ICMIEE-PI-140253

Application of Branch and Bound algorithm for solving flow shop scheduling problem comparing it with tabu search algorithm.

* *Mr. Subrata Talapatra[1], Md. Shamsur Rahman[2], Chandnee Das[3], Utpal Kumar Dey[4]*
[1, 2, 3] Department of Industrial Engineering and Management (IEM)
[4]Department of Computer Science and Engineering
Khulna University of Engineering & Technology (KUET)
Khulna-9203, Bangladesh.

ABSTRACT
The leading position in contributing to the economics of many countries is hold occupied by the garments factory and it has great opportunity to enhance its area .In Bangladesh, the garment factory is the top of the organization, which takes the vital role in the economic sector. As the number of jobs and machines increase, the flow shop scheduling problems in the industry approaches to difficulty. Consider a regular flow shop cell with several bottleneck stages. If such were the case, the industry owners would provide more resources to these bottleneck stages. In this case it is so much important to eliminate the bottleneck in production section and improve the total productivity of the industry. This paper deals with the Branch and Bound technique for solving M machines and N jobs in flow-shop scheduling problem. Here the optimal sequence of jobs is obtained through minimizing the total elapsed time by a lower Bounding (LB) method based on the Branch and Bound algorithm. The working of the algorithm has been illustrated by numerical example and also a C++ code was used to generate an algorithm for finding the optimal solution. The input parameters are process time and operation sequence for each job in the machines provided. This research ensures the makespan optimal values of the schedules comparing with the Tabu search method.

Keywords: Flow-Shop, Branch and Bound, Scheduling, Makespan, C++ code, Tabu search algorithm.

1. Introduction
Consider n different jobs that need to be processed on m machines in the same order. Each job has one operation on each machine and the operation of job i on machine j has processing time Pij. This problem is called a flow shop problem [1,5].In a flow shop problem all jobs have the same ordering sequence on all machines. An optimal permutation schedule does not produce an appreciably worse performance than the optimal general flow shop schedule [4]. Also schedules are attractive from a practical point of view since they are easier to implement. The flow shop problem is NP-hard for m >=3 [3,7]. Optimal solutions can only be obtained via enumeration techniques such as Branch and Bound[6] . However, these methods may take a prohibitive amount of computation even for medium-size problems and become intractable for large problems. This leads to the development of many heuristic procedures. Heuristics for solving the flow shop scheduling problem can be divided into two categories: sequence generating heuristics and improvement heuristics. The former methods generate a schedule from scratch. Most of these methods are either extensions or based on the ideas behind Johnsons' s well known algorithm for solving two- and three-machine problems[2,4,8,9]. Starting with a solution produced by some sequence generating heuristic, improvement heuristics provide a scheme for obtaining a new sequence with improved performance measure. Methods of this type include neighborhood search techniques[5] such as simulated annealing and tabu search. Tabu search is a local search based optimization method which has been successfully used to solve many difficult combinatorial optimization problems, particularly in the scheduling area. It also exhibited considerable robustness.

2. Literature review
Many applied and experimental situations, which generally arise in manufacturing concern to get an optimal schedule of jobs in set of machines, diverted the attention of researchers and engineers. In flow-shop scheduling, the objective is to obtain a sequence of jobs which when processed in a fixed order of machines, will optimize some well defined criteria. Various researchers have done a lot of work in this direction. Johnson , first of all gave a method to minimise the makespan for n-job, two-machine scheduling problems. The work was further extended by Ignall and Scharge, Cambell[2], Maggu and Dass, Heydar,Yoshida and Hitomi, Lomnicki Palmer Bestwick and Hastings,Nawaz et al.Sarin and Lefoka , Koulamas,Dannenbring , etc. by considering various parameters.Yoshida and Hitomi considered two stage flow shop problem to minimize the makespan whenever set uptimes are separated from processing time. The basic concept of equivalent job for a job block has been introduced by Maggu and Dass. Singh T.P. and Gupta Deepak studied the optimal two stage production schedule in which processing time and set uptime both were associated with probabilities including job block criteria. Heydari dealt with a flow shop scheduling problem where n jobs are processed in two disjoint job blocks in a string consists of one job block in which order

* Corresponding author. Tel.: +88-01727555334
E-mail address: sub_ksy@yahoo.com

of jobs is fixed and other job block in which order of jobs is arbitrary. Lomnicki introduced the concept of flow shop scheduling with the help of Branch and Bound method. Further the work was developed by Ignall and Scharge , Chandrasekharan, Brown and Lomnicki , with the Branch and Bound technique to the machine scheduling problem by introducing different parameters. In practical situations processing times are not always deterministic so we have associated probabilities with their processing times of all the jobs on all the three machines. This hence the problem discussed here is wider and has significant use of theoretical results in process industries.

