Production Systems and Information Engineering
Volume 12 (1), pp. 29–40.
https://doi.org/10.32968/psaie.2024.1.3

EFFICIENCY ANALYSIS OF THE ANT SYSTEM ALGORITHM ON THE FLOW SHOP SCHEDULING PROBLEM

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

Abstract: This article analyzes the effectiveness of the Ant System (AS) on the benchmark dataset of a production scheduling task, the Flow Shop Scheduling Problem (FSS). The Ant System (AS) algorithm is one of the algorithms of the Ant Colony Optimization (ACO) family, which is based on the behavior of ants. This is a population algorithm that iteratively improves individual elements of the population until the stopping condition is met. Flow Shop Scheduling is a task in which specific jobs must be performed on specific machines and the goal is to minimize the makespan. The article presents the FSS task, the AS algorithm, and the methods used for efficiency analyses. The tests showed that the Ant System algorithm is effective for the Flow Shop Scheduling task as a discrete production scheduling task.

Keywords: *Ant System, Flow Shop Scheduling, efficiency analysis, fitness landscape analysis*

1. Introduction

Efficient production is an important driver of industrial development. In this article, a production scheduling problem, the Flow Shop Scheduling Problem [1] is presented. The Ant System [2] algorithm was used for the solution. The Ant System algorithm is a discrete optimization problem, originally developed for the Traveling Salesman Problem, but over the years it has been applied to many other problems. It first appeared in Marco Dorigo's article "Optimization, Learning and Natural Algorithms" in 1992. The algorithm is a member of the Ant Colony Optimization (ACO) family. *Figure 1* illustrates the publications published in connection with the Ant System from 2010, for which the data was extracted with the help of Google Scholar.

Regarding the number of publications, the number of articles increased greatly from 2011 to 2013. From 2021 to 2023, however, the number of articles published by researchers decreased significantly.

1740 articles were published in 2010, 1790 in 2015, 1800 in 2020, and 1610 in 2013. The Ant System algorithm has been used for many discrete optimization problems over the years, such as Job-Shop Scheduling [2], Vehicle Routing Problem [3], Quadratic Assignment Problem [4], Network Design Problem [5], Traveling

Salesman Problem [6], University Course Timetabling Problem [7], Sequential Ordering Problem [8] etc.

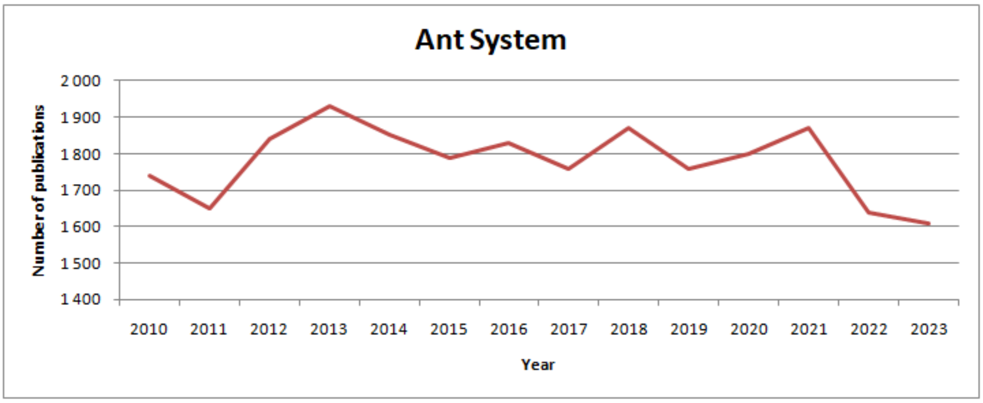

Figure 1. Literature review for the Ant System

The article consists of the following parts. The second chapter describes the Flow Shop Scheduling Problem and the Ant System algorithm. The third chapter presents the test runs of the article and their analysis. The last chapter contains conclusions and future research direction.

