone evaporation, transferring pheromone information to the next iteration, generating a fresh list of unvisited vertices, deploying ants on these vertices, and facilitating the creation of new tours guided by updated pheromone and visibility cues.

RESULT AND DISCUSSION

The optimization of the makespan value in this flowshop scheduling task relies on data sourced from the operations of ED Aluminium. Three algorithms, namely SA, LNS, and ACO, are utilized for computations. The total processing time employed in this task is calculated by multiplying the number of products scheduled for production by their respective job processing times across each machine. The forecasted demand data for each product is derived from figures recorded in August 2015. The original production process time data for ED Aluminium, before being multiplied with the demand forecast, is outlined in Table 2. This multiplication involves the quantity of products slated for production and the specific job processing times for each machine, culminating in the subsequent table.

Table 2. Existing production process data obtained by multiplying the processing time with the demand forecast

No	Product	Operation Time (Second)				Projected demand (unit)
		Printing Machine	Milling Machine	Grinding Machine	Lathe Machine	
1	Ordinary Pan 12	2153	120568	94732	38754	2153
2	Ordinary Pan 13	2446	146760	74236	49335	2446
3	Ordinary Pan 14	2205	130095	78277	46305	2205
4	Ordinary Pan 15	1950	122850	130650	35100	1950

5	Ordinary Pan 16	2101	165979	134464	12606	2101
6	Ordinary Pan 18	1237	106382	76694	8659	1237
7	Ordinary Pan 20	629	59755	55981	22644	629
8	Ordinary Pan 26	138	19734	11868	2622	138
9	Super Dinar Pan 15	404	36360	26664	8148	404
10	Super Dinar Pan 16	496	39680	17608	10416	496
11	Super Dinar Pan 18	480	36000	19200	4800	480
12	Super Dinar Pan 20	368	44896	32752	13248	368
13	Super Dinar Pan 22	383	53237	33704	4596	383
14	Super Dinar Pan 24	254	36322	29210	3302	254
15	Cauldron 26	68	7684	7684	680	68
16	Cauldron 28	106	12826	12826	1060	106
17	Cauldron 45	261	37845	37845	3915	261
18	Pot 40	63	7308	7308	819	63

Result of Simulated Annealing

The flowshop problem stands as a combinatorial optimization challenge, involving the processing of a series of jobs across several machines in a defined sequence. The primary aim revolves around identifying the sequence that minimizes the total completion time. Displayed in Table 3 are the outcomes derived from the production sequence, inclusive of the computed makespan obtained through the simulated annealing technique.

Table 3. Job Sequence using SA Algorithm

Job sequence	Product Code	Name of Product
1	Product 6	Ordinary Pan 18
2	Product 15	Cauldron 26
3	Product 4	Ordinary Pan 15
4	Product 14	Super Dinar Pan 24
5	Product 13	Super Dinar Pan 22
6	Product 5	Ordinary Pan 16
7	Product 17	Cauldron 45
8	Product 2	Ordinary Pan 13
9	Product 9	Super Dinar Pan 15
10	Product 11	Super Dinar Pan 18
11	Product 3	Ordinary Pan 14
12	Product 18	Pot 40
13	Product 16	Cauldron 28
14	Product 1	Ordinary Pan 12
15	Product 12	Super Dinar Pan 20
16	Product 7	Ordinary Pan 20
17	Product 10	Super Dinar Pan 16
18	Product 8	Ordinary Pan 26
Best Makespan	In seconds	599,150
	In minutes	9,985.83
	In hours	166.43
	Workdays	20.8

Result of Large Neighborhood Search

The subsequent algorithm applied is LNS, which operates by dividing the problem into smaller segments termed fragments. These chosen fragments are extracted from the initial solution and substituted with a fresh random solution. This replacement solution undergoes enhancement through a local search algorithm to secure an improved outcome. Leveraging computations conducted using the Python programming language, the resulting job sequences for each product are delineated below.

Table 4. Job Sequence using LNS Algorithm

Job sequence	Product Code	Name of Product
1	Product 5	Ordinary Pan 16
2	Product 4	Ordinary Pan 15

3	Product 3	Ordinary Pan 14
4	Product 1	Ordinary Pan 12
5	Product 2	Ordinary Pan 13
6	Product 7	Ordinary Pan 20
7	Product 6	Ordinary Pan 18
8	Product 17	Cauldron 45
9	Product 11	Super Dinar Pan 18
10	Product 14	Super Dinar Pan 24
11	Product 16	Cauldron 28
12	Product 15	Cauldron 26
13	Product 9	Super Dinar Pan 15
14	Product 12	Super Dinar Pan 20
15	Product 13	Super Dinar Pan 22
16	Product 10	Super Dinar Pan 16
17	Product 18	Pot 40
18	Product 8	Ordinary Pan 26
Best Makespan	In seconds	1,212,306.88
	In minutes	20,205.11
	In hours	336.75
	Workdays	42.09

Result of Ant Colony Optimization

This algorithm draws inspiration from ants seeking the shortest path between food sources, adapting here to explore the briefest duration within the processing times across multiple machines. The outcome, displaying the sequence of products slated for production following the ACO algorithm, is depicted in Table 5. Table 6 exhibits the replication outcomes stemming from the ACO method, which underwent a total of 10 repetitions for analysis.

