15th Triennial World Congress, Barcelona, Spain

A COMBINED ALGORITHM FOR SOLVING JOB SHOP SCHEDULING PROBLEMS

Yajie Tian, [*] **Nobuo Sannomiya,** [**] **Changzheng Xie** [**] **and Tetsuo Sawaragi** [***]

[*] *Graduate School of Informatics, Kyoto University, Japan*
[**] *Kyoto Institute of Technology, Japan*
[***] *Graduate School of Engineering, Kyoto University, Japan*

Abstract: In this paper, a combined algorithm consisting of two stages of approaches is proposed for solving a Job Shop Scheduling Problem (JSSP). The first stage of approach called Semi-active Scheduling Approach (SSA) is proposed for obtaining a local optimal solution in a short time by using the improved local search algorithm and the neighborhood search technique proposed in our earlier paper. The second stage of approach called Active Scheduling Approach (ASA) is proposed for improving the solution obtained from SSA and preventing the solution from trapping in the local minimum by reducing the idle time in the processes and making good use of the resources. Both of the approaches focus their improvement efforts on reducing production expense. The proposed algorithm is applied to solving JSSPs. A large number of computational experiments show that the combined algorithm can overcome the disadvantages of the respective approaches, SSA and ASA, and obtain a better solution as compared with single use of such approaches for solving the complicated JSSPs. In addition, the proposed algorithm is compared with Genetic Algorithm (GA), and the result shows that in the case of limited total search points, the proposed algorithm can converge to a suboptimal solution faster than GA.

Keywords: Optimization, Local search method, Job shop scheduling problem, Semi-active scheduling, Active scheduling.

1. INTRODUCTION

In the site of production, an optimal algorithm with simple method and short computation time is always necessary and well received, because almost real world problems have a rigorous due date and are restricted with complicated conditions. In this view, local search algorithm (LS) is as one of the simplest method usually used in the site. However, LS is easy to trap in a local minimum, and the solution obtained from LS often cannot be satisfied by the managers.

For this reason, many researchers focus their efforts on improving local search techniques, and many methods such as simulated annealing (SA) (Kirkpatrick *et al.*, 1983), tabu search (TS) (Glover, 1986) and so on have been proposed for the purpose. SA has been proposed to avoid being trapped in a local minimum, but its disadvantage is to have a possibility of getting back to solutions already visited. TS has been proposed to avoid the disadvantage of simulated annealing by storing the only accepted solutions in a tabu list. However, storing all visited solutions in a tabu list and testing the tabu list is generally too consuming both in the term of memory and computational time. Therefore, many efforts on overcoming the disadvantage have been done by improving the tabu list and the neighborhood.

In the paper (Tian *et al.*, 2000), a tabu search with a new neighborhood was proposed for reducing the term of memory and computational time. In another paper (Nakano *et al.*, 2001), an improved local search method (ILS) with two search stages was proposed for solving JSSP. An optimal size of neighborhood was obtained from a large number of statistic experiments. But it cannot solve a problem of determining the best switching time from the first stage to the second stage.

In this paper, a combined algorithm consisting of two stages of approaches is proposed for solving JSSPs. The first stage of approach called semi-active scheduling approach (SSA) is proposed for obtaining a local optimal solution in a short time by using ILS and the neighborhood search technique proposed in the paper (Nakano *et al.*, 2001). The second stage of approach called active scheduling approach (ASA) is proposed for improving the solution obtained from SSA and for preventing the solution from trapping in the local minimum by making good use of the resources with reducing the idle time in the processes. A new method to determine the switching time from SSA to ASA is proposed in this paper by using the concept of a check buffer, which is used to store the failed negotiating procedures. An optimal size of check buffer is found by carrying out a large number of computational experiments.

As we know, any real world system has at least one constraint to give the system a restriction from achieving its objective. For the production scheduling problems, the constraint is considered as a bottleneck process, in which the demanded resource exceeds the holding resource of the process or the resources cannot be used in a good way. In this paper, we try to make good use of the resources by reducing the idle time existing in the processes and obtain an optimal assignment between SSA and ASA. In addition, we assume that the total number of search points is limited in this paper. Therefore, the assignment of the local search effort and the global search effort, and the optimal size of the neighborhood are adopted according to the results obtained from the paper (Nakano *et al.*, 2001). The proposed algorithm is applied to solving classical JSSPs, and the effectiveness is examined by comparing it with GA.

