Proceedings of the 5th WSEAS Int. Conf. on APPLIED INFORMATICS and COMMUNICATIONS, Malta, September 15-17, 2005 (pp221-226)

A High-Speed Integrated Scheduling System with Tabu Search for Large-Scale Job Shops Problems with Group Constraints

YOSHITOMO IKKAI, HIRONORI OKA, NORIHISA KOMODA
Graduate School of Information Science and Technology
Osaka University
Yamada-Oka 2-1, Suita, Osaka, 565–0871, JAPAN

abstract: - The target of this research is a job shop problem of about 2000 jobs with group constraints where jobs are grouped and processed. In addition to difficulties of large-scale problems, an evaluation function of this problem is not perfect practically, because there are many another evaluation factors, which human experts judges empirically. Therefore, the final goal of this research is creation of the solution which has almost minimum value of the evaluation function and to which human experts' empirical demand of modification is reflected for two hours with minimum human works.
To accomplish our purpose, we propose an integrated scheduling system with tabu search. This system consists of a basic scheduling engine and a parallel scheduling, a parameter tuning and a modification module. This system uses empirical dispatching rules effectively, tunes parameter automatically, and makes modification rapidly.
Through several experimentations, it is confirmed that this integrated scheduling system can generate final solution which satisfies all constraints and humans demands, and whose evaluation value is almost equal to the experts' one for two hours without human works for setting time of rule application rates.
key-words: - job shop problem, group constraints, tabu search, parallel search, modification, parameter tuning

1 Introduction

The target of this research is a job shop problem [1] with group constraints. The constraint is that jobs have to be grouped and processed together because a line features jobs with the same process. Minimum and maximum size restrictions exist in the total size of jobs in a group. Since this problem deals with about 2000 jobs, there are enormous group combinations. Additionally, once a group composition is changed, a schedule may also change drastically. Therefore, an effective search of many group combinations is necessary. Furthermore, an evaluation function of this problem is not perfect practically, because there are many other evaluation factors, which human experts judges empirically. Therefore, the final goal of this research is creation of the solution which has almost minimum value of the evaluation function and to which human experts' empirical demand of modification is reflected for two hours with minimum human works.

To accomplish our purpose, we propose an integrated scheduling system with tabu search. This system consists of a basic scheduling engine and a parallel scheduling, a parameter tuning and a modification module. The basic scheduling engine uses empirical dispatching rules effectively by using tabu search, and parallel scheduling module accelerates search speed for a large-scale problem. Parameter tuning module helps setting good rule application rates in order to make the basic scheduling engine high performance rapidly, and modification module modifies schedules in order to satisfy human demands rapidly.

2 A large-scale job shop problem with group constraints

2.1 Outline of the problem

The target of this research is a job shop problem with group constraints. Jobs dealing with the same process have to be grouped together for every line process because of equipment constraints and efficiency, as shown in the upper right of Fig. 1. Each group has a process code showing which process have to be excecuted. The total size of the jobs contained in a group is called the group size, and it is constrained by minimum and maximum values. When a group follows another group on the line that had a different process code, it requires additional time to change the machine setting of the line.

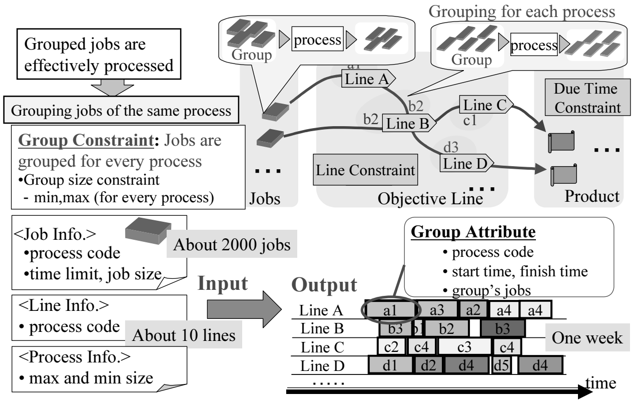

Figure 1: Job shop problem with group constraints

As shown at the bottom of Fig. 1, job information, process information, and line information are the inputs of this problem. The attribute values and constraints of each input in this problem are as follows: job information consists of job size, production time limit and process codes. Process information consists of the minimum and maximum constraints of the total size of jobs in all process codes, and group connection constraints which show whether groups can be processed continuously after another process codes. Line information is a set of process codes that show executable processes on the line.

As a result, the schedule for every group is output as shown at the bottom of Fig. 1. Group attributes consist of a process code, start time, finish time, and a group's jobs.

