Production Systems and Information Engineering
Volume 13 (1), pp. 13-22 13
doi: https://doi.org/10.32968/psaie.2025.1.2

RANK-BASED VERSION OF ANT SYSTEM IN PRODUCTION SCHEDULING

ANITA AGÁRDI[1]
University of Miskolc, Hungary
Institute of Information Technology
agardianita@iit.uni-miskolc.hu

Abstract. The paper presents the application of the Rank-Based Version of Ant System algorithm to a common production scheduling problem. This problem is Flow Shop Scheduling. The paper presents the results using the Taillard benchmark dataset. The Rank-Based Version of Ant System (RBAS) is an improved version of the original Ant System (AS) algorithm. The goal of RBAS is to improve the convergence of the original algorithm. The algorithm uses rank-based pheromone updating. In this method, not all ants contribute equally to the pheromone deposit. Ants that find the best solutions have a greater influence on the search process. Ants that find weaker solutions have a smaller impact on the creation of new solutions. As a result, the algorithm can have faster convergence and provide high-quality solutions. The paper presents the Flow Shop Scheduling, the Rank-Based Version of Ant System, and the running results on the Taillard benchmark dataset.

Keywords: Rank-Based Version of Ant System, Production Scheduling, Flow Shop Scheduling, Taillard benchmark

1. Introduction

Industry 4.0 means the fourth industrial revolution. It transforms manufacturing and business processes by integrating smart technologies and digital solutions. This era is built on the coordinated application of automation, machine learning, artificial intelligence (AI), IoT (Internet of Things) and big data. It thus enables the emergence of intelligent and autonomous systems. Industry 4.0 aims to increase productivity, reduce costs and create more flexible and efficient manufacturing processes. In smart factories, devices and machines communicate with each other in real time, facilitating predictive maintenance and optimized production processes. This form of digitalization not only affects the manufacturing industry, but also fundamentally changes supply chains, logistics and customer relationships. Industry 4.0 is therefore not just a technological development, but a new business approach in which data and networked systems play a strategic role.

Production scheduling is one of the most important elements in optimizing manufacturing processes. It determines the order in which individual production tasks are performed and when. Its goal is to increase production efficiency, reduce costs, and optimize resource utilization. Proper scheduling helps minimize delays, lead times, and machine downtime, while ensuring that customer needs are met in the shortest possible time. The scheduling process takes into account many factors.

[1] The author used ChatGPT (OpenAI) to assist in the textual formulation of several section of the manuscript. The author reviewed and edited the AI-generated content for clarity, accuracy and coherence, and take full responsibility for the final version.

For example, production capacity, availability of raw materials, sequence of work processes, and delivery deadlines. In the modern production environment, digital systems and artificial intelligence-based algorithms are playing an increasingly important role.

Ant Colony Optimization (ACO) [1] is a metaheuristic search method inspired by nature. It was developed by Marco Dorigo in the 1990s. The algorithm is based on the behavior of ants. Ants use pheromones to find optimal routes to food sources. ACO is suitable for solving combinatorial optimization problems. It is especially good in cases where the search for the best solution is complex and has a large search space. The advantage of ACO is that it dynamically adapts to changing conditions. In the meantime, it performs parallel searches, thus providing efficient and robust solutions.
The paper further presents the Flow Shop Scheduling [2] task, the Rank-Based Version of Ant Sytem [3]. Section 3 shows the test results and compares them with the results of other researchers. The last section also provides the conclusion and future research direction.

2. Materials and methods

2.1. Flow Shop Scheduling Problem

The Flow Shop Scheduling Problem (FSSP) [2] is one of the best-known and most frequently studied types of scheduling problems. This problem occurs in manufacturing environments where workpieces pass through a given number of workstations in a predetermined sequence. One of the main challenges of the FSSP is to find the optimal sequence for performing production tasks. This minimizes the total production time (makespan), reduces waiting times, and maximizes productivity.
In Flow Shop Scheduling, each workpiece passes through the same workstations and must be processed in the same order at each workstation. However, different workpieces may require different processing times at each workstation.
In Flow Shop Scheduling, various objective functions can be optimized, such as:

- Minimization of the total machining time (makespan).
- Reducing lead times.
- Maximization of the workstation utilization.
- Minimization of the delays and missed deadlines.

