36

SENSITIVITY ANALYSIS OF CRITICAL PATH AND ITS VISUALIZATION IN JOB SHOP SCHEDULING

Ryosuke Tsutsumi and Yasutaka Fujimoto
Department of Electrical and Computer Engineering, Yokohama National University, Japan

Abstract: A visualization of a critical path in a job shop scheduling for manufacturing system is presented. The visualization is based on influence factor of the jobs in a given job shop scheduling. The influence factor is defined for each job as quantity of contribution to the critical path that is given by a sensitivity analysis of the path. The information of the sensitivity helps operator to improve the makespan of existing schedule easily.

Key words: Job Shop Scheduling, Tabu Search, Sensitivity Analysis, Critical Path

1. INTRODUCTION

In recent years, as scheduling problem getting more and more large-scaled and complicated, approximation algorithms such as Tabu Search (TS), Simulated Annealing (SA) and Genetic Algorithm (GA) are used for such problems[1][2][3]. In general, the job shop scheduling problem (JSSP) is well known to be one of the particularly hard optimization problems among the NP-hard combinatorial problems. It is difficult to find the optimal solution to the large-scale, practical problem within practical time. Thus the human interaction in combination with a heuristic algorithm is one of the important factors in the practical job shop scheduling systems. In combination with human expertise, the appropriate solution in the practical field is obtained.

In this paper, we propose sensitivity visualization tool to improve solution by human interaction. The sensitivity in this paper means a degree of influence of each operation on the whole schedule. In this system, a feasible schedule is obtained first by use of tabu search scheduling algorithm. The schedule manager can interactively tune the schedule based on visual information.

The structure of this paper is described as follows. In section 2, the model of job shop scheduling and conventional TS is defined. Also the new neighborhood selection in use of sensitivity is proposed. In section 3, the idea of sensitivity analysis and its algorithm are proposed. In section 4, the simulation of TS by conventional method and by proposed method is performed, and the visualization is presented.

2. PROBLEM DEFINITION

2.1 Job Shop Scheduling Problem (JSSP)

The JSSP has complexly intertwined two set of constrains, one is for jobs and the other is for machines. The problem is defined as finding a sequence of operations on each machine that minimizes an objective function, e.g., makespan, under a given sequence of operations for each job. The problem is considered as a particular hard combinatorial optimization problem. The JSSP is briefly described as follows. There are a set of jobs and a set of machines. Each jobs consists of a sequence of operations, and each of them needs fixed duration on one of the machines. Each machine can process at most one operation at one time, each operation is not interrupted if once it has started. The purpose of this problem is to find a minimal makespan schedule.

The model of the JSSP is described as follows. Consider a manufacturing system that consists of m-machine and n-jobs. The sequence of operations is defined for both jobs and machines. Let a set of jobs be $J = \{1,\ldots,n\}$, a set of machines be $M = \{1,\ldots,m\}$, a set of sequence of operations for jobs be $P = \{p_1,\ldots p_n\}$, and a set of sequence of operations for machines $Q = \{q_1,\ldots,q_n\}$. The sequence $p_i = \{f_{i,1},\ldots.f_{i,\mu_i}\}$, $f_{i,k} \in M$ for job i consists of a finite sequence of machines, which means the job is processed on machines in order of the sequence. The sequence $q_j = \{g_{j,1},\ldots,g_{j,v_j}\}$, $g_{j,l} \in J$ for machine j consists of a finite sequence of jobs, that means the machine processes the jobs in order of the sequence, as well. The relation between P and Q must be feasible. Then the start time of the kth operation for job i is denoted by variable $s_{i,k}$. Also the start time of the lth operation for machine j is denoted by variable $\tau_{j,l}$. If $s_{i,k} = \tau_{j,l}$ then $f_{i,k} = j$ and $g_{j,l} = i$ is satisfied. The uninterrupted processing time for job i on machine j is denoted by $\mu_{i,j}$. The setup times are ignored in this model. The constraints are described by

$$s_{i,k} + \mu_{i,f_{i,k}} \leq s_{i,k+1} \quad (1)$$

$$\tau_{j,l} + \mu_{g_{j,l},j} \leq \tau_{j,l+1}\,. \tag{2}$$

Fig. 1 and Fig. 2 show an example of Gantt-Chart of a feasible solution.

