MEDWIN PUBLISHERS
Committed to Create Value for Researchers

Advances in Robotic Technology
ISSN: 2997-6197

Utilizing Idle Time for Job Clusters Reduces Inventory and Total Job Completion Time for Single Machine Scheduling

Janaki E[1], Kathirvel A[2]*, Naren AK[3] and Ismail M[4]

[1]Department of Mathematics, Panimalar Engineering College, India
[2]Department of Computer Science and Engineering, Tagore Engineering College, India
[3]Department of AIDS, Anand Institute of Higher Technology, India
[4]Department of Mathematics, Sathyabama Institute of science and Technology, India

***Corresponding author:** A Kathirvel, Department of Computer Science and Engineering, Tagore Engineering College, India, Email: ayyakathir@gmail.com

Research Article
Volume 2 Issue 1
Received Date: August 06, 2024
Published Date: September 05, 2024
DOI: 10.23880/art-16000119

Abstract

The algorithm The algorithm must decide how much to increase or modify the completion times of each job because the completion times in the original schedule are the earliest potential completion times. Since each cluster's jobs are all shifted by the same amount, the goal of this research can be satisfied by computing the best shift for each cluster. By inserting idle time for job clusters, inventory and overall completion time for single machine scheduling are minimized. According to the results of the numerical experiment, if idle time is properly included before jobs, Earliness and Tardiness cost values can be greatly decreased. Finding the best schedule to minimize the fines for being early or late is the issue under consideration here. The article suggests a method for fitting idle time in while still meeting deadlines. The suggested algorithm reduces the penalty by inserting idle time up until the point where further reduction of the objective function is not possible.

Keywords: Clustering; Earliness and Tardiness; Inventory Cost; Idle Time; Single Machine

Introduction

Job that finishes ahead of schedule must be kept in inventory until its due date in a JIT scheduling system, but a job that finishes after its due date may interfere with a customer's activities. A perfect schedule consists of one in which every task is completed precisely on the scheduled due date. Scheduling methods with both earliness and tardiness (E/T) costs tackle the important scheduling aspect of the JIT approach. Producing fact, JIT involves a far wider set of concepts than those relevant to due dates. The creation of no delay programs has been the focus of the vast bulk of scheduling research. However, the insertion idle time (IIT) schedules are a class of schedules that we are interested in since they are more broadly speaking.

The use of JIT, or just-in-time delivery, which attempts to deliver the customer's order at the precise requested moment, has recently grown in importance. This is an earliness/tardiness (E/T) model in the scheduling sector [1]. Since it is usually preferable to include idle time in the schedule than not, several academics. Pinedo ML [2] have pointed out that the assumption of no idle time is incompatible with the E/T model. Feng G, et al. [3] investigated at the multi-machine scheduling problem with early and late charges and schedule based setup durations, and found that the problem may be divided into two sub problems: scheduling and timetabling. By using a lexicographical mixed-integer linear programming model and an improved simulated annealing technique, Cheng L, et al. [4] studied an Assembly job-shop scheduling problem with the aim of minimizing both a regular primary objective, the total completion time, and an irregular additional intended, the overall inventory time. Review the existing literature on inserted idle time scheduling after Kanet JJ, et al. [5] developed a classification of the situations in which inserted idle time scheduling is applicable. In order to reduce the overall completion time of all jobs, He N, et al. [6] and Tsai CY, et al. [7] assumed that the degradation of processing time and setup time is of a linear increasing function with regard to the starting time. To reduce the overall completion time and total energy cost under time-of-use electricity tariffs, Rubaiee S, et al. [8] suggested a preemptive planning issue on a single machine, which is a mixed-integer multiobjective mathematical programming model. A single-machine scheduling issue with weighted earliness and tardiness penalties is investigated by Chen JY, et al. [9]. Jobs may have several due dates and may have idle time in between them Babor M, et al. [10] looked at three production lines from small- and medium-sized bakeries to determine the ideal idle time for the makespan and ovens. reduction of the highest anticipated cost for stochastic processing time whereas five factors influence processing times. Mahnam M, et al. [11] tackled the issue of minimising the sum of maximum earlyness and tardiness on a single machine with unequal release times and idle insert based on realistic assumptions and real-world production settings. The existing research on the above scheduling problem is divided into three main groups by: Senthilkumar P & Narayanan S [12] offline scheduling, online scheduling, and other scheduling. The vertex cover problem and the single-machine scheduling problem were linked [13]. Specifically, for series-parallel precedence restrictions. To reduce the weighted sum of completion times, Mastrolilli M, et al. [14] investigated the single machine scheduling problem. By substituting a set of potential vectors, referred to as scenarios, for the numerical values in the description of the instance, we model uncertainty. In the worst-case situation, the schedule with the lowest value must be found. Yazdani M, et al. [15] created a mixed integer linear programming (MIP) formulation of the job shop scheduling issue using the new goal function, and he also researched The goal function's greatest earliness and tardiness requirements for the job shop scheduling issue. Seda M [16] suggests mathematical solutions to the problems of permutation flow shop scheduling and job shop scheduling