There are several variations of the Flow Shop Scheduling problem:

- Permutation Flow Shop Scheduling – PFSP [4]
- Non-Permutation Flow Shop Scheduling – NPFS [5]
- Stochastic Flow Shop Scheduling [6]
- Flexible Flow Shop Scheduling [7]
- Hybrid Flow Shop Scheduling [8]

Flow Shop Scheduling is an NP-hard optimization problem. For larger problems, computing the exact solution can be extremely time-consuming. Therefore, various heuristic and metaheuristic approaches are used:

- Exact algorithms: Dynamic programming [9], Branch and Bound method [10], Linear programming [11], Heuristic methods [12].
- Johnson's rule [13]
- Simple scheduling heuristics: Shortest Processing Time [14], Earliest Due Date [15]

- Metaheuristic algorithms:
  1. Genetic algorithm (GA) [16]
  2. Ant Colony Optimization (ACO) [17]
  3. Particle Swarm Optimization (PSO) [18]
  4. Tabu Search (TS) [19]
  5. Simulated Annealing (SA) [20]

2.2. Rank-Based Version of Ant System

The Rank Based Version of Ant System (RBVAS) [21] is an improved version of Ant System (AS). In the algorithm, the better a given solution (the shorter a given path segment) is, the more pheromone it will receive, thus the more likely it is that ants will choose that path segment during their journey.
The steps of the algorithm:

1. **Parameter initialization**
   ρ: evaporation coefficient
   α: effect of pheromone on route selection
   β: the effect of heuristic information on route selection
   w: weights that contribute to the ranks of each ant
2. **Contructing the route of the ants**
   Each ants construct their routes based on the following probability:

$$P_{ij}^{k} = \begin{cases} \dfrac{\tau_{ij}^{\alpha} \cdot \eta_{ij}^{\beta}}{\sum_{l \in N_i^k} \tau_{il}^{\alpha} \cdot \eta_{il}^{\beta}} \\ 0, \quad else \end{cases}$$

   where N_i^k represents the nodes reachable from i that have not yet been selected by ant k.
3. **Ranking and weighting**
   After each ant has created its route, the solutions are ranked based on their fitness values:

$$w_r = \max(0, w - r)$$

   where $r = 1,2, \ldots, m$.
   Only the path of the first w ants are updated with the pheromone

4. **Pheromone updating:** the following factors are considered when refreshing pheromones.
   Evaporation:

$$\tau_{ij} \leftarrow (1 - \rho) \cdot \tau_{ij}$$

   Pheromone deposition: only the best solution so far and the best solution of the iteration are modified here.

$$\tau_{ij} \leftarrow \tau_{ij} + \sum_{r=1}^{w} w_r \cdot \Delta\tau_{ij}^{r}$$

   where:

$$\Delta\tau_{ij}^{r} = \begin{cases} \dfrac{1}{L_r}, if\ edge\ (i,j)\ is\ part\ of\ the\ path\ of\ ant\ r \\ 0, else \end{cases}$$

   where L_r is the cost of solution r
5. **Termination condition:** it can be reaching a certain number of iterations, convergence, or reaching a certain running time.