2. JOB SHOP SCHEDULING PROBLEM

A classic job shop scheduling problem is solved in this paper. A job shop has H machines M_j $(j = 1, 2, \cdots, H)$, each of which can do several types of operations but can process only one activity at a time. N jobs J_i $(i = 1, 2, \cdots, N)$ are variables for processing at time zero. Job J_i consists of H operations Q_{ij} $(j = 1, 2, \cdots, H)$ in series. The input tables of processing time P_{ij} of operation Q_{ij} and the machine routing of operation Q_{ij} must be given. The machine routing expresses the sequence of machine M_j on which operation Q_{ij} is to be run. Q is the set of the total operations and the number of elements in Q is NH.

The basic constraints for assigning an activity to start are as follows:

- It cannot start until the job's preceding operation is finished.
- It cannot start until the machine is free.
- Two operations cannot be processed on a machine at the same time and the operations $(Q_{ij};\ j = 1, 2, \cdots, H)$ must be processed according to the ascending order of j.

The objective of the problem is to find a suboptimal solution to minimize the makespan. The objective function is denoted by

$$Z = \max_{ij}\{C_{ij}\} \tag{1}$$

where $C_{ij} = S_{ij} + P_{ij}$. S_{ij} and C_{ij} are the start time and the completion time of operation Q_{ij}, respectively.

3. COMBINED ALGORITHM

A combined algorithm consisting of two stages of approaches is proposed for solving JSSPs. In the first stage of approach (SSA), the improved local search method is used for searching a local optimal solution in a short time. In the second stage of approach (ASA), the operation negotiating procedure and the method of reducing idle time are used for preventing being trapped in a local minimum. Both approaches improve the solution by using operation negotiating procedure. The operation is called a manager if it is chosen as the negotiator and the operation is called as a contractor if it is chosen as a subject of the manager.

The definition of a neighborhood is always important for improving the accuracy of solutions. A definition of neighborhood was proposed in the paper (Nakano *et al.*, 2001) and an optimal size K of neighborhood was estimated by a large number of computational experiments.

Figure 1 shows the definition of neighborhood. In this figure, a job sequence is given as

$$J_1, J_2, J_3, J_4, J_3, J_2, J_1, J_4, \cdots, J_3, J_1 \tag{2}$$

for the case where $(N, H) = (4, 4)$. The corresponding operation sequence is

$$Q_{11}, Q_{21}, Q_{31}, Q_{41}, Q_{32}, Q_{22}, Q_{12}, \mathbf{Q_{42}}, \cdots, Q_{34}, Q_{14} \tag{3}$$

$\mathbf{Q_{42}}$ is chosen as the manager and $K = 7$ is chosen as the neighborhood size. Then operations

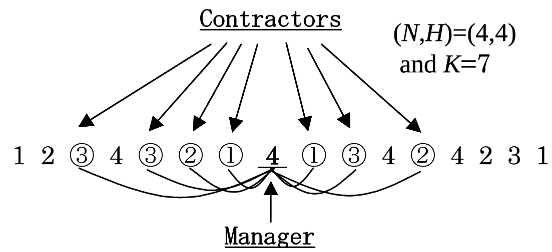

Fig. 1. The definition of neighborhood

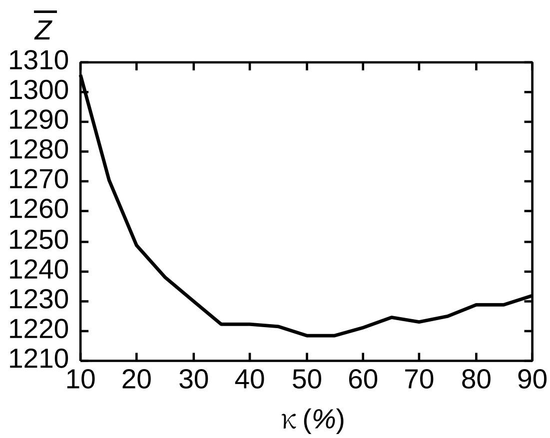

Fig. 2. Variation of the suboptimal objective value with κ for the case of $(N, H) = (20, 15)$

$\{Q_{31}, Q_{32}, Q_{22}, Q_{12}, Q_{13}, Q_{33}, Q_{23}\}$ are neighboring elements of the manager $\mathbf{Q_{42}}$ and they are the contractors.