A Critical Path (CP) is defined as a set of operations that affect the makespan of its schedule directly. In other words, the delay of the operations on the critical path means the delay of whole schedule. Thus the manager of schedule has to pay the most attention to the operations on the critical path. A series of operations on the CP that are processed on same machine successively is called critical block (CB). The critical path for the initial sequence P and Q is obtained as shown in *Fig. 3* by solving the earliest schedule for $s_{i,k}$ and $\tau_{j,l}$ and checking the activity of constraints.

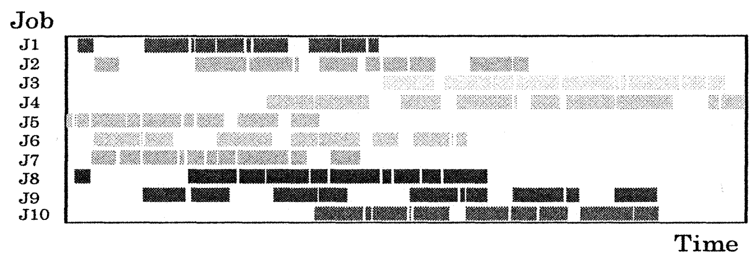

Figure 1. Gantt-Chart for each job

Figure 2. Gantt-Chart for each resource

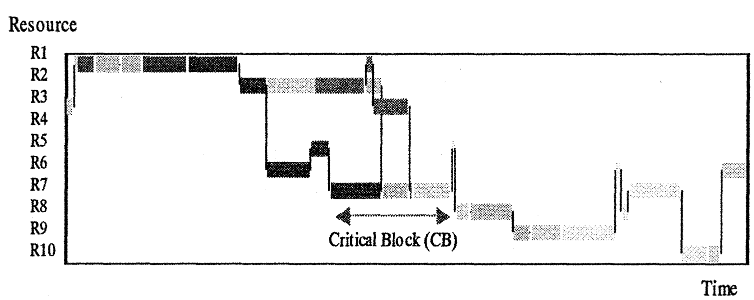

Figure 3. Critical path and critical block

2.2 Tabu Search (TS)

The TS is a heuristic search framework for solving hard combinatorial problems such as job shop scheduling, graph coloring (related), the Traveling Salesman Problem (TSP), the capacitated arc routing problem, etc. The TS iterates a swap of a specific neighborhood to improve the objective function. The TS has two lists to realize efficient convergence to quasi-minimal point. One is Tabu List, another is Superior Solution List[1].

2.2.1 Tabu List

As the TS consists of iterations of neighborhood swapping, it is very important to escape from locally optimal but not globally optimal solutions efficiently. It is also important to prevent the swap cycling. Movements in a certain past term are stored in Tabu List to make the backwards moves taboo.

2.2.2 Superior Solution List

When a superior solution is found, the current state of the solution is stored in the Superior Solution List. The state consists of the processing sequence of the solution, current tabu list, and neighborhood except next candidate. If the tentative best solution is not improved after enough iteration, the state of the past superior solution is restored according to the list that realizes Backtrack. This handling makes the convergence speed fast by searching the neighbor of the past good solutions.

2.2.3 New Neighborhood

A set of schedules generated from current schedule by interchange of a pair of the operations is called neighborhood of current schedule. As mentioned above, one feasible schedule should have the critical path that determines its makespan. In other word, the improvement of sequence on the critical path is necessary to improve the whole solution. Thus the interchange of the operations on the CB is used to improve the solution. The full neighborhood is generally too large to search completely. In a conventional TS method, neighborhood means an interchange of an adjacent pair of the operations on the CB. However, it is afraid that it takes large computation time to improve the solution because only a very narrow range of neighborhood is searched in the conventional method.

Thus the new neighborhood based on sensitivity analysis is introduced in this paper. The most sensitive operation, in other words the operation that should be paid attention extremely, on each of CB is detected first. Then the schedules generated by swapping the most sensitive operation for the other operations on the same CB is defined as new neighborhood. By use of this method, large range of neighborhood can be searched and the calculation cost is reduced.