2.2 Evaluation Function

The evaluation parameters of this problem are average manufactured time needed for completion of all jobs, the number of group changes, and the number of idle line periods. The manufactured time needed for completion is calculated from the start time of the first group containing the job to the finish time of the last group containing the job. The number of group changes means the number of times that the group process code is changed. The number of idle line periods is the number of idle times because no work is being done. The number of group changes and the number of idle line periods have weights for every line because of differences in performance.

There is a trade-off between shortening the average manufactured time and decreasing the number of the group changes and idle line periods. Therefore, the weighted sum of those three evaluation factors is used as a total evaluation value. The main purpose of this research is minimizing the evaluation value V, which is displayed as following expression:

$$V = W_T \times T + W_C \times C + W_R \times R, \qquad (1)$$

where the average manufactured time is expressed by T, its weight is W_T, the number of group changes of all lines is C, its weight is W_C, the number of idle line periods of all lines is R, and its weight is W_R.

However, this evaluation function is not perfect practically, because there are many another evaluation factors, which human experts judges empirically. Therefore, the total goal of this research is creation of the solution which has almost minimum value of the evaluation function and to which human experts' empirical demand of modification is reflected for two hours with minimum human works.

2.3 Difficulties of this problem

This problem is large-scale, because the number of object is about 2,000. Moreover, group compositions and order of groups makes a huge number of combinations. Therefore, combinatorial explosion occurs.

Additionally, human demand for modification is sensuous and empirical. So that, several repetitions of inputting human demands with trial and error and searching solutions which satisfies the human demands rapidly. Human interface and the scheduling method which searchs them in the limited hours are necessary.

3 A integrated scheduling system with tabu search

3.1 Configuration of the proposed system

In order to cope with those difficulties mentioned in Subsection 2.3, an integrated scheduling system with tabu search is proposed. The configuration of the system is shown in Fig. 2.

The core of this system is a basic scheduling engine using tabu search. In this engine, first, an initial solution is generated using heuristics based on flexibility, then the solution is improved by applying tabu search with empirical dispatching rules.

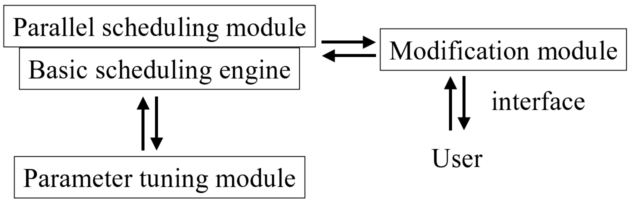

Figure 2: Configuration of an integrated scheduling system with tabu search

However, the target problem is too large scale, only the scheduling engine, calculation performance is not enough rapid. Therefore, a parallel scheduling module is added for rapid scheduling.

Furthermore, in parameter tuning of rule application rates, because they makes very important roles in quality of schedule and shortening calculation time, and it takes a large amount of time by manual tuning, automatic tuning module is developed.

Finally, for modification by human experts' demand, a modification module is proposed. There, human interface for inputting the demand and a control method of dispatching rules of tabu search for rapid modification, are prepared.

In the following subsections, the detail of algorithm of those engine and modules are described.

3.2 Basic scheduling engine

In a basic scheduling engine, 2 steps are performed in order to create a solution; namely, generation of an initial solution by analyzing job flexibility and repeated improvement of a solution by tabu search[2].

3.2.1 Generation of an initial solution by analyzing job flexibility

Based on an analysis of given job data, an initial solution is rapidly scheduled. When jobs are assigned to the schedule, each job has the influence on the whole schedule as a bottleneck. The degree of the influence on the whole schedule is defined as "flexibility." High job flexibility means that the job has less influence on the entire schedule, and low flexibility means a bigger influence. Flexibility consists of the following two elements:

- The number of processes
  The number of processes for completion is different for each job. A job with a large number of required processes has to be efficiently assigned because it is processed through many lines.
- Operating ratio
  Assuming that all jobs are assigned, the time required for processing each line is defined as operating time. The ratio of operating time of each line to the longest operating time in all lines is defined as the operating ratio. Since lines with large operating ratios process many jobs, a change of a schedule of the line can has large influence on another lines.

