Journal of Mathematics and Statistics 6 (4): 416-420, 2010
ISSN 1549-3644

Scheduling Algorithm to Optimize Jobs in Shop Floor

[1]T. Hemamalini, [2]A.N. Senthilvel and [3]S. Somasundaram
[1]Department of Mathematics, Tamilnadu College of Engineering, Karumathampatty, 641659, Tamil Nadu, India
[2]Department of Computer Science and Engineering,
[3]Department of Mathematics,
Coimbatore Institute of Technology, Coimbatore, 641 014, Tamil Nadu, India

Abstract: Problem statement: The ratio scheduling algorithm to solve the allocation of jobs in the shop floor was proposed. The problem was to find an optimal schedule so as to minimize the maximum completion time, the sum of distinct earliness and tardiness penalties from a given common due date d. **Approach:** The objective of the proposed algorithm was to reduce the early penalty and the late penalty and to increase the overall profit of the organization. The proposed method was discussed with different possible instances. **Results:** The test results showed that the algorithm was robust and simple and can be applied for any job size problem. **Conclusion:** The proposed algorithm gave encouraging result for the bench mark instances when the due date is less than half of the total processing time.

Key words: Ratio scheduling, job shop scheduling, Np-hardness, penalty, restrictive problem

INTRODUCTION

In the Job Shop Scheduling Problem (JSP), a finite set of jobs has to be processed for a specified duration on a single machine around a common due date. The machine can process at most one job at a time and no preemption was allowed. Scheduling decision should be made according to a certain measure of performance, or scheduling criterion which is elaborated by Sule (2007). Industries job shop scheduling is a complex phenomenon to be solved with novel computational methods. Early finished jobs can lead to inventory loss and finishing jobs late lead to customer dissatisfaction. In general, a constructive optimization method tries to give the best possible solution. On the other hand, an iterative process improvises an assumed initial solution, which may take a long time, since we do not know which would give the optimal solution in a stipulated time frame. Hence we used the constructive method of optimization, barring the time taken to get the solution. The JSP is NP-hard discussed by Lenstra and Rinnooy Kan (1979) has continuously challenged to computational researchers. As the due date approaches processing speed of the jobs cannot be altered as discussed like in the Bender *et al*. (2007). Hong *et al.* (2007) developed LPT and PT algorithms for flexible flow-shop problem. Two-stage scheduling problem which minimizes total completion time was discussed by Yang and Lin (2009). A scheduling algorithm to solve sub models for complex scheduling problem was developed by Ghoul *et al.* (2007). Ismail and Loh (2009) developed ASO to minimize operational cost of an industry.

A set of n independent jobs has to be scheduled in a single machine, which can handle one job at a time. Assuming that there is no preemption of jobs and the machine is available from time t = 0 onwards. Let Jobs J_i(i = 1,2,…n) having processing time p_i, earliness penalty α_i and tardiness penalty β_i are non symmetric. Every job has the Common Due Date d.

If S being the optimal schedule then the objective is to minimize:

$$F(S) = F(E_S)+F(L_S)$$

Where:

$$F(E_s) = \sum_{i=1}^{m}\{(d - \sum_{j=1}^{i} p_j)\alpha_i\}$$

$$F(L_s) = \sum_{i=m+1}^{n}\{(\sum_{j=1}^{i} p_j - d)\beta_i\}$$

The Common Due Date d and the ratio between early and late penalties are the decision variables Cheng and Gupta (1989) and Dileepan (1993). The problem is a restricted problem Feldmann and Biskup (2003)

Corresponding Author: T. Hemamalini, Department of Mathematics, Tamilnadu College of Engineering, Karumathampatty, 641659, Tamil Nadu, India

studied the restricted Earliness and Tardiness problem in which $d < \sum_{i=1}^{n} p_i$. Hoogeveen and van de Velde (1991) discussed the earliness and tardiness penalties are taken as symmetric $\alpha_i = \beta_i$ for all jobs. Bagchi *et al.* (1987) discussed non symmetric case with all α_i are equal and β_i are equal. They developed an algorithm which takes O(n log n) time to schedule the non symmetric case. Biskup and Feldmann (2001) created the bench marks for scheduling jobs on a single machine by considering the Common Due Date as decision variable. Figure 1-5 in the Materials and Methods are drawn by using the bench mark test instances. In this study we discussed about restricted problem with common due date and the early and late penalties being non symmetric and distinct.

