

Application of plant growth simulation algorithm

Zhengen Fan^{1,a}

¹Gansu Normal University for Nationalities, Hezuo, gansu, 747000, China

^azhengtwn@163.com

Keywords: Plant Growth Simulation Algorithm; Permutation flow shop scheduling problem; Total flow time; Heuristics

Abstract: Permutation flow shop scheduling problem with total flow time criterion has been increasingly dealt with in recent years. A number of heuristics and meta-heuristics have been developed for this problem. However, the calculation of total flow time for job permutations spends too much computational time in the algorithms. Based on Plant Growth Simulation Algorithm, we propose a novel algorithm for solving permutation flow shop scheduling problems. The computation experiments based on the well-known benchmarks are provided. Simulation results that the Plant Growth Simulation Algorithm (PGSA) has better feasibility and validity for solving permutation flow shop scheduling problem.

Introduction

The permutation flow shop scheduling problem has been widely studied in the recent decades. An flow shop system consists of m machines in a serial layout where a set of n jobs has to go through first machine one, then machine two, and so on until machine m . That is, all the jobs have identical routes. Each job requires an operation with uninterrupted processing time on each of the machines. Each job can be preceded at most one machine at the same time. The processing of a job on a machine cannot be interrupted. All jobs are independent and are available for processing at time 0; and etc. The aim is to find a sequence for processing all jobs on all machines so that the given criterion is minimized. Although the process constraint of the model is relatively simple, but it has proven more than 3 machine of permutation flow shop scheduling problem is NP problem^[1].

Two criteria are widely addressed for permutation flow time scheduling problem in the literature. These are make span and total flow time. Minimizing make span is important in situations where a simultaneously received batch of jobs is required to be completed as soon as possible or the utilization of resources is essential^[2-6]. There are other real life situations in which each completed job is needed as soon as it is processed. In such situations, one is interested in minimizing the total or mean completion time of all jobs rather than minimizing makespan. Minimizing total flow time can not only reduce inventory and holding cost, but also lead to stable or even use of resources^[7-9]. Therefore, it is of significance to develop solution techniques to minimize total flow time for permutation flow shop scheduling.

The paper proposes a new effective hybrid algorithm is proposed which is based on plant growth simulation algorithm for solving the permutation flow shop scheduling problem, simulation example verifies the correctness and effectiveness of the algorithm.

Mathematical description of permutation flow shop scheduling problem

Permutation flow shop scheduling problem studies the flow process of n work pieces on m machines. If the scheduling objective is the maximum completion time, the case of Permutation flow shop scheduling problem mathematics is described as follows: n represents the number of jobs, m represents the machine number, $prmu$ shows that all the workpiece after each machine processing consistency, C_{max} represents the maximum completion time of the work pieces, t_{ij} denotes the processing time of workpiece i on machine j , $C(j_i, k)$ represents the completion time of workpiece j_i on machine k , π represents a sort of all jobs, T is collection of all sort. Assume that the workpieces

are processed according to machine 1 to m , then the completion time of n work pieces on m machines can be obtained by formula(1) to formula(5).

$$c(j_1, 1) = t_{j_1, 1} \quad (1)$$

$$c(j_i, 1) = c(j_{i-1}, 1) + t_{j_i, 1} \quad i = 2 \dots n \quad (2)$$

$$c(j_1, k) = c(j_1, k-1) + t_{j_1, k} \quad k = 2 \dots m \quad (3)$$

$$c(j_i, k) = \max\{c(j_i, k-1), c(j_{i-1}, k)\} + t_{j_i, k} \quad i = 2 \dots n, k = 2 \dots m \quad (4)$$

$$c_{\max}(\pi) = c(j_n, m) \quad (5)$$

$$\pi^* = \arg\{c_{\max}(\pi) = c(j_n, m)\} \rightarrow \min, \forall \pi \in T \quad (6)$$

Which formula (5) is the maximum completion time, formula (6) represents the corresponding scheduling scheme of minimize the maximum completion time.

The plant growth simulation algorithm

The plant growth simulation algorithm is a bionicrandom algorithm [8]. It looks at the feasible region of integer programming as the growth environment of a plant and determines the probabilities to grow a new branch on different nodes of a plant according to the change of the objective function, and then makes the model, which simulates the growth process of a plant, rapidly growth towards the light source (global optimum solution).

In the growth process, in order to get enough sunlight for photosynthesis, the plants strive to breed more branches and leaves. This character of plants is called phototropism. There are two facts for the growth law of plants.

1. The higher the morphactin concentration of a node, the greater the probability to grow a new branch on the node;

2. The morphactin concentration of any node on a plant is not given beforehand and is not fixed; however, it is determined by the environmental information of the node which depends on its relative position on the plant. The morphactin concentrations of all nodes of a plant are allotted again according to the new environment information after it grows a new branch.

It was proved by biological experiments that a new branch that is able to grow depends on its morphactin concentration when the plant has more than one node. The node which has larger value of morpheme concentration has more growth opportunities than the smaller one, morphactin concentration are not pre-assigned to the nodes, but formed according to their location information which we call that plant showed the feature of plant's Phototropism. After the node has been formed, morphactin concentration will be readjusted according to changing in the environment of a new growth node else.

We could describe the characteristics of plant growth in term of the mathematical view basis on plant growth analysis of the above. Suppose the length of tree trunk is T , the length of branch is L , there are W growing nodes $S_T = (S_{T1}, S_{T2}, \dots, S_{TW})$ in the trunk, morphactin concentration of those nodes is $P_T = (P_{T1}, P_{T2}, \dots, P_{TW})$ respectively. There are q growth nodes $S_T = (S_{T1}, S_{T2}, \dots, S_{Tq})$ in the branch. Morphactin concentration of those nodes is $P_T = (P_{T1}, P_{T2}, \dots, P_{Tq})$ respectively. The morphactin concentration which grown in the trunk and branch is calculated as follows:

$$P_{Ti} = \frac{f(x_0) - f(S_{Ti})}{\sum_i^W (f(x_0) - f(S_{Ti})) + \sum_j^q (f(x_0) - f(x_{Tq}))} \quad (7)$$

Where x_0 represents the root of plant (the initial node), $f(*)$ is the information function of the node in the environment, The smaller value of the function as possible shows the better environment where the node located, it will help to grow new branch.

From formula (7), the value of morphactin concentration in every node depends on relative position of the relative to the initial root and the location of the environmental information, the

mechanism consistent with the formation of the morphactin concentration in the plant cell. We can deduced from formula (7) as follows:

$$\sum_{i=1}^T \sum_{j=1}^q (P_{Ti} + P_{Tj}) = 1 \quad (8)$$

We can produce random number in the interval [0, 1]. The random number must fall within one of the state space $(P_1, P_2, \dots, P_{T+q})$, where the corresponding node will have a prior right to grow new branch. A simple example of morphactin concentration state space displays in the fig. 1. The value of morphactin concentration in all nodes will change after the new branch has grown up. Computation formula of the value of morphactin concentration in other nodes will add the relevant node on the new branch, and remove the grown node, which basis on formula (7). The process will be repeated until there is no new branches growing.

The analysis of simulation

In order to explain the performance of the proposed algorithm, we use the test bed presented by Taillard^[10] for permutation flow shop scheduling problem, which consists of a total of 120 problems of various sizes, having 20, 50, 100, 200, and 500 jobs and 5, 10, or 20 machines. These sets are denoted according to their sizes as follows: 20×5, 20×10, 20×20, 50×5, 50×10, 50×20, 100×5, 100×10, 100×20, 200×10, 200×20 and 500×20. We compare the NEH and the proposed algorithm of the paper. Table 1 reports the CPU time of these heuristics respectively.

Table 1 The computation time of the compared algorithms

Instance	NEH	The proposed algorithm of the paper
20*5	0.000	0.000
20*10	0.002	0.000
20*20	0.002	0.000
50*5	0.005	0.001
50*10	0.009	0.002
50*20	0.013	0.006
100*5	0.031	0.01
100*10	0.064	0.032
100*20	0.217	0.135
200*20	0.477	0.313
500*20	5.874	4.133

From the test data can be seen, for the selected Taillard problems, the plant growth simulation algorithm has good optimization performance in the field of combinatorial optimization, and it is an effective tool for solving the permutation flow shop scheduling problem.

Conclusions

Permutation flow shop scheduling problem with total flow time criterion has a very strong engineering background in the modern manufacturing systems. This paper proposes an improve plant growth simulation algorithm to evaluate the total flow time for a given permutation, which is a key component of the heuristics and metaheuristics for the problem. The computation experiments demonstrated the effectiveness of the improve plant growth simulation algorithm.

References

- [1] Garey M R, Johnson D S, Sethi R. The complexity of flow shop and job shop scheduling[J]. Mathematics of operations research, 1976, 1(2): 117-129.
- [2] Ruiz R, Maroto C. A comprehensive review and evaluation of permutation flowshop heuristics. European Journal of Operational Research 2005, 165: 479-494.

- [3] Pan Quan-Ke, Suganthan PN, Tasgetiren MF, Chua TJ. A novel artificial bee colony algorithm for a lot-streaming flow shop scheduling problem [J]. Information sciences, 2010, DO: 10.1016/j.ins.2009.12.025.
- [4] Pan Quan-Ke, Wang Ling, Qian Bin. A novel differential evolution algorithm for bi-criteria no-wait flow shop scheduling problems [J]. Computers and Operations Research, 2009, 36(8):2498-2511.
- [5] Pan Quan-Ke, Tasgetiren MF, Liang YC. A discrete particle swarm optimization algorithm for the no-wait flowshop scheduling problem [J]. Computers and Operations Research 2008, 35(9):2807-2839.
- [6] Wang L, Pan QK, Suganthan PN, Wang WH, Wang YM. A novel hybrid discrete differential evolution algorithm for blocking flow shop scheduling problems. Computers & Operations Research, 2010, 37(3): 509-520. [9] Liao C. J., Tjandradjaja E., Chung T. P. An approach using particle swarm optimization and bottleneck heuristic to solve hybrid flow shop scheduling problem[J]. Applied Soft Computing, 2012, 12(6): 1755-1764.
- [7] Wang L, Li BB. Quantum-inspired genetic algorithms for flow shop scheduling. In: Quantum Inspired Intelligent Systems, Nedjah N, Coelho LDS and Mourelle LDM, Eds. Berlin: Springer, 2008. Studies in Computational Intelligence (SCI), 2008, 121: 17-56.
- [8] Pan Quan-Ke, Tasgetiren MF, Liang YC. A discrete differential evolution algorithm for the permutation flowshop scheduling problem [J]. Computers and Industrial Engineering [J], 2008, 55(4):795-816.
- [9] Framinan JM, Leisten R, Ruiz-Usano R. Comparison of heuristics for flowtime minimisation in permutation flowshops. Computers & Operations Research 2005, 32(5):1237-1254.
- [10] Taillard E. Benchmarks for basic scheduling problems. European Journal of Operational Research 1993,47:67-74.