2. The Flow Shop Scheduling Problem and the Ant System algorithm

2.1. Flow Shop Scheduling Problem

The Flow Shop Scheduling Problem [1] is a discrete production scheduling problem. The number of machines and jobs are given during the task. Each job must be performed once on each machine during the task. A single machine can do one job at a time, and once it has started, it must finish it, and only then can it start another job. Each job and machine has a processing time (makespan). The goal is to minimize the makespan. Therefore, a work order must be created where the makespan is minimal.

2.2. Ant System Algorithm

Ant System [2] is a member of the Ant Colony Optimization (ACO) family of algorithms. ACO algorithms maintain a population of solutions. ACO algorithms are inspired by the natural behaviour of ants. Ants deposit pheromones along their route. When choosing a route, ants are more likely to choose the route with higher pheromone content, because it is more attractive to them. However, the pheromone also evaporates, so the algorithm must also take care of updating the pheromone.
The Ant System algorithm is an improvement of the Ant Colony System, some steps and formulas are the same in the two algorithms.

Ant System only performs one type of pheromone update. The route construction formula is the same as the Ant Colony System formula, i.e.:

$$p_{ij}^{k} = \frac{\left[\tau_{ij}(t)\right]^{\alpha} * [\eta_{ij}]^{\beta}}{\sum_{l \in N_i^k} [\tau_{il}(t)]^{\alpha} * [\eta_{il}]^{\beta}} \quad \text{if } j \in N_i^k$$

The formula has the following notations:
$\eta_{ij} = \frac{1}{d_{ij}}$: the reciprocal of the distance between the two routes, where d_{ij} means the distance
$\tau_{ij}(t)$: pheromone content
(i, j): edge pair
α and β: the effects of the pheromone and the distance are determined by these parameters.
N_i^k: the nodes that the ant has not yet visited.

The pheromone update formula is:

$$\tau_{ij}(t+1) = (1-\rho) * \tau_{ij}(t) + \sum_{k=1}^{m} \Delta\tau_{ij}^{k}(t)$$

where:
$0 < \rho \leq 1$: gives the rate of evaporation
$\Delta\tau_{ij}^{k}(t) = \begin{cases} \frac{1}{L^k(t)} & \text{if ant k. travels through the edge } (i,j) \\ 0 & \text{else} \end{cases}$:
$L^k(t)$: the route of ant k.
$\tau_{ij}(t)$: pheromone content of edge (i, j) in iteration t

3. Test results

This chapter presents the test results. I performed two types of tests, during one I ran the tests on the benchmark data, and during the other, I used fitness landscape analysis techniques. I used the Taillard benchmark dataset [9] to run the tests. The Taillard benchmark dataset comprises problems of various sizes, with job numbers ranging from 20 to 500 and machine numbers from 5 to 20. Variable-sized problems imply that smaller ones can be completed more quickly, while larger ones demand a longer processing time. This results in a combinatorial explosion, causing the number of potential schedules to increase exponentially. As a result, identifying the optimal solution for larger datasets is an exceptionally challenging task.

The AS column indicates the result of the presented Ant System algorithm. In addition, I compared the results of algorithms published by other authors who also ran their algorithms on this Taillard benchmark data set. These are the following:

- HMM-PFA: Hormone Modulation Mechanism Flower Pollination Algorithm [10]
- HGA: Hybrid Genetic Algorithm [11]

- IIGA: Improved Iterated Greedy Algorithm [11]
- DSOMA: Discrete Self-Organizing Migrating Algorithm [11]
- HGSA: Hybrid Genetic Simulated Annealing [11]
- IWO: Invasive Weed Optimization Algorithm [12]

The table shows the relative performances.