Hemamalini *et al.* (2010) proposed DMGS algorithm to solve job of scheduling in m machines.

The sequence of jobs J_i(i = 1,2.. n) are partitioned to two subsets E(J) and L(J) according to the ratio $\frac{\alpha_i}{\beta_i}$. To minimize F(S) two sets are created E(J) and L(J). The set E(J) which have jobs with $\frac{\alpha_i}{\beta_i} < 1$, to be completed before the due date in an optimal sequence and the set L(J) which has jobs with $\frac{\alpha_i}{\beta_i} > 1$, to be completed after the due date d.

In section 2 we discussed about job scheduling according to the ratio $\frac{\alpha_i}{\beta_i}$ and the sum of the processing times of E(J) which is less than the common due date d. In section 3 we proved the properties of the optimal sequence according to the ratio $\frac{\alpha_i}{\beta_i}$ and the sum of the processing times of E(J) which is greater than the common due date d. In section 4 ratio scheduling algorithm is developed based on the ratio $\frac{\alpha_i}{\beta_i}$ which gives an optimal sequence with minimum penalty and the results are illustrated.

MATERIALS AND METHODS

Case 1: In a sequence if $\sum_{j\in(J)} p_i < d$ then there is a time gap (d-Δ) (where $\Delta = \sum_{j\in(J)} p_i$) before the due date d. Also if $\forall J_i \in L(J)$, its $p_i < (d-\Delta)$ then the starting time of the global optimal sequence is either $t = |\min(0, \Delta - d)|$ or $|\min(0, \Delta + \tau_k - d)|$ which depends on $\gamma_i = \frac{\alpha_i}{\beta_i} p_i$ and τ_k is the processing time of a job in L(J) with min γ_i.

Proof: In the optimal schedule the m jobs of E(J) has to be completed before the common due date d and n-m jobs of L(J) has to be completed after the common due date d. Since Δ<d and each job of L(J) has lesser processing time than (d-Δ), but it is not beneficial if a job J_k of L(J) is completed before the due date d (since in the set $L(J), \frac{\alpha_i}{\beta_i} > 1$) which will increase the penalty. Therefore the job whose completion time coincides with common due date d belongs to either E(J) or L(J) which depends on $\gamma_i = \frac{\alpha_i}{\beta_i} p_i$. If $\sum\gamma_i$ of L(J)>d then starting time of the schedule is $t = |\min(0, \Delta - d)|$. And $\sum\gamma_i$ of L(J)>d than starting time of the optimal schedule is a job in L(J) with min γ_i. Figure 1 demonstrates this scenario. Thus if $t = |\min(0, \Delta - d)|$, then the minimum penalty is:

$$F(s) = \sum_{i=1}^{m-1} \{(d - \sum_{j=1}^{i} p_i)\alpha_i\} + \sum_{i=m+1}^{n} \{(\sum_{j=1}^{i} p_i - d)\beta_i\}$$

i.e., the completion time of the last job of E(J) in the optimal schedule coincides with the due date d with nil penalty.

and if $t = |\min(0, \Delta + \tau_k - d)|$, then the minimum penalty is:

$$F(s) = \sum_{i=1}^{m} \{(d - \tau_k - \sum_{j=1}^{i} p_i)\alpha_i\} + \sum_{i=m+2}^{n} \{(\sum_{j=1}^{i} p_i + \tau_k - d)\beta_i\}$$

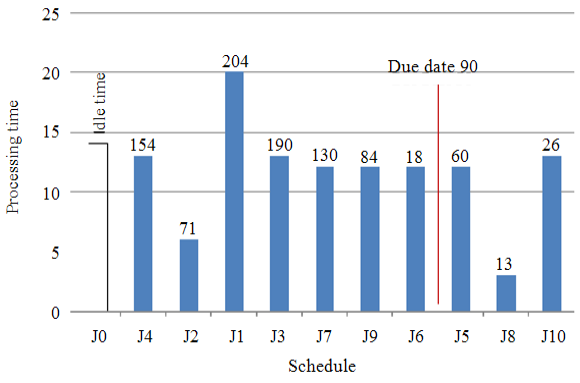

Fig. 1: For Δ<d and p_i<d-Δ$\forall J_i$ of L(J), idle time was inserted before starting the process

i.e., the completion time of a job of L(J) with min γ_i coincides with the common due date d with nil penalty.