Table 5. Job Sequence using ACO Algorithm

Job sequence	Product Code	Name of Product
1	Product 9	Super Dinar Pan 15
2	Product 18	Pot 40
3	Product 16	Cauldron 28
4	Product 8	Ordinary Pan 26
5	Product 15	Cauldron 26
6	Product 17	Cauldron 45
7	Product 14	Super Dinar Pan 24
8	Product 5	Ordinary Pan 16
9	Product 13	Super Dinar Pan 22
10	Product 10	Super Dinar Pan 16
11	Product 3	Ordinary Pan 14
12	Product 2	Ordinary Pan 13
13	Product 4	Ordinary Pan 15
14	Product 11	Super Dinar Pan 18
15	Product 7	Ordinary Pan 20
16	Product 1	Ordinary Pan 12
17	Product 12	Super Dinar Pan 20
18	Product 6	Ordinary Pan 18
Best Makespan	In seconds	1,303,033
	In minutes	21,717.22
	In hours	361.95
	Workdays	45.24

Table 6. Replication Results Using ACO Method

Replication Number	Best Solution	Makespan	Computational time (second)
1	15-18-16-8-17-14-13-10-11-1-9-12-3-7-5-4-2-6	1,330,746	3.54
2	15-14-16-18-11-8-10-13-9-17-3-7-1-5-12-4-2-6	1,322,314	3.51
3	18-15-13-16-11-10-8-14-2-17-4-9-6-12-5-1-3-7	1,345,177	4.79
4	15-18-11-16-8-10-13-6-9-14-1-17-5-4-12-2-3-7	1,345,177	3.94
5	18-16-15-14-11-17-8-10-13-2-9-6-5-7-4-1-3-12	1,341,244	3.42
6	15-16-8-18-14-11-12-10-9-3-13-17-1-4-5-2-6-7	1,325,133	3.61
7	15-16-18-13-8-14-11-10-17-12-5-9-6-4-1-2-3-7	1,345,177	4.60
8	18-16-15-9-8-10-5-4-6-12-14-17-2-1-3-13-11-7	1,308,194	3.52
9	15-16-18-11-17-12-14-13-8-5-4-3-10-1-9-6-2-3-7	1,345,177	3.83
10	9-18-16-8-15-17-14-5-13-10-3-2-4-11-7-1-12-6	1,303,033	4.21

According to the data presented in Table 6, the production completion involving 18 products required a total of 1,303,033 seconds, equivalent to approximately 45.24 working days. However, this outcome falls short compared to the performance achieved by the LNS and SA algorithms.

CONCLUSION

This research introduces innovative metaheuristics tailored to tackle the permutation flowshop scheduling issue within ED Aluminium, a company specializing in crafting kitchen utensils. The principal aim is to optimize the product processing sequence, effectively minimizing both makespan and total time. In this study, three distinct algorithms—Simulated Annealing, Large Neighborhood Search, and Ant Colony Optimization—were applied. The results obtained were 20.8 days for the makespan of simulated annealing, 42.09 days for large neighborhood search, and 45.24 days for Ant Colony Optimization. In comparison to a prior study conducted by Isnaini (2016), which yielded a result of 23.6 days, the simulated annealing algorithm outperformed, achieving a shorter makespan. This shorter makespan signifies heightened production efficiency and improved workstation utilization, positioning simulated annealing as a more effective approach.

REFERENCES

Isnaini, W. (2016). Minimasi Makespan Penjadwalan Produksi Dengan Metode Palmer, Dannenbring, Dan Campbell Dudek Smith (Cds) Pada Sistem Produksi Flowshop (Studi Kasus: IKM ED Aluminium Yogyakarta). *Tesis*. Yogyakarta: Universitas Gadjah Mada.

Liantoni, F. (2015). Deteksi Tepi Citra Daun Mangga Menggunakan Algoritma Ant Colony Optimization. *Seminar Nasional Sains dan Teknologi Terapan III*, Vol. 3, 411-418.

Mohan, B. C. and Baskaran, R. (2012). A survey: Ant Colony Optimization Based Recent Research and Implementation on Several Engineering Domain. *Expert System and Application*, Vol. 39 No. 4, 4618-4627.

Muharni, Y., Febianti, E., dan 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. *Jurnal Industrial Services*, Vol. 4. No. 2.

Nurdiansyah, R. (2011). Pengembangan Algoritma Differential Evolution Untuk Penjadwalan Flow Shop Multi Obyektif Dengan Banyak Mesin. *Prosiding Konferensi Nasional Inovasi dalam Desain dan Teknologi.*

Rachmawati, N. L. and Lentari, M. (2022). Penerapan Metode Min-Max untuk Minimasi Stockout dan Overstock Persediaan Bahan Baku. Jurnal INTECH, Vol. 8. No. 2.