Table 1. Test result comparisons

		Relative performance					
Instance	**AS**	**HMM-PFA [10] %**	**HGA [11] %**	**IIGA [11] %**	**DSOMA [11] %**	**HGSA [11] %**	**IWO [12] %**
Ta001	1297	114.57	111.72	114.57	105.94	102.08	107.09
Ta002	1367	111.78	106.8	111.78	103	105.49	–
Ta003	1140	128.07	121.58	128.07	112.28	96.32	–
Ta004	1375	115.49	110.62	115.49	105.31	106.84	–
Ta005	1254	115.55	111.88	115.55	106.94	102.95	–
Ta006	1241	119.34	115.23	119.34	109.83	112.09	–
Ta007	1259	117.79	116.04	117.79	109.69	103.18	–
Ta008	1258	117.81	113.91	117.81	109.62	102.7	–
Ta009	1284	114.41	108.88	114.41	106.93	101.71	–
Ta010	1165	118.2	113.65	118.2	110.13	105.84	–
Ta011	1692	120.8	115.54	118.85	100.35	101.24	130.44
Ta012	1769	122.44	120.01	122.44	103.62	97.12	–
Ta013	1610	120.5	118.76	120.5	104.1	96.58	–
Ta014	1468	123.37	121.39	123.37	105.31	103.27	–
Ta015	1550	124.71	124.71	124.71	104.32	101.48	–
Ta016	1508	125.46	121.15	125.46	105.44	96.62	–
Ta017	1583	124.01	122.8	124.01	102.46	102.46	–
Ta018	1624	126.66	123.52	126.66	106.59	107.7	–
Ta019	1682	117.3	113.44	117.3	103.86	96.55	–
Ta020	1704	120.36	117.43	120.36	104.58	101.06	–

The table shows that Ta001 was given a fitness value of 1297 by the Ant System. It was 14% better than HMM-PFA, 11% better than HGA, 14% better than IIGA, 5% better than DSOMA, 2% better than HGSA, and 7% better than IWO. The authors of the IWO article only published the results of some benchmark data. For Ta002, the AS score was 1367, which is 11% better than HMM-PFA, 6% better than HGA, 11% better than IIGA, 3% better than DSOMA, and 5% better than HGSA. The AS algorithm gave 1140 results for the Ta003 dataset. It was 28% better than HMM-PFA and IIGA, but 3% worse than HGSA. For Ta004 the AS gave a fitness value of 1375. This was 15% better than HMM-PFA and IIGA.

The table shows that the HSGA results are the closest to the Ant System results, and in some cases, HSGA gives better results than the Ant System. Such are the benchmark cases Ta003, Ta012, Ta013, Ta016, Ta019. Then DSOMA approximates Ant System results, followed by HMM-PFA, HGA, and IIGA algorithms. In the case of IWO, the results could only be compared with two test runs.

Table 2. Test results

Algorithm	Number of data rows (on which the comparisons were made)	Number of better results
HMM-PFA	20	20
HGA	20	20
IIGA	20	20
DSOMA	20	20
HGSA	20	15
IWO	2	2

The table shows how many data sets could be compared with each comparison algorithm, and for how many data sets Ant System gave better results.

It can be seen that, except for HGSA, the Ant System algorithm provided better results than the other algorithms during all test runs. The Ant System algorithm from HGSA performed better in most of the runs, in 15 out of 20 cases. Where HSGA was better, it was also better by only a few percent.

The results of the Ant System (AS) algorithm regarding the search space are shown in the table below. The following metrics were defined [13]: fitness values (FV), average of fitness distances (AFD), average of Hamming distances (AHD), average of basic swap distances (ABSD), fitness distances of the best solution (FDBS), Hamming distances of the best solution (HDBS), basic swap sequence distances of the best solution (BSSDS), cost density (CD), fitness distance of filtered global optima (FDFGO), Hamming distance of filtered global optima, (HDFGO), basic swap sequence distance of filtered global optima (BSDFGO).

Table 3. Fitness landscape results

Ant System (AS)			
		Ta001	
	Distance	LB	UB
FV		1324	1370
AFD	Fitness	8.38	37.62
AHD	Hamming	6.12	17.56
ABSD	BSS	5.09	15.57
FDBS	Fitness	8.38	37.62
HDBS	Hamming	6.12	17.56
BSSDS	BSS	5.09	15.57
CD		1.0	64.0
FDFGO	Fitness	8.38	37.62
HDFGO	Hamming	6.12	17.56
BSDFGO	BSS	5.09	15.57
		Ta002	
	Distance	LB	LB
FV		1383	1383
AFD	Fitness	2.48	2.48
AHD	Hamming	6.41	6.41
ABSD	BSS	5.69	5.69
FDBS	Fitness	2.48	2.48
HDBS	Hamming	6.41	6.41
BSSDS	BSS	5.69	5.69
CD		1.0	1.0

FDFGO	**Fitness**	2.48	2.48
HDFGO	**Hamming**	6.41	6.41
BSDFGO	**BSS**	5.69	5.69
		Ta003	
	Distance	**LB**	**LB**
FV		1163	1163
AFD	**Fitness**	16.13	16.13
AHD	**Hamming**	9.64	9.64
ABSD	**BSS**	7.97	7.97
FDBS	**Fitness**	16.13	16.13
HDBS	**Hamming**	9.64	9.64
BSSDS	**BSS**	7.97	7.97
CD		2.0	2.0
FDFGO	**Fitness**	16.13	16.13
HDFGO	**Hamming**	9.64	9.64
BSDFGO	**BSS**	7.97	7.97
		Ta004	
	Distance	**LB**	**LB**
FV		1405	1405
AFD	**Fitness**	5.98	5.98
AHD	**Hamming**	8.01	8.01
ABSD	**BSS**	7.53	7.53
FDBS	**Fitness**	5.98	5.98
HDBS	**Hamming**	8.01	8.01
BSSDS	**BSS**	7.53	7.53
CD		1.0	1.0
FDFGO	**Fitness**	5.98	5.98
HDFGO	**Hamming**	8.01	8.01
BSDFGO	**BSS**	7.53	7.53
		Ta005	
	Distance	**LB**	**LB**
FV		1311	1311
AFD	**Fitness**	0.14	0.14
AHD	**Hamming**	0.2	0.2
ABSD	**BSS**	0.18	0.18
FDBS	**Fitness**	0.14	0.14
HDBS	**Hamming**	0.2	0.2
BSSDS	**BSS**	0.18	0.18
CD		1.0	1.0
FDFGO	**Fitness**	0.14	0.14
HDFGO	**Hamming**	0.2	0.2
BSDFGO	**BSS**	0.18	0.18

For the Ant System algorithms, the results of Ta001 are between 1370 and 1324, the results of Ta002 are between 1420 and 1383, the fitness values for Ta003 are between 1216 and 1163, the values of Ta004 are between 1465 and 1405, while the values of Ta005 are between 1325 and 1311.

The Cost Density values are quite high, which means that, for example, for Ta001 the algorithm had 64 equal solutions during the iterations, for Ta002 67 equal solutions, for Ta003 46 equal solutions, for Ta004 54 solutions, while Ta005 had almost all the same solutions, barely the algorithm improved during the iterations.

The lower bounds of the average fitness distances are small, while the upper bounds are larger, which also indicates that larger improvements have been made during each iteration. The lower limits of the Hamming and BSS distances are also small, and the upper limits are large, which indicates that the algorithm maps the search space well.

The following figures show the results of the above table for Ta001.

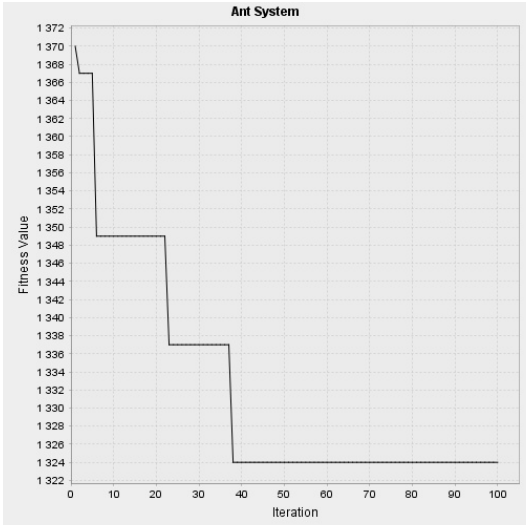

Figure 2. Iteration diagram

Figure 2 shows the iteration diagram. It can see how much the results of the algorithm improved in the first 100 iterations. In the first iteration, there were still 1370 values, until the 40th iteration, the values continued to decrease, up to 1324. From the 40th iteration, we can't see any improvement.

