

GENETIC LOCAL SEARCH ALGORITHMS FOR SINGLE MACHINE SCHEDULING PROBLEMS WITH RELEASE TIME

Jihchang Hsieh¹, Peichann Chang², Shihhsin Chen²

¹Department of Industrial Management, Vanung University, Chung-Li 32061, Tao-Yuan, Taiwan, R.O.C., Email: hjc@vnu.edu.tw. ²Department of Industrial Engineering and Management, Yuan-Ze University, Ne-Li, Tao-Yuan, Taiwan, R.O.C..

Abstract: Single machine scheduling problems with release time are the prototypes for other complicated scheduling systems. Genetic local search algorithms are proposed to minimize total weighted completion time. Numerical experiments to evaluate the three embedding local search techniques are done. The results indicate that the $2r_i+t_i$ rule proposed by Liu and MacCathy (1991) outperform ERT and WSPT.

Key words: Genetic Local Search Algorithm, Single Machine Scheduling, Release Time.

1. INTRODUCTION

Production scheduling is to arrange job sequences under some manufacturing resources. In this paper, a single machine scheduling with release time is discussed. Although single machine describes one of the most basic production environments, the results can be extended in many other complicated production systems. Therefore it is the most applicable. However, the single machine scheduling problem with release time does not receive much attention in the contrast of the prosperous developments of other scheduling problems.

The traditional ideas developed to solve single machines scheduling with release time is to search optimum solutions. Branch-and-bounds algorithms were ever proposed by Dessouky and Deogun (1981), Deogun (1983), Hariri and Potts (1983), Chu (1992), Chand et al. (1996b), Chang et al. (2004). Rinnooy Kan (1976) has shown that single machine scheduling with release

Please use the following format when citing this chapter:

Hsieh, Jihchang, Chang, Peichann, Chen, Shihhsin, 2006, in International Federation for Information Processing (IFIP), Volume 207, Knowledge Enterprise: Intelligent Strategies In Product Design, Manufacturing, and Management, eds. K. Wang, Kovacs G., Wozny M., Fang M., (Boston: Springer), pp. 875-880.

time is NP-hard. Therefore to search optimum solutions is not efficient. To achieve efficiency, Liu and MacCarthy (1991), Reeves (1995), Chand et al. (1996a) proposed heuristic methods. Those heuristics work for some specified assumptions. Once one of the assumptions is violated, the performance is not guaranteed. In order to sustain stable and satisfactory performance, genetic algorithms are applied in this paper.

Genetic algorithms have widely applied in the optimization problems, of course in scheduling problems as well. However, in the open literature, genetic algorithms only have few applications on single machine scheduling with release time including Liu and Tang (1999), Madureira et al. (2001). In contrast to the rich applications of genetic algorithms on the other fields, it is expected that genetic algorithms should have more contributions in solving the single machine scheduling problems with release time. Therefore a genetic algorithm embedding local search techniques is presented.

2. PROBLEM DEFINITION

A single machine scheduling problem investigates the possible arrangements for jobs that need to be processed on only one machine under one or some specified constraints. Baker (1974) indicated that the research is very important because it may provide good ideas for complex systems. Most of the prior research commonly assumed that jobs are available at time zero. Such an assumption simplifies the complexity of scheduling. But this is not practical since not all jobs are available at time zero in the real-world cases. Therefore, to consider release time is meaningful in scheduling problems.

The following assumptions are used to characterize the scenario of this research.

1. There is a set of independent, single-operation jobs.
2. Release times are positive and not all zeros.
3. Setup times are independent of sequence and included in processing time.
4. The attributes of each job, i.e. release time, processing time and weight, are known in advance.
5. Only one machine is continuously available.
6. No job preemption.

In the shop floor, it is important to keep the work-in-process low in order to control the holding cost. Completion time is an index to show the level of work-in-process. Moreover, each job has its priority to be completed. The priority can be represented in terms of weights. Therefore the total weighted

completion time is a suitable objective. In order to minimize the weighted completion time, a genetic local search algorithm is proposed.

3. GENETIC LOCAL SEARCH ALGORITHMS

A genetic local search algorithm is developed to solve the single machine scheduling problem with release time and the detailed procedure is briefly described as follows:

Encoding: Integer encoding.

Generating initial population: The chromosomes in the initial population are generated by random number generation.

Fitness function: Since this problem focuses on minimizing the total completion time, the fitness function needs to be transformed as the following form:

$$fit(x) = MAX - f(x)$$

To prevent the fitness from being zero, the MAX is set as:

$$MAX = f_{max}(\Psi) \times 1.1$$

Where $f_{max}(\Psi)$ means the maximum objective value in population Ψ .

Reproduction: Roulette wheel method (Goldberg, 1989).

Crossover: Two-point crossover (Goldberg, 1989).

Mutation: Shift mutation (Goldberg, 1989).

Local Search: Three heuristic-based local search rules are proposed in this research.

1. $2r_i+t_i$: Liu and MacCathy (1991) proposed a heuristic rule to sequence the jobs by $2r_i+t_i$, where r_i and t_i respectively denote the release time and processing time of job i . It is proven that $2r_i+t_i$ rule obtains optimum solutions when there are only two jobs. Therefore, this property is applied to local search. It is expected to improve the solution quality through $2r_i+t_i$ rule.

2. ERT: Earliest release time (ERT) rule is applied to local search.

3. WSPT: Weighted shortest processing time (WSPT) rule is applied to local search.

Any two consecutive jobs are randomly selected in each sequence after mutation. Then the jobs are checked if the heuristic-based local search rule is met. If the rule is not met then the positions of the two jobs are interchanged, else check the next sequence till all the sequences are checked off.

Replacement: An elite preserve strategy is applied. The top 10%

sequences are preserved for the next generation.

Termination: Once the maximum number of generations reaches a prespecified number, the algorithm is terminated.

4. NUMERICAL EXPERIMENTS

To compare the performance of each local search, numerical experiments are performed. Test instances are generated according to the following steps:

- (i) Number of jobs: 20 and 40.
- (ii) The processing time, weight, and release time follow uniform distribution. The parameters of the uniform distributions are reported in Table 1.

Table 1. Parameters of the Uniform Distributions to Generate Test Instances

Processing time	Weight	Release time
U (1,100)	U (1,10)	U (0, 50.5×n×R) where R equals 0.4

*R is a tardy factor that controls the release times to be tight or loose.

The parameters used in genetic local search algorithm including population size, maximum number of generations, crossover rate, and mutation rate are reported in Table 2.

Table 2. Parameters Used in Genetic Local Search Algorithm

Population size	50
Maximum number of generations	20000
Crossover rate	0.9
Mutation rate	0.1

For each test instance case, 10 test instances are generated and the average is used to be the result for each case. In Table 3, the results of solving 20 jobs are reported. On average, the $2r_i+t_i$ rule proposed by Liu and MacCathy (1991) outperform ERT and WSPT.

Table 3. ATWCT of Each Local Search (20 Jobs)

Local Search	$2r_i+t_i$	ERT	WSPT
ATWCT	13449.2	13551.1	13646.56

* ATWCT: Average Total Weighted Completion Time

In the 40 jobs instances, the performance is the same as Table 3. $2r_i+t_i$ rule is the most effective, ERT is worse, and WSPT is the worst.

Table 4. ATWCT of Each Local Search (40 Jobs)

Local Search	$2r_i+t_i$	ERT	WSPT
ATWCT	59299.66	60518.1	60847.1

* ATWCT: Average Total Weighted Completion Time

Two kinds of number of jobs are used to evaluate the performance of local search techniques. As shown in Figure 1, the result is consistent that $2r_i+t_i$ rule has the minimum ATWCT.