3. Assumptions

a) No passing is allowed.
b) Each operation once started must performed till completion.
c) A job is entity, i.e. no job may be processed by more than one machine at a time.

Notations:
We are given n jobs to be processed on three stage flow shop scheduling problem and we have used the following notations:
Ai: Processing time for job i on machine A
Bi: Processing time for job i on machine B
Ci: Processing time for job i on machine C
Pi: Expected processing time for job i on machine A
pi2: Expected processing time for job i on machine B
pi3: Expected processing time for job i on machine C
Cij: Completion time for job i on machines A, B and C.
S0: Optimal sequence
Jr: Partial schedule of r scheduled jobs.
Jr′: The set of remaining (n-r) free jobs

4. Mathematical Development

Consider n jobs say i=1, 2, 3 … n are processed on three machines A, B & C in the order ABC. A job i (i=1,2,3…n) has processing time Ai , Bi & Ci on each machine respectively, assuming their respective probabilities pi , qi & ri such that $0 \leq pi \leq 1, \Sigma pi = 1$, $0 \leq qi \leq 1$, $\Sigma qi = 1$, $0 \leq ri \leq 1$, $\Sigma ri = 1$. The mathematical model of the problem in matrix form can be stated as:

Step 1: Calculate expected processing time Pi1, Pi2 &Pi3 on machines A, B & C respectively
Step 2: To calculate
(i) LB1 =Max Pi1+Min (Pi2+pi3)
(ii) LB2=MinPi1+Max Pi2+Min Pi3
(iii) LB3=Min (Pi1+Pi2) +MaxPi3
Step 3: To calculate LB
Step 4: Continuing this way, until we reach at the end of the tree. Finally we get the optimal schedule of the jobs.

Consider 3 machines and 4 jobs are used in flow shop scheduling. Applying Branch and Bound method for reducing flow shop scheduling problem. The processing time of these 3 machines is Pi1, Pi2, and Pi3.Calculate the completed time by this method using this Branch and Bound steps. Calculation is done in below:

4.1. Algorithm development for branch &bound method

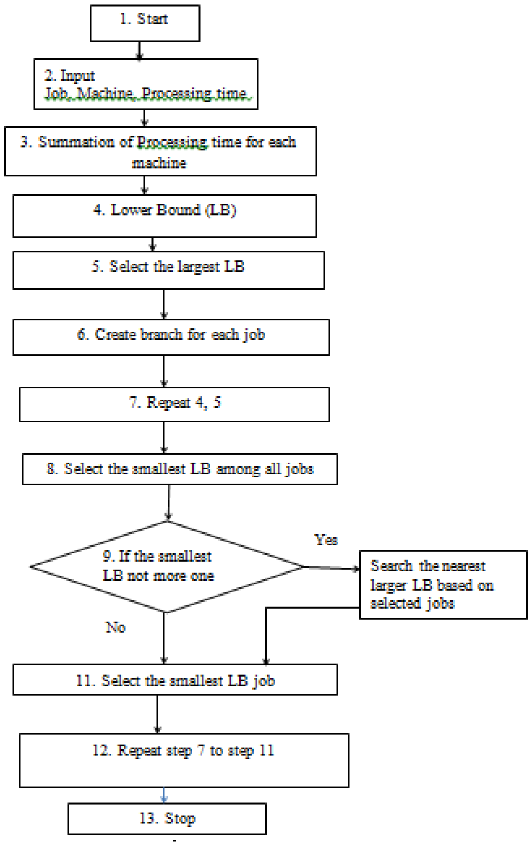


Fig1: Flow diagram of Branch-and-Bound algorithm

5. Data Analysis & Calculation

Job\Machine	M1	M2	M3
Processing time	Pi1	Pi2	Pi3
J1	77	11	82
J2	34	92	8
J3	88	36	30
J4	1	98	9
total	200	237	129

So the makespan is more than or equal to 237.