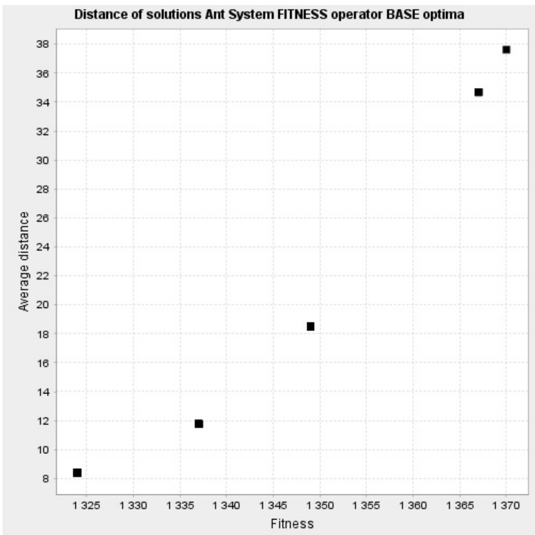

Figure 3. Fitness distance solutions

According to the fitness distance of the solutions diagram *(Figure 3)*, the values are condensed into a few points. The x-axis represents the fitness values, and the y represents the average distance of that fitness value from the other solutions. The largest is at 1370, here this distance is 38.

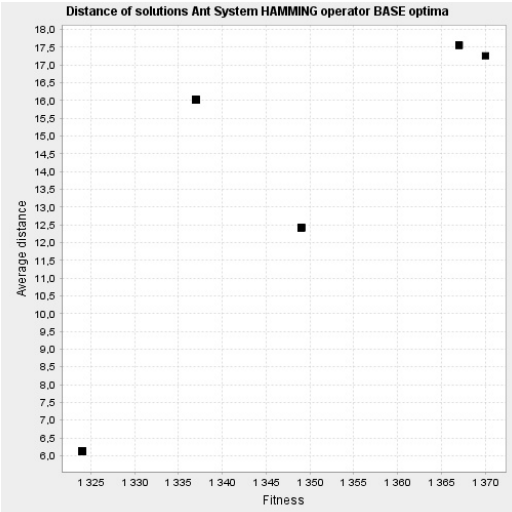

Figure 4. Average Hamming distance of solutions

Figure 4 shows the average Hamming distances. The x-axis represents the fitness values and the y-axis represents the average Hamming distances. It can be seen that here, too, the results are condensed into a few points. At a fitness value of 1325, the average Hamming distance is 6, at 1370 it is around 17.

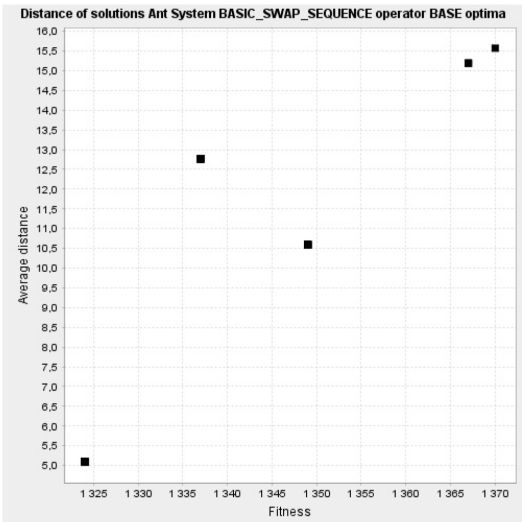

Figure 5. Average Basic Swap Sequence distance of solutions

Figure 5 shows the average basic swap sequence distance. The x-axis shows the fitness values and the y-axis shows the average basic swap sequence distances. It can be seen that the average distance at 1325 fitness is 5, at 1350 fitness it is 10.5, at 1357 fitness it is 12.5, and at 1370 fitness it is 15.5.