3. Test results

This section contains the test results on the Taillard dataset [22]. First, the algorithm statistics (maximum, average, minimum) are presented. Then, comparisons with published results by researchers follow. The comparisons were made for the following algorithms [23-24]:

- HMM-PFA (Hybrid Metaheuristic Model with Population Feature Adaptation)
- HGA (Hybrid Genetic Algorithm)
- IIGA (Improved Intelligent Genetic Algorithm)
- DSOMA (Differential Self-Organizing Migrating Algorithm)
- HGSA (Hybrid Gravitational Search Algorithm)

Table 1. Test results for the Rank Based Version of Ant System

Instance	RBVAS		
	Max	Avg	Min
Ta001	1305	1300	1297
Ta002	1367	1366.6	1366
Ta003	1155	1142.6	1127
Ta004	1389	1375.4	1368
Ta005	1285	1278.8	1276
Ta006	1244	1239.6	1230
Ta007	1261	1258.8	1254
Ta008	1283	1269.6	1255
Ta009	1295	1288.4	1285
Ta010	1170	1164.8	1157
Ta011	1700	1688.4	1672
Ta012	1784	1779	1776
Ta013	1624	1607.8	1586
Ta014	1499	1489.8	1481
Ta015	1553	1540.2	1533
Ta016	1505	1498.6	1494
Ta017	1587	1578	1560
Ta018	1658	1654.2	1649
Ta019	1694	1682	1678
Ta020	1709	1701.8	1695

Table 1 presents the performance of the Rank Based Version of Ant System (RBVAS) algorithm on various test cases. Each row shows the best (Max), average (Avg), and worst (Min) values achieved fitness on the given data set. The RBVAS algorithm performed with small deviation in all cases. This suggests that the algorithm works consistently.

Table 2. The minimum values of the test results for the Rank Based Version of Ant System

Instance	RBVAS	HMM-PFA %	HGA %	IIGA %	DSOMA %	HGSA %
Ta001	1297	14.57	11.72	14.57	5.94	2.08
Ta002	1366	11.86	6.88	11.86	3.07	5.56
Ta003	1127	29.55	22.98	29.55	13.58	-2.57

Ta004	1368	16.08	11.18	16.08	5.85	7.38
Ta005	1276	13.56	9.95	13.56	5.09	1.18
Ta006	1230	20.41	16.26	20.41	10.81	13.09
Ta007	1254	18.26	16.51	18.26	10.13	3.59
Ta008	1255	18.09	14.18	18.09	9.88	2.95
Ta009	1285	14.32	8.79	14.32	6.85	1.63
Ta010	1157	19.01	14.43	19.01	10.89	6.57
Ta011	1672	22.25	16.93	20.28	1.56	2.45
Ta012	1776	21.96	19.54	21.96	3.21	-3.27
Ta013	1586	22.32	20.55	22.32	5.67	-1.95
Ta014	1481	22.28	20.32	22.28	4.39	2.36
Ta015	1533	26.09	26.09	26.09	5.48	2.61
Ta016	1494	26.64	22.29	26.64	6.43	-2.48
Ta017	1560	25.83	24.62	25.83	3.97	3.97
Ta018	1649	24.74	21.65	24.74	4.97	6.06
Ta019	1678	17.58	13.71	17.58	4.11	-3.22
Ta020	1695	21.00	18.05	21.00	5.13	1.59

Table 2 compares the minimum values achieved by the Rank Based Version of Ant System (RBVAS) with the results of other optimization algorithms.
Each cell presents percentage values, which indicate how much the given algorithm performed better or worse compared to RBVAS.
According to the table, the RBVAS algorithm gave 10-20% better results than the HMM-PFA, HGA and IIGA algorithms in several cases. The largest positive differences can be observed in the cases Ta003 (29.55%), Ta006 (20.41%), Ta015 (26.09%) and Ta016 (26.64%).