Case 2: If $\sum_{J_i \in E(J)} p_i < d$ and some $p_i \in L(J), p_i > (d-\Delta)$ then the starting time of optimal schedule is t = 0 and a job of L(J) with max γ_i will be completed before the common due date d.

Proof: Here also $\Delta<d$, but some of the jobs of the set L(J) has the processing time $p_i>(d-\Delta)$. Assuming that there is no job with $p_i = (d-\Delta)$ (which will be discussed in next case). In all the other cases except the case 1, it is advantageous only if the starting time is t = 0. Since there is a time gap $d-\Delta$, it is possible to processes a job of L(J) before the due date d, which depends on γ_i. i.e., a job of L(J) with max γ_i moves to E(J) and then scheduled according to its processing time and early penalty. Therefore some jobs of E(J) will be completed after the due date d (Fig. 2). Therefore in this case, the number of jobs completed before and after the due date d is less than or equal to m. And the remaining jobs of L(J) will be processed only after all the jobs of E(J) along with a job of L(J) with max γ_i are completed.

Case 3: If $\Delta<d$ and if at least one job of L(J) with $p_i = (d-\Delta)$ then the starting time of the schedule is t = 0 and the completion time of such a job in L(J) coincides with the common due date d.

Proof: In the set E(J) if ($\Delta<d$), then L(J) be the set of jobs which has to be completed after the due date. Suppose L'(J) be the jobs in which $p_i = (d-\Delta)$ then choose a job in L'(J) with min γ_i moves to E(J) and it is scheduled according to its processing time and early penalty. Therefore m jobs will be completed before the due date d and the job of L(J) with min γ_i, is completed exactly at the due date d with nil penalty. And combine the remaining jobs L'(J) with L(J) which will be completed after the common due date d (Fig. 3). Now in this case the early penalty is:

$$F(E_s) = \sum_{i=1}^{m} \{(d - \sum_{j=1}^{i} p_j)\alpha_i\}$$

and the late penalty is:

$$F(L_s) = \sum_{i=m+2}^{m} \{(\sum_{j=1}^{i} p_j - d)\beta_i\}$$

Then $F(S) = F(E_s) + F(L_s)$.

If the sum of the processing time of the jobs of E(J) is greater than the common due date. i.e., $\sum_{p_i \in E(J)} p_i > d$ then it is obvious that the starting time of the sequence is t = 0 and the completion time of the last job of E(J) in the optimal schedule is after the common due date.

Case 1: If there exists some jobs of E(J) with the processing time p_i such that $\Delta-p_i<d$, then the job with max γ_i will be completed after the due date d in the optimal sequence.

Proof: Let E'(J) be the non empty subset of E(J) such that $E'(J) = \{J_i/\Delta-p_i<d\}$. Since $\Delta<d$ and E'(J) is non empty, exactly one of the job in E'(J) will be completed after the due date d with late penalty L_i. The remaining jobs of E'(J) are processed before the common due date d with early penalty E_i. Here also γ_i is the decision variable that the job of E'(J) with max γ_i will move to L(J) and scheduled according to its processing time and late penalty (Fig. 4). Therefore n-m+1 jobs will be completed after the due date d:

$$\therefore F(S) = \sum_{i=1}^{m-1} \{(d - \sum_{j=1}^{i} p_j)\alpha_i\} + \sum_{i=m}^{n} \{(\sum_{j=1}^{i} p_j - d)\beta_i\}$$

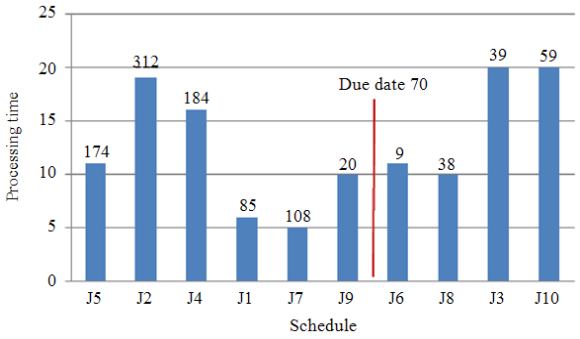

Fig. 2: For $\Delta<d$ and not all $p_i<d-\Delta$ of L(J), then the process starts at time t = 0