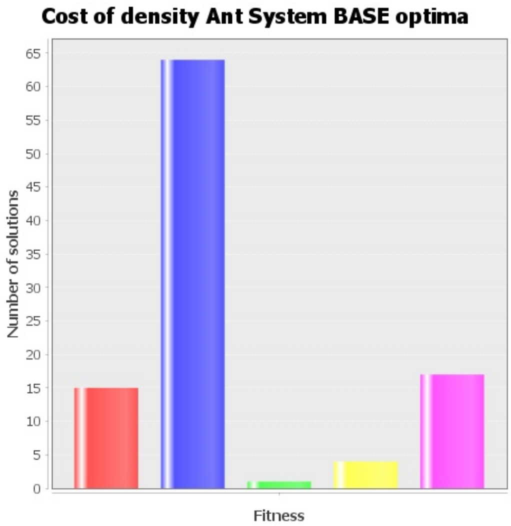

Figure 6. Cost density values

Figure 6 shows the cost density values. The bar chart shows that there are 5 different fitness values. One fitness value is contained by 15 solutions, the other by 65, the third by 1, the fourth by 5, and the fifth by 19.

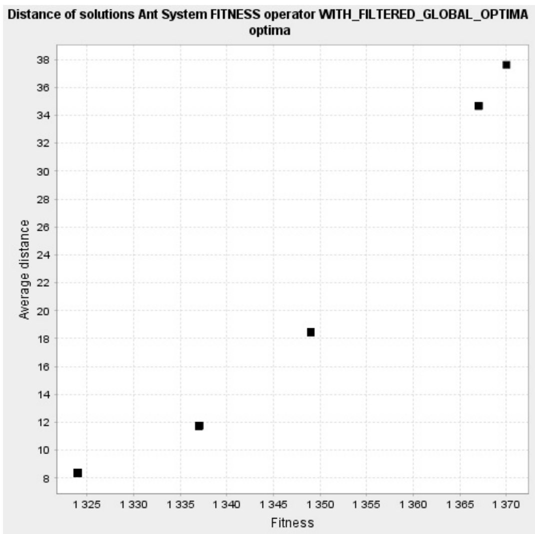

Figure 7. Fitness distance of filtered global optima

The result of the fitness distance of filtered global optima is shown in *Figure 7*. The figure shows the fitness values on the x-axis and the average fitness distances on the y-axis. These are also condensed into a few points, and their value is between 8–38.

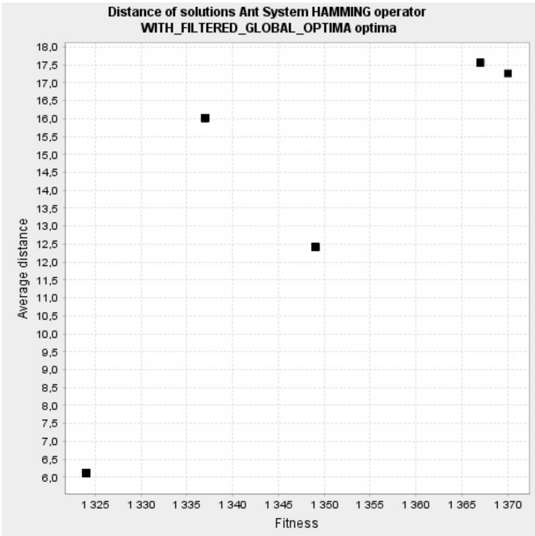

Figure 8. Hamming distance of filtered global optima

The Hamming distance of the filtered global optima figure is shown in *Figure 8*. The x-axis is the fitness values and the y-axis is the average distances. The average Hamming distances here are between 6 and 18.

