Production Systems and Information Engineering
Volume 13 (1), pp. 1–13.
https://doi.org/10.32968/psaie.2025.1.1

APPLICATION OF THE MAX-MIN ANT SYSTEM ON FLOW SHOP SCHEDULING PROBLEMS

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

Abstract. The paper investigates the effectiveness of one of the algorithms in the Ant Colony Optimization algorithm family, the MAX-MIN Ant System algorithm, on the Flow Shop Scheduling. The main feature of the algorithm is the pheromone constraint, which defines a lower and upper bound on the pheromone values. This prevents the search space from converging too quickly towards a single solution. Another important feature of MMAS is that usually only the best ant (globally best or iteration best) updates the pheromone values. This helps the algorithm to find optimal or near-optimal solutions faster, while reducing the chance of getting stuck in local minima.

Keywords: *Flow Shop Scheduling Problem, MAX-MIN Ant System*

1. Introduction

Manufacturing and production scheduling play a key role in increasing the efficiency of modern industrial systems. During manufacturing, raw materials and semi-finished products are converted into finished products through various technological processes. The development of optimal production processes is essential in order to reduce costs, minimize lead times and efficiently utilize resources.

Production scheduling is an important task in manufacturing. It determines how different work processes, machines and labor should be distributed in time and space for the best possible performance. Heuristic and metaheuristic algorithms, such as Genetic Algorithms (GA) [1–3], Ant Colony Optimization (ACO) [4–6] or Harmony Search (HS) [7–9], are often used for optimal scheduling.

Genetic algorithms (GAs) are particularly useful in the field of production scheduling, as they can effectively handle the combinatorial complexity of problems. GAs are capable of parallel exploration of large and complex search spaces.

[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.

Population-based search ensures that multiple scheduling solutions are explored simultaneously, which reduces the chance of getting stuck in local optima. The genetic algorithm can be easily customized to different manufacturing environments, such as machine capacities, working time constraints, lead times, or priority rules. The coding scheme (e.g., permutation coding) and the fitness function specification allow the integration of problem-specific conditions. Genetic algorithms also allow the optimization of multiple objective functions simultaneously (e.g., minimizing lead time, maximizing machine utilization, and reducing costs).

The Ant Colony Optimization (ACO) algorithm models the collective behavior of ants and is excellent for solving production scheduling problems. In the ACO algorithm, “artificial ants” build the schedule step by step, similar to how ants search for a route to food. This approach is particularly advantageous in scheduling problems where the order of decisions is critical (e.g., assigning jobs to machines). A key element of ACO is the updating of pheromone trails. Paths representing high-quality scheduling solutions receive higher pheromone levels, making them more likely to be chosen by ants in subsequent iterations. This positive feedback quickly improves the efficiency of the search process. ACO is easily adaptable to different manufacturing environments, such as multi-machine scheduling, taking into account transition times between workstations, or dynamic conditions (machine failures, unexpected orders).

Harmony Search (HS) is a metaheuristic algorithm inspired by musical improvisation. The HS algorithm is based on simple steps: a “harmony memory” stores possible scheduling solutions, and then combines them, generates random new solutions, and fine-tunes them to create new, better solutions. This flexibility makes it easy to adapt to different manufacturing environments. HS is able to simultaneously consider multiple scheduling considerations, such as minimizing lead times, increasing resource utilization, and reducing costs. This makes it suitable for complex industrial environments where decision-making involves multiple, often conflicting objectuves. The harmony memory’s update rules allow the algorithm to find near-optimal schedules even after relatively few iterations. This is especially beneficial in high-volume or time-sensitive manufacturing systems.

In recent years, the digitalization and automation of manufacturing processes have become increasingly important. One of its defining concepts is Industry 4.0, also known as the fourth industrial revolution. This concept encompasses the application of advanced technologies such as IoT (Internet of Things), artificial intelligence (AI), big data analytics, cyber-physical systems (CPS), and cloud computing. Intelligent manufacturing systems are capable of real-time data collection and analysis. This enables dynamic optimization of scheduling and more efficient use of resources.

