PARETO-OPTIMAL SOLUTION OF A SCHEDULING PROBLEM ON A SINGLE MACHINE WITH PERIODIC MAINTENANCE AND NON-PRE-EMPTIVE JOBS

Bhaba R. Sarker[1], Junfang Yu[2*], Deniz Mungan[3], Mohammad A. A. Rahman[4], and Sultana Parveen[5]

[1,3,4]Department of Industrial Engineering, Louisiana State University, Baton Rouge, LA 70803, USA
[2]Department of Engineering Management, Information & Systems, Southern Methodist University Dallas, TX 75205, USA,
[5]Department of Industrial and Production Engineering, Bangladesh University of Engineering and Technology, Dhaka,
[1]bsarker@lsu.edu, [2]yuj@engr.smu.edu, [5]sparveen@ipe.buet.ac.bd

ABSTRACT

A Pareto-optimal solution is developed in this paper for a scheduling problem on a single machine with periodic maintenance and non-preemptive jobs. In literature, most of the scheduling problems address only one objective function; while in the real world, such problems are always associated with more than one objective. In this paper, both multi-objective functions and multi-maintenance periods are considered for a single machine scheduling problem. To avoid the complexities involved in solving an explicit multi-objective optimization problem, multiple objective functions are consolidated and combined into a single objective function after they are weighted and assigned proper weighting factors. On the other hand, periodic maintenance schedules are also considered in the model. The objective of the model addressed in this paper is to minimize the total job flow time, the maximum tardiness, and the machine idle time in a single machine environment. A heuristic algorithm is developed to solve this multiple criterion problem. The parametric analysis of the trade-offs of all solutions with all possible weighted combination of the criterions is performed. Results are provided to explore the best schedule among all the Pareto-optimality sets.

Keywords: Job scheduling, Multi-objective, Periodic maintenance, Non-preemptive job

1. INTRODUCTION

It is always challenging to solve multi-criterion scheduling problems due to the lack of suitable solution techniques. Therefore, such problems are usually transformed into a single-objective problem. A solution called *Pareto-optimal* if it is not possible to decrease the value of one objective without increasing the value of the other (Pinedo, 2002). The difficulty that arises with this approach is the rise of a set of Pareto-optimal solutions, instead of a single optimum solution.

Traditionally, the most common way to deal with a multi-objective problem is to consolidate the multiple criterions into a single scalar value by using weighted aggregating functions based on a set of predefined preferences and then to find a compromise solution that reflects these preferences (Deb, 2001). However, in many real cases involving multi-criterion scheduling problems, it is preferable to present a set of promising solutions to the decision-makers so that the most adequate schedule can be chosen. This is why there are increasing interests in investigating the application of Pareto-optimization techniques to multi-criterion scheduling problems. The focus of Pareto-optimization is to find a set of compromised solutions that represent a good approximation to the Pareto-optimality (Pinedo, 2002). In recent years, several Pareto-optimization related algorithms have been published (Kasprzak and Lewis, 2001, Gupta and Sivakumar, 2002). This is due to the existing of multi-objective optimization problems in many different domains.

In addition to the consideration of multiple objectives, periodic maintenance is also considered in this scheduling problem. An unexpected breakdown of production system will make the production behavior hard to predict, and thereby will reduce the system efficiency. Maintenance can reduce the breakdown rate with minor sacrifices in production (Liao and Chen, 2003).

A modified Pareto-optimality algorithm is developed in this paper for the multi-criterion scheduling problem with periodic maintenance. The algorithm is used to determine the trade-offs between total completion time and maximum lateness. The specific problem considered in this paper is to minimize the total flow time, the maximum lateness and the machine idle time. These multiple objectives are transformed into a single objective function, cost function, by aggregating them using a set of predefined weighting factors. In addition, various sets of Pareto-optimality sequences are generated against various maintenance plans, which contain various time intervals between two consecutive maintenances and various amount of time to perform the maintenance. Various maintenance plans give the decision-maker more flexibility in making his or her decision.

2. THE PROBLEM

We assume that there are *n* independent non-preemptive jobs, that is, once a job is started it must be completed to process them on a single machine without interruption. Each job *j* becomes available for processing at ready time zero and has a due date d_j . At every *T* unit of time, the machine is seized to hold for maintenance. A number of jobs that are grouped together to fit in every *T* amount of time is a *batch*. There could be machine idle time, I_b , before the maintenance starts after the last job in a batch is completed. The maintenance period is *M* which is also a fixed time. The total machine idle time is obtained by adding the idle time of all the batches.