When the number of required processes of job i is L_i, the l-th process of job i is processed in line X_l, weights of those 2 elements are W_{f1} and W_{f2}, and operating ratio of line X_l is O_{X_l}, flexibility of job i is defined as:

$$\text{Flexibility of job } i = -(W_{f1} \times L_i + W_{f2} \times \sum_{l=1}^{L_i} O_{X_l}). \tag{2}$$

3.2.2 Improvement of a solution by tabu search

Tabu search is a general-purpose repetition improvement technique for solving combinational problems [3]. In a tabu search, at first, the neighborhood set $N(x)$ is generated, and defined as the set of solutions obtained by adding slight changes to a tentative solution x. Then the best solution y in $N(x)$ is selected as the candidate for the next tentative solution as shown in Fig. 3. Even when y is inferior to x, a tabu search continues with y as the next solution. To avoid returning to previous solutions, a "tabu list" is used. A tabu list is the set of the latest k solutions. The number of latest solutions k is called a size of the tabu list. The neighborhood set $N(x)$ is generated by excluding solutions in the tabu list.

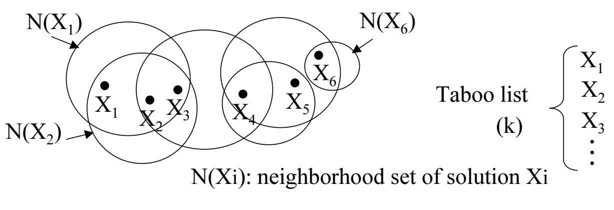

Figure 3: Outline of tabu search

When tabu search is applied to this problem, focusing attention on three kinds of evaluation parameters, three rules that can expect improvement of each evaluation parameter are prepared. Application of one of the rules will generate a neighborhood set. Moreover, the rule used when improving a schedule is randomly selected according to given their rule application rates.

3.3 Parallel scheduling module

In order to accelerate the basic scheduling engine, a parallel scheduling module is introduced[4]. Here by several computers, parallel calculation is performed. Each step, mentioned in Section 3.2, is improved, so that generation of plural initial solutions and parallel search method are introduced. Fig. 4 shows an outline of this module.

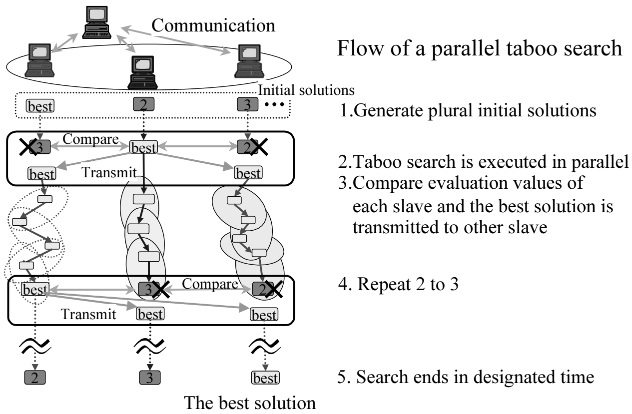

Figure 4: Outline of parallel scheduling module

3.3.1 The plural initial solution scheduling method

The basic scheduling engine has created an initial solution using the flexibility mentioned in Subsection 3.2.1. In order to generate plural initial solutions and to find a better solution in early stage of parallel searching, the margin of production time limit is introduced as a new component to determine flexibility. Here, the margin of production time limit is slack time of required all process until finish time of the last process of a job. Whether the margin of production time limit of a job is prior, average manufactured time has a various value. Therefore, by introducing the margin of production time limit, initial solutions with various average manufactured time can be created. New flexibility is defined as the following expressions:

$$\text{Flexibility of job } i = -(W_{f1} \times L_i + W_{f2} \times \sum_{l=1}^{L_i} O_{X_l}) + \frac{2S+j}{2S} W_{f3} \times D_i \quad (j = -S+1, \cdots, -1, 0, 1, \cdots, S). \tag{3}$$

where the margin of production time limit is D_i, and the number of initial solutions is $2S$. By changing j from $-S+1$ to S, $2S$ expressions of flexibilities can be created and plural initial solutions are generated. The initial solutions are used in a parallel search method explained in the next subsubsection.

3.3.2 A parallel search with tabu search

In parallel search, a master computer gets problem data as input, controls some slave computers, and outputs a schedule. In each slaves, tabu search with different random factor is performed in order to prevent a solution from being in local optimum. When a slave finds the best solution, the solution is send to another slaves through the master as a new tentative solution.

An example is illustrated in Fig 5, which shows that Slave 2 has begun a search from solution X_{10} of Slave 1. To avoid returning to the solutions that Slave 1 has already found, the size of the tabu list of Slave 2 changes to $T(> k)$ and receives the tabu list of Slave 1 with five recent solutions (X_5, X_6, X_7, X_8, X_9). On the other hand, the size of the tabu list of Slave 1 continues searching for the remaining k (Constant k is the size of the tabu list explained in Section 3.2.2). Slave 2 begins the first search from solution A_{11} and the size of tabu list of Slave 2 keeps T until T-th search. After $(T+1)$th search, the size of the tabu list is changed to k because the neighborhood set of Slave 2 is separated from Slave 1's.