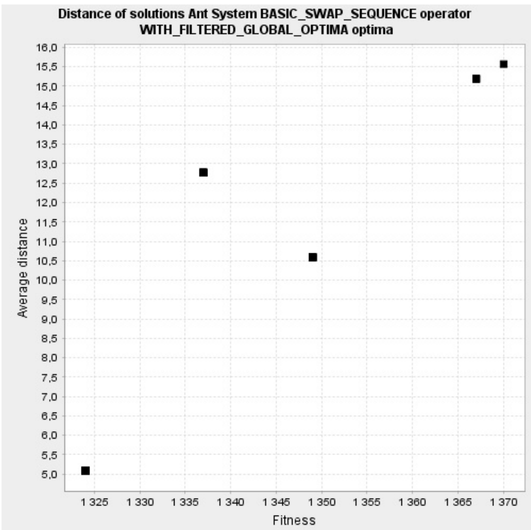

Figure 9. Basic Swap Sequence distance of filtered global optima

Figure 9 shows the basic swap sequence distance of filtered global optima. The x-axis shows the fitness values and the y-axis shows the average distances. Average distances are between 5 and 15.5.

4. Conclusions and future work

This paper examines the effectiveness of the Ant Sytem algorithm for the Flow Shop Scheduling Problem. The Ant System algorithm, the Flow Shop Scheduling Problem, was presented in the article. Then test results were detailed. First, a table is presented, which contains the fitness values of the AS and the comparison of the values with other algorithms. Here we can see that the Ant System algorithm solved the task efficiently. Then, the fitness landscape was analyzed for the iterations of the Ant System. Based on the running tests, the Ant System algorithm efficiently solved this production scheduling task. The future research direction is the investigation of new heuristic algorithms for discrete production scheduling tasks, and the evaluation of their results.

Acknowledgment

Supported by the ÚNKP-23-4-II. New National Excellence Program of the Ministry for Culture and Innovation from the source of the National Research, Development and Innovation Fund.

Új Nemzeti Kiválósági Program

Ministry of Culture and Innovation

References

[1] Kulcsár, G., & Erdélyi, Ferenc (2006). Modeling and solving of the extended flexible flow shop scheduling problem. *Production Systems and Information Engineering*, 3, pp. 121–139.

[2] Colorni, A., Dorigo, M., Maniezzo, V., & Trubian, M. (1994). Ant system for job-shop scheduling. *JORBEL-Belgian Journal of Operations Research, Statistics, and Computer Science*, 34 (1), pp. 39–53.

[3] Bullnheimer, B., Hartl, R. F., & Strauss, C. (1999). Applying the ant system to the vehicle routing problem. In: *Meta-heuristics: Advances and trends in local search paradigms for optimization*. pp. 285–296. https://doi.org/10.1007/978-1-4615-5775-3_20

[4] Maniezzo, V., & Colorni, A. (1999). The ant system applied to the quadratic assignment problem. *IEEE Transactions on knowledge and data engineering*, 11 (5), pp. 769–778. https://doi.org/10.1109/69.806935

[5] Poorzahedy, H., & Abulghasemi, F. (2005). Application of ant system to network design problem. *Transportation*, 32 (3), pp. 251-273. https://doi.org/10.1007/s11116-004-8246-7

[6] 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) IEEE*, pp. 309–314. https://doi.org/10.1109/ICEC.1997.592327

[7] Socha, K., Knowles, J., & Sampels, M. (2002). A max-min ant system for the university course timetabling problem. In: *Ant Algorithms: Third International Workshop*. ANTS 2002 Brussels, Belgium, September 12–14, 2002 Proceedings 3, Springer Berlin Heidelberg, pp. 1–13. https://doi.org/10.1007/3-540-45724-0_1

[8] Gambardella, L. M., & Dorigo, M. (1997). *HAS-SOP: Hybrid ant system for the sequential ordering problem*. Istituto Dalle Molle Di Studi Sull Intelligenza Artificiale.

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

[10] 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

[11] 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), p. 2621. https://doi.org/10.3390/app8122621

[12] Zhou, Y., Chen, H., & Zhou, G. (2014). Invasive weed optimization algorithm for optimization no-idle flow shop scheduling problem. *Neurocomputing*, 137, pp. 285–292. https://doi.org/10.1016/j.neucom.2013.05.063

[13] Agárdi, A., Kovács, L., & Bányai, T. (2021). The fitness landscape analysis of the ant colony system algorithm in solving a vehicle routing problem. Academic Journal of Manufacturing Engineering, 19 (2).