A large number of computational experiments were carried out by using ILS and changing the parameter $\kappa(= \frac{NH}{K})$ (Nakano *et al.*, 2001). The optimal size of neighborhood was obtained as $\kappa = 0.54$. As an example, Figure 2 shows the variation of the mean of suboptimal objective values with the size κ for the case of $(N, H) = (20, 15)$. The value of $\bar{Z}$ is the average result for five problems and each of which was run 100 times. We find that $\bar{Z}$ reaches its smallest value when $\kappa_{opt} \cong 0.5$, that is $K_{opt} \cong 0.5NH$. It is observed that the large size of neighborhood is not optimal for SSA. Decreasing the size K of neighborhood means increasing the iteration number $t^* = \frac{L}{K}$ and the opportunity of searching a suboptimal solution in the case of limited total number L of search points.

In this paper, a set $B_c = \{\hat{Q}_{ij}\}$ called a check buffer is introduced in the algorithm for checking the improving situation of the solution. $\hat{Q}_{ij}$ is the operation which is chosen as the manager and whose negotiating procedures are failed. The size of B_c indicates the measure of difficulty for improving the solution. A large value of B_c shows that the solution tends to a converging situation of search process. The situation of $B_c = Q$ indicates that any operation chosen as a manager cannot improve the solution by using the SSA. In this case, it is considered that the solution is trapped in a local minimum and then continuation of SSA is useless. Consequently, the active scheduling approach (ASA) should be executed for escaping from the local minimum.

Therefore, it is important to determine the best conversion chance from SSA to ASA. Too early starting of ASA will lead to increasing computational time and too late starting of ASA will lead to losing the opportunity to search an optimal solution. In this paper, a parameter μ ($\in (0, 1]$) is defined as $|B_c| = \mu|Q|$) and is used to determine the optimal conversion of ASA. A large number of computational experiments are carried out by changing μ from 0.3$\sim$1.0 for finding an optimal value of μ.

The Combined Algorithm is mentioned as follows.

Combined Algorithm

Step1. Generate an initial solution randomly. Then the order of operation is Π_0 and the corresponding objective value is Z_0. Set the iteration number as $t = 1$ and the check buffer $B_c = \Phi$. Set $flag = 0$.

Step2. If $|B_c| = |Q|$ and $flag = 1$, then go to Step8. Otherwise, choose a manager $Q_{nm} \in \Pi_{t-1}$ randomly. If $Q_{nm} \in B_c$, then repeat Step2. If $Q_{nm} \notin B_c$, $B_c \Leftarrow B_c \cup Q_{nm}$.

Step3. Choose K_{opt} operations Q_{ij}, $i \neq n$ as the contractors according to the distance from the manager Q_{nm}. Carry out the negotiating procedures between the manager and the respective contractors. Then obtain K_{opt} orders of operations Π^k, $k = 1, 2, \cdots, K_{opt}$. If $flag = 0$ and $|B_c| < \mu|Q|$, go to Step4. Otherwise, let $flag = 1$ and go to Step5.

Step4. (SSA) Calculate the objective functions Z^k corresponding to Π^k, $k = 1, 2, \cdots, K_{opt}$ obtained from Step3 by using SSA. Choose the best order Π^{best} such that $Z^{best} = \min_k\{Z^k\}$, and go to Step7.

Step5. (ASA) Calculate the objective functions Z^k corresponding to Π^k, $k = 1, 2, \cdots, K_{opt}$ obtained from Step3 by using ASA. Choose the best order Π^{best} such that $Z^{best} = \min_k\{Z^k\}$.

Step6. (ASA) Searching the available idle time from each process in Π^{best} and make use of it. If there exist more than one available idle times, choose the one having the smallest value of the objective function.

Step7. If $Z^{best} < Z_{t-1}$, then set $Z_t = Z^{best}$, $\Pi_t = \Pi^{best}$ and $B_c = \Phi$. If $Z^{best} = Z_{t-1}$, then set $Z_t = Z^{best}$ and $\Pi_t = \Pi^{best}$. If $Z^{best} > Z_{t-1}$, then set $Z_t = Z_{t-1}$ and $\Pi_t = \Pi_{t-1}$.