3. SENSITIVITY ANALYSIS

The structure of schedule at any given time is paid attention. The influence of each operation on the critical path on whole schedule is defined as sensitivity. The constraints that satisfy the equality (1) (2) are defined as *active constraints*. The sensitivity of the operations on critical path is obtained by use of this active concept. The algorithm is described as follows.

1. Select one operation O^* on the critical path. On forwardly justified (earliest) schedule, let the operations whose start time is later than the finish time of O^* be a set A. Let the other operations except O^* be a set B. O^* belong to neither A nor B.
2. Then perform a next modification for an arbitrary operation $a \in A$, where a and $^\exists b \in B$ have a same active constrains.
   $A := A - \{a\}$, $B := B + \{a\}$
   This modification is iterated until a becomes empty.
3. Find a minimal slack in the constrains between A and B, and let it be $l_{\min}$. Then reduce the start time of the operation that belongs to A by $l_{\min}$.
4. Iterate procedure 2, 3 until a becomes an element of the critical path. The sensitivity of O^* is defined as $\sum l_{\min}$.
   Perform above procedures for all operations on the critical path.

The sensitivity expresses quantity of contribution to the makespan of the given schedule. Therefore, improvements of high sensitivity operations have possibility of yielding better solutions.

4. SIMULATIONS AND VISUALIZATION TOOL

4.1 Simulation (Tabu Search Scheduling)

Simulation for typical 10 machines 10 jobs (10x10) benchmark problem designed by Muth and Thompson[4] is performed to compare proposed method with conventional method. The procedure is described as follows.

1. Make initial schedule by use of activation algorithms (see appendix)
2. Detect the critical path by PERT
3. Search the neighbors (by conventional method and by proposed method).
4. Estimate the neighbor and move to one of them.
5. In case the best solution is not improved after enough iterations of move, restore the state of a past superior solution according to the list.

4.2 Simulation Results

Fig. 4 shows the convergence tendency of both by proposed method and by conventional method in the process of search. Conventional method needs a lot of iterations to escape from local optimal solution, however the proposed method by use of sensitivity analysis escapes from local optima solutions efficiently as shown in *Fig. 4.* The average number of evaluation of the objective function is reduced to 60% as shown in *Table. 1.*

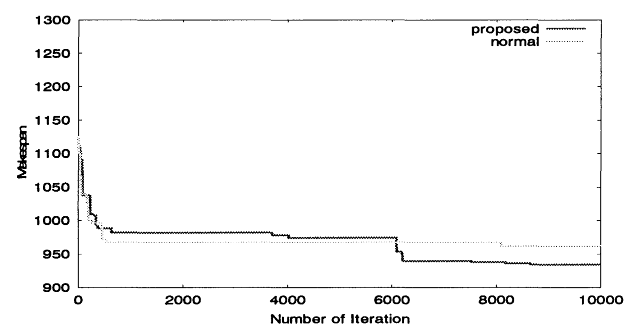

Figure 4. A convergent tendency

Table 1. The average and maximum number of evaluation of the objective function

	Average	Maximum
Conventional Method	11.9	29
Proposed Method	7.1	22

4.3 Visualization

Scheduling system with 3-D Gantt-Chart is developed (*Fig. 5*). This visualization tool aids the user to make the schedule as he wished. As the user can understand the schedule visually and can modify the schedule by use of this tool, the interaction between human and a heuristic algorithm is realized. The function of the system is described as follows. When the scheduling problem is given, the tentative quasi-optimal schedule is searched by use of aforementioned TS. Then the critical path and sensitivities are displayed to the user. The large influence factor (sensitivity) for job *i* expresses the easiness to improve the makespan by modifying the sequence of the neighborhood of the job on the CB. The paths are visualized by density plots on the Gantt-Chart as shown in *Fig. 6*.