A key element of Industry 4.0 is the smart factory, in which machines and sensors are in constant communication with each other. This enables autonomous decision-making and adaptive production processes. Such systems can respond quickly to changing market demands, reduce downtime and increase productivity.

2. MAX-MIN Ant System

The Max-Min Ant System (MMAS) [10] is an improved Ant Colony Optimization algorithm developed by Thomas Stützle and Holger Hoos in 1997. The algorithm is published first in the following article: Max-Min Ant System and Local Search for the Traveling Salesman Problem.
The number of articles published annually (Figure 1.) in the topic of the MAX-MIN Ant System algorithm has increased over the years, approximately doubling between 2010 and 2024. In 2010, 1,260 articles were published, in 2015, 1,650, in 2020, 2,060, and in 2024, 2,110.

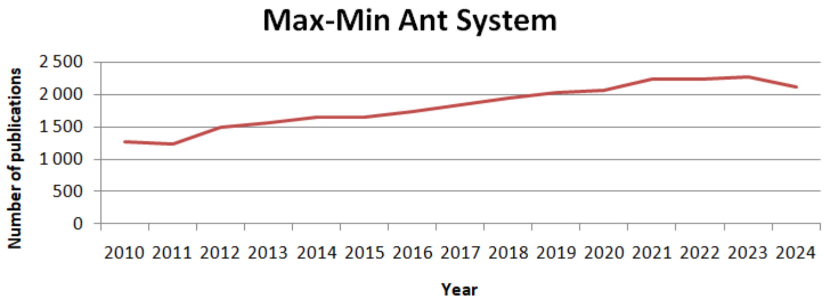

Figure 1. Publication frequency of the keyword “Max-Min Ant System”

One of the main features of MMAS is pheromone constraints. The algorithm defines a lower and upper bound on the pheromones of the edges. This ensures that the algorithm continuously maps the search space. The pheromone constraints can change dynamically during the iterations. In this case, higher diversity is observed at the beginning of the search process, while in the final stage, the best solution is fine-tuned.

Another important innovation is that MMAS usually updates the pheromones only for the best ant (based on the iterative best or global best solution). The MMAS algorithm is widely applied to combinatorial optimization problems. For example, the Traveling Salesman Problem (TSP) [11], Flow Shop Scheduling [12], Vehicle Route Problem (VRP) [13], and other complex scheduling problems.

The Flow Shop Scheduling Problem (FSSP) is a typical permutation combinatorial optimization problem, where the goal is to find the optimal order of jobs on multiple machines, such that, for example, the makespan (total throughput time) is minimized. The search space of the FSSP grows very quickly with the size of the problem, so classical exact computation methods (e.g., branch-and-bound) can only be applied to small instances. The Max–Min Ant System is effective on this problem for several reasons:

- *Permutation-based solution construction:* MMAS builds permutations naturally. Ants select the next job in the partial order step by step. Pheromones learn pairwise preferences, which fits well with the nature of the flow shop.

- *Balance of intensification and diversification:* Pheromone constraints prevent early stagnation. They maintain high diversity at the beginning of the search (exploring a wide range of orders), and at the end they strongly focus on fine-tuning the best solutions. This feature is important because FSSP often contains several local optima close to each other.
- *Flexibility in integrating heuristics:* It can be easily combined with domain-specific heuristics. Incorporating them can significantly speed up the search for good permutations.
- *Robustness and scalability:* MMAS performs well on larger instances, because more ants can explore the search space in parallel. Pheromone boundaries and iterative best update stabilize the search even with increasing problem sizes.
- *Empirical evidence:* Several studies have shown that MMAS-based approaches often find better or similar quality solutions than other ACO variants or metaheuristics (e.g. PSO, GA) for FSSP, while converging faster or showing less parameter sensitivity.

