



## META HEURISTIC APPROACH FOR EARLY-TARDY SINGLE MACHINE SCHEDULING PROBLEM

GUNERI, A[li] F[uat]; OEZGUERLER, S[enim] & PARLAK, M[uhammed]

**Abstract:** *The objective of this study is to develop algorithm for single machine early/tardy problems based on due dates. For this reason, greedy algorithm has developed. The working mechanism of greedy algorithm is mentioned in this study. Test problems are created for comparison of these algorithms performances. After operations, it is found that greedy algorithm improves cost for single machine early/tardy problems based on tight due date.*

**Key words:** *single machine, early, tardy, greedy algorithm*

### 1. INTRODUCTION

Nowadays, many just in time production systems, production facilities are organized in accordance with the contents. Just in time production scheduling strategies adopted by developing companies are trying to deliver to its customers orders on time. Thus, companies are finishing early or late finishing jobs to reduce costs arising from the direct and indirect, is working to eliminate. In this paper it is examined performances of heuristic methods greedy algorithm for the early/tardy single machine scheduling problem.

### 2. EARLY/TARDY SCHEDULING

Researches in single machine scheduling with earliness and tardiness penalties became one of the major areas of interest in scheduling theory (Ying, 2008), (Lin et al., 2007). Researchers have proved that Early / Tardy scheduling problem are NP-hard.

In the area of Early / Tardy scheduling, due date is the most important factor in the single machine problem. The researches of early/tardy Schedule problems has been classified types of due dates: common due dates common due window and distinct due dates.

Early/Tardy problem with common due date that considers a situation where several items represent a single customer's order or they might be part of a final product and all the parts should be ready at the same time for the final assembly to avoid undesired delay (Khader, 2008).

Early / Tardy problem with common due date, in many circumstances, jobs are some times associated with a period of time within which they incur no penalty since customers usually allow for a time interval within which products are delivered.

Early / Tardy problem with distinct due dates that considers  $n$  jobs ready at time zero and each job has its own distinct due date, earliness penalty and tardiness penalty with the objective to minimize the total penalties of earliness and tardiness.

### 3. EARLY/TARDY MODEL

Many manufacturing systems have developed scheduling policies with the objective of producing each customer order at the required time, due to direct and indirect costs incurred by completing jobs before or after their due dates. For example, inventory carrying costs, such as storage and insurance costs arise when jobs are finished before their due dates. Shortage

costs as for example, backorder and transportation expediting costs are incurred when jobs are completed after their due dates (Mazzini & Armentano, 2001).

### 3.1 The Model of Early Tardy Single Machine

Single machine early /tardy problem considers scheduling  $n$  jobs ready at time zero on a single machine with the objective to minimize the total penalties of earliness and tardiness. It is defined that  $n$  jobs to be processed on a single machine, each job  $j$  have a processing time  $P_j$  and a due date  $d_j$ .  $C_j$  is defined that the completion time for a job  $j$ .  $E_j$  represents earliness and  $T_j$  represents tardiness of job  $j$  respectively, where they defined by the following formulas:

$$E_j = \max(0, d_j - C_j) \quad \text{or} \quad E_j = (d_j - C_j)^+ \quad (1)$$

$$T_j = \max(0, C_j - d_j) \quad \text{or} \quad T_j = (C_j - d_j)^+ \quad (2)$$

With each of the objective functions, there exists a penalty:  $\alpha_j$  that it represents the earliness penalty and  $\beta_j$  that it represents the tardiness penalty. With the assumption that the penalty functions are linear, the Early/Tardy objective function can be written as follows:

$$f(s) = \sum_{j=1}^n \left[ \alpha_j (d_j - C_j)^+ + \beta_j (C_j - d_j)^+ \right] \quad (3)$$

or

$$f(s) = \sum_{j=1}^n \left[ \alpha_j E_j + \beta_j T_j \right] \quad (4)$$

### 3.4 Problem Assumptions and Limitations

The main standard assumptions in scheduling are: 1. Job cancellation is not allowed, 2. Preemption is not allowed, 3. Machines are always available, 4. Processing times are independent of the schedule, 5. Machines are able to process one job at a time, 6. All jobs are available for processing at time zero, 7. Each job is independent of the other jobs, 8. Job is started on a machine it must continue to completion, 9. Job processing times include all ancillary processing times such as set up, transportation or operator time, 10. Machine breakdown is not considered throughout the scheduling period, 11. Jobs have due dates and early/tardy penalties.

### 4. GREEDY ALGORITHM

The greedy method is a well known approach to problems involving the selection and/or ordering of elements is from a given set so as to optimize a given objective function (Alidaee et al., 2001). A greedy algorithm is a meta heuristic algorithm when it makes decisions for each step based on what seems best at the current step. Moreover, once a decision is made, it is never revoked. It may seem that this approach is rather limited. Nevertheless, many important problems have special features

that allow correct solutions using this approach. When applying this algorithm seems a logical decision for the moment, perhaps the result will be worse for future situations. Since it does not that revoke our greedy decisions, such algorithms tend to be simple and efficient (Khader, 2008).

**4.1 Greedy Algorithm Solutions**

We have given problem with n jobs. It is needed to apply the greedy algorithm to find the best possible sequence of these n jobs. So, we start initial sequence. We choice sequence SPT (shortest process time) and EDD (early due date) rules for observe the initial sequence performance for best sequence with minimizing cost. The greedy algorithm goes through interchanges of jobs in all pair of positions. If an interchange does not results an improvement in the performance, we reverse it back and continue the interchanges. If it results an improvement, it is kept the new position of the jobs and continuo the sequence of interchanges till the end.

**4.2 Greedy Algorithm Mechanism**

First, according to choice initial rule we start feasible sequence. We calculate cost according to cost objective function. The initial sequence is chosen best sequence and initial cost is chosen best cost at this step. After this step, we swap first job with second job. Cost function is calculated again. If after swapping we find better cost, the current solution is best solution. So, the current sequence is best sequence and current cost is best cost. If the current solution is not better than initial solution, we do not exchange solution. So, we update best sequence every all swapping process. After doing all the possible swaps the best sequence and the best cost are printed out.

**5. CREATING TEST PROBLEMS**

Test problems are created for a greedy algorithm. We used spt and edd rules for initial sequence. We are trying to reach the best possible sequence for minimizing objective cost function.

Four parameters must be identified for each job for test problems. They are the processing time ( $P_i$ ), work the due date, ( $d_i$ ), the cost of early completion ( $\alpha_i$ ), the cost of late completion ( $\beta_i$ ).

*Processing time:* Processing time is the time required for that job be processed. In test problems, we generate random processing times which are integers between 1 and 20.

*Due date:* The date the job is promised to the customer. We randomly generated due dates according to processing times in test problems. We mentioned before in test problems, we generate random processing times which are integers between 1 and 20. To generate a problem with n jobs, firstly, we start by generating n different processing times. Then we find minimum processing time and we sum all processing times of n jobs. After computing, we generate randomly n number between minimum processing times and sum of processing times. We generate due dates following formula.

$$Due\ date = relaxation\ factor * random\ (between\ min.\ processing\ times,\ sum\ of\ processing\ times)$$

In our test problems, 0,5 is a relaxation factor. It is accepted that problems have tight due dates.

*Earliness penalties:* is the penalty occurred when the job completes before its due date. In our problems, the earliness penalty is randomly generated integers between 1 and 10.

*Tardiness penalties:* is the penalty occurred when the job completes after its due date. In our problems, the earliness penalty is randomly generated integers between 1 and 10.

The sizes of our problems are 5 jobs, 10 jobs, 15 jobs, 20 jobs. Every single job has the four different parameters and they are randomly generated.

**6. RESULTS**

We use Microsoft Excel to solve the problems. In this study, the sizes of problems are 5, 10, 15 and 20 jobs. We swap double jobs in neighbor relation. The numbers of swaps are known: 10 swaps when n=5, 45 swaps when n=10, 105 swaps when n=15, 190 swaps when n=20. Table 1 shows a summary of the percentage improvement of the greedy algorithm. We analyze the performance of greedy algorithm performances for SPT rules and EDD rules.

It is concluded from the table that the greedy algorithm SPT gives better sequence than greedy algorithm EDD. The objective of cost function is decreased using swap greedy.

NUMBER OF JOBS	NATURAL SCHEDULE	GREEDY SPT SCHEDULE	PERCENTAGE IMPROVEMENT	GREEDY EDD SCHEDULE	PERCENTAGE IMPROVEMENT
5	187	69	%63.101	69	%63.101
10	1229	600	%51.117	665	%45.8980
15	1934	1150	%40.537	1183	%38.831
20	3574	1065	%70.201	1092	%69.445

Tab. 1. Summary of Results

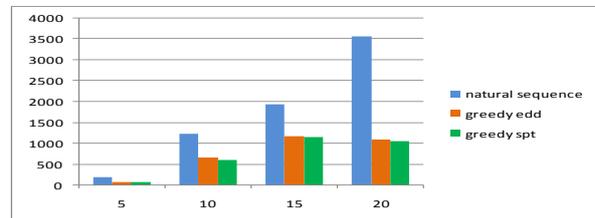


Fig. 1. Summary of Results

**7. CONCLUSION**

In this paper, it is considered that the problem of scheduling n jobs which are ready at time zero on a single machine with the objective of minimizing the total penalties of earliness and tardiness. It is addressed the minimum earliness–tardiness single machine scheduling problem with distinct due dates. 5, 10, 15 and 20 test problems are created and solved these problems sets by greedy algorithm. The developed algorithm will be for finding an optimal sequence of jobs. It is shown that for initial solution using SPT rules and then using greedy algorithm improve that minimizing cost function than initial solution using EDD rules and then using greedy algorithm. For further research other meta-heuristic techniques can be used and obtained results also can be compared with each other.

**8. REFERENCES**

Alidaee et al., (2001). Greedy Solutions of Selection and Ordering Problems. *European Journal of Operational Research*, No. 134, 203-215

Khader, M., (2008). A Solution Approach for The Single Machine Early/Tardy Scheduling Problems. MS. King FAHD University of Petroleum and Minerals, MAI 46/06, Dec 2008, p. 3421, 1456209

Lin et al., (2007). A Sequential Exchange Approach for Minimizing Earliness-Tardiness Penalties of Single-Machine Scheduling With a Common Due Date. *European Journal of Operational Research*, No. 177, 1294-1301

Mazini, R. & Armentano, V. (2001). A Heuristic for Single Machine Scheduling with Early and Tardy Costs. *European Journal of Operational Research*, No. 128, 129-146

Ying, K. (2008). Minimizing earliness–tardiness penalties for common due date single-machine scheduling problems by a recovering beam search algorithm. *Computers & Industrial Engineering*, No. 55, 494-502