The algorithm presented here for the problem combines all the criterions together in one schedule. The new approach starts with an initially obtained set of Pareto-optimal schedule for flow time and maximum tardiness minimization problem. It then includes machine idle time *I* and maintenance time *M* in each of these initially found sequences. Once the rescheduling of machine maintenance and idleness period of machine is completed, it then calculates the new values of flow time, maximum tardiness and machine idle time for which the assigned weights are w_1, w_2, and w_3, respectively.. All possible weight combinations, satisfying $w_1 + w_2 + w_3 = 1$, are assigned for criterions in each schedule of Pareto-optimality set. The minimum total cost among all the Pareto-optimal sequences according to certain weighted parameters gives the best sequence for the problem. It is clear that the problem is *NP*-hard since the problem that minimizes the maximum tardiness subject to periodic maintenance period and non-presumable jobs is *N*P-hard (Liao and Chen, 2003).

For two or more contradictory criterions, each criterion corresponds to a different optimal solution, but none of these trade-off solutions is optimal with respect to all criterions (Deb 2001). Thus, multi-criterion optimization does not try to find one optimal solution but a set of trade-off solutions. The fundamental difference is that multi-objective optimization deals with a set of Pareto-optimal solutions. The best schedule among the set that gives the most promising result for a particular set of weighted criterions is found.

Notations:

n	Number of jobs for processing at time zero
J_j	Job number *j, (j=1,2,…,n)*
p_j	Processing time of job *j*
p_{ji}	Processing time of job *j* in batch *i*
C_j	Completion time of job *j*
d_j	Due date of job *j*
L_j	Lateness of job *j*, where $L_j = C_j - d_j$
T_j	Tardiness of job *j*, where $T_j = max\{0; L_j\}$, $L_{max} = \max_j\{T_j\}$
T	Time interval between two maintenance periods
M	Amount of time to perform one maintenance
I_{bi}	Machine idle time in batch *i*, (*i*= 1, 2, …, *r*)
I	Total machine idle time of a schedule
T_{bi}	Total processing time for scheduled jobs in batch *i*, (*i*= 1, 2, …,*r*)
m	Iteration number.

3. THE SOLUTION ALGORITHM

When there are multiple objectives, the concept of Pareto-optimality plays a role in scheduling. A schedule is Pareto-optimal if it is impossible to improve on one of the objectives without making at least one other objective worse. The scheduler may want to view a set of Pareto-optimal schedules before deciding which schedule to select, when there are multiple objectives. In this paper, the algorithm of determining trade-offs between total completion time and maximum lateness (Pinedo, 2002) is modified by including periodic maintenance. In addition to the total completion time and the maximum lateness, the machine idle time is also considered as the third objective. A set of Pareto-optimal schedules represent the trade-offs between total completion time, maximum lateness and machine idle time.

There are many sequencing rules that can be applied to the jobs through the machines in a job shop according to the preferences. Two of those basic sequencing rules, shortest processing time (SPT) and earliest due date (EDD) are adapted in the modified Pareto-optimality algorithm. For explanatory convenience, we define two terms that are needed in the algorithm.

Definition 1.
The machine idle time of a batch, I_{bi}, is defined as the time by subtracting the total processing time for scheduled jobs in a batch, T_{bi}, from the time interval between two maintenance periods *T* (i.e., $I_{bi} = T - T_{bi}$).

Definition 2.
The total machine idle time of all machine in a schedule is defined as $I = \sum_{i=1}^{k} I_{bi}$ for all batches.

Pareto-optimality algorithm determines the trade-offs between total completion time and maximum lateness only (Pinedo, 2002). A third objective, machine idle time, is added and the stated algorithm is modified accordingly. The steps of the *modified* Pareto-optimality algorithm are outlined as follows:

