



Journal Homepage: - www.journalijar.com
**INTERNATIONAL JOURNAL OF
 ADVANCED RESEARCH (IJAR)**

Article DOI:10.21474/IJAR01/4871
 DOI URL: <http://dx.doi.org/10.21474/IJAR01/4871>



RESEARCH ARTICLE

SOLVING OSSP WITH RELEASE DATES TO MINIMIZING THE MAKESPAN AND RESOURCE IDLENESS – A HYPOTHETICAL CASE.

Dr. S. Jayakumar¹ and R. Meganathan².

1. Assistant Professor & Head, PG & Research Department of Mathematics, AA Govt. Arts College, Cheyyar, Tamil Nadu- 604 407.
2. Assistant Professor, PG & Research Department of Mathematics, Shanmuga Industries Arts & Science College, Tiruvannamalai, Tamil Nadu – 606 601.

Manuscript Info

Manuscript History

Received: 17 May 2017
 Final Accepted: 19 June 2017
 Published: July 2017

Key words:-

Heuristic, OSSP, Makespan, Resource idleness, hypothetical case.

Abstract

Scheduling has become one of the major fields within Operations Research with several hundred papers published in every year. In this paper we are considering the open shop scheduling problem with release dates for the objective of minimizing makespan and resource idleness of machines for the hypothetical situation when some jobs need not processed on some machines. Here pre-emption is not allowed. In 2014, we developed an algorithm for the OSSP problem with release dates to minimize makespan and now we tested whether the same algorithm performs well for the objective of resource idleness of the machines for the hypothetical case also. It was found that the proposed algorithm performs better than the existing algorithm with respect to both the objectives.

Copy Right, IJAR, 2017,. All rights reserved.

Introduction:-

Scheduling has its origin in manufacturing industries. Scheduling is a decision making process which is used on regular basis in many manufacturing industries. Shop scheduling problems are widely used in the modelling of industrial production process and are receiving an increasing amount of attention from researchers. Scheduling is the allocation of resources over a period of time to perform a collection of tasks. Scheduling problem exists almost everywhere in real industrial world situations. In the theory of scheduling, there are three basic types, namely Flow-shop, Job-shop and Open-shop scheduling problems. If the jobs have different process sequence, the problem is known as Job-shop scheduling problem. If the jobs have same process sequence, the problem is known as Flow-shop scheduling problem. In Open shop scheduling problem (OSSP), jobs have no predetermined processing sequence. i.e., jobs can be processed in any conceivable order. The OSSP is similar to the job shop scheduling problem with the exemption that there are no precedence relations between the operations of each job. The OSSP has considerably larger solution space than the other scheduling problems (Flow-shop & Job-shop) and seems to receive less attention in the literature, although it is an important and universal problem.

An open shop model finds numerous applications in the real world situations. For example, the timetable problem is the generalisation of the OSSP. The professors are the machines, the jobs are the classes and the objective is to find times at which the professors can instruct their classes without any professor teaching more than one class at a time and any class meeting with more than one professor at a time.

Corresponding Author:- R. Meganathan.

Address:-Assistant Professor, PG & Research Department of Mathematics, Shanmuga Industries Arts & Science College, Tiruvannamalai, Tamil Nadu – 606 601.

A schedule is preemptive, if the execution of any operation may arbitrarily often be interrupted and resumed at a later time. Otherwise it is known as non-preemptive schedule. i.e., each operation is executed continuously from start to completion without interrupted. In this paper, we consider a hypothetical case open shop scheduling problem with release dates for the objective of minimizing the makespan and resource idleness of machines, when pre-emption not allowed. The rest of the paper is organized as follows: In section 2, we gave the statement of the problem; section 3 is devoted for literature review; section 4 details objective and the scope of the problem. In section 5, we gave our methodology for solving the proposed problem followed by an example in section 6; results and discussion was given in section 7.