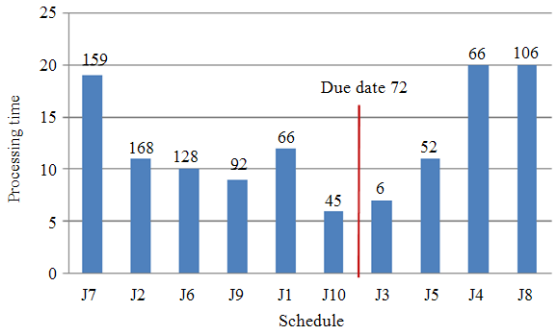

Fig. 3: For $\Delta<d$ and at least one $p_i = d-\Delta$ of L(J), then the completion time of that job coincides with the common due date

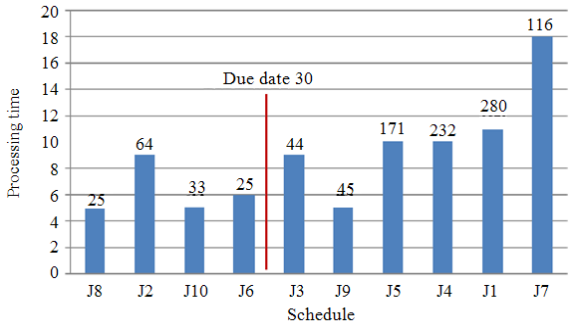

Fig. 4: For Δ>d, then jobs of L(J) moves to E(J)

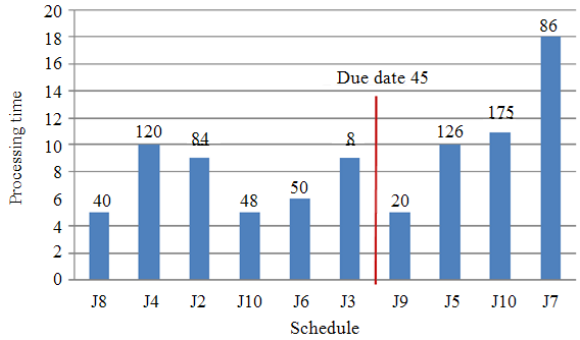

Fig. 5: For Δ>d and at least one p_i = d-Δ of E(J) then the completion time of that job coincides with common due date

Corollary 1: If there exists some job of E(J) such that Δ-p_i = d then E'(J) = {J_i/Δ-p_i = d}. Therefore in the optimal sequence the job of E(J) with max γ_i combines with L(J), also a job of E(J) is completed exactly on the due date d with nil penalty. This can be proved in a similar way.

Corollary 2: If the set E'(J) = {J_i/Δp_i<d} is empty in E(J), then there exist two or more jobs in E(J), which are completed after the common due date d with late penalty such that γ_i of those jobs are greater than γ_i of the jobs which are completed before the due date d in an optimal sequence (Fig. 5).

Corollary 3: It is obvious that the above case is true if L(J) is empty (i.e.,) in the given sequence all the jobs with (α_i/β_i)<1 for the restricted problem.

Corollary 4: If there exists some jobs with $\frac{\alpha_i}{\beta_i}=1$, Then $S(J)=\{J_i / \frac{\alpha_i}{\beta_i}=1\}$. But in our algorithm we have partitioned the given set of jobs as E(J) and L(J), so in order to move the jobs of S(J) to E(J) or L(J),let us use the following conditions:

- If Δ<d, then the job with max γ_i of S(J) move to E(J), the process repeats until the gap (Δ-d) is zero or p_i of max γ_i is greater than (Δ-d) and the remaining jobs of S(J) combines with L(J)
- If Δ>d, then all the jobs of S(J) combines with L(J). Thus the given set of jobs can be partitioned to E(J) and L(J)

Scheduling algorithm:

```
algorithm scheduleJobs(list of jobs, common due date)
{       sort list of jobs based on ratio_factor in non increasing order
        {
                Update ratio list
        } where γi >∅;∅-higher ratio should be scheduled before lower ratio in early order split ratio_list into two lists early_list and late_list
        for (i = 0;i<NumberOfJobs;i++)
        {
                If ∑ (j=1 to i) pi < d
                        Update early_list
        Else
                        Update early_list
}
for (i = 0;i<NumberOfJobsEarlyJobs;i++)
{       sort early_list in non decreasing order
                jobi value (γi) is higher than jobi+1 value(γi) if and only if (jobi cost * jobi+1 early penalty ) > (jobi+1 cost * jobi early penalty)
}
for (i = 0;i<NumberOfJobsEarlyJobs;i++)
{       sort late_list in ascending order
        jobi value (γi) is higher than jobi+1 value (γi) if and only if (jobi cost * jobi+1
        late penalty ) > (jobi+1 cost * jobi late penalty)
    }
}
```

RESULTS

Results are demonstrated with the set of jobs to be sequenced before the due and after the due date. Jobs displayed before the red mark should be scheduled before due date d. Sequence was highlighted in the X-axis. Processing time was placed in the Y-axis. Each cell was updated with the penalty. Figure 1-5 illustrated with all possible cases.

