

Cost Optimal Production Scheduling Model Based on VNS-NSGA-II Hybrid Algorithm—Study on Tissue Paper Mill

Huanhuan Zhang ¹, Jigeng Li ¹, Mengna Hong ^{1,2}, Yi Man ¹ and Zhenglei He ^{1,*}

¹ State Key Laboratory of Pulp and Paper Engineering, South China University of Technology, Guangzhou 510640, China.

² China Singapore International Joint Research Institute, Guangzhou 510006, China.

* Correspondence: hezhenglei@scut.edu.cn

1. Encoding and Decoding

For multi-objective optimization problems, coding is a fundamental and important task. A good coding scheme will be able to obtain high-quality solutions quickly, and an unreasonable coding scheme will make the algorithm inefficient and may lead to poor algorithm optimization. Due to the first stage (papermaking stage) and the second stage (converting stage), the order of processing of the jobs is the same. Therefore, the processing job is used for encoding. All processing jobs constitute a chromosome, and each processing job is a gene on the chromosome. For example, chromosomes [1, 2, 3, 4] are used to indicate that there are 4 processing jobs. When decoding, the four processing jobs are sequentially assigned to the available machines according to the machine's earliest machinable time. Taking chromosome [1, 2, 3, 4] as an example, the first step is to find the machine with the shortest processing time in all available machines. The second step is to assign job 1 to the machine for processing. It is the earliest process time to update this machine. Repeat these three steps until you have scheduled 4 processing jobs.

2. Population Initialization

In this paper, the population is initialized by the random generation method [4]. The length of the chromosome in the population is determined by the amount of job. For example, if there are 50 processing jobs, the length of the chromosome is equal to 50.

3. Population Classification

A fast sort algorithm for non-dominated sorting is used in this study, details of which can be seen from the literature [21].

4. Selection operator

In the selection operation, this paper adopts a simple and efficient binary tournament mechanism [23]. In the binary tournament, the first step is to randomly select two individuals $N1$ and $N2$ in the current population N . The second step is to compare the Pareto ranks of $N1$ and $N2$. If the $N1$ level is higher than $N2$, the high-level $N1$ is directly stored in the new population space N' . If the $N1$ rank is lower than $N2$, $N2$ is directly deposited into the new population space. If the $N1$ rank is equal to $N2$, then the crowding distance values of individuals are compared, and the individual with high crowding distances value are stored in the new population space. The above process is repeated until the number of individuals in the new population N' is the same as the previous population N .

5. Crossover Operator

The cross-operation process of this paper is as follows: (1) Randomly select two parent individuals from the population, which are recorded as Parent 1 and Parent 2. (2) Randomly select several genes to form a gene segment S , and the number of genes in the gene segment S is about half of the length of the parent individual. (3) Find the genes in the gene segment S in Parent 1 and Parent 2, respectively, and record the order of the genes.

(4) First, delete the same gene as the gene segment S in Parent 1 and Parent 2, and then insert the green part of Parent 1 and the orange part of Parent 2 into the gaps of parent 2 and parent 1, respectively, to generate two new offspring individuals (Child 1 and Child 2). The crossover operation is shown in Figure S1.

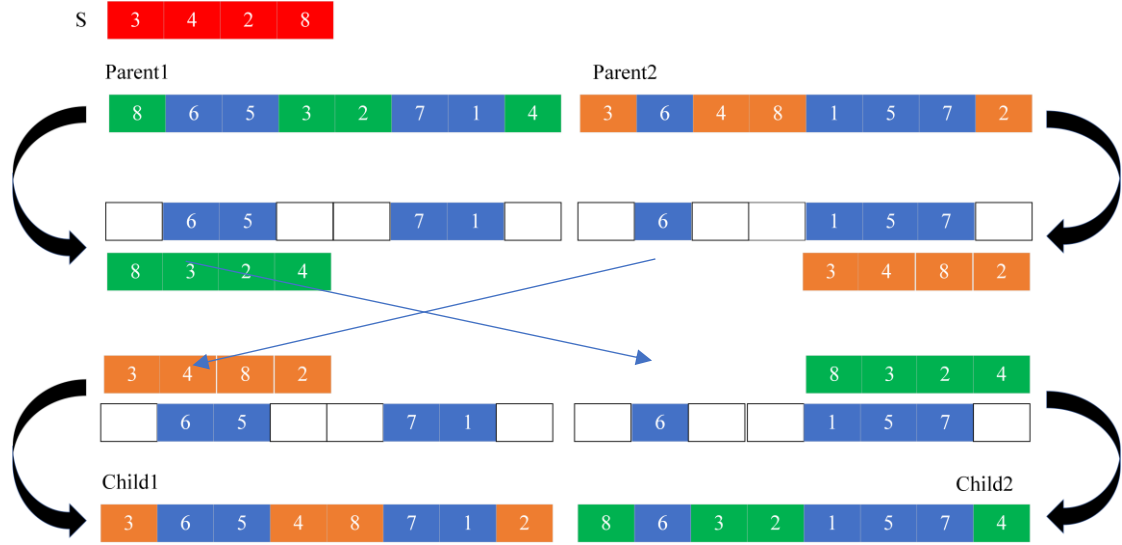


Figure S1. Cross operation diagram.

6. Mutation Operator

In this paper, the mutation operation is performed according to the exchange method. First, randomly select a chromosome. Second, on this chromosome, two genes were randomly selected. Finally, exchange the location of these two genes. The variation diagram is shown in Figure S2.

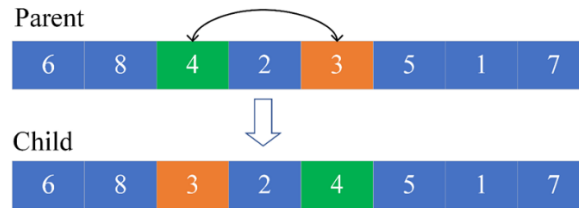


Figure S2. Mutation operation diagram.

7. Principle of hypervolume calculation

For the calculation of the hypervolume index, this paper uses the calculation method proposed by Fonseca et al. [26]. The main idea is to process one target (one dimension) of all points at a time. First, the points in the solution set are sorted in descending order according to the one-dimensional target. Secondly, the whole hypervolume is sliced according to the value of this dimension from large to small, and each slice is a d-1 dimension (d is the target function). The super volume of the number; finally calculate the super volume of each d-1 dimension slice, and multiply the depth of the corresponding slice in the first dimension to obtain the value of the d dimension sub-volume. The values of all these sub-hypervolumes are added to obtain the value of the entire d-dimensional hypervolume.

References

4. Li, Z.; Yang, H.; Zhang, S.; Liu, G. Unrelated parallel machine scheduling problem with energy and tardiness cost. *Int. J. Adv. Manuf. Technol.* **2016**, *84*, 213–226.
21. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T.A.M.T. A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2001**, *6*, 182–197.
23. Leung, J.Y.T.; Lee, K.; Pinedo, M.L. Bicriteria scheduling with machine assignment costs. *Int. J. Prod. Econ.* **2012**, *139*, 321–329.
26. Fonseca, C.M., Paquete, L., López-Ibáñez, M. An improved dimension-sweep algorithm for the hypervolume indicator. In *Conf. on Evolutionary Computation*: IEEE Press: Los Alamitos, CA, USA, **2006**, 1157–1163.