Statement of the problem:-

In an OSSP, a set of n jobs J_1, J_2, \dots, J_n has to be processed on a set of m machines M_1, M_2, \dots, M_m . Every job consists of m operations each of which must be processed on a different machine for a given processing time. The operations of each job can be processed in any order. At any time one operation can be processed on each machine, and almost one operation of each job can be processed. There is no precedence relation between the operations. All machines are continuously available. The process of a job cannot be interrupted. There are infinite buffer between machines (i.e., a job needs a machine that is occupied it can wait indefinitely until the machine becomes idle again). There are no transportation times between machines. It is assumed that the processing times of all operations are assumed to be given in advance. $O(i, j)$ denotes operation of job j on machine i . The processing time of job j on machine i , $i = 1, 2, \dots, m$ is denoted by $p(i, j)$. It is assumed that the processing times are bounded by P_{max} and *i. i. d.* (independently and identically distributed) random variables. $R(i, j)$ is the starting time of operation $O(i, j)$ and the completion time of job j on machine i is denoted by $C(i, j)$. For each job J_i , there may be given a release date $r_j \geq 0$ which is the earliest possible time when the first operation of this job may start. The maximum completion time of all the jobs is known as makespan of the schedule and is denoted as C_{max} . In addition to the above assumptions we include one more assumption that some of the jobs may not require certain machines for processing. This may occur in many industrial situations which are the beauty of the applications of open shop scheduling problems. An OSSP with this assumption is called the hypothetical case problems. In this paper we consider this problem and our objective is to find a sequence of jobs with the given processing times on each machine to minimize the makespan and idleness of machines as well.

Literature Review:-

In the literature of OSSP, most of the attention has been paid to the minimization of makespan without considering release dates or due dates. Most of the researchers focus on the problem with the assumption that all jobs are available at time zero, whereas we consider the release dates also for our problem.

Graham R L (et. al., 1979) provided the standard notation for scheduling problem. With that the general OSSP problem can be described as $Om//C_{max}$, where m is the number of machines. Pinedo M (2002) presented a priority rule, Longest Alternate processing time first (LAPT) for $2//C_{max}$, with which the optimal schedule can be found in polynomial time. For the case $m = 2$ or $n = 2$, a polynomial time algorithms is provided by Gonzalez T & Sahni S (1976), to the OSSP with arbitrary number of jobs and machines and preemption allowed. Also they proved that the problem $Om/2/C_{max}$ is strongly NP-hard.

Lawler E L (et. al., 1993) proved that the problem $Om/2/C_{max}$, is strongly NP-hard, which means that the optimal solution of the problem cannot be obtained in polynomial time. For small scale problem, branch and bound algorithms are used to solve it [Brucker P (et. al., 1997), Dorndorf U (et. al., 2001)]. For large scale problems, constructing heuristic algorithms is an effective way to obtain the approximately optimal solution. Jayakumar S (2000) focus attention in OSSP with the objective of minimizing the makespan and resource idleness and invented that in OSSP, a heuristic approach with longest processing time (LPT) perform better than the shortest processing time (SPT).

A feasible schedule for the open shop problem is called dense when any machine is idle if and only if there is no job which currently could be processed on that machine. This concept was introduced by Racsmany (cf Barany I and Fiala T 1982) and it has been shown that the makespan of any dense schedule is almost twice the optimum makespan. This result can also be derived as a corollary from a more general result by Aksjonov V A (1988). Blazewicz J (et. al., 2004) considered OSSP with a common due date where the goal is to minimize total weighted late work.

Chen B and Strusevich V A (1993) conjectured that for every $m \geq 2$, dense schedules are almost a factor of $2 - \frac{1}{m}$ away from the optimum, and they proved this conjecture for $m = 3$. Strusevich V A (1998) proved that when jobs are pre-ordered, the DS is improved and the new algorithm is almost $2 - \frac{1}{m+1}$ times of the optimal solution for problem $Om/r_j/C_{max}$. It is also noted that there is no polynomial time approximation algorithm with the worst-case performance ratio strictly less than $\frac{5}{4}$ unless $P = NP$ by Williamson D P (et. al., 1997).

The jobs are available only after its arrival in real world problems. The OSSP which seeks minimizing makespan occurs with release dates. If r_j be the release date for job j , then the problem can be described as $Om/r_j/C_{max}$. Lawler E L, Lenstra J K and Rinnooykan A H G (1981) pointed out that the problem $O2/r_j/C_{max}$ is strongly NP-hard. Chen R (2003) proved that the worst-case performance ratio of DS is $\frac{3}{2}$ for the problem $O2/r_j/C_{max}$ and conjectured that the ratio is bounded by $2 - \frac{1}{m}$ where m is arbitrary. Chen R, Huang W and Tang G (2008) proved that the worst-case performance ratio of DS is bounded by 2 for problem $O3/r_j/C_{max}$ and showed that the ratio can reach $\frac{5}{3}$ for some special case. A survey of algorithms of on-line scheduling problems was provided by Sgall J (1998). Chen B (et. al., 1998) generalized the DS to schedule the on-line version of the problem $Om/r_j/C_{max}$ and proved that the conjecture that the worst competitive ratio of DS for the two-machine case is bounded by $2 - \frac{1}{m}$. The basic terminologies of scheduling theory can be found in (French s, 1982). In 2016, we considered the hypothetical case OSSP with release dates for the objective of makespan minimization. In this paper, we focus the hypothetical case problems with multiple objectives which is very much interesting in nature and has wide applications to the real world industrial situations.

