ISSN: 0974-5823 Vol. 6 No. 3 December, 2021

International Journal of Mechanical Engineering

Ameliorated Ant Algorithm(AAA) for load balanced Task Scheduling in Manufacturing Grid

Avijit Bhowmick[1]
[1]Budge Budge Institute of Technology

Arup Kumar Nandi[2]
[2] Senior Principal Scientist, CSIR-CMERI

Goutam Sutradhar[3]
[3]Director, National Institute of Technology, Manipur

***Corresponding author:** Avijit Bhowmick, Email: mr.avijit.bhowmick@gmail.com*

Avijit Bhowmick, Arup Kumar Nandi, Goutam Sutradhar/ Ameliorated Ant Algorithm(AAA) for load balanced Task Scheduling in Manufacturing Grid

Abstract

In a Grid-based environment like Manufacturing Grid, one of the most critical issues is how to divide the workload equally. Heuristic techniques are necessary since locating the optimal schedules in this situation is an NP-hard task. One of the Shared, enlarged, and error-absorbing features of the heuristic scheduling algorithms is known as the Ant algorithm. Based on its projections for future resource utilization, it creates timetables. To maximize the use of available resources, this is a must. As part of this study, we provide an improved ant algorithm named Ameliorated ant algorithm(AAA) for scheduling activities to optimize throughput while controlling costs. Aside from that, several scheduling approaches are examined and contrasted in simulations. It was revealed that the our novel improved ant algorithm method(AAA) performed effectively and efficiently with respect to specific parameters.Because of the ant-price algorithm's factor, our this new scheduling strategy is better suited to a broader range of applications of MGrid.

1. Introduction

Although computing and networking have made tremendous development in the last decade and there are still problems in research, engineering, and business that can't be handled with today's most capable supercomputers. The Grid concept is comprised of a large range of resources, including computing systems, data storage, data sources, and sophisticated manufacturing equipment that are scattered out globally and belong to different enterprises.The Internet serves as a conduit for connecting all of these resources.There are different types of Grid like Data Grid,Computational Grid,Manufacturing Grid.A typical specific kind of grid that is grid-oriented and built on collaborative manufacturing environments is known as a manufacturing Grid(MGrid). Global users must be able to utilise a resource or service in an MGrid in an equitable and systematic manner, just as if they were consuming locally available resources and services.Now,tasks assigned to the Manufacturing Grid system must be completed on time. An automated task scheduler in more advanced Grid systems will help select which machines are most suited to perform a given task.

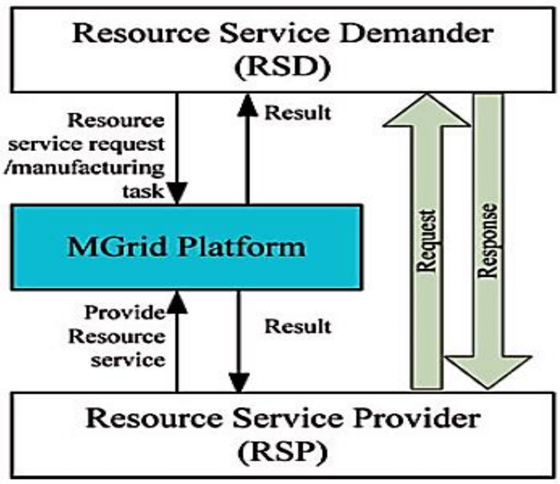

Fig.1 Manufacturing Grid platform[1]

According to Tao et al.[1] there are primarily two types of users in MGrid: (a) resource enterprises

(b) user enterprises. In the first case, resource enterprises make available in MGrid its idle resources, products, manufacturing capabilities & also offers the service associated to manufacturing to satisfy customer needs and demands. In order to create the virtual cooperation manufacturing network, resource users looks for the best-fit resource of manufacturing & service.So,the task scheduling approach significantly impacts this resource selection, which is why it is critical to decrease the duration and expense of activity execution. The Grid system's throughput and resource efficiency are determined by the scheduling rules implemented by the scheduler. The scheduler ensures that the system's resources are distributed equitably so that the system's throughput and the use of its resources can be maximized.

Problem Statement :

The intention of the job scheduling issue is to identify the best way to distribute n tasks amongst m resources in order to enhance performance as a whole.We'll now provide the task allocation problem's formulation.Let the i resources are denoted $R = \{R_1, R_2, R_3, \ldots R_i\}$ and j tasks are represented by $T = \{T_1, T_2, T_3 \ldots T_j\}$.Now,the time for execution of each and every task on every resource is retained in a matrix structure.The objective to find an optimal solution to have the best load balance and minimizing makespan as well as cost reduction.

Related work :

In the year 1991, ant colony technique was first introduced through his research paper by Dorigo[2].It is inspired by how ant colonies hunt [3][4] and [5]. It is based on just how ants actually behave for food searching or other activitiesThe quantity of pheromone on each route determines the possibility that other ants would take that way, which is how blind insects like ants find food. A little amount of pheromone is released along the route by each passing ant, which causes the pheromone on a shorter path to climb fast. All the ants will eventually decide on the quickest route. The most wellknown problem like travelling salesman problem (TSP), internet backbone routing, automotive tracking problem, and other optimization-related issues have all been solved using ACO [6]. ACO do always try to discover the quickest route between ant nests and sources of food. Each ant emits pheromone while moving along a pheromone density.The pheromone that has a dense population in short pathways is produced by this activity. Furtheremore pheromone is less dense across greater distances. [7][8]. The most prominent features and benefits of ACO [6][9][10][11][12] comprise competent effective solution for the Traveling Salesman Problem (TSP) and related issues and speedy finding of excellent solutions. ACO's [5, 8, 9, and 12] drawbacks include their lengthy operating times and susceptibility to becoming trapped in a local optimal solution. Many studies [14][15][10][16] by lot of researchers boosted probing multiplicity by incorporating the mutation approach into the ant colony process so as to address the limitations of this method. The experiment findings from these studies revealed that mutation methodologies can improve standard ACO's searching performance and yield better results. To enhance searching performance from lot of unknown objects with parameters, many algorithms , such as Particle Swarm Optimization [17][18] include the different kind of mutation strategy.

An array of NP problems have been solved using this method, including the TSP, for assignment kind of Problem, the Scheduling Problem, and the problems of Graph coloring. A higher level of complexity can be achieved using the algorithm's inherent parallelism. It is possible to scale, share, and deal with errors with the algorithm which is based on novel empirical, extrapolative scheduling method.

Assumptions and factors are considered :

The following factors are taken into account in the suggested scheduling method: Each job's processing requirementsthe current status of the resources that are available, the load that is now being placed on those resources, and the computing costs related to those resources.All the nodes connected to the Manufacturing Grid are CAD computing machine.Here we are Processing-

1. Modifications of fundamental Ant algorithm for scheduling jobs :

The resources with a high probability of being available are free, and the task may be assigned to a resource with a low probability of being accessible. As the method is designed to maximize efficiency, this is problematic. While other resources may be available, a resource with a high probability of becoming free will be assigned duties, and these tasks will have to wait until that resource is free, even if all other resources are available. It can be prevented by always giving jobs to the help with the best likelihood of being available at completion time. However, if this isn't the case, the scheduler will pick another resource and repeat these steps. The first step is to choose a threshold, and this is a crucial one. A faster processing time is achieved by considering resources with the best chance of being used. However, the cost of processing may be higher than when utilizing the ant method.

The conventional ACO is then supplemented with a new parameter termed cutoff probability in order to improve the computing environment. The threshold probability value is selected using standard ACO from a list of determined values.The acceptance of an object having a high pheromone value is assisted by a high probability value, which decreases the time required for jobs to execute on the chosen cluster resource, hence increasing resource usage. This number functions as a constraint in the proposed method which is needed that when selecting a resource, the transitioning probability value must be higher than the threshold probability value.

Proposed Task Scheduling Algorithm:

If any new resource "℘ " enters into Manufacturing Grid(MGrid), a request is issued for the resource to transmit information about its performance metrics. There are a total number of PEs and the amount of MIPS each PE has i.e each CAD machine. The links pheromone is set up as follows when the resource monitor confirms that the parameters are correct: Every each time a whenever a new resource is added to the MGrid, an existing resource fails, a job is allocated. The pheromone on the route from the scheduling centre to the pertinent resource changes when a study is returned.

Proposed Task Scheduling Algorithm

1. A resource '℘' must give its performance characteristics, number of processing units, and other information when it enrols into the MGrid environment. The links pheromone is initialised by resource analyzer, and these parameters are tested for reliability.

$$\mathscr{L}(0) = x * \eta + m * x + s/t_t \qquad \text{---(1)}$$

Where x = No. of processing unit

η = processing capacity /second

s = parameter size

t_t = transfer time

m = memory or storage size

$$\mathscr{L}_{\wp}^{\ new} = \boldsymbol{\beta}.\ \mathscr{L}_{\wp}^{\ old} + \Delta\ \mathscr{L}_{\wp}. \qquad \text{---(2)}$$

$\Delta\ \mathscr{L}_{\wp}$ = change of pheromone on the path

$\boldsymbol{\beta}$ = permanence ($0 < \boldsymbol{\beta} < 1$)

($1- \boldsymbol{\beta}$) – Pheromone evaporation whenever a task is issued to '℘',

$\Delta\ \mathscr{L}_{\wp} = -y$, y is the computation and transmission quality of the task.

As soon as the task fruitfully coming back from the resource '℘'

$\Delta\ \mathscr{L}\wp$ = Ee $*$ ý, Ee – factor concerned encouragement

When task is not brought back fruitfully from the resource '℘',

$\Delta\ \mathscr{L}_{\wp} = \mathbf{P}_f * y$, $\mathbf{P}_f$ – Penalize factor

A new computation of the task allocation probability for each resource would be get ready according to

$$\mathscr{L}\wp(t) = [\ \boldsymbol{\beta}(\mathbf{t})\boldsymbol{a}\ ] * [Z(t)\mathbf{b}\ ] \ /\ \sum[\ \mathscr{L}_v(t)^a * [z]^b \qquad (3)$$

= 0 others

$\mathscr{L}\wp(t)$ is the level of pheromones on the route from the planned centre to the resource '℘'

$Z(t)$ is the resource's inherent performance, that is $\mathscr{L}\wp$ (0).

The parameter values are considered here as : $\boldsymbol{a}$= 0.5, $\mathbf{b}$= 0.5, $\boldsymbol{\beta}$=0.5, Ee = 1.1, $\mathbf{P}_f$=0.8.

$\boldsymbol{a}$ - the significance of pheromone

$\mathbf{b}$ - the significance of resource distinctive characteristics

This provides resource probabilities for said grid, which aids in job scheduling further.

4 . The scheduler discovers '℘' where new-fangled task 'ɱ' is needs to be scheduled at time 't' with probability equal to its conforming $\boldsymbol{\beta}^{ɱ}(t)$, till $\boldsymbol{\beta}_i^{\ ɱ}(t) - \boldsymbol{\beta}\wp^{\ ɱ}(t) \leq Q_i$

Where i becomes that resource which is having maximum $\boldsymbol{\beta}^{ɱ}(t)$, Qi is the cutoff threshold point that will be used as 1/ (no. of resources.)

5 . The scheduler locates a resource named "u," taking each resource once at a time in the rising sequence of $\boldsymbol{D}_u /\ \mathrm{Mips}_u$, until $|\ \boldsymbol{\beta}\wp^{\ ɱ}(t) - \boldsymbol{\beta}_u^{ɱ}(t)\ | \leq Q_u * e$

$Q_{u=}$ Qi- Qi ($\boldsymbol{\beta} i$ ^{ɱ}(t)- $\boldsymbol{\beta}\wp$ ^{ɱ}(t) and ɭ is the factor concerns with cost lessening and this is nominated by the users of the concerned Grid who actually responsible of task submission 0 ≤ ɭ ≤1

6 .The scheduler assigns the newly created task ‘ɱ’to the resource "u."

2. Implementation using GridSim

The GridSim toolkit has a scheduling mechanism that can be used to run simulations. Information about Grid users and resources is provided during the simulation's initialization procedure. To begin the simulation, resources must be generated, and these resources must be used to meet users' needs. To build their tasks, the user must select their specifications The GridSim simulation will then begin.

It begins the procedure by gathering information about the Grid resources that have been established in the system's resource creation section. Every time work has to be scheduled, the scheduler estimates the existing MGrid resources that are potentially accessible based on the pheromone values of each resource (which are obtained from the scheduling hub).The scheduler then uses the algorithm to select a resource to assign to the job and adds it to the schedule. Pheromone value decreases when resources are given a job due to the quality of the work's computation and transfer procedures. Grid resources handle assigned tasks, which report their findings to the user who gave them the assignment in the first place.

The system will then compile all of the tasks that have been accomplished and deliver them to the appropriate user. The pheromone value of the resource to which it was transferred may rise due to the transfer. The resource's pheromone level will drop if the job doesn't go as anticipated.

3. Experiments, Results, and Discussion

Various scheduling methods are studied and evaluated using simulations, which consider both the amount of time it takes and the amount of money it costs.Jobs are received from the users U1 to U50 and resources are available R1 to R25.Tables provide definitions of the attributes of the client tasks and the resources utilized during the simulation. The database for the simulation has these tables.

Resources	**PEs**	**MIPS**	**Comn. Rate**	**Cost**
R1	2	160	20	1
R2	4	350	40	2
R3	6	500	50	2
R4	1	80	20	1
R5	1	80	20	1
R6	2	350	20	1
R7	4	400	60	3
R8	6	550	80	4
R9	2	180	30	2
R10	4	550	60	3
R11	2	180	20	1
R12	4	560	80	4
R13	8	700	80	4
R14	6	520	50	3
R15	8	720	80	4
R16	6	490	60	3
R17	2	160	20	2
R18	1	80	20	1
R19	1	80	20	1
R20	2	160	20	1
R21	4	560	80	4
R22	1	80	20	1
R23	2	160	20	1
R24	4	400	60	3
R25	2	180	30	3

Table 1 Resources and characteristics

Users	**Length of Job**	**Size of Job**	**Size of Output**
U1 + 26	80000	200	80
U2+27	75000	120	45
U3+28	50000	80	35
U4+29	65000	90	40
U5+30	45000	80	20
U6+31	70000	40	50

U7+32	4400	80	10
U8+33	67000	70	30
U9+34	55000	80	50
U10+35	75000	60	20
U11+36	5000	60	20
U12+37	40000	80	50
U13+38	50000	50	30
U14+39	65000	40	20
U15+40	30000	80	40
U16+41	45000	40	20
U17+42	54000	45	35
U18+43	60000	55	34
U19+44	7000	45	25
U20+45	8000	35	20
U21+46	10000	20	10
U22+47	40000	40	20
U23+48	56000	34	30
U24+49	45000	20	25
U25+50	55000	40	20

Table 2 Tasks associated Users

Simulations for 50 grid users of various methods in reaction to an increase in the overall number of resources available on the grid have produced the following results, summarised below. We'll pretend for the time being that each user is only engaged in one activity.

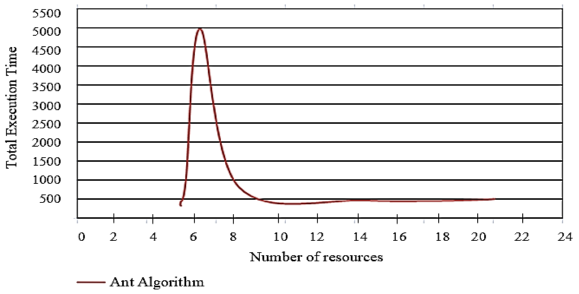

Fig 2 Total execution time with number of resources

Fig. 2 shows the total computation time associated with number of resources as increases.

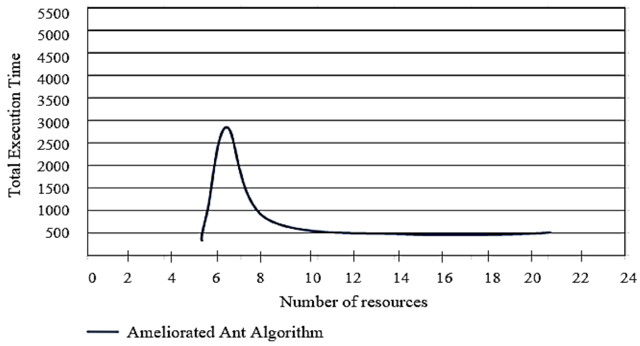

Fig 3 Total execution time with number of resources

It has been observed from experiment that the execution time in Ameliorated ant algorithm becomes much less than that traditional conventional ant algorithm.As result cost is reduced in the developed ant algorithm.

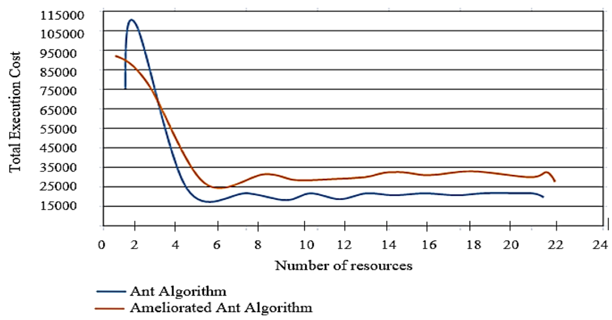

Fig 4. Total execution cost along number of resources

It has been observed from experiment that the modified ant algorithm has a substantially lower overall execution cost than the standard ant colony method, providing superior performance and results.

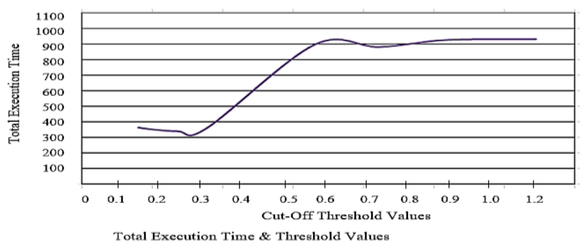

Fig 5 Total execution time with cut-off threshold value

The result in Fig. 5 above displays the overall execution time for a range of cut-off threshold values in the scheduling algorithm. The algorithm's output resembles that of a conventional algorithm if the threshold value increases by a specified quantity

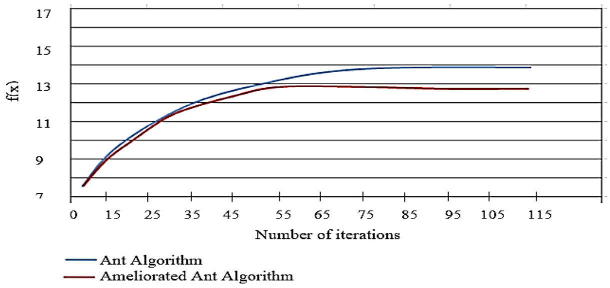

Fig. 6 Convergence results

The convergence of our modified ant algorithm and traditional conventional standard ant algorithms is seen in Figure 6. We can see that the traditional conventional standard ant algorithm converges after 75 iterations, but our modified ant algorithm converges after about 65 iterations. As a result, our improved method converges more quickly than the original ACO algorithm.

When the price component is taken into account, the redesigned ant algorithm computes more quickly than the ant algorithm.Because both of these algorithms consider the price, this is why this is the case. As illustrated our the new method has far more control over its processing costs than the reworked ant-algorithm, which ignores them. As a result, the cost of developing the algorithm has been considered.We can see from the graphic how long it takes the scheduling approach to complete each step for different threshold values. If the threshold value is increased by a certain amount, the algorithm's outcomes will be equivalent to those of the ant algorithm.

Conclusion

Traditional ant colony algorithms have some significant shortcomings that our unique AAA algorithm has resolved, including longer searching times, local optimums, and the stagnation problem.As an example of an application that can benefit from the proposed scheduling method, consider one that uses a lot of coarse-granularity activities, such as parameter sweeps. Because of the application's coarse-granularity duties, the recommended technique is effective. If the work's load is small, each resource's likelihood of selection is almost equal; if certain resources are heavily loaded, they are selected with the lowest likelihood.This considerably decreases both the computation time and the cost associated with execution of the jobs, based on the selection made by the grid user submitting the work.Job grouping may reduce transmission costs and delays in lightweight applications before load balancing operations onto resources.Distribution scheduling systems that consider how resources are organized inside groups will be developed in the not-too-distant future.

References :

1. Tao, F., Hu, Y.F., Ding, Y.F., Sheng, B.Y. and Zhou, Z.D., 2006. Resources publication and discovery in manufacturing grid [J]. Journal of Zhejiang University (Science), 7 (10), 1676–1682.
2. Colorni, M. Dorigo, V. Maniezzo, “Distributed optimization by ant colonies,” Proceedings of European Conference on Artificial Life, Paris, France, pp. 134-142, 1991.
3. J. L. Denebourg, J. M. Pasteels, and J. C. Verhaeghe, “Probabilistic behavior in ants: A strategy of errors?,” J. Theoret. Biol., vol. 105, pp. 259-271, 1983.
4. J. L. Denebourgand S. Goss, “Collective patternsand decisionmaking,” Ethology, Ecology & Evolution, vol. 1, pp. 295-311, 1989.
5. S. Goss, R. Beckers, J. L. Denebourg, S. Aron and J. M. Pasteels, “How trail laying and trail following can solve foragingproblems for ant colonies,” in Behavioral Mechanisms of Food Selection, R. N. Hughes Ed., NATO-AS1 Series. Berlin: Springer-Verlag, vol. 20, 1990.
6. V. Selvi, and R. Umarani, “Comparative Analysis of Ant Colony and Particle Swarm Optimization Techniques,” International Journal of Computer Applications, vol. 5., pp. 1–6, August 2010.
7. M. Dorigo, V. Maniezzo, A. Colorni, “Ant system: optimization by a colony of cooperating agents,” IEEE Transactions on Systems, Man, and Cybernetics, vol. 26, pp. 693–698, pp.1-13, February 1996.
8. M. Dorigo, L.M. Gambardella, “Ant colony system: a cooperative learning approach to the traveling salesman problem,” IEEE Transactions on Evolutionary Computation, vol 1, pp. 53 - 66, April 1997.

9. Y. Zhang, Z. Pei, J. Yang, Y. Liang, “An Improved Ant Colony Optimization Algorithm Based on Route Optimization and Its Applications in Travelling Salesman Problem,” IEEE, BIBE, pp. 693– 698, 2007.
10. X. Song, B. Li, and H. Yang, “Improved Ant Colony Algorithm and its Applications in TSP,” Sixth International Conference on Intelligent Systems Design and Applications, october 2006.
11. M. Dorigo and, LM. Gambardella, “Ant colony system: A cooperative learning approach to the traveling salesman problem,” IEEE Transactions on Evolutionary Computation, pp. 53–66, 1997.
12. M. Dorigo and, LM. Gambardella, “A study of some properties of antQ . In: Voigt H-M, Ebeling W, Rechenberg I, Schwefel H-S, eds. Proceedings of the 44th International Conference on Parallel Problem Solving from Nature,” Berlin: Springer-Verlag, pp. 656–665, 1996.
13. Colorni and, M. Dorigo, “Heuristics from nature for hard combinatorial optimization problems,” International Trans Operational Research, pp. 1–21, 1996.
14. R. S. Jadon, U. Datta, “Modified ant colony optimization algorithm with uniform mutation using self-adaptive approach for travelling salesman problem,” Computing, Communications and Networking Technologies, 2013.
15. N. Zhao, Z. L. Wu, Y. Q. Zhao, and T. F. Quan, “Ant colony optimization algorithm with mutation mechanism and its applications,” Expert Syst. Appl. , vol. 37, pp. 4805-4810, July 2010.
16. Q. WU, and J. ZHANG, “ACO With the Characteristic of Mutation”, Computer research and development, pp. 240-245, 1999.
17. C. Ratanavilisagul, and B. Kruatrachue, “A modified particle swarm optimization with dynamic mutation period,” Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, July 2014.
18. C. Ratanavilisagul, and B. Kruatrachue, “A modified particle swarm optimization with mutation and reposition,” International Journal of Innovative Computing, Information and Control, vol. 10, pp. 2127– 2142, December 2014.
19. Kundakcı, N.; Kulak, O. Hybrid genetic algorithms for minimizing makespan in dynamic job shop scheduling problem. Comput. Ind. Eng. 2016, 96, 31–51.
20. Jung, S.; Woo, Y.B.; Kim, B.S. Two-stage assembly scheduling problem for processing products with dynamic component-sizes and a setup time. Comput. Ind. Eng. 2017, 104, 98–113.
21. Gao, K.Z.; Suganthan, P.N.; Pan, Q.K.; Chua, T.J.; Chong, C.S.; Cai, T.X. An improved artificial bee colony algorithm for flexible job-shop scheduling problem with fuzzy processing time. Expert Syst. Appl. 2016, 65, 52–67.
22. Wang, L.; Cai, J.; Li, M.; Liu, Z. Flexible Job Shop Scheduling Problem Using an Improved Ant Colony Optimization. Sci. Program. 2017, 2017.
23. Xiong, W.; Fu, D. A new immune multi-agent system for the flexible job shop scheduling problem. J. Intell. Manuf. 2015, 29, 857–873.
24. Helo, P.; Phuong, D.; Hao, Y. Cloud manufacturing—Scheduling as a service for sheet metal manufacturing. Comput. Oper. Res. 2018, 110, 1–12.
25. Barak, S.; Moghdani, R.; Maghsoudlou, H. Energy-efficient multi-objective flexible manufacturing scheduling. J. Clean. Prod. 2020, 283, 124610.
26. Zan, X.; Wu, Z.; Guo, C.; Yu, Z. A Pareto-based genetic algorithm for multi-objective scheduling of automated manufacturing systems. Adv. Mech. Eng. 2020, 12, 1–15.
27. Mohammadi, S.; Al-e-Hashem, S.M.J.M.; Rekik, Y. An integrated production scheduling and delivery route planning with multi-purpose machines: A case study from a furniture manufacturing company. Int. J. Prod. Econ. 2020, 219, 347–359.
28. Baxendale, M.; McGree, J.M.; Bellette, A.; Corry, P. Machine-based production scheduling for rotomoulded plastics manufacturing. Int. J. Prod. Res. 2020, 1–18.
29. Guo, J.; Shi, Y.; Chen, Z.; Yu, T.; Shirinzadeh, B.; Zhao, P. Improved SP-MCTS-Based Scheduling for Multi-Constraint Hybrid Flow Shop. Appl. Sci. 2020, 10, 6220.