

Chapter 10

The Local Search and Experiments of Job-Shop Scheduling Problem

Ning Zhao and Si-yu Chen

Abstract The Job-shop Scheduling Problem is a typical scheduling problem, which is regarded as NP-hard because of its complexity. Improvement based on traditional local search methods is realized with the combination of neighborhood and critical path. Firstly, feasible solution is achieved by heuristic approaches or just randomization. Secondly, makespan optimized solution is achieved by improved local search. Finally, the improved local search method is proved effectively by experiments, moreover the searching process is visible and suits man-machine interaction.

Keywords Local search • Job-shop scheduling problem • Man-machine interaction • Critical path

Introduction

JSP stands for the Job-shop Scheduling Problem. It means to combine time with jobs closely in order to meet the requirements of the productions' progress. The result of scheduling directly influences the completion time of products, the arrangement of workers and the utilization rate of machines. In the context of diversified market demand, in order to meet the production modes of multispecies, singleton and small-lot, Flexible Manufacturing and Agile Manufacturing emerged. The production line is more complicated, but the delivery time is shortened. In order to resolve these conflicts, researchers from different fields start to focus on the efficient and accurate JSP.

The deterministic job-shop scheduling problem can be briefly described as follows: given a finite set of jobs, each is a finite sequence of operations subjected to precedence constraints, each operation needs to be processed exclusively on a machine from a

N. Zhao (✉) • S.-y. Chen
School of Mechanical Engineering, University of Science and Technology Beijing,
Beijing 100083, China
e-mail: zhning@sina.com

finite set of machines for a prescribed time interval, the goal is to find a schedule which can complete all operations in the shortest time (Murovec and Suhel 2004).

As the augment of the number of jobs and machines, the number of solutions can increase exponentially. Researchers regard this kind of problems as NP-hard (non-deterministic polynomial) (Jain and Meeran 1999).

The history of JSP's study can date back to half a century ago. It is the result of development. Its valuable and challengeable arouse a lot of researchers' interests to study. The methods to solve JSP have experienced the process of simple to complex, single to multiple and theoretical to practical, and have yielded many excellent results. These algorithms can be divided into two categories: Optimization Algorithms and Approximation Algorithms (Shao-li Dai et al. 1999).

Optimization Algorithms can assure an optimal result. It includes methods of operations research (such as branch-and-bound algorithm, linear programming, dynamic programming and nonlinear programming) and enumeration. The methods of operations research use equalities and inequalities to express the problem, and solve the objective function under the constraints. Enumeration, however, tries to list all possible solutions, and then find the feasible ones and the optimal one among them.

Approximation Algorithms play an important role recently to solve JSP. They are classified to three groups in this paper, Constructive Method, Artificial Intelligence and Local Search.

Constructive Method can obtain a feasible solution quickly, but the solution is always in bad condition. A common one is the Priority Distribution Rule, it is to construct a rule set, and the operation which meets the rules can be written on the schedule first, by such analogy, other operations can be on the schedule at last. Artificial Intelligence is to use the artificial theories and techniques to guide searches and provide effective searching program to find better solutions. Ant System (Xiao-rong Wang 2003) and Neural Network are representatives.

The method of Local Search means to start from several initial solutions, and realize optimization by searching neighborhood and replacing current solutions. At first, the search is only limited to the local. It is easily realized, but the searching results largely depend on the neighborhood functions and initial solutions (Ling Wang 2003). In recent years, in order to get better global optimized results, the improved Genetic Algorithm, Taboo Search and Simulated Annealing improve it from different perspectives with different searching mechanisms and strategies. Moreover, a new trend of solving JSP recently is to combine different kinds of local search methods into mixed heuristic algorithms, and has achieved some results.

Although different kinds of algorithms are put forward, and among them we can find effective ones, till now, none of the algorithms can completely solve JSP. The drawbacks of the algorithms mentioned above are listed as follows:

1. Optimization Algorithms

Optimization algorithms can obtain the optimal solutions, but the cost is searching all solutions that is possible to be optimal. So it is time wasting for large-size problems and is well known for its low efficiency.

2. Approximation Algorithms

Approximation algorithms can overcome the drawbacks of Optimization algorithms and good at solving large-scale optimization problems. But they also have some shortages.