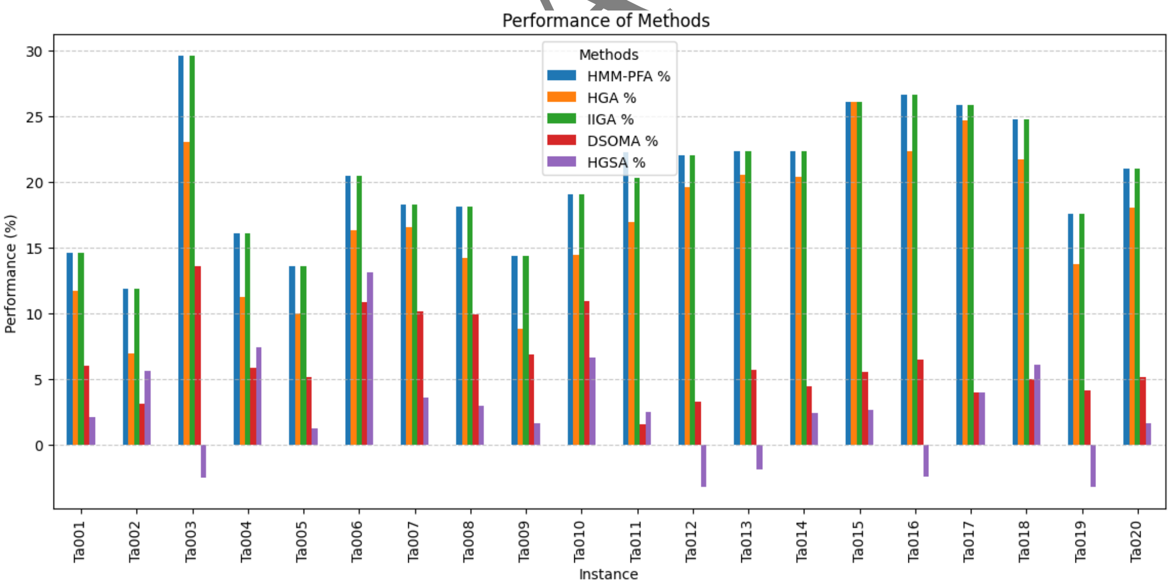

Figure 1. The minimum values of the test results for the Rank Based Version of Ant System

Figure 1 compares the minimum of the values of the Rank Based Version of Ant System with the results published by the researchers in the form of a bar chart.

Table 3. The average values of the test results for the Rank Based Version of

Ant System

Instance	RBVAS	HMM-PFA %	HGA %	IIGA %	DSOMA %	HGSA %
Ta001	1300	14.31	11.46	14.31	5.69	1.85
Ta002	1366.6	11.81	6.83	11.81	3.03	5.52
Ta003	1142.6	27.78	21.30	27.78	12.03	-3.90
Ta004	1375.4	15.46	10.59	15.46	5.28	6.81
Ta005	1278.8	13.31	9.71	13.31	4.86	0.95
Ta006	1239.6	19.47	15.36	19.47	9.95	12.21
Ta007	1258.8	17.81	16.06	17.81	9.71	3.19
Ta008	1269.6	16.73	12.87	16.73	8.62	1.76
Ta009	1288.4	14.02	8.51	14.02	6.57	1.37
Ta010	1164.8	18.22	13.67	18.22	10.15	5.86
Ta011	1688.4	21.06	15.79	19.11	0.57	1.46
Ta012	1779	21.75	19.34	21.75	3.04	-3.43
Ta013	1607.8	20.66	18.92	20.66	4.24	-3.28
Ta014	1489.8	21.56	19.61	21.56	3.77	1.76
Ta015	1540.2	25.50	25.50	25.50	4.99	2.13
Ta016	1498.6	26.25	21.91	26.25	6.10	-2.78
Ta017	1578	24.40	23.19	24.40	2.79	2.79
Ta018	1654.2	24.35	21.27	24.35	4.64	5.73
Ta019	1682	17.30	13.44	17.30	3.86	-3.45
Ta020	1701.8	20.52	17.58	20.52	4.71	1.19

Table 3 shows the average performance of the RBVAS algorithm. It compares the algorithm with other optimization algorithms. The percentage values here also show how much each algorithm performed better or worse than RBVAS.

RBVAS was generally better or similar to the other algorithms. The average difference was 10-20% better in several cases. The best results were achieved by RBVAS in the following test cases: Ta003 (27.78%), Ta015 (25.50%), Ta016 (26.25%).