The steps of the MMAS algorithm is the following:

1. Initialization of the parameters

τ_{min}: lower limit of pheromone
τ_{max}: upper limit of pheromone
ρ: evaporation coefficient
α: effect of pheromone on route selection
β: the effect of heuristic information on route selection
Initially, the pheromone values are set toτ_{max}

2. Construction of the route of the ants

Each ants construct their routes based on the following probability:

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

where N_i^k represents the nodes reachable from i that have not yet been selected by ant k.

3. Pheromone updating: the following factors are considered when updating pheromones.

The following formula ensure the pheromone evaporation:

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

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

$$\tau_{ij} \leftarrow \tau_{ij} + \Delta\tau_{ij}$$

where:

$$\Delta\tau_{ij} = \begin{cases} \frac{1}{L_{best}}, if\ edge\ (i,j)\ is\ part\ of\ the\ best\ solution \\ 0, else \end{cases}$$

where L_{best} is the cost (fitness) of the best solution
Pheromone restriction: pheromone values can only be between a certain τ_{min} and τ_{max}.

4. **Termination condition:** it can be reaching a certain number of iterations, convergence, or reaching a certain running time.

3. Test results

This section presents the test results. The test results were run on the Taillard [14] data set. First, the article examines the maximum, average and minimum values of the runs for each data set. Then, the individual results (maximum, average, minimum) are compared with other published results [15–16].

Table 1
Test results for the Max-Min Ant System

Instance	MMAS		
	Max	**Avg**	**Min**
Ta001	1297	1297	1297
Ta002	1368	1366.4	1364
Ta003	1161	1147.6	1132
Ta004	1377	1365.4	1359
Ta005	1283	1276.2	1265
Ta006	1254	1245.4	1241
Ta007	1267	1260.4	1251
Ta008	1283	1274.2	1265
Ta009	1300	1290	1275
Ta010	1175	1163.4	1142
Ta011	1702	1697.2	1692
Ta012	1793	1777.4	1770
Ta013	1624	1613.8	1588
Ta014	1498	1488	1477
Ta015	1557	1549.2	1542
Ta016	1503	1496.6	1491
Ta017	1595	1578	1554
Ta018	1667	1651.6	1639
Ta019	1684	1676.6	1673
Ta020	1711	1703.8	1693

The table above shows the results of the Max-Min Ant System (MMAS) algorithm. The columns contain the best (Max), average (Avg), and worst (Min) solutions of the iterations. Based on the results, the MMAS algorithm provides stable performance. The difference between the minimum, average, and maximum values obtained for each instance is relatively small. This indicates that the algorithm consistently finds good solutions. The average values are close to the best solutions found.

Minor datasets (Ta001–T010): For example, in the case of Ta001, all test runs gave the same result. In the case of Ta002, the difference between the maximum and minimum values is only 4 fitness value. The average result is also very close to the maximum. The other instances, such as Ta003 and Ta004, also perform well, although the differences between the maximum and minimum increase a bit.

Medium problems (Ta011–Ta015): In the case of Ta011 and Ta012, the differences between the maximum and minimum are slightly larger. In the case of Ta011, the maximum value was 1702, the average was 1697.2, while the minimum was 1692. In the case of Ta012, the maximum value was 1793, the average was 1777.4, while the minimum was 1770.

Larger problems (Ta016–Ta020): In the case of larger problems, the differences increase. In the case of Ta016, the maximum value is 1503, the average is 1496.6, while the minimum is 1491. In the case of Ta017, the maximum is 1595, the average is 1578, while the minimum is 1554. The situation is similar in the case of Ta018 and Ta019.

Table 2

Comparison of the maximum test results for the Max-Min Ant System

Instance	MMAS	HMM-PFA %	HGA %	IIGA %	DSOMA %	HGSA %
Ta001	1297	14.57	11.72	14.57	5.94	2.08
Ta002	1364	12.02	7.04	12.02	3.23	5.72
Ta003	1132	28.98	22.44	28.98	13.07	–3.00
Ta004	1359	16.85	11.92	16.85	6.55	8.09
Ta005	1265	14.55	10.91	14.55	6.01	2.06
Ta006	1241	19.34	15.23	19.34	9.83	12.09
Ta007	1251	18.55	16.79	18.55	10.39	3.84
Ta008	1265	17.15	13.28	17.15	9.01	2.13
Ta009	1275	15.22	9.65	15.22	7.69	2.43
Ta010	1142	20.58	15.94	20.58	12.35	7.97
Ta011	1692	20.80	15.54	18.85	0.35	1.24
Ta012	1770	22.37	19.94	22.37	3.56	–2.94
Ta013	1588	22.17	20.40	22.17	5.54	–2.08
Ta014	1477	22.61	20.65	22.61	4.67	2.64
Ta015	1542	25.36	25.36	25.36	4.86	2.01
Ta016	1491	26.89	22.54	26.89	6.64	–2.28
Ta017	1554	26.32	25.10	26.32	4.38	4.38
Ta018	1639	25.50	22.39	25.50	5.61	6.71
Ta019	1673	17.93	14.05	17.93	4.42	–2.93
Ta020	1693	21.15	18.19	21.15	5.26	1.71

Table 2 presents the minimum of the Max-Min Ant System test results, also compared to the published results by researchers.

Smaller problems (Ta001–Ta010): For smaller problems, the MMAS results show very good stability and low minimum values. These are significantly better than the published results by researchers. For example, for Ta001, the MMAS algorithm has a minimum value of 1297. For Ta002, the MMAS minimum is 1364.

Medium problems (Ta011–Ta015): MMAS also outperforms medium problems in several cases. In cases Ta011 and Ta012, the minimum value of MMAS is 20.80% and 22.37% lower than the results achieved by the researchers. In case Ta013, the minimum value is 1588. This is about 22% better than the results published by the researchers. In case Ta015, MMAS produced a result that is about 25% better than the minimum achieved by the researchers.

Larger problems (Ta016–Ta020): In case Ta016, the minimum of MMAS is 1491. This is about 26% better than the published results. In case Ta017 and Ta018, a significant advantage is also shown.

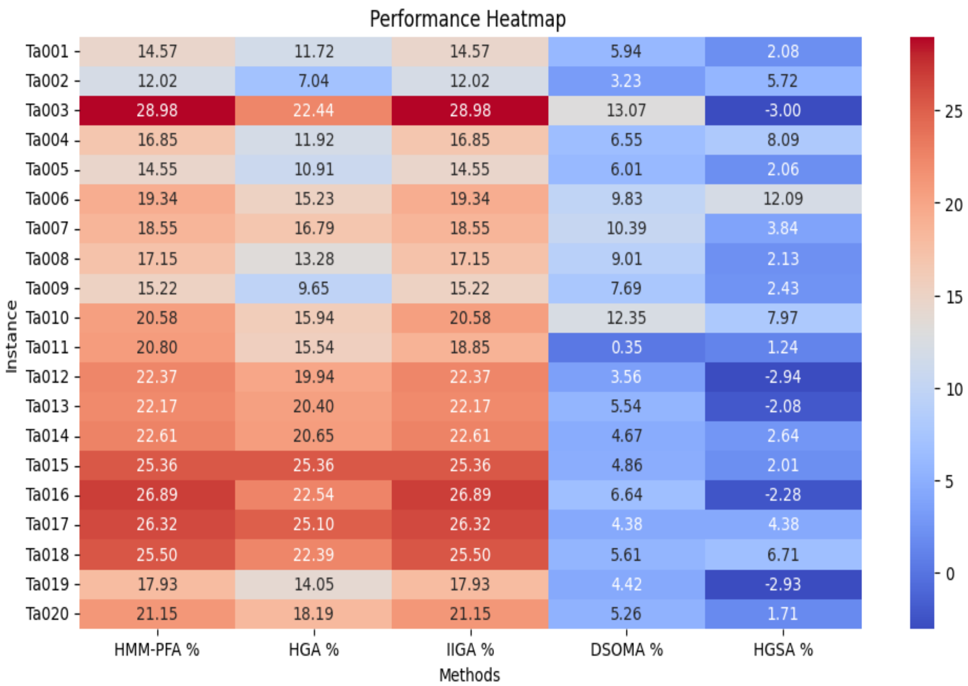

Figure 2. Heatmap of the maximum test results for the Max-Min Ant System

Figure 2 compares the maximum of the running values of the MAX-MIN Ant System algorithm and the results published by the researchers with a heat map.

Table 3

Comparison of the average test results for the Max-Min Ant System

Instance	MMAS	HMM-PFA %	HGA %	IIGA %	DSOMA %	HGSA %
Ta001	1297	14.57	11.72	14.57	5.94	2.08
Ta002	1366.4	11.83	6.85	11.83	3.04	5.53
Ta003	1147.6	27.22	20.77	27.22	11.54	–4.32
Ta004	1365.4	16.30	11.40	16.30	6.05	7.59
Ta005	1276.2	13.54	9.94	13.54	5.08	1.16
Ta006	1245.4	18.92	14.82	18.92	9.44	11.69
Ta007	1260.4	17.66	15.92	17.66	9.57	3.06
Ta008	1274.2	16.31	12.46	16.31	8.22	1.40
Ta009	1290	13.88	8.37	13.88	6.43	1.24
Ta010	1163.4	18.36	13.80	18.36	10.28	5.98
Ta011	1697.2	20.43	15.19	18.49	0.05	0.93
Ta012	1777.4	21.86	19.44	21.86	3.13	–3.34
Ta013	1613.8	20.21	18.48	20.21	3.85	–3.64
Ta014	1488	21.71	19.76	21.71	3.90	1.88
Ta015	1549.2	24.77	24.77	24.77	4.38	1.54
Ta016	1496.6	26.42	22.08	26.42	6.24	–2.65
Ta017	1578	24.40	23.19	24.40	2.79	2.79
Ta018	1651.6	24.55	21.46	24.55	4.81	5.90
Ta019	1676.6	17.68	13.80	17.68	4.20	–3.14
Ta020	1703.8	20.38	17.44	20.38	4.59	1.07

Table 3 shows the average performance of the MMAS (Max-Min Ant System) algorithm. The table compares the results with the published results by researchers. The average results of MMAS show outstanding efficiency against various comparison algorithms (HMM-PFA, HGA, IIGA, DSOMA, HGSA).

Minor problems (Ta001–Ta010): On problem Ta001, MMAS achieved an average result of 1297. On problem Ta002, MMAS achieved an average result of 1366.4. The other problems also presented good test results.

Medium problems (Ta011–Ta015): MMAS continued to perform well on medium-sized problems. However, the improvement was smaller than on smaller problems. For Ta011, MMAS achieved an average score of 1697.2. For the other medium problems, MMAS continued to achieve outstanding results.

Larger problems (Ta016–Ta020): MMAS also performed well on larger problems, but the differences were not as large in favor of MMAS. For example, Ta016 achieved a 26.42% better result than the published result. For Ta017 and Ta018, MMAS also performed well on average, with improvements of 24.40% and 24.55%. For Ta019 and Ta020, MMAS achieved improvements of 17.68% and 20.38%.