- (1) Priority Distribution Rules: it can build a feasible solution quickly, but the solution is always bad.
- (2) Ant system: the amount of computation is large, and the ability of describing complex issues is weak.
- (3) Neural Network: can only solve small-scale scheduling problem, the calculated complexity is large and it can't promise the best solution.
- (4) Taboo Search, Simulated Annealing and Genetic Algorithm: they can achieve better results, but the quality of the results depends on the selection of parameters, and they are always time-consuming. And they can't promise the best solution.

Furthermore, all the algorithms mentioned are highly mechanized. Once they start to calculate on the computer, one cannot participate in it. In practical, the experience of human-being is very important for scheduling. In order to solve dynamic real-time scheduling problem and give full play to human-being in scheduling, we must develop an interactive scheduling system (Qing Zhang 2004). Among all the algorithms mentioned above, local search is the best one which suits to man-machine interaction, but need to be improved.

Improved Local Search by Critical Path

Definition 1: Critical Path, it means the uninterrupted processing path running through the period from the starting time of the first operation to the finishing time of the last operation.

On critical path, every operation's completion time and the next operation's start time are closely linked. In order to explain the definition vividly, we cite the graph raised by (Nowicki and Smutnicki 1996) here (Fig. 10.1):

The graph above is typical JSP's Gantt Graph. The critical path in the graph is: 1.1-4.2-5.2-2.2-2.3-1.2-3.1-4.3-5.3-5.4-6.4-1.3-3.3-3.4-2.4-1.4-4.4.

The operations on the critical path are Critical Operations. As the length of critical path determines makespan, we must shorten critical path's length so that we can get the shortest makespan.

Definition 2: Critical Block, it means that on the critical path, every machine has one or more adjacent operations, the adjacent operations which belong to the same machine are called a critical block.

For example, in the graph above, for M2, 1.1-2.2 is a block, and on M1, 2.3-5.3 is a critical block.

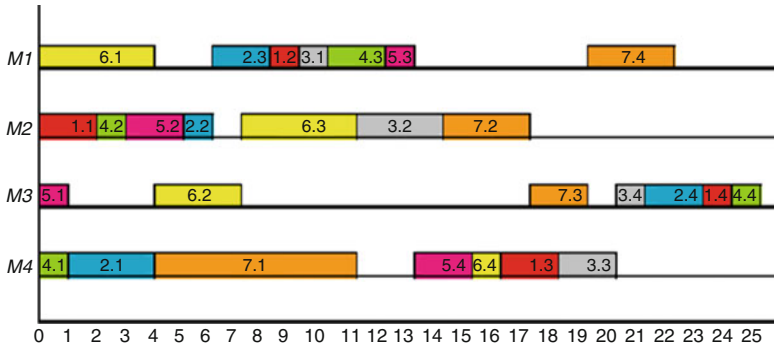


Fig. 10.1 Neighborhood of Nowicki and Smutnicki

It is also to be noted that every scheduling result may have more than one critical path.

The meaning of Local Search is the procedure of starting from the current results, utilizing the neighborhood functions (specific changing pattern), searching the neighborhood of current results and replacing them. In this paper, the local or the neighborhood means two adjacent operations on the same machine. Studies show that we can generate a new schedule if we swap the sequence of operations on the same machine (Jian-shuang Cui and Ke-tie Li 2009). The swap here is also a neighborhood function. As a result, for JSP problem, local search in neighborhood function can be replaced as Local Swap. If we make some local swap on current solutions, we can reach some new feasible solutions. These solutions are neighbors of current solution.

Of course, not all local swaps can result in feasible solutions or optimal solution. Some swaps may lead infeasible ones. And as the scale improves, the number of neighborhoods increases explosively. How to deduce compute capacity and avoid some infeasible local search is the point of studying.

Experiments

We major in Single-step Optimization in this paper. The Single-step Optimization means that we can only change one operation’s place each time. If we have some initial solutions, one or more Single-step Optimization may obtain the optimal solution.

We conduct these experiments on IESS (Industrial Engineering Scheduling System) which is self-developed for experimenting. The system is visible and interactive. Our experiments are to make some manual adjustments on different initial solutions and observe the effectiveness and drawbacks of local swap.

Table 10.1 FT06

Initial solution	Final optimal solution	Steps from initial solution to local optimal solution	Global optimal solution
89	79	2	55
71	62	4	55
86	65	5	55
83	69	4	55
79	73	1	55

Table 10.2 FT10

Initial solution	Final optimal solution	Steps from initial solution to local optimal solution	Global optimal solution
1,420	1,300	9	930
1,369	1,369	0	930
1,387	1,273	9	930
1,326	1,166	12	930
1,291	1,282	2	930

In order to make the experiments more persuasive, we set some swapping rules here:

1. Distinguish critical paths and critical blocks

We find the critical paths and critical blocks from the back to the forward in current solution. If there is more than one critical path, we should choose the one of job predecessor.

2. Operation swapping

Every critical block and critical operation of initial solution is studied first. In principle, we should try every two operations of initial solution that can be local swapped and calculate makespan, the best one is kept as local optimal solution. The local optimal solution is set to be initial solution and new local optimal solution can be reached by the same way. If new local optimal solution is equal to initial solution, the local optimal solution is kept as the final optimal solution.

During the search procedure, man can easily involved in searching by interaction, but the question is whether the final optimal solution is good enough to practical use.

In the experiment, we test on two classical test samples: FT06 and FT10. We obtain the initial solutions by randomization, and conduct five experiments on every sample. The results are as follows:

The final optimal solution means that when using the swap rules above in the current solution, every swap on the critical path will make the current makespan worse. We can see from the results that if create an initial solution randomly, one can easily jump into the final optimal solution no matter the problem is FT06 or the more complex FT10, and can't move towards the global optimal result.

As the experiments are based on randomly generated results, in order to prove how initial results influence the final schedule, we add a rule of making initial result: "operation which its job has the most remnant operations" + "random", which means

Table 10.3 FT06

Initial solution	Local optimal solution	Steps from initial solution to local optimal solution	Global optimal solution
66	59	4	55
62	59	1	55

Table 10.4 FT10

Initial solution	Local optimal solution	Steps from initial solution to local optimal solution	Global optimal solution
1,221	1,119	13	930
1,219	1,162	5	930

that the system will choose the operation meet the first condition, if there are more than one operation can be chosen, then choose one randomly. It can be proved that this method of constructing solutions can generate better solutions to the FT samples, because they have the same amount of operations, and every job's total processing time is more or less the same.

We can see that the initial solutions are better than the randomly generated ones, and the improvements on the initial solutions can influence the local optimal solutions on a large scale.

If the conclusion above is true, then we can believe that, every JSP must have a rule set to construct a good initial result, and on the basis of the result we can jump to the optimal solution with single-step optimization. But the rule set is very difficult to build. For example, to FT06, we can find a rule set: "the operation which has the longest surplus time" + "operation which its job has the most remnant operations" + "random", these rules' priority decrease progressively. The initial result is 58, and we can get the global optimal solution 55 only by two single-step optimizations. But if we add the rule set onto FT10, the initial result is 1,191, which is better than all the experiment results above, and we can get the local optimal solution 1,090 by ten steps, but there is still a certain distance to the global optimal solution 930.

Conclusion

We can prove from the experiments that on IESS, based on the initial solutions which are obtained by several rules, we can get the global or local optimal solution with the help of Single-step Optimization. But this method is not developed well enough. We still need intensive study in these aspects:

1. With what rule set can we construct better initial solutions, and jump to the optimal solution by less steps.
2. Put forward a set of complete swapping rules so that humans' affection can be reduced.
3. Realize automatic Single-step Optimization on IESS.

This paper is sponsored by the Fundamental Research Funds for the Central Universities of China FRF-TP-12-071A.

References

- Jain AS, Meeran S (1999) Deterministic job-shop scheduling: past, present and future [J]. *Eur J Oper Res* (113):390–434
- Jian-shuang Cui, Ke-tie Li (2009) Global local search method to solve JSP [J]. *Comput Integr Manuf Syst* 15(7)
- Ling Wang (2003) Job-shop scheduling and genetic algorithm [M]. Tsinghua University Press, Beijing
- Murovec B, Suhel P (2004) A repairing technique for the local search of the job-shop problem [J]. *Eur J Oper Res* (153):220–238
- Nowicki E, Smutnicki C (1996) A fast taboo search algorithm for the job-shop problem [J]. *Manag Sci* 42(6):797–813
- Qing Zhang (2004) Man machine interactive job-shop dynamic scheduling method [A]. *Mech Manuf* 42(476)
- Shao-li Dai, Yue-jin Tan, Hao Wang (1999) System research in job-shop scheduling methods [J]. *Syst Eng* (1):41–45
- Xiao-rong Wang (2003) Theoretical model of ant system and application research in job-shop scheduling. Ph.D. thesis, ZheJiang University