The worst performances were achieved by the HGSA and IIGA algorithms on the following data: Ta012 (-3.43%), Ta019 (-3.45%), Ta003 (-3.90%). The HGSA algorithm performed better than RBVAS for some data sets. For example: Ta003 (-3.90%), Ta019 (-3.45%), Ta012 (-3.43%).

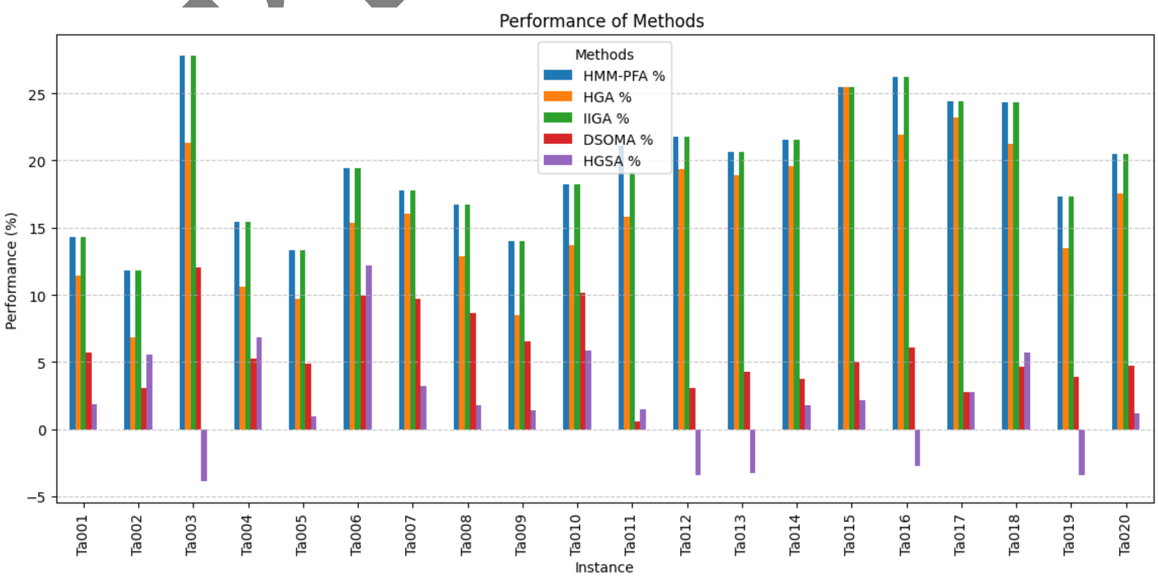

Figure 2. The average values of the

test results for the Rank Based Version
of Ant System

Figure 2 shows the results of the Rank Based Version of Ant System and compares them with the published results in the form of a bar chart.

Table 4. The maximum values of the test results for the Rank Based Version of Ant System

Instance	RBVAS	HMM-PFA %	HGA %	IIGA %	DSOMA %	HGSA %
Ta001	1305	13.87	11.03	13.87	5.29	1.46
Ta002	1367	11.78	6.80	11.78	3.00	5.49
Ta003	1155	26.41	20.00	26.41	10.82	-4.94
Ta004	1389	14.33	9.50	14.33	4.25	5.76
Ta005	1285	12.76	9.18	12.76	4.36	0.47
Ta006	1244	19.05	14.95	19.05	9.57	11.82
Ta007	1261	17.61	15.86	17.61	9.52	3.01
Ta008	1283	15.51	11.69	15.51	7.48	0.70
Ta009	1295	13.44	7.95	13.44	6.02	0.85
Ta010	1170	17.69	13.16	17.69	9.66	5.38
Ta011	1700	20.24	15.00	18.29	-0.12	0.76
Ta012	1784	21.41	19.00	21.41	2.75	-3.70
Ta013	1624	19.46	17.73	19.46	3.20	-4.25
Ta014	1499	20.81	18.88	20.81	3.14	1.13
Ta015	1553	24.47	24.47	24.47	4.12	1.29
Ta016	1505	25.71	21.40	25.71	5.65	-3.19
Ta017	1587	23.69	22.50	23.69	2.21	2.21
Ta018	1658	24.07	20.99	24.07	4.40	5.49
Ta019	1694	16.47	12.63	16.47	3.13	-4.13
Ta020	1709	20.01	17.09	20.01	4.27	0.76

Table 4 shows the maximum of the Rank Based Version of Ant System algorithm runs.