Redi, A. A. N. P. and Redioka, A. A. N. A. (2019). Algoritma Simulated Annealing untuk Optimasi Rute Kendaraan dan Pemindahan Lokasi Sepeda pada Sistem Public Bike Sharing. *Jurnal Sistem dan Manajemen Industri*. Vol. 3 No. 1, 50-58.

Widodo, D. S., Santoso, P. B., and Siswanto, E. (2014). Pendekatan Algoritma Cross Entropy-Genetic Algorithm Untuk Menurunkan Makespan Pada Penjadwalan Flow Shop. *JEMIS (Journal of Engineering & Management in Industrial System)*, Vol 2 No. 1.

Winarno and Redi, A. A. N. P. (2020). Analisa Perbandingan Metode Simulated Annealing dan Large Neighborhood Search Untuk Memecahkan Masalah Lokasi dan Rute Kendaraan Dua Eselon. *Jurnal Manajemen Industri dan Logistik*, Vol. 4 No. 1, 35-46.

Widyaningsih, R. R., Sulistyo, B., and Astuti, M. D. (2018). Perbaikan Penjadwalan Aktivasi Starter Pack Untuk Meminimasi Keterlambatan Dengan Menggunakan Metode Earliest Due Date Pada PT XYZ. *JATI UNIK: Jurnal Ilmiah Teknik dan Manajemen Industri*, Vol. 1 No. 2, 60-69.

Nurdiansyah, R. (2011). Pengembangan algoritma differential evolution untuk penjadwalan flow shop multi obyektif dengan banyak mesin. Prosiding Konferensi Nasional "Inovasi dalam Desain dan Teknologi".

Rifai, A. P., Kusumastuti, P. A., Mara, S. T. W., Norcahyo, R., and Dawal, S. Z. (2021). Multi-operator hybrid genetic algorithm-simulated annealing for reentrant permutation flow-shop scheduling. *ASEAN Engineering Journal*, Vol. 11 No. 3, 109-126.

Kusumaningsih, D. A., Azim, A. F., Albab, D. A. I. U., Hans, F. R., Korin, F., Pohan, R. N. A. M., Ananta, V. S., and Rifai, A. P. (2022). Simulated Annealing untuk Perancangan Tata Letak Industri Furniture dengan Model Single dan Double Row Layout. *Jurnal Media Teknik Dan Sistem Industri*, Vol. 6 No. 1, 60-67.

Agista, A. B., Natuna, A. P., Wangsa, H. B., Fernanda, J., Akmal, N. N., and Rifai, A. P. (2021). Perancangan Tata Letak Fasilitas UKM Kerajinan Kayu Dengan Metode Simulated Annealing. *Journal of Industrial and Manufacture Engineering*, Vol. 5 No. 2, 137-147.

Fathollahi-Fard, A. M., Govindan, K., Hajiaghaei-Keshteli, M., and Ahmadi, A. (2019). A green home health care supply chain: New modified simulated annealing algorithms. *Journal of Cleaner Production*, Vol. 240, 118200.

Rifai, A. P., Nguyen, H. T., and Dawal, S. Z. M. (2016). Multi-objective adaptive large neighborhood search for distributed reentrant permutation flow shop scheduling. *Applied Soft Computing*, Vol. 40, 42-57.

Rifai, A. P., Mara, S. T. W., and Sudiarso, A. (2021). Multi-objective distributed reentrant permutation flow shop scheduling with sequence-dependent setup time. *Expert Systems with Applications*, Vol. 183, 115339.

Mara, S. T. W., Rifai, A. P., dan Sopha, B. M. 2022. An adaptive large neighborhood search heuristic for the flying sidekick traveling salesman problem with multiple drops. *Expert Systems with Applications*, Vol. 205, 117647.

Rifai, A. P., Sutoyo, E., Mara, S. T. W., and Dawal, S. Z. M. (2022). Multiobjective Sequence-Dependent Job Sequencing and Tool Switching Problem. *IEEE Systems Journal*, *Vol. 17 No. 1, 1395-1406.*

Rokbani, N., Kumar, R., Abraham, A., Alimi, A. M., Long, H. V., Priyadarshini, I., and Son, L. H. (2021). Bi-heuristic ant colony optimization-based approaches for traveling salesman problem. *Soft Computing*, Vol. 25, 3775-3794.

Deng, W., Xu, J., and Zhao, H. (2019). An improved ant colony optimization algorithm based on hybrid strategies for scheduling problem. *IEEE access*, Vol. 7, 20281-20292.

Kurdi, M. (2022). Ant colony optimization with a new exploratory heuristic information approach for open shop scheduling problem. *Knowledge-Based Systems*, Vol. 242, 108323.