Objective and scope of the problem:-

In the modern real world problems it may happen that some jobs may not be required to process in all the available machines. In that situations also we need to find a feasible sequence that optimize our objective or objectives. Complexity of solving the OSSP problems is high if we add more constraints to the basic assumptions and expect to optimize multiple objectives. In this paper we include the hypothetical situation to the OSSP with release dates and tested our DLPT – DS (Dynamic longest processing time-Dense schedule) algorithm for the multiple objectives minimizing the makespan and reducing the resource idle time of machines.

Definition (Chen R, 2008) An idle interval $[b, e)$ on machine $i, i = 1, 2, \dots, m$ for a given schedule S is called reasonable if one of the following conditions holds for job $j, j = 1, 2, \dots, n$,

- (1) Job j has been finished on machine i before time b i.e., $C(i, j) \leq b$; or
- (2) Job j is being processed on a machine other than i at any time t in $[b, e)$,

$$\text{i.e., } [b, e) \subseteq \bigcup_{i' \neq i} [R(i', j), C(i', j));$$

- (3) Job j released after time e , i.e., $r_j \geq e$.

A schedule is dense if all its idle intervals are reasonable. It is supposed that any idle interval does not traverse any release date, i.e., if there is an idle interval $[b, e)$ in which there is a release date $b < r_j < e$, then we denote $[b, e)$ by two idle intervals $[b, r_j)$ and $[r_j, e)$.

If the jobs are indexed according to their arriving sequence, i.e., $r_1 \leq r_2 \leq \dots \leq r_n$, then the following lower bound for problem $Om/r_j/C_{max}$ can be easily obtained by observation.

$$C_{LB} = \text{Max} \left\{ \max_{1 \leq j \leq n, 1 \leq i \leq m} \left\{ r_j + \sum_{g=j}^n p(i, g) \right\}, \max_{1 \leq j \leq n} \left\{ r_j + \sum_{i=1}^m p(i, j) \right\} \right\}$$

Theorem 1 (Bai D and Tang L, 2013) Let release date r_j be nonnegative random variables, $j = 1, 2, \dots, n$, and the processing time $p(i, j)$ of job $j, j = 1, 2, \dots, n, i = 1, 2, \dots, m$, be independent random variables and have the same continuous distribution with nonzero bounded density $\varphi(\cdot)$. Then, for a series of randomly generated instances of problem $m/r_j/C_{max}$, with probability one, we have

$\lim_{n \rightarrow \infty} \frac{C_{LB}}{n} = \lim_{n \rightarrow \infty} \frac{C_{max}(S^*)}{n} = \lim_{n \rightarrow \infty} \frac{C_{max}(DS)}{n}$, where $C_{max}(S^*)$ and $C_{max}(DS)$ denote the objective values obtained by the optimal schedule and the DS, respectively.

Theorem 2 (Bai D and Tang L, 2013) The sequence of operations in a DS does not influence the asymptotic optimality.

Methodology:-

A heuristic algorithm called DLPT – DS was developed by the authors (Jayakumar S and Meganathan R, 2014) for solving the OSSP with the consideration of release dates to minimize the makespan. We further tested the algorithm developed for two objectives (Minimizing the makespan and resources idleness for OSSP with release dates, accepted for publication in International journal of pure and applied mathematical sciences, 2017), where the second objective is reducing the resource idleness. In this paper we test our algorithm for the hypothetical case problems. The computational results were compared to the algorithm available in the literature called DSPT – DS (Bai D and Tang L, 2013). The proposed heuristic is described as follows.

DLPT-DS Heuristic (Jayakumar S and Meganathan R, 2014)

Let $B = O(i, j)_{m \times j}$, $1 \leq j \leq n$, denote the operations that are available at time t , $t \geq 0$ and $R(i, j)$ be the starting time of operation $O(i, j)$.