Algorithm 1: *Modified Pareto-Optimality Algorithm*

```
Step 1: Set $m = 1$ (number of iteration)
  a) Schedule the jobs by SPT rule and apply EDD rule
     to the jobs with same processing time as schedule
     $S_{SPT/EDD}$
  b) Compute $L_{max}(SPT/EDD)$
```

```
    c) Go to Step 8 to find machine idle time in the
       schedule $S_{SPT/EDD}$ and the revised $S_{SPT/EDD}$ is now
       called $S^*_{SPT/EDD}$ when maintenance time is included.
Step 2: Set $m = 2$
    a) First schedule the jobs by EDD rule, and apply SPT
       rule to the jobs with same due date, as schedule
       $S_{EDD/SPT}$
    b) Compute $L_{max}(EDD/SPT)$
    c) Go to Step 8 to find machine idle time in the
       schedule $S_{EDD/SPT}$ and, on inclusion of maintenance
       time the revised $S_{EDD/SPT}$ is called $S^*_{EDD/SPT}$.
Step 3: Iteration $m = 3$.
Set $L_{\max} = L_{\max}(EDD)$ and $\bar{d}_j = d_j + L_{\max}$.
Step 4:
    Set $k = n$, $J^c = \{1,...,n\}$, $\tau = \sum_{j=1}^{n} p_j$ and $\delta = \tau$.
Step 5:
    Find $j^*$ in $J^c$ such that $\bar{d}_{j*} \ge \tau$, and $p_{j*} \ge p_l$ for
    all jobs $l$ in $J^c$ such that $\bar{d}_l \ge \tau$.
    Put job $j^*$ in position $k$ of the sequence.
Step 6:
    If there is no job $\ell$ such that $\bar{d}_\ell < \tau$ and $p_\ell > p_{j*}$,
    go to Step 7.
  Otherwise find j** such that $\tau - \bar{d}_{j**} = \min_{\ell}(\tau - \bar{d}_\ell)$
    for all $\ell$ such that $\bar{d}_\ell < \tau$ and $p_\ell > p_{j*}$,
        Set $\delta^{**} = \tau - d_{j**}$.
        If $\delta^{**} < \delta$, then $\delta = \delta^{**}$.
Step 7:
    Set $k \leftarrow k - 1$, and $\tau \leftarrow \tau - p_{j*}$.
    Update the set as $J^c = J^c - j^*$.
        If $k \ge 1$ go to Step 5.
Step 8:
    Generate a batch by grouping a set of jobs such that
    $\sum_{j=1}^{n} p_j \le T$
    Repeat grouping of the remaining jobs to form
    other batches.
    Set $b_i$ = number of batches in one schedule, where $i = 1, 2,\ldots,r$.
    Find machine idle time for one batch
        $I_{bi} = T - \sum_{j=1}^{n} p_{ji}$
    Find machine idle time for one schedule, $I = \sum_{i=1}^{r} I_{bi}$
    Revise the schedule by adding the amount of time to
    perform maintenance, $M$, to the end of each batch.
    Compute $\sum_{j=1}^{n} C_j^*$, $L_{\max}^*$, and $I^*$.
Step 9:
    Set $L_{\max} = L_{\max} + \delta$.
    If $L_{\max} \le L_{\max}(SPT/EDD)$
        set $m = m + 1$, $\bar{d}_j = \bar{d}_j + \delta$, and go to Step 4.
        Otherwise STOP.
```

In Step 1, the algorithm starts with sequencing the jobs in SPT order. If two jobs have the same processing time, the job with smaller due date is placed earlier. Then $L_{max}(SPT/EDD)$ is calculated for this generated SPT/EDD schedule. This $L_{max}(SPT/EDD)$ value indicate when to stop the iterations in the algorithm. For the schedule of SPT/EDD ($S_{SPT/EDD}$), batches are generated by grouping sets of jobs according to $\sum_{j=1}^{n} p_j \le T$.

The first Pareto-optimal schedule ($S^*_{SPT/EDD}$) is obtained after adding the maintenance time to the generated batches in $S_{SPT/EDD}$. The second Pareto-optimal schedule ($S^*_{EDD/SPT}$) is obtained in Step 2 which is similar to Step 1. The only difference is that instead of starting with SPT order, the procedure starts with EDD order and follows the same idea as in Step 1. In Step 3, due dates of the jobs are increased by $L_{max}(EDD/SPT)$ for the next iteration. Step 4 calculates the total processing time for n jobs and assigns that value to δ. Step 5 generates a Pareto-optimal schedule that minimizes $\sum_{j=1}^{n} C_j$ in which job k is scheduled last, if and only if

(i) $\bar{d}_k \ge \sum_{j=1}^{n} p_j$, and

(ii) $p_k \ge p_\ell$ for all jobs ℓ in J^c such that $\bar{d}_\ell \ge \sum_{j=1}^{n} p_j$.

Step 6 determines the minimum increment δ in the L_{max} that would allow for a decrease in the minimum $\sum_{j=1}^{n} C_j$ from the new generated Pareto-optimal schedule. Maintenance time is included to the generated Pareto-optimal schedule after forming the batches in Step 8. Three objective values ($\sum_{j=1}^{n} C_j^*, L_{\max}^*, I_m^*$) are also calculated at this point for all the Pareto-optimal schedules with periodic maintenance.

4. COMPUTATIONAL RESULTS

Consider a single-machine scheduling problem with nine jobs, as given in Table 1 (taken from Liao and Chen, 2003). The time interval between two consecutive maintenances, T, is 8 hours and the amount of time to perform one maintenance, M, is 2 hours. Now, all possible Pareto-optimal schedules are generated to determine average flow time of jobs, $\bar{F}^*$ (= $\sum_{j=1}^{9} C^*{}_j / 9$), job tardiness L_{max}^*, and machine idle time I^* for nine jobs.

Starting with Step 1, an optimal schedule $S_{(SPT/EDD)}$: <1-5-6-3-7-8-9-2-4 > is found by arranging jobs in SPT order and followed by $d_{[j]} \le d_{[k]}$ if $P_j = P_k$ and job j and k are adjacent (see Table 2). The maximum lateness, $L_{\max}(SPT/EDD)$, equals to 11 corresponding to job 2. Now Step 8 is applied to find the machine idle time after the insertion of maintenance.

*Table 1: The processing time and due dates for 9-job problem (in hour)**

Jobs	1	2	3	4	5	6	7	8	9
P_j	1	5	3	5	2	2	3	4	4
D_j	1	13	2	30	10	13	20	12	14

* Liao and Chen (2003).

Table 2: Pareto-optimal schedule, $S_{(SPT/EDD)}$

Jobs	1	5	6	3	7	8	9	2	4
P_j	1	2	2	3	3	4	4	5	5
d_j	1	10	13	2	20	12	14	13	30
C_j	1	3	5	8	11	15	19	24	29
L_j	-	-	-	6	-	3	5	11	-

Continue with the rest of the steps and iterations. Table 3 summarizes the results from all the iterations.

Table 3: All the iterations of the algorithm

Iteration m	Schedule	Pareto-Optimal Sequence	$\sum_{j=1}^{9} C_j^*, L_{\max}^*, I_m^*$	Current $d_j + \delta$	δ
1	SPT/EDD	<1-5-6-3-7-8-9-2-4>	151, 22, 8	30, 20, 14, 13, 13, 12, 10, 2, 1	-
2	EDD/SPT	<1-3-5-8-6-2-9-7-4>	182, 20, 8	30, 20, 14, 13, 13, 12, 10, 2, 1	7
3	S_1	<1-5-6-3-8-2-9-7-4>	172, 20, 8	37, 27, 21, 20, 20, 19, 17, 9, 8	1
4	S_2	<1-5-6-3-8-9-2-7-4>	137, 12, 0	38, 28, 22, 21, 21, 20, 18, 10, 9	2
5	S_3	<1-5-6-3-7-8-2-9-4>	151, 20, 8	40, 30, 24, 23, 23, 22, 20, 12,11	1
6	S_4	<1-5-6-3-7-8-9-2-4>	151, 22, 8	41, 31, 25, 24, 24, 23, 21, 13,12	Stop

Table 4: A Pareto-Optimality Set

Iteration, m	Schedule	Pareto-Optimal Sequence	$\sum_{j=1}^{9} C_j^*, L_{\max}^*, I_m^*$	$c(\overline{F}^*, L_{\max}^*, I^*)$
1	SPT/EDD	<1-5-6-3-7-8-9-2-4>	151, 22, 8	17.989
2	EDD/SPT	<1-3-5-8-6-2-9-7-4>	182, 20, 8	18.911
3	S_1	<1-5-6-3-8-2-9-7-4>	172, 20, 8	18.356
4	S_2	<1-5-6-3-8-9-2-7-4>	137, 12, 0	12.411*
5	S_3	<1-5-6-3-7-8-2-9-4>	151, 20, 8	17.189
6	S_4	<1-5-6-3-7-8-9-2-4>	151, 22, 8	17.989

* The best schedule with the minimum weighted function

After generating all possible Pareto-optimal schedules with respect to job completion time $\overline{F}^*$, job tardiness L_{max}^*, and machine idle time I^* for nine jobs, the total weighted function,

$$c(\overline{F}^*, L_{\max}^*, I^*) = w_1(\sum_{j=1}^{9} C_j^*/9) + w_2 L_{\max}^* + w_3 I^*$$

, where $w_1 = 0.5$, $w_2 = 0.4$, $w_3 = 0.1$ (arbitrarily chosen) for 6 schedules is calculated [see Table 4]. The minimum-weighted schedule is
$S^* = S_2$: <1-5-6-3-8-9-2-7-4>
corresponding to $c(\overline{F}^*, L_{\max}^*, I^*)$ = 12.411.

To compare the modified Pareto-optimality algorithm with the neighborhood search heuristic (Baker, 1998), the same instance with parameters $T = 8$, $M = 2$, $w_1 = 0.5$, $w_2 = 0.4$ and $w_3 = 0.1$ is used. The neighborhood search heuristic gives the best near-optimal schedule as S: <1-5-6-3-7-2-8-9-4> with the minimum weighted function equals to 13.100. On the other hand, the modified Pareto-optimality algorithm gives the best near-optimal schedule as S: <1-5-6-3-8-9-2-7-4> with the minimum weighted function equals to 12.410. It can be concluded that the modified Pareto-optimality algorithm provides a better result than the neighborhood search heuristic for this instance.

The same instance shown in the example is repeated for four levels of T (7, 8, 9, 10) and two levels of M (2, 3) as given in Liao and Chen (2002) to show the performance of various sets of Pareto-optimality schedules for different maintenance plans. Various maintenance plans give more flexibility to the scheduler (or decision-maker) to make a decision according to the preference and available maintenance alternatives. It can be concluded that the objective $c(\overline{F}^*, L_{\max}^*, I^*)$ is depended on T and M, and this shows the importance of a good maintenance plan. As the time to perform maintenance, M increases, both the completion time and the maximum lateness increases too. On the other hand, as the time interval between two maintenance periods, T changes, all three objectives are changing as well because of forming different batches.

5. CONCLUSION

A multi-criterion, non-preemptive, and periodically maintained single machine scheduling problem is studied in this paper. Three criterions are considered: reduction of flow time, maximum tardiness and machine idle time.

The trade-offs between the flow time and maximum tardiness is comparatively simple. Sometimes a single sequence (with no trade-off) or a set of Pareto-optimal sequences (with trade-off) minimizes the flow time and maximum tardiness, but the trade-off between minimum flow time, maximum tardiness and machine idle time is a complex problem. In this study a new kind of approach that allows the use of simple recombination of the criterions is presented. The new approach started with an initially obtained set of Pareto-optimal schedule for flow time and maximum tardiness minimization problem. It then introduces machine idle time and maintenance time in each of these initially found sequences. Once the rescheduling of machine maintenance and idleness period of machine is completed, it then calculates the new values of flow time, maximum tardiness and machine idle time. The search for the minimum total cost among all the Pareto-optimal schedules with the assigned weights on criterions is obtained. Finally, a promising sequence is chosen that gives the minimum total cost for a particular set of weights on the criterions.

The computational results have shown that the modified Pareto-optimality algorithm provides a better solution than the neighborhood search heuristic (Sarker *et al.*, 2007) and this shows the efficiency of the modified Pareto-optimality algorithm. Direct application of this study may be applied to the industries where performance of machine maintenance is a routine work and worthwhile as well. Chemical processing equipments, boilers, furnaces, mechanical machineries etc. are the examples of such implications.

6. REFERENCES

1. Baker, K. R. (1998), *Elements of Sequencing and Scheduling,* John Wiley, Inc., NY.

2. Gupta, A. K. and Sivakumar, A. I. (2002), “Simulation based multiobjective schedule optimization in semiconductor manufacturing,” *Proceedings of the 2002 Winter Simulation Conference*, Singapore, 1862-1870.

3. Kasprzak, E.M. and Lewis, K.E. (2001), “Pareto analysis in multiobjective optimization using the colinearity theorem and scaling method,” *Structural and Multidisciplinary Optimization,* **22**(3): 208-218.

4. Liao, C.J. and Chen, W.J. (2003), “Single-machine scheduling with periodic maintenance and nonresumable jobs,” *Computers & Operations Research*, **30**: 1335-1347.

5. Pinedo, M. (2002), *Scheduling, Theory, Algorithms, and Systems* (2nd edition), Prentice Hall, Inc., Upper Saddle River, New Jersey.

6. Sarker, B.R., Yu, J., Mungan, D. and Mohammad, A.A.R. (2007), “A Multiple Criterion Scheduling on a Single Machine with Periodic Maintenance and Non-preemptive Jobs,” *Working Paper*, Department of Industrial Engineering, Louisiana State University, Baton Rouge, LA 70803-6409.