The percentage differences show several times a 10–20% advantage in favor of RBVAS (for example, in the case of Ta001, 13.87% for HMM-PFA, 11.03% for HGA, and in the case of Ta006, 19.05% for HMM-PFA and 14.95% for HGA). This suggests that in many cases, RBVAS produced significantly better solutions than the compared algorithms.

Ta001–Ta005: These data sets show maximum values between 1285–1367 for RBVAS. The HMM-PFA and IIGA columns show an approximately 12–14% advantage in favor of RBVAS. The value for HGA is 9–11%. In the case of DSOMA and HGSA, the percentage values are smaller, 5.29–1.46%, so the RBVAS algorithm was less better than these algorithms.

Ta003: The maximum value of RBVAS is 1155. The percentage advantages for HMM-PFA and IIGA are 26.41%. For HGA, it is 20.00%, which indicates that the RBVAS algorithm is an outstanding improvement. However, for HGSA, it is -4.94%, which shows that the published HGSA result was better.

Ta006–Ta010: In these cases, the results of RBVAS range from 1170–1244. The advantages over HMM-PFA, HGA, and IIGA algorithms are in the range of 14–19%. For DSOMA, the results are 6–10%. For HGSA, it can be seen a difference of 0.7–11%. In particular, for Ta006, the positive value of 11.82% in the HGSA column indicates that RBVAS has achieved a significant improvement over its competitor.

Ta011–Ta014: RBVAS shows higher maximum values for these data sets: 1499–

1700. The advantage of RBVAS over HMM-PFA and IIGA algorithms is still in the range of 15–22%.
Ta015–Ta020: In the last five instances, RBVAS results are between 1505–1709. Here, the advantages for HMM-PFA and IIGA are 20–26%.