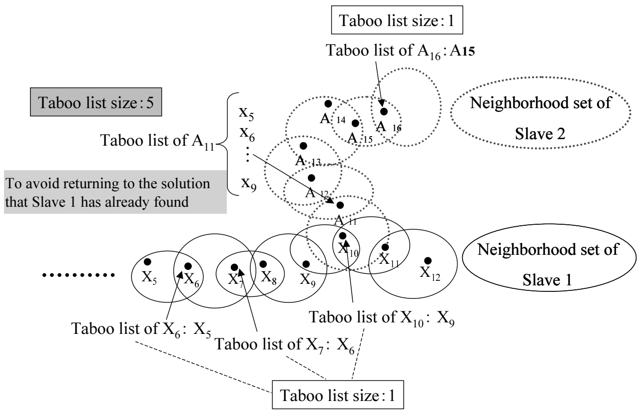

Figure 5: Division of search space ($k = 1, T = 5$)

3.4 Parameter tuning module

Rule application rates in scheduling engine make very important roles in quality of schedule and shortening calculation time. Additionally, a large amount of time and human experts' work are necessary in manual tuning. Therefore, automatic tuning module is developed.

In this module, line data, several sets of job data, and initial rule application rates are input, and new rule application rates suited to solve the sets of job data are output by repeatedly tuning.

Issues of this automatic tuning is increasing the application rates of the rules, whose improving abilities are high or the rules which improve solutions well by the combination with another rules.

The automatic tuning algorithm is described as follow: First, The basic schedule engine are executed with current rule application rates against each set of job data. From the result, improved evaluation values by application of dispatching rules in each repetition are extracted, and the following two factors are calculated;

- average of decreased evaluation value when every rule is applied once
- percentage when evaluation value is decreased by application of every rule

From the products of those two factors of each rule, the following tuning procedure is executed. All the products are normalized between $-p$ and p. p is decreased as tuning repetition advances. The normalized product of each rule is added to current application rate of it.

Finally, evaluation value of the final solution is compared to the one of the previous repetition of tuning. If current value is worse than the previous one, the added value in the previous repetition is subtracted from the current application rate of each rule randomly with probability 1/2.

3.5 Modification module

A modification module has an interactive user interface, where new constraints using evaluation factors and another sub-evaluation factors can be input. A solution which satisfy all those constraints have to be found by the basic engine. The dispatching rules that improve each input demand are prepared to improve solution for tabu search. However, such rules have different feature and have bad influences to another rules and there is also a trade-off among such rules. So that, a method of dynamic change of applying rate of rules in order to apply plural rules in effective order is proposed.

When a dispatching rule is selected in tabu search of the basic scheduling engine, the application rate of each rule are changed whether to satisfy the corresponding demands. Where total number of rules is N, rule weights are $W1_i$(in the case that the corresponding demand i is satisfied) and $W2_i$(in the case that the corresponding demand isn't satisfied), and flag is S_i(if demand i is satisfied then S_i=0, otherwise S_i=1), application rate L_k of rule k is defined as follow:

$$L_k = \frac{W1_k \times S_k + W2_k \times (1 - S_k)}{\sum_{i+1}^{N}(W1_i \times S_i + W2_i \times (1 - S_k)}$$

When the demands are satisfied, rule application rate changes dynamically by the above formula. As shown in Fig. 6, when demand A is satisfied, the weight of rule A becomes zero, and rule application rate changes. Then rule B,C can be applied in high probabilities.

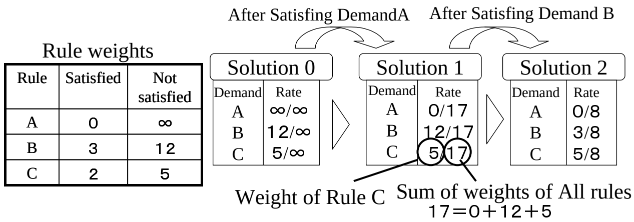

Figure 6: Dynamic change of rule application rates

4 An application to practical problems

4.1 Parameters of the proposed method

The proposed method was applied to a real large-scale job shop problem with group constraints. The problem has about 2000 jobs and 13 production lines.

The experiment was executed with one master (OS: Windows 2000, CPU: Pentium IV 2.0GHz, MEMORY: 512 MB) and several slaves (OS: Windows 2000, CPU: Pentium IV 2.4GHz, MEMORY:512 MB). The weights of evaluation function shown in Section 2.2 are set up with $W_T = 1, W_C = 5$, and $W_R = 1$.