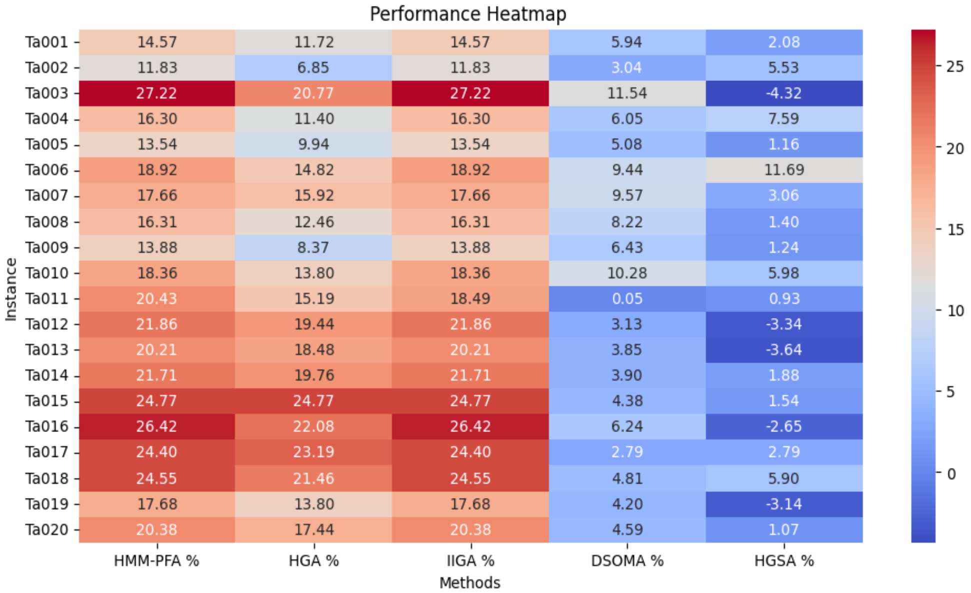

Figure 3. Heatmap of the maximum test results for the Max-Min Ant System

Figure 3 illustrates the averages of the MAX-MIN Ant System run results and the results published by the researchers in the form of a heat map.

Table 4

Comparison of the minimum test results for the Max-Min Ant System

Instance	MMAS	HMM-PFA %	HGA %	IIGA %	DSOMA %	HGSA %
Ta001	1297	14.57	11.72	14.57	5.94	2.08
Ta002	1368	11.70	6.73	11.70	2.92	5.41
Ta003	1161	25.75	19.38	25.75	10.25	–5.43
Ta004	1377	15.32	10.46	15.32	5.16	6.68
Ta005	1283	12.94	9.35	12.94	4.52	0.62
Ta006	1254	18.10	14.04	18.10	8.69	10.93
Ta007	1267	17.05	15.31	17.05	9.00	2.53
Ta008	1283	15.51	11.69	15.51	7.48	0.70
Ta009	1300	13.00	7.54	13.00	5.62	0.46
Ta010	1175	17.19	12.68	17.19	9.19	4.94
Ta011	1702	20.09	14.86	18.16	-0.24	0.65
Ta012	1793	20.80	18.40	20.80	2.23	–4.18
Ta013	1624	19.46	17.73	19.46	3.20	–4.25
Ta014	1498	20.89	18.96	20.89	3.20	1.20
Ta015	1557	24.15	24.15	24.15	3.85	1.03
Ta016	1503	25.88	21.56	25.88	5.79	–3.06
Ta017	1595	23.07	21.88	23.07	1.69	1.69
Ta018	1667	23.40	20.34	23.40	3.84	4.92
Ta019	1684	17.16	13.30	17.16	3.74	–3.56
Ta020	1711	19.87	16.95	19.87	4.15	0.64

Table 4 shows the maximum running values of the MAX-MIN Ant System. In several cases, the results published by the researchers exceeded the maximum running values of the algorithm. But in most cases, MMAS still showed 10-20% better results than those published by the researchers.

Smaller problems (Ta001–Ta010): The MMAS algorithm continued to be effective for smaller problems. In most cases, it produced 10-20% better results than those published. For example, for problem Ta001, the maximum value of 1297 achieved by MMAS is 14.57% better than the published value. For problem Ta002, MMAS achieved an 11.70% improvement. For problem Ta003, MMAS also presented a 25.75% better value. For problems Ta004, Ta005, and Ta006, MMAS also achieved an improvement of more than 10%. For problems Ta007, Ta008, and Ta009, MMAS achieved improvements of 17.05%, 15.51%, and 13.00%, respectively.

Medium problems (Ta011–Ta015): For Ta011, MMAS produced a 20.09% better maximum, while for Ta012 and Ta013, it achieved 20.80% and 19.46% improvements. For Ta014 and Ta015, MMAS also produced good results.