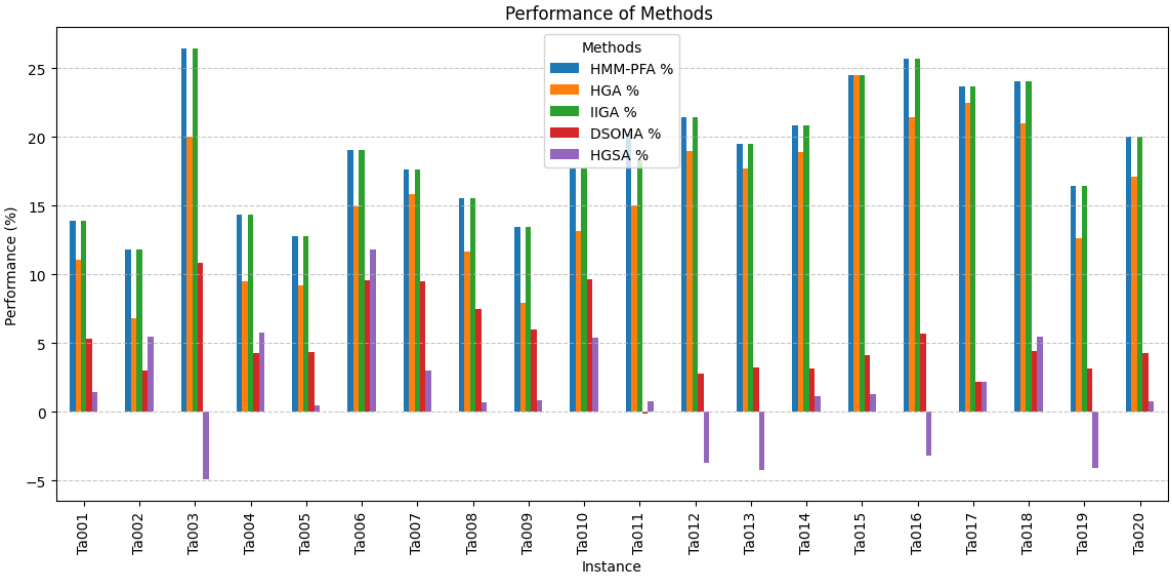

Figure 3. The average values of the test results for the Rank Based Version of Ant System

Figure 3 presents the maximum running values of the Rank Based Version of Ant System, comparing it with the results published by the researchers in the form of a bar chart.

4. Conclusions and future research directions

The paper investigated the efficiency of the Rank-Based Version of Ant System algorithm on a common production scheduling problem, which is the Flow Shop Scheduling Problem. For this problem, researchers often use the Taillard dataset to test the developed algorithms. Based on the results of the RBVAS algorithm, it can be concluded that the Rank Based Version of Ant System is significantly more efficient than the other compared optimization methods. During the tests, the research repeatedly experienced 10–20% improvement compared to the HMM-PFA, HGA and IIGA algorithms. However, it can be observed that for some individuals (e.g. Ta003, Ta012, Ta016, Ta019) the HGSA algorithm showed better performance.
The results of the RBVAS algorithm show that the algorithm exhibits strong competitiveness in the field of combinatorial optimization. However, several questions arise that require further research:

- Parameter optimization and adaptivity
- Development of hybrid approaches
- Real-time and dynamic environments
- Multi-objective optimization
- Scalability and large-scale problems

Acknowledgement. Supported by the University Research Scholarship Program of the Ministry for Culture and Innovation from the source of the National Research, Development and Innovation Fund.”

EKÖP Egyetemi Kutatói Ösztöndíj Program

Ministry of Culture and Innovation

References

[1] Dorigo, M., Birattari, M., & Stutzle, T. (2007). Ant colony optimization. IEEE computational intelligence magazine, 1(4), 28-39. https://doi.org/10.1109/MCI.2006.329691

[2] Rossit, D. A., Tohmé, F., & Frutos, M. (2018). The non-permutation flow-shop scheduling problem: a literature review. Omega, 77, 143-153. https://doi.org/10.1016/j.omega.2017.05.010

[3] Yousefikhoshbakht, M., Didehvar, F., & Rahmati, F. (2016). AN EFFECTIVE RANK BASED ANT SYSTEM ALGORITHM FOR SOLVING THE BALANCED VEHICLE ROUTING PROBLEM. International Journal of Industrial Engineering, 23(1).

[4] Framinan, J. M., Gupta, J. N., & Leisten, R. (2004). A review and classification of heuristics for permutation flow-shop scheduling with makespan objective. Journal of the Operational Research Society, 55(12), 1243-1255. https://doi.org/10.1057/palgrave.jors.2601784

[5] Rossit, D. A., Tohmé, F., & Frutos, M. (2018). The non-permutation flow-shop scheduling problem: a literature review. Omega, 77, 143-153. https://doi.org/10.1016/j.omega.2017.05.010

[6] Gourgand, M., Grangeon, N., & Norre, S. (2003). A contribution to the stochastic flow shop scheduling problem. European Journal of Operational Research, 151(2), 415-433. https://doi.org/10.1016/S0377-2217(02)00835-4

[7] Wang, H. (2005). Flexible flow shop scheduling: optimum, heuristics and artificial intelligence solutions. Expert systems, 22(2), 78-85. https://doi.org/10.1111/j.1468-0394.2005.00297.x

[8] Linn, R., & Zhang, W. (1999). Hybrid flow shop scheduling: a survey. Computers & industrial engineering, 37(1-2), 57-61. https://doi.org/10.1016/S0360-8352(99)00023-6