Parameters used in parallel tabu search was determined as the following:

- The size of tabu list: $k = 1$
- Weights of flexibility: $W_{f1} = 1000, W_{f2} = 100$, and $W_{f3} = 0.1$
- The size of tabu list in parallel tabu searches: $T = 5$
- The number of initial solutions: $2S = 10$
- Search time: one hour

4.2 Evaluation of parallel tabu search

In order to confirm effectiveness of the parallel tabu search, two experiments were executed. One is the parallel tabu search by three slaves. The other is single tabu searches with different random factors in which 3 slaves do not exchange solutions. The result is shown in Table 1.

Table 1: Evaluation value of parallel and single tabu search using data 2

	Slave (Single)			Parallel
	1	2	3	
Result 1	13436	13774	14776	12218
Result 2	13961	13523	14628	12990
Result 3	12826	13589	14819	12200

In Table 1, the solution by parallel search is better than the best solution of the three slaves in a single search. In the results of another job data, it can be confirmed that the parallel search is superior to the single search.

4.3 Evaluation of parameter tuning

Against 7 dispatching rules and 3 sets of job data, experiments of parameter tuning were performed. Averages evaluation value of scheduled solutions of those 3 sets of job data by using the tuned application rates is shown in Table 2. This tuning takes 18 hours. The last and 6th repetition is the best.

The proposed method is compared to manual tune made for one week. Both of tuned application rates are applied to another 2 sets of job data for evaluate. The average value of the proposed method and manual tune are 9979 and 10097 respectively. It is also confirmed that the value of the proposed method is almost same with experts' one, but it is much better in tuning time.

Table 2: Average values of the evaluation function and factors by automatic tuning

tuning times	evaluation function	man. time	group change	idle periods
1	12029	6392	202	23
2	12086	6168	211	22
3	11803	6009	208	19
4	11790	5698	217	23
5	12135	6216	212	23
6	11313	5456	209	24

4.4 Evaluation of modification

The experiment about calculation time was performed. 2 set of input demands are prepared. Fig. 7 shows comparison of time applying the rules by constant rates and applying rules by the proposed method.

The proposed method, decrease much calculation especially against many demands. As a number of demand increases, calculation time increases linearly in the proposed method, while time increase exponentially in the constant probability. From these result, it can be concluded that several interactive modification can be executed within one hour.

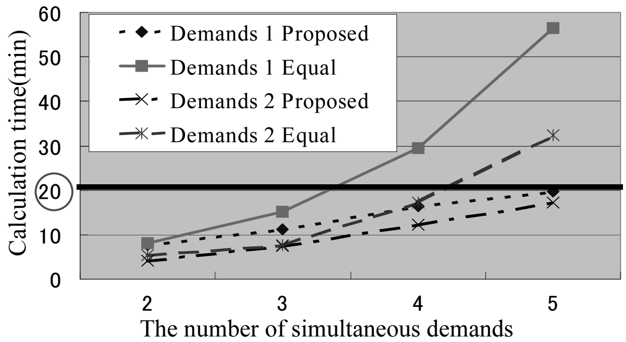

Figure 7: Calculation time to the demands

5 Conclusion

In this research, for large-scale job shop problems with group constraints, an integrated scheduling system with tabu search is proposed. It consists of a basic scheduling engine and a parallel scheduling, a parameter tuning and a modification modules.

Through several experimentations, it is confirmed that this integrated scheduling system can generate final solution which satisfies all constraints and humans demands, and whose evaluation value is almost equal to the experts' one for two hours without human works for setting time of rule application rates.

References

[1] M. Pinedo, *Scheduling Theory, Algorithms, and systems*, Prentice Hall, 1995.

[2] H. Ohmae, K. Ishihara, Y. Ikkai, K. Horiuchi, and H. Hamamoto: A High Speed Scheduling Method by Analyzing Job Flexibility and taboo Search for A Large-Scale Job Shop Problem with Group Constraints, *in Proc. of 9th IEEE Int. Conf. on Emerging Technologies and Factory Automation*, Vol.2, 2003, pp.227-232.

[3] J. W. Barnes and J. B. Chambers, "Solving the Job Shop Scheduling Problem with Tabu Search," *IIE Transactions*, Vol.27, 1995, pp.257-263.

[4] N. Kawamura, Y. Kawakita, Y. Ikkai, and N. Komoda, "A High Speed Scheduling Method by Parallel and Asymptotic Taboo Search for A Large-Scale Job Shop Problem with Group Constraints," *2004 Int. Conf. on Systems, Man and Cybernetics*, 2004, pp.1452-1457.