Inserted Idle Time

In this work, The creation of heuristics for creating inserted idle time schedules is crucial for the following reasons:

(i) Minimizing Inventory Problems

(ii) IIT scheduling issues are highly complex, making it unlikely that particular solutions will ever be used

(iii) In actuality, the problem definition is constantly changing due to the unpredictable nature of real-life scheduling conditions, making quick fixes absolutely necessary

Scheduling jobs

Scheduling issues are a typical occurrence in daily life. The majority of scheduling algorithms forbid keeping resources idle while jobs wait their turn. However, when early penalties are appreciably high, it is preferable to schedule jobs with idle time insertion. Delaying a task for a bit instead of starting it right away may make it easier to complete it by the deadline.

Single machine

We discuss the next circumstance. It is necessary to plan A set of jobs on a single machine that is always accessible from time zero onward. One job at a time is the most that the machine can manage. Job Ij (j = 1,..., n) should ideally be finished exactly on the due date dj and requires a positive integral uninterrupted processing time Pj. Each job's completion time is specified in a schedule so that the jobs don't run concurrently. The job sequence is the order in which the machine completes the jobs. Using a just-in-time (JIT) inventory system, suppliers can place orders for raw materials that are directly in line with production schedules. By purchasing only the things they actually need for the production process, businesses can cut down on inventory expenses while improving productivity and reducing loss. the fundamental single-machine model's effects of the E/T criterion. A scheduling target which costs a job according to the amount of time it takes to complete a job compared to the due date can be used to achieve the aim of finishing every job on time. Let E j and T j stand in for job j's punctuality and tardiness, respectively.

Notation

Pj- Processing time
Pb- Probability for Processsing time
EP- Expected Processing Time
Dj- Due date
Cj- Completion Time
 - Earliness Penalty
 - Tardiness Penalty, j= 1,2,3 L

$$\Lambda_j = \sum_{L=t}^{j} \alpha_L{}' - \sum_{L=j+1}^{l} \beta_L{}' \quad , \text{ j= 1,2,3 ......L}$$

$$\Lambda(r) = Max(\Lambda_j)$$

Proposed Algorithm

- **Step 1:** Find the expected Processing time for each job
- **Step 2:** The given job sequence j= 1, . . . , n can be decomposed into a set of m clusters
- **Step 3:** $\tau_{1,}$ $\tau_{2,......}$ τ_m with each cluster consisting job sequence

Step 4: Calculate Completion time for each cluster

Step 5: Find Earliness and Tardiness For Each Cluster and Select the minimum Earliness $\delta(r)$ among the Earliness and select $\Lambda(r) = Max(\Lambda_j)$

Step 6: Identify the least such that $\sum_{r=1}^{s} \Lambda(r) \le 0$

Step 7: For first 's' clusters the completion time remains same .If this s is equal to number of jobs then stop the iteration ; else, move on to Step 8.

Step 8: If there are none, move on to Step 10.

Step 9: Eliminate the list's first s clusters. Re index all active jobs and clusters.

Step 10: Consider the collection of remaining clusters by moving on to Step5.

Step 11: Compute Minimum Ej ,j= 1,2,3…. n. All Cn should be increased by min(E(1),…, E(n)).

Step 12: Remove all jobs that were early but are now late.

Step 13: Refresh E(r) and (r).

Step 14: Step 5 is next.

Problem Description

Consider 10 jobs associated with probability and assigned with earliness and tardiness penalty . A block is described as a series of clusters that are continuously processed.Think about the block . One or more clusters may come before such a block, but none may come after it. Let jr represent the final early task in cluster ór, or the early job with the smallest earliness (Tables 1-3).

Job	1	2	3	4	5	6	7	8	9	10
Pj	20	60	40	40	70	60	40	40	50	50
Pb	0.2	0.05	0.2	0.1	0.1	0.05	0.05	0.05	0.1	0.1
Dj	10	10	14	20	18	30	10	15	35	20
α'_j	5	10	5	5	10	30	5	10	5	5
β'_j	10	10	10	12	12	10	5	2	10	10

Table 1: Processing time with probability.

Position	1	2	3	4	5	6	7	8	9	10
job	1	2	3	4	5	6	7	8	9	10
EPj	4	3	8	4	7	3	2	2	5	5
Dj	10	10	14	20	18	30	10	15	35	20
Cj	4	7	15	19	26	29	31	33	38	43
E	6	3	0	1	0	1	0	0	0	0
T	0	0	1	0	8	0	21	18	3	23

Table 2: Earliness and Tardiness for each job.

	τ_1			τ_2		τ_3			τ_4	
J	1	2	3	4	5	6	7	8	9	10
Cj	4	7	15	19	26	29	31	33	38	43
Ej	6	3	0	1	0	1	0	0	0	0
$\delta(r)$	3			1		1			inf	
Λ_j	-15	5	0	-7	0	23	0	0	-20	
$\Lambda(r)$		5		-7		23			-20	

Table 3: Clustering a given job.