DISCUSSION

The existing algorithms like Tabu search and Genetic Algorithm, the time complexity is more i.e., in worst case time complexity is O(n!). But the proposed algorithm time complexity is 2 log (n). So the proposed algorithm outperforms in many instances of the test cases. The proposed algorithm is also suitable for unrestricted problem against single machine, the algorithm can also be extended to m machine scheduling problems.

CONCLUSION

Based on proposed algorithm we present some of the properties of the jobs whose completion time coincides with the common due date and the instances in which early jobs moves after the due date and the late jobs before the due date. The proposed algorithm gives encouraging result for the bench mark instances when the due date is less than half of the total processing time. The algorithm was implemented using Java platform. The authors are gladly willing to distribute the jar file by email.

REFERENCES

Bagchi, U., R.S. Sullivan and Y.L. Chang, 1987. Minimizing mean squared deviation of completion times about a common due date. Manage. Sci., 33: 894-906. Dhttp://www.jstor.org/stable/2632140

Bender, M.A., R. Clifford and K. Tsichlas, 2007. Scheduling algorithms for procrastinators. J. Schedul., 11: 95-104. DOI: 10.1007/s10951-007-0038-4

Biskup, D. and M. Feldmann, 2001. Benchmarks for scheduling on a single machine against restrictive and unrestrictive common due dates. Comput. Operat. Res., 28: 787-801. DOI: 10.1016/S0305-0548(00)00008-3

Cheng, T.C.E. and M.C. Gupta, 1989. Survey of scheduling research involving due date determination decisions. Eur. J. Operat. Res., 38: 156-166. DOI: 10.1016/0377-2217(89)90100-8

Dileepan, P., 1993. Common due date scheduling problem with separate earliness and tardiness penalties. Comput. Operat. Res., 20: 179-184. DOI: 10.1016/0305-0548(93)90073-R

Feldmann, M. and D. Biskup, 2003. Single-machine scheduling for minimizing earliness and tardiness penalties by meta-heuristic approaches. Comput. Ind. Eng., 44: 307-323. DOI: 10.1016/S0360-8352(02)00181-X

Ghoul, R.H., A. Benjelloul, S. Kechida and H. Tebbikh 2007. A scheduling algorithm based on Petri nets and simulated annealing. Am. J. Applied Sci., 4: 269-273. http://www.scipub.org/fulltext/ajas/ajas45269-273.pdf

Hemamalini, T., A.N. Senthilvel and S. Somasundaram, 2010. Deep memory greedy search and allocation of job shop scheduling for optimizing shop floor performance. Asian J. Ind. Eng., 2: 9-16. DOI: 10.3923/ajie.2010.9.16

Hoogeveen, J.A. and S.L. van de Velde, 1991. Scheduling around a small common due date. Eur. J. Operat. Res., 55: 237-242. DOI: 10.1016/0377-2217(91)90228-N

Hong, T.P., P.Y. Huang, G. Horng and C.L. Wang, 2007. Three algorithms for flexible flow-shop scheduling. Am. J. Applied Sci., 4: 887-895. http://www.scipub.org/fulltext/ajas/ajas411889-896.pdf

Ismail, Z. and S.L. Loh, 2009. Ant colony optimization for solving solid waste collection scheduling problems. J. Math. Stat., 5: 199-205. http://www.scipub.org/fulltext/jms2/jms253199-205.pdf

Lenstra, J.K. and A.H.G. Rinnooy Kan, 1979. Computational complexity of discrete optimization problems. Ann. Discrete Math., 4: 121-140. DOI: 10.1016/S0167-5060(08)70821-5

Sule, D.R., 2007. Production Planning and Industrial Scheduling: Examples, Case Studies and Applications. 2nd Edn., CRC Press, USA., ISBN: 10: 1420044206, pp: 560.

Yang, M.F. and Y. Lin, 2009. The coordinated scheduling support system of production and delivery. Am. J. Applied Sci., 6: 601-607. http://www.scipub.org/fulltext/ajas/ajas64601-607.pdf