Larger problems (Ta016–Ta020): For larger problems, such as Ta016, Ta017, and Ta018, the researchers' results showed smaller differences.

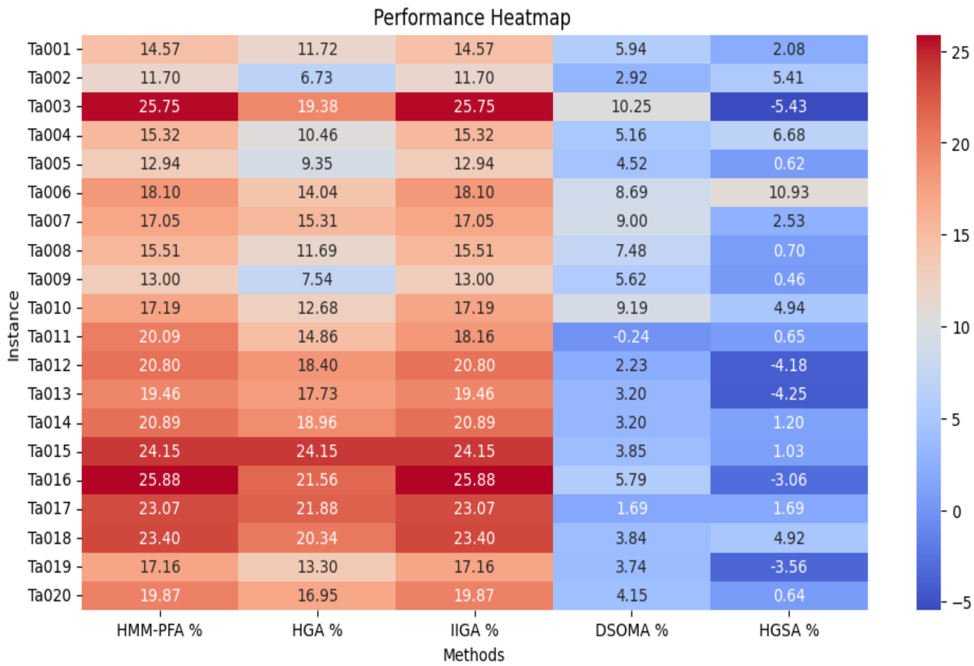

Figure 4. Heatmap of the maximum test results for the Max-Min Ant System

Figure 4 illustrates the maximum of the MAX-MIN Ant System running results and the results published by the researchers with a heat map.

4. Conclusions and future research directions

Various manufacturing problems, such as flow shop scheduling, play an important role in industrial optimization, especially in the design and operation of manufacturing systems. Flow Shop Scheduling is a scheduling task in which multiple machines perform operations on workpieces in a series. All workpieces must be processed through the same machines in a specific order.

In this paper, the performance of the Max-Min Ant System (MMAS) algorithm is investigated. During the research, the Taillard benchmark dataset is solved. Based on the results, a detailed comparison of the algorithm's performance is performed. The test runs are analyzed based on three key indicators, minimum, average and maximum values.

The results of the runs (maximum, average and minimum makespan) for the MMAS algorithm on the Taillard dataset are as follows:

- *Observations on stability:* MMAS provided consistent performance across instances, as the difference between the maximum and minimum values is generally small. For example, for Ta001, all runs gave the same result (Max = Avg = Min = 1297), while for larger and more complex instances (e.g. Ta013) the difference between the best and worst solution was 36 units (Min = 1588, Max = 1624). This indicates that the algorithm is stable, but there is less variability for more difficult instances, which is a typical feature of ACO variants governed by Max–Min pheromone constraints.
- *Average performance:* The average makespan values are close to the minimum in all instances, suggesting that MMAS consistently finds near-optimal solutions. For example, for Ta010, the minimum is 1142, the average is 1163.4, and the maximum is 1175; the variance is therefore only ~33 units, which is a relatively small difference for an example consisting of 10 jobs.
- *Trend analysis with increasing problem size:* Harder problems (e.g. Ta011–Ta020) result in larger makespan values, but even for these, the difference between the average and the minimum is usually around 10–20 units. This shows that MMAS scales well and is able to avoid the traps of local optima, even for larger instances.

One promising direction for future research is to refine the MMAS algorithm and apply it to more complex versions of different manufacturing systems. Although MMAS performed well in solving flow shop scheduling problems in this research, increasing the efficiency of the algorithm could be one of the further research directions. Another research direction is industrial application. Solving problems in a real industrial environment.

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] Wang, L., Zhang, L., & Zheng, D. Z. (2006). An effective hybrid genetic algorithm for flow shop scheduling with limited buffers. *Computers & Operations Research*, 33 (10), 2960–2971. https://doi.org/10.1016/j.cor.2005.02.028

[2] Chen, C. L., Vempati, V. S., & Aljaber, N. (1995). An application of genetic algorithms for flow shop problems. European Journal of Operational Research, 80 (2), 389–396. https://doi.org/10.1016/0377-2217(93)E0228-P

[3] Li, B. B., & Wang, L. (2007). A hybrid quantum-inspired genetic algorithm for multiobjective flow shop scheduling. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 37 (3), 576–591. https://doi.org/10.1109/TSMCB.2006.887946

[4] Yagmahan, B., & Yenisey, M. M. (2008). Ant colony optimization for multi-objective flow shop scheduling problem. *Computers & Industrial Engineering*, 54 (3), 411–420. https://doi.org/10.1016/j.cie.2007.08.003

[5] Alaykýran, K., Engin, O., & Döyen, A. (2007). Using ant colony optimization to solve hybrid flow shop scheduling problems. *The International Journal of Advanced Manufacturing Technology*, 35, 541–550. https://doi.org/10.1007/s00170-007-1048-2

[6] Yagmahan, B., & Yenisey, M. M. (2010). A multi-objective ant colony system algorithm for flow shop scheduling problem. *Expert Systems with Applications*, 37 (2), 1361–1368. https://doi.org/10.1016/j.eswa.2009.06.105

[7] Wang, L., Pan, Q. K., & Tasgetiren, M. F. (2011). A hybrid harmony search algorithm for the blocking permutation flow shop scheduling problem. *Computers & Industrial Engineering*, 61 (1), 76–83. https://doi.org/10.1016/j.cie.2011.02.013

[8] Wang, L., Pan, Q. K., & Tasgetiren, M. F. (2010). Minimizing the total flow time in a flow shop with blocking by using hybrid harmony search algorithms. *Expert Systems with Applications*, 37 (12), 7929–7936. https://doi.org/10.1016/j.eswa.2010.04.042

[9] Pan, Q. K., Wang, L., & Gao, L. (2011). A chaotic harmony search algorithm for the flow shop scheduling problem with limited buffers. *Applied Soft Computing*, 11 (8), 5270–5280. https://doi.org/10.1016/j.asoc.2011.05.033

[10] Stützle, T., & Hoos, H. H. (2000). MAX–MIN ant system. *Future Generation Computer Systems*, 16 (8), 889–914. https://doi.org/10.1016/S0167-739X(00)00043-1

[11] Stutzle, T., & Hoos, H. (1997, April). MAX-MIN ant system and local search for the traveling salesman problem. In *Proceedings of 1997 IEEE International Conference on Evolutionary Computation (ICEC'97)*, 309–314, IEEE. https://doi.org/10.1109/ICEC.1997.592327

[12] Yagmahan, B., & Yenisey, M. M. (2010). A multi-objective ant colony system algorithm for flow shop scheduling problem. *Expert Systems with Applications*, 37 (2), 1361–1368. https://doi.org/10.1016/j.eswa.2009.06.105

[13] Tang, J., Ma, Y., Guan, J., & Yan, C. (2013). A max–min ant system for the split delivery weighted vehicle routing problem. *Expert Systems with Applications*, 40 (18), 7468–7477. https://doi.org/10.1016/j.eswa.2013.06.068

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

[15] 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*. https://doi.org/10.1155/2018/1973604

[16] 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