[9] de Souza, E. A. G., Nagano, M. S., & Rolim, G. A. (2022). Dynamic Programming algorithms and their applications in machine scheduling: A review. Expert Systems with Applications, 190, 116180. https://doi.org/10.1016/j.eswa.2021.116180

[10] Gmys, J., Mezmaz, M., Melab, N., & Tuyttens, D. (2020). A computationally efficient Branch-and-Bound algorithm for the permutation flow-shop scheduling problem. European Journal of Operational Research, 284(3), 814-833. https://doi.org/10.1016/j.ejor.2020.01.039

[11] Bülbül, K., & Kaminsky, P. (2013). A linear programming-based method for job shop scheduling. Journal of Scheduling, 16, 161-183. https://doi.org/10.1007/s10951-012-0270-4

[12] Zheng, D. Z., & Wang, L. (2003). An effective hybrid heuristic for flow shop scheduling. The International Journal of Advanced Manufacturing Technology, 21, 38-44. https://doi.org/10.1007/s001700300005

[13] Ku, P. S., & Niu, S. C. (1986). On Johnson's two-machine flow shop with random processing times. Operations Research, 34(1), 130-136. https://doi.org/10.1287/opre.34.1.130

[14] Riezebos, J., & Gaalman, G. J. (1998). Time lag size in multiple operations flow shop scheduling heuristics. European journal of operational research, 105(1), 72-90. https://doi.org/10.1016/S0377-2217(97)00025-8

[15] Gere Jr, W. S. (1966). Heuristics in job shop scheduling. Management Science, 13(3), 167-190. https://doi.org/10.1287/mnsc.13.3.167

[16] Engin, O., Ceran, G., & Yilmaz, M. K. (2011). An efficient genetic algorithm for hybrid flow shop scheduling with multiprocessor task problems. Applied Soft Computing, 11(3), 3056-3065. https://doi.org/10.1016/j.asoc.2010.12.006

[17] Engin, O., & Güçlü, A. (2018). A new hybrid ant colony optimization algorithm for solving the no-wait flow shop scheduling problems. Applied Soft Computing, 72, 166-176. https://doi.org/10.1016/j.asoc.2018.08.002

[18] Kuo, I. H., Horng, S. J., Kao, T. W., Lin, T. L., Lee, C. L., Terano, T., & Pan, Y. (2009). An efficient flow-shop scheduling algorithm based on a hybrid particle swarm optimization model. Expert systems with applications, 36(3), 7027-7032. https://doi.org/10.1016/j.eswa.2008.08.054

[19] Ben-Daya, M., & Al-Fawzan, M. (1998). A tabu search approach for the flow shop scheduling problem. European journal of operational research, 109(1), 88-95. https://doi.org/10.1016/S0377-2217(97)00136-7

[20] Van Laarhoven, P. J., Aarts, E. H., & Lenstra, J. K. (1992). Job shop scheduling by simulated annealing. Operations research, 40(1), 113-125. https://doi.org/10.1287/opre.40.1.113

[21] Bullnheimer, B., Hartl, R. F., & Strauss, C. (1999). A New Rank Based Version of the Ant System--A Computational Study. Central European Journal of Operations Research, 7(1).

[22] E. Taillard, "Benchmarks for basic scheduling problems", EJOR 64(2):278-285, 1993. https://doi.org/10.1016/0377-2217(93)90182-M

[23] Qu, C., Fu, Y., Yi, Z., & Tan, J. (2018). Solutions to no-wait flow shop scheduling problem using the flower pollination algorithm based on the hormone modulation mechanism. Complexity, 2018 https://doi.org/10.1155/2018/1973604

[24] Wei, H., Li, S., Jiang, H., Hu, J., & Hu, J. (2018). Hybrid genetic simulated annealing algorithm for improved flow shop scheduling with makespan criterion. Applied Sciences, 8(12), 2621 https://doi.org/10.3390/app8122621