Step 1: At time t , $t \geq 0$ process the operation with the longest processing time, say $O(i_1, j_1)$ among all the available ones in matrix B. If some operations simultaneously satisfy the condition, give preference to the operation with smallest $O(i_1, j_1)$ index. Update the starting times of the operations, which are at the same column and row with $O(i_1, j_1)$, to $t + p(i_1, j_1)$ in matrix B. Remove operation from matrix B.

Step 2: If some jobs arrive, go to step3; if matrix B becomes empty, go to step4.

Step 3: Sort the operations of the arrivals into matrix B, and update the starting time of each new operation to the longest starting time of its row in matrix B. Then go to Step1.

Step 4: Let the machines remain idle until a job arrives, and go to step 3 .If the scheduling is completed, terminate the program.

Example:-

We illustrate our algorithm by considering the problem of scheduling four jobs on four machines. The processing times of job $J_j, j = 1, 2, 3, 4$ on machine $M_i, i = 1, 2, 3, 4$ and the release dates r_j are given below.

	J_1	J_2	J_3	J_4
M_1	2	5	3	-
M_2	4	3	1	5
M_3	2	1	7	8
M_4	3	2	1	4
r_j	3	2	1	4

If one use to schedule the operations by using DSPT-DS algorithm, we obtain the makespan value as 23 units of time and idle time for machine 1, machine 2, machine 3, and machine 4 as 5, 10, 2 and 4 units of time respectively, totally 21 units (see figure 6.1.1). Where as if we schedule the operations by using DLPT-DS algorithm, we obtain the makespan value as 22 units of time and idle time for machine 1, machine 2, machine 3, and machine 4 as 1, 9, 2 and 5 units of time respectively, totally 17 units (see figure 6.1.2).

Result and Discussion:-

In general, an algorithm which reduces the makespan value will always reduce the idle time of machines also. But sometimes it may not be true for the case of OSSP. i.e., an algorithm which optimizes the makespan criteria does not provide any guarantee for the reduction of the idle time. After solving the problem, it was found that our proposed DLPT – DS algorithm performs well than the algorithm available in the literature (Bai D and Tang L, 2013) for both the objectives. The former algorithm reduces not only makespan criteria but also reduces the idle times of the machines. So for solving OSSP with release dates with the objective of minimizing makespan and resource idleness, one may choose our algorithm for better results.

13. Gonzalez T & Sahni S (1976). Open shop scheduling to minimize finish times. *Journal of association for Computer science*, 23, 665-679.
14. Graham R L, Lawler E L, Lenstra J K & Rinnooykan A H G (1979). Optimization and approximation in deterministic machine scheduling; a survey. *Annals of Discrete Mathematics*, 5, 287-326.
15. Jayakumar S (2000). A heuristic approach for solving open shop scheduling problems, minimizing makespan and resource idleness. Ph.D., thesis, Anna University, Chennai, Tamil Nadu.
16. Jayakumar S & Meganathan R (2014). Heuristic approach for OSSP to minimize makespan with release dates. *Industrial Engineering Journal*, 7(8), 17-22.
17. Lawler E L, Lenstra J K & Rinnooykan A H G (1981). Minimizing maximum lateness in a two-machine open shop. *Mathematics of Operations Research*, 6, 153-158.
18. Lawler E L, Lenstra J K, Rinnooykan A H G & Shmoys D B (1993). Sequencing and Scheduling: algorithms and complexity, in: Graves S C, Rinnooykan A H G, Zipkin P H (Eds.), *Handbook in Operations Research and Management Science, Logistics of Production and Inventory*, North-Holland, Amsterdam, 4, 445-522.
19. Pinedo M (2002). *Scheduling: Theory, Algorithms and Systems*. second ed., Prentice-Hall, New Jersey.
20. Sgall J (1998). On-line scheduling, in: Fiat A, Woeginger G J (Eds.) (1998), *Online algorithms: The state of the Art. Lecture Notes in Computer Science*, Springer, Berlin, 1442, 196-231.
21. Sevastianov S V & Woeginger G J (1998). Makespan minimization in open shops: a polynomial time approximation scheme. *Mathematical programming*, 82, 191-198.
22. Strusevich V A (1998). A greedy open shop heuristic with job priorities. *Annals of Operations Research*, 83, 253-270.
23. Williamson D P, Hall L A, Hoogeveen H A, Hurkens C A J, Lenstra J K, Sevast'janov S V & Shmoys D B (1997). Short shop schedules. *Operations Research*, 45, 288-294.