Results and Discussion

The algorithm must decide how much to extend each job's completion time because the completion times in the original schedule are the earliest feasible times. In reality, since each cluster's workloads are all shifted by the same amount, computing the best shift for each cluster is adequate. The given job sequence j= 1. . . n can be decomposed into a set of m clusters.With each cluster consisting job sequence. A block is described as a series of clusters that are continuously processed.

In Table 4, completion time remains same for first two clusters. After removing first two cluster. Apply step (v), and hence by step (x), Increase all completion time in the 3rd and 4th cluster by one time unit.

Table 5 shows the fixed current completion time for above jobs. Hence after inserting the idle time the completion time of given jobs 4,7,15,19,26,30,32,34,39,44.

Cluster	τ_1	τ_2	τ_3	τ_4
$\delta(r)$	3	1	1	inf
$\Lambda(r)$	5	**-7**	23	-20

Table 4: Completion time.

	τ_3			τ_4	
J	6	7	8	9	10
Cj	30	32	34	39	44
Dj	30	10	15	35	20
Ej	0	-ve	-ve	-ve	-ve
$\delta(r)$	inf		inf		
$\Lambda(r)$	-17		-20		

Table 5: Inserting idle Time.

Conclusion

Our clustering experiments have demonstrated the efficacy of our scheduling approach, particularly in situations with predictable conditions. The approach excels in reducing material usage and time required for single machine scheduling by accounting for idle time within work clusters. Moreover, our extension of this research to stochastic settings showcases its adaptability to more uncertain scenarios. Overall, our work highlights the potential of our approach to optimize scheduling challenges across various contexts, offering improved efficiency and resource utilization.

Acknowledgment

Data for this article was gathered from Geebon Small Scale Industry in Chennai.

Conflict of Interest

There is no conflict of interest.

Funding

This research did not receive any financial support.

References

1. Baker KR, Trietsch D (1943) Principles of Sequencing and Scheduling. John Wiley & Sons, Hoboken, New Jersey, Canada, USA, pp: 1-483.
2. Pinedo ML (2008) Scheduling: Theory, algorithms, and systems. 3rd (Edn.), Springer, New York City, pp: 698.
3. Feng G, Lau HC (2008) Efficient algorithms for machine scheduling problems with earliness and tardiness penalties. Annals of Operations Research 159(1): 83-95.
4. Cheng L, Tang Q, Zhang L, Li Z (2022) Inventory and total completion time minimization for assembly job-shop scheduling considering material integrity and assembly sequential constraint. Journal of Manufacturing Systems 65: 660-672.
5. Kanet JJ, Sridharan V (2000) Scheduling with inserted idle time: problem taxonomy and literature review. Operations Research 48(1): 99-110.
6. He N, Qiao Y, Wu N, Qu T (2017) Total completion time minimization for scheduling of two-machine flow shop with deterioration jobs and setup time. Advances in Mechanical Engineering 9(4): 1-12.
7. Tsai C-Y, Wang Y-C (2014) A Mixed Integer Linear Programming Solution for Insertion of Idle Time in the Jobs Scheduling. Proceedings of the 2nd International Conference on Soft Computing in Information Communication Technology 71: 53-56.
8. Rubaiee S, Yildirim MB (2019) An energy-aware multiobjective ant colony algorithm to minimize total completion time and energy cost on a single-machine preemptive scheduling. Computers and Industrial Engineering 127: 240-252.
9. Chen JY, Lin SF (2002) Minimizing weighted earliness and tardiness penalties in single-machine scheduling with idle time permitted. Naval Research Logistics

49(8): 760-780.

10. Babor M, Paquet-Durand O, Kohlus R, Hitzmann B (2023) Modeling and optimization of bakery production scheduling to minimize makespan and oven idle time. Scientific Reports 13: 235.

11. Mahnam M, Moslehi G, Ghomi SMTF (2013) Single machine scheduling with unequal release times and idle insert for minimizing the sum of maximum earliness and tardiness. Mathematical and Computer Modelling 57(9-10): 2549-2563.

12. Senthilkumar P, Narayanan S (2010) Literature Review of Single Machine Scheduling Problem with Uniform Parallel Machines. Intelligent Information Management 2(8): 457-474.

13. Correa JR, Schulz AS (2005) Single-machine scheduling with precedence constraints. Mathematics of Operations Researchm 30(4): 1005-1021.

14. Mastrolilli M, Mutsanas N, Svensson O (2013) Single machine scheduling with scenarios. Theoretical Computer Science 477: 57-66.

15. Yazdani M, Aleti A, Khalili SM, Jolai F (2017) Optimizing the sum of maximum earliness and tardiness of the job shop scheduling problem. Computers and Industrial Engineering 107: 12-24.

16. Seda M (2007) Mathematical Models of Flow Shop and Job Shop Scheduling Problems. International Journal of Physical and Mathematical Sciences 1(7): 307-312.