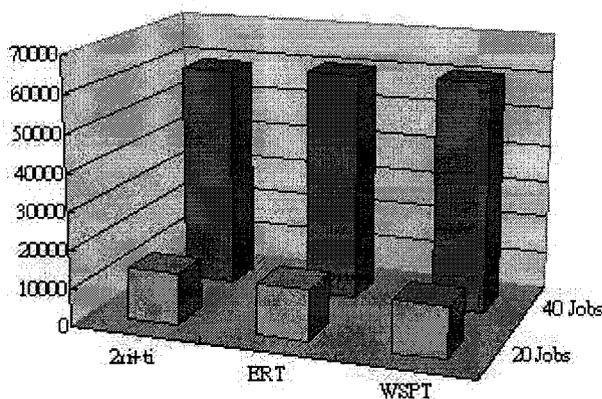


Figure 1. The ATWCT for Solving 20 and 40 Jobs

5. CONCLUSION

Single machine scheduling problems with release time are the prototypes for other complex scheduling systems. The results can be extended to many practical cases. However, this topic does not receive much attention. In this paper, genetic local search algorithms are proposed for this problem. Numerical experiments are done. The results show that the $2r_i+t_i$ rule proposed by Liu and MacCathy (1991) outperform ERT and WSPT.

This paper is a pilot study for single machine scheduling problems with release time. In the future, we may include more test instance cases to further verify the generalized performance of this idea. Moreover, we may extend this result to further extensive application areas.

6. REFERENCES

1. Baker, K., (1974), *Introduction to Sequencing and Scheduling*, New York: Wiley.
2. Chand, S., Traub, R., Uzsoy, R., (1996a), An iterative heuristic for the single machine dynamic total completion time scheduling problem, *Computers and Operations Research*, Vol. 23, pp.641-651.
3. Chand, S., Traub, R., Uzsoy, R., (1996b), Single-machine scheduling with dynamic arrivals: decomposition results and an improved algorithm, *Naval Research Logistics*, Vol. 43, pp.709-719.
4. Chang, P.C., Chung, Y.K., Hsieh, J.C., (2004), On single-machine scheduling with release times to minimize total weighted completion time, *Journal of the Chinese Institute of Industrial Engineers*, Vol. 21, pp.567-575.
5. Chu, C., (1992), A branch-and-bound algorithm to minimize total flow time with unequal release dates, *Naval Research Logistics*, Vol. 39, pp.859-875.
6. Deogun, J.S., (1983), On scheduling with ready times to minimize mean flow time, *Computer Journal*, Vol. 26, pp.320-328.
7. Dessouky, M.I., Deogun, J.S., (1981), Sequencing jobs with unequal ready times to minimize mean flow time, *SIAM Journal of Computing*, Vol. 10, pp.192-202.
8. Goldberg, D.E., (1989), *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Publishing Company, Inc.
9. Hariri, A.M.A., Potts, C.N., (1983), An algorithm for single machine sequencing with release dates to minimize total weighted completion time, *Discrete Applied Mathematics*, Vol. 5, pp.99-109.
10. Liu, J., MacCarthy, B.L., (1991), Effective heuristics for single machine sequencing problem with ready times, *International Journal of Production Research*, Vol. 29, pp.1521-1533.
11. Liu, J., Tang, L. (1999), A modified genetic algorithm for single machine scheduling, *Computers and Industrial Engineering*, Vol. 37, pp.43-46.
12. Madureira, A., Ramos, C., do Carmo Silva, S., (2001), A GA based scheduling system for dynamic single machine problem, *Proceedings of the 4th IEEE International Symposium on Assembly and Task Planning*, Soft Research Park, Fukuoka, Japan, May 28-29, pp.262-267.
13. Reeves, C., (1995), Heuristics for scheduling a single machine subject to unequal job release times, *European Journal of Operational Research*, Vol. 80, pp.397-403.
14. Rinnooy Kan, A.H.G., (1976), *Machine Scheduling Problem: Classification, Complexity, and Computations*, The Hague: Martinus Nijhoff.