Step8. If $t = t^*$, or if $B_c = Q$ and $flag = 1$, then the solution obtained from Step7 is adopted as the suboptimal solution and stop the

Table 1. Benchmark problems used in the computational experiments

(N,H)	Examples				
(15,10)	la21	la22	la23	la24	la25
(15,15)	ta01	ta02	ta03	ta04	ta05
(20,15)	abz7	abz8	abz9	swv06	swv07
(20,20)	yam01	yam02	yam03	yam04	ta25

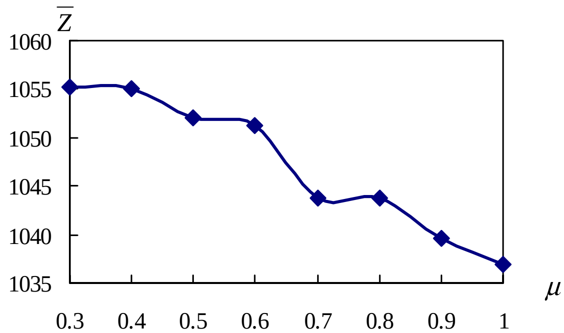

Fig. 3. Variation of the mean suboptimal objective value with μ ($|B_c| = \mu|Q|$) for the case of $(N,H) = (15,10)$

computation. Otherwise, $t \Leftarrow t+1$ and go back to Step2.

In the above algorithm, the stop condition is related to the check buffer B_c and iteration number t^* for both SSA and ASA. The first stage of approach (SSA) is stopped when $|B_c| = \mu|Q|$ or $t = t^*$. The second stage of approach (ASA) is stopped when $|B_c| = |Q|$ ($i.e. \mu = 1$) or $t = t^*$. The switching time to ASA is decided when $|B_c| = \mu|Q|$ in the first stage. The negotiating procedure is successful, only if the maximum makespan does not become worse. The check buffer B_c is made clear only if the solution is improved. In the proposed algorithm, the computation can stop automatically according to the size of check buffer B_c. Therefore, even though the total number of search points is set to be a large value, the real search points does not depend on it, but depend on the size of the check buffer B_c.

4. COMPUTATIONAL EXPERIMENTS

The proposed algorithm is applied to solving classical JSSPs. A large number of computational experiments are carried out by changing the parameter μ of the check buffer from 0.3 to 1. Table 1 shows the job shop scheduling problems (Mattfeld and Gambardella, 1995; Shi *et al.*, 1996) solved in this paper. The sizes of problems are $(N,H) = (15,10)$, $(15,15)$, $(20,15)$ and $(20,20)$. The number L is set to be 45000, 60000, 200000 and 600000 for the corresponding problem size. The size of neighborhood for both SSA and ASA is set to be $K_{opt} = 0.54|Q|$ (Nakano *et al.*, 2001).

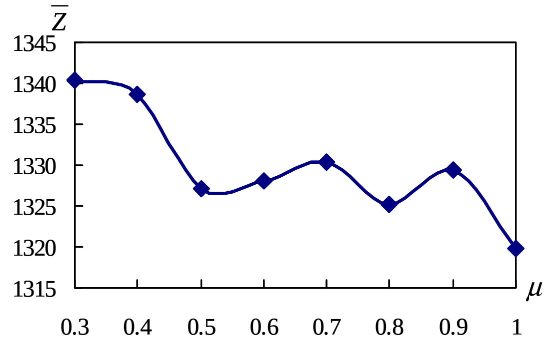

Fig. 4. Variation of the mean suboptimal objective value with μ ($|B_c| = \mu|Q|$) for the case of $(N,H) = (15,15)$

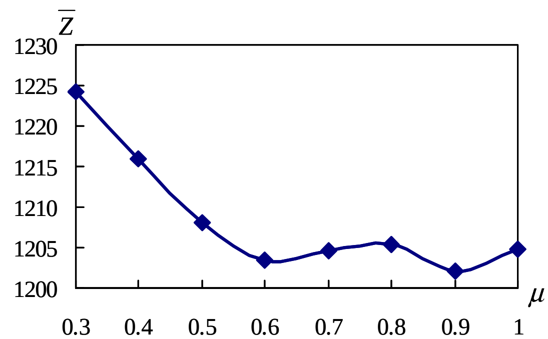

Fig. 5. Variation of the mean suboptimal objective value with μ ($|B_c| = \mu|Q|$) for the case of $(N,H) = (20,15)$

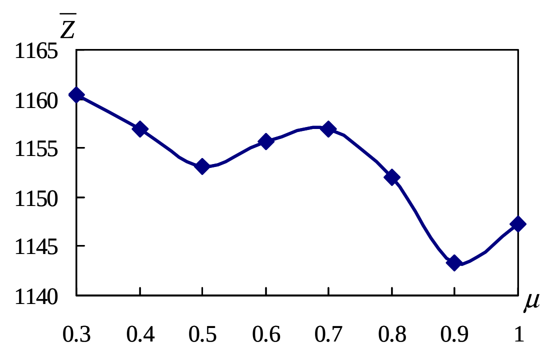

Fig. 6. Variation of the mean suboptimal objective value with μ ($|B_c| = \mu|Q|$) for the case of $(N,H) = (20,20)$

Figures 3 to 6 show the variation of the mean suboptimal objective values for the respective problem sizes. As shown in Table 1, five examples are used for the corresponding size. Each example is run 10 times and the average value is calculated. Consequently, each point of these figures shows the average value of $5 \times 10 = 50$ trials.

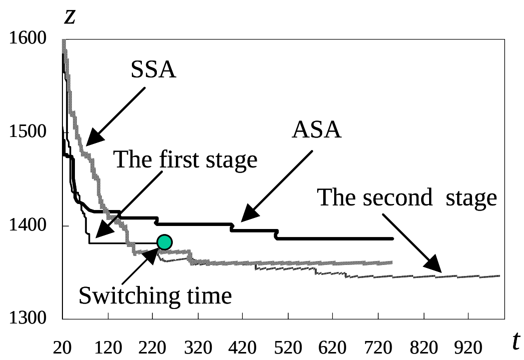

Fig. 7. Comparison of the suboptimal objective values obtained from SSA, ASA and the proposed algorithm ($|B_c| = |Q|$, ta02 $(N, H) = (15, 15)$)

As shown in Figures 3 to 6, the experimental results show that the average value $\bar{Z}$ of the objective function varies with the parameter μ. For problems of $(N, H) = (15, 10)$ and $(15, 15)$, the minimum value of $\bar{Z}$ is obtained when $\mu = 1$. For problems of $(N, H) = (20, 15)$ and $(20, 20)$, the minimum value of $\bar{Z}$ is obtained when $\mu = 0.9$. It is observed that for all the problems solved in this paper, the objective function can reach its minimum value when μ takes a large value, 0.9 or 1 in the sense of average.

This means that the best switching time to ASA is $B_c \to Q$. That is, for searching an optimal solution by using the combined algorithm, it is necessary to make good use of the improving opportunities of SSA. The reason is that the solutions obtained from ASA have a low diversity because it improves the solution by using the operation negotiating procedure and making good use of the available idle time from a provisional solution. As mentioned above, the available idle time decreases with the solutions being improved. On the other hand, the solutions obtained from SSA have a high diversity because it searches a solution by using the random negotiating procedures without any special deterministic means. Since there exists much idle time in the processes, the solution is easy to be improved. Therefore, taking much more iterations to carry out SSA, i.e. $B_c \to Q$, and then using ASA to improve the deadlock situation of SSA is effective to find a suboptimal solution in the case of limited total number L of search points. The converging situations of the three algorithms, SAS, ASA and the proposed algorithm are shown in Figure 7 for the problem ta03 of $((N, H) = (15, 15))$.

We compare the proposed algorithm with single use of SSA or ASA subject to the same total number L of search points and the same size K_{opt} of neighborhood. The results are shown in Table 2. Each case is run 10 times, and the average value and the best value obtained are shown. The results show that under the condition of limited L, the solution obtained from the proposed algorithm is much better than those obtained from either SSA or ASA. The reason is considered as follows. SSA can converge fast in a short time, but the solution is easily trapped in a local minimum. ASA can improve the deadlock situation of local minimum, but the degree of freedom is limited by its special search procedure. On the other hand, the combined algorithm can make good use of both advantages of SSA and ASA. Thus it is more effective for solving JSSPs than each of SSA and ASA.

The comparison between the proposed algorithm and GA is also carried out under the assumption of limited total number L of search points. The results are also shown in Table 2. We find that the solution obtained from the proposed algorithm is better than that obtained from GA in the case of limited L. The reason is considered as follows. GA is essentially a global search method, which deals with a population (a set of solutions) instead of a single solution. Consequently, a large number of search points is necessary. As shown in the paper (Nakano *et al.*, 2001), if the total number L of search points is increased, GA can obtain a better solution than ILS. However, most of real world problems have rigorous due dates, and convergence to a suboptimal solution in the limited short time is significant and important. In this view, the proposed combined algorithm is effective and practically important for solving JSSPs.

5. CONCLUSION

A combined algorithm consisting of two stages of approaches has been proposed for solving job shop scheduling problems. In the first stage of approach (SSA), an improved local search algorithm and the method of neighborhood search are used for seeking a local optimal solution in a short time. In the second stage of approach (ASA), a method of the negotiating procedure with good use of the available idle time in each process is used to dissolve the deadlock situation caused in the first stage of approach.

In the case of limited resources such as limited total number of search points, we have compared the results of applying the proposed algorithm, GA, SSA and ASA to 4 kinds of 20 problems. The computational results show that the proposed algorithm obtains a better solution than others. The reason is considered as follows. GA is a probabilistic algorithm, which applies the operations such as crossover, mutation and selection to a set

Table 2. Comparison of suboptimal objective value

(i) $(N, H) = (15, 10)$ and $L = 45000$

Case	SSA+ASA	SSA	ASA	GA	Optimal solution
la23	1046/1032	1058/1032	1072/1032	1052/1032	1032
la24	1000/963	1010/954	1027/990	1004/988	935

(ii) $(N, H) = (15, 15)$ and $L = 60000$

Case	SSA+ASA	SSA	ASA	GA	Optimal solution
ta01	1331/1266	1351/1314	1373/1309	1353/1322	1231
ta02	1325/1285	1375/1301	1388/1348	1348/1316	1244

(iii) $(N, H) = (20, 15)$ and $L = 200000$

Case	SSA+ASA	SSA	ASA	GA	Lower bound
abz7	713/694	724/693	743/722	736/720	655
abz8	728/707	731/718	757/741	758/737	638

(iv) $(N, H) = (20, 20)$ and $L = 600000$

Case	SSA+ASA	SSA	ASA	GA	Lower bound
yam02	983/951	995/966	1022/1000	1008/975	861
yam03	974/935	975/939	1019/1005	971/947	827

Note: */* expresses [average value]/[best value obtained]

of candidates for a solution. An optimal solution obtained from GA needs a large number of search points. Thus, it is hard for GA to obtain an optimal solution in a limited computation time. SSA uses an improved local search algorithm to search a locally optimal solution. It can obtain a locally optimal solution fast, but is easy to trap in a local minimum. Therefore, it is difficult to obtain a good suboptimal solution by using SSA only. ASA pays attention to dissolve the deadlock situation caused by SSA through making good use of the available idle time. Since the available idle time decreases with the improvement of solutions, the opportunities of searching a suboptimal solution decrease gradually. On the other hand, since the combined algorithm makes good use of both advantages of SSA and ASA, it can obtain a good suboptimal solution more easily than SSA and ASA. At the same time, it can reach a suboptimal solution earlier than GA in the case of limited search time.

As we know, most of real world problems have rigorous due date and all such problems must be solved within a limited computation time. Therefore, it is more important to obtain a suboptimal solution fast than to obtain the optimal solution in sufficiently much computation time. In this view, the proposed combined algorithm is effective and applicable to solving JSSPs in the real world.

6. REFERENCES

Glover, F. (1986). Tabu search-part 1. *ORSA J. Computing* **1**, 190–206.

Kirkpatrick, S.Jr., C.D. Gelatt and M.P. Vecchi (1983). Optimization by simulated annealing. *Science* **220**, 671–680.

Mattfeld, D.C. and L.M. Gambardella (1995). Is there a way out?. *Proc. of Genetic Algorithms* pp. 506–511.

Nakano, T., Y. Tian and N. Sannomiya (2001). An improved local search method for soving job shop scheduling problems. *Trans. of IEE Japan* **121C**(10), 1627–1633 (in Japanese).

Shi, G., H. Iima and N. Sannomiya (1996). A new encoding scheme for solving job shop scheduling problems by genetic algorithm. *Proc. of the 35th IEEE Conf. on Decision and Control* **4**, 4395–4400.

Tian, Y., N. Sannomiya and Y. Xu (2000). A tabu search with a new neighborhood search technique applied to flow shop scheduling problems. *Proc. of the 39th IEEE Conf. on Decision and Control* pp. 4606–4611.