This tool provides some of the quasi-optimal schedules that searched as candidates. The user can choice the most convenient schedule from them and also can modify the schedule freely by clicking the operations in the window. The system notifies the user whether the modifications is possible and suggests the idea of better schedule. The tool also can make the modified schedule active by the activation algorithm. (See appendix)

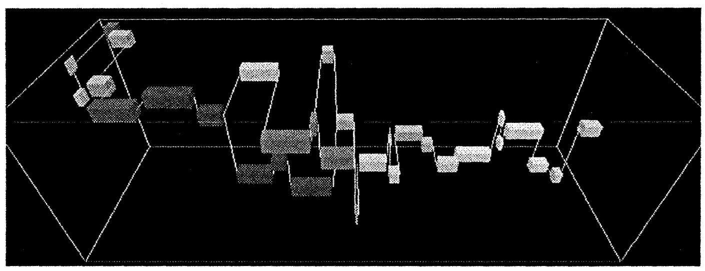

Figure 5. 3-D Gantt-Chart (Critical Path Mode)

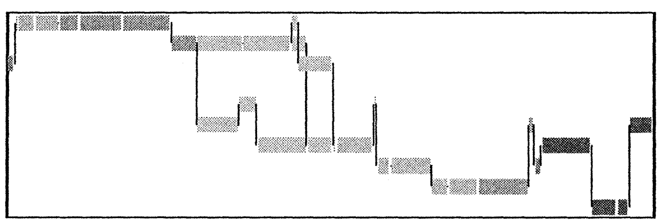

Figure 6. 2-D Visualization of sensitivity (density plot; The dark colored jobs are sensitive. The light colored jobs are insensitive)

5. CONCLUSION

It is very difficult to find the optimal solution to the large-scale for hard combinatorial problems such as JSSP within practical time. Thus the human interaction is considered as one of the important factors to solve such a hard problems.

In this paper, we presented an idea of sensitivity analysis and proposed a visualization system based on the TS with new neighborhood. Proposed system is considered as the useful interactive tool between human expertise and a heuristic algorithm. This system enables the user to improve the schedule effectively. For the future work, it is necessary to consider what kind of function is required for the interface.

6. REFERENCES

1. E. Nowicki, C. Smutnicki (1996). A Fast Taboo Search Algorithm for the Job Shop Problem, *Management Science*, **42**(6), 797-813
2. Takeshi Yamada, Ryohei Nakano (1996). Job-Shop Scheduling by Simulated Annealing Combined with Deterministic Local Search, *Journal of Information Processing Society of Japan,* **37**(4), 597-604. (in Japanese)
3. Takeshi Yamada, Ryohei Nakano (1997). Job-Shop Scheduling by Genetic Local Search, *Journal of Information Processing Society of Japan,* **38**(6), 1126-1138. (in Japanese)
4. J. E. Beasley (1990). OR-Library: Distributing Test Problems by Electronic Mail, *J. Opl. Res. Soc*, **41**(11), 1069-1072
5. B. Giffler, G. L. Thompson (1960). Algorithms for Solving Production Scheduling Problems, *Operations research*, **8**, 487-503
6. Muth, J. F., G. L. Thompson (1963). Industrial Scheduling, Prentice-Hall

APPENDIX

[Active Schedule] To find an initial solution or to improve a given solution, an idea of active schedule proposed by B. Giffler and G. L. Thompson is used[5][6]. An active schedule is defined as a feasible schedule that has a property where no operation can be made to start sooner by permissive left shifting. Algorithm to get active schedule is described as follows.

<Activation Algorithm>

1: Let $ES(O)$ and $EC(O)$ be the earliest start time of operation and the earliest finish time of operation. Both of these are the earliest time without upsetting the schedule that is already decided. Let a set of operations that is both not yet attended and feasible from the viewpoint of technical procedure be X . Let one of the operations whose $EC(O)$ is the earliest in $O \in X$ be O^* and the machine that process O^* be F^* .

2: In the operations $O \in X$ which is processed on the machine F^* , if $ES(O) \leq EC(O^*)$ then add the operation O to the set Y .

3: select one of the operations $O \in (Y \cup O^*)$ for the next processed operation on machine F^* . Iterate this procedure until X become the empty set.

The optimal schedule is obviously active schedule. Thus a better solution is obtained if the activation algorithm is applied to a given schedule.