LB1:200+66=266
=max Pi1+min (Pi2+pi3)
LB2:MinPi1+max Pi2+Min Pi3
=1+237+8=246
LB3: Min (Pi1+Pi2) +MaxPi3=88+129=217

5.2. Branch & Bound's search tree diagram

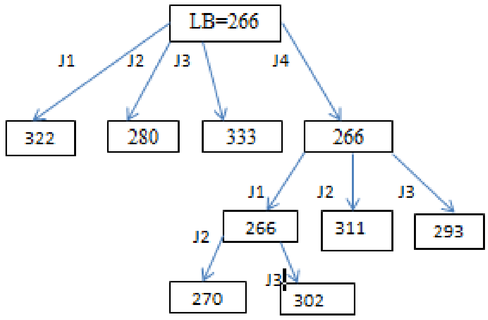


Fig2: Branches for the optimal sequence

Table 1 Verification between old sequence and new sequence job

Job\Machine	M1	M2	M3
J1	77	88	170
J2	111	203	211
J3	199	239	269
J4	200	337	346

Here, without using Branch and Bound method, total completion time is 346 minute for initial sequence of job.

Job\Machine	M1	M2	M3
J4	1	99	108
J1	192	110	192
J2	112	204	212
J3	270	240	270

After calculating by Branch and Bound method, we get the job sequence is J4-J1-J2-J3.The total completion time of this sequence is 270 minute. So, makespan is reduced.

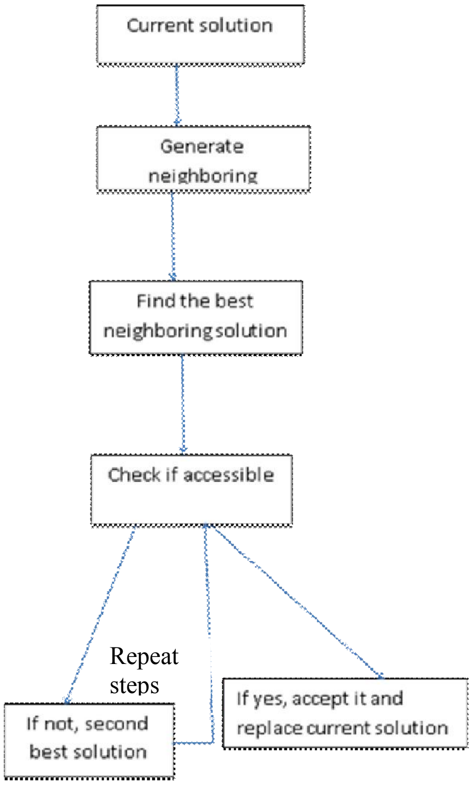
Now this Branch and Bound result compare with tabu search algorithm results.

6. Tabu Search algorithm

A tabu search approach for solving the permutation flow shop scheduling problem. The proposed implementation of the tabu search approach suggests simple techniques for generating neighborhoods of a given sequence and a combined scheme for intensification and diversification that has not been considered before. These new features result in an implementation that improves upon previous tabu search implementations that use mechanisms of comparable simplicity. Tabu search is a local search based optimization method which has been successfully used to solve many difficult combinatorial optimization problems, particularly in the scheduling area. It also exhibited considerable robustness.
In its simplest form, tabu search requires the following ingredients:

a) initial sequence;
b) mechanism for generating some neighborhood of the current sequence;
c) tabu list;
d) Stopping criteria.

6.1. Flow diagram of proposed tabu search algorithm:



6.2. Step development of this Tabu search algorithm

a) Initial solution:

To get an initial starting solution here Branch and Bound method is applied.

b) Neighborhood structure:

Given a sequence s, we define N(s) as being the set of all sequences which can be obtained from s using one of the following schemes:
(1) Swapping: given a sequence s, let i and j be two positions in the sequence s. A neighbor of s is obtained by interchanging the jobs in positions i and j. The positions i and j can be specified in one of two ways:

a) Positions i and j are selected randomly; or
b) They are enumerated in some systematic way. Such as interchanging adjacent pair wisely.

(2) Insertion:
given a sequence s, let i and j be two positions in the sequences. A neighbor of s is obtained by inserting the job in position i in position j. The positions i and j can be specified in one of two ways:

a) Positions i and j are selected randomly; or

b) they are enumerated in some systematic way such as inserting every job in every position.

c) Selecting the best neighbor in the candidate list

The objective function is the makespan. Thus, we define ‘ best’ simply by reference to the objective function and the current tabu conditions, the best neighbor in the candidate list is the sequence that yields the smallest makespan. Possible ways to select the best neighbor include:

i. Choose the first sequence that improves makespan.
ii. Consider a subset of neighbors and’ search for the best sequence.
iii. Search the whole neighborhood and choose the best sequence.

In our algorithm, we adopted the first alternative, that is, we examine the neighborhood and take the first sequence which improves the current solution. If there is no move that improves the solution then we examine the whole neighborhood.

d) Tabu list:

The size of the tabu list is a very important parameter of a tabu search algorithm. The tabu list can be either fixed or variable.

e) Aspiration criteria

In order to override the tabu list when, there is a good tabu move, we introduced the aspiration crite-rion concept. We used the simplest form of aspiration criterion which is stated as follows: a tabu move is accepted if it produces a solution better than the best obtained so far.

f) Intensification and diversification scheme

As previously noted, an intensification scheme often takes the form of reinforcing attributes of good solutions while a diversification scheme typically consists of driving the search into regions not yet explored.

The random restarting approach, as its name suggests, consists of restarting the algorithm by a sequence that is randomly generated. The second scheme uses a frequency matrix which is constructed as follows. The frequency matrix is an n x n matrix whose columns are the n positions in a sequence and its rows are the n jobs. Each entry fij is incremented by 1 whenever job i visits position j. Using this frequency matrix, the sequence used to restart the algorithm is generated using the following procedure:

Step1. Let k = 1

Step2. Find the entry fij of the frequency matrix having the maximum value. The algorithm is stopped when there is no improvement between two consecutive calls of the diversification scheme. Other criteria could be used such as stopping after some maximum number of iterations or stopping after some maximum number of calls of the diversification scheme. This completes the description of our implementation of the tabu search algorithm. In the next section, we report and analyze our computational experiments.

Step3. Assign job i to position j

Step4. Delete row i and column j

Step5. Set k = k + 1

Step6. If k> n stop, otherwise go to Step 1.

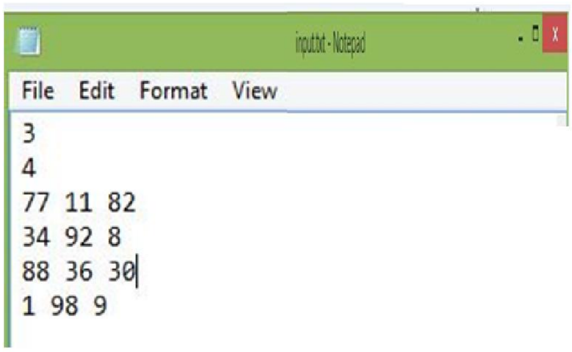
g) Stopping criteria

The algorithm is stopped when there is no improvement between two consecutive calls of the diversification scheme. Other criteria could be used such as stopping after some maximum number of iterations or stopping after some maximum number of calls of the diversification scheme. This completes the description of our implementation of the tabu search algorithm. In the next section, we report and analyze our computational experiments.

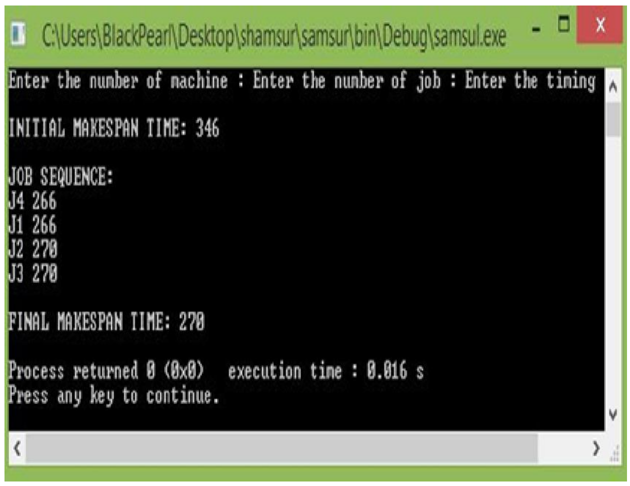
After using Branch & Bound method and Tabu search method with c++ codding software, we get the following completion time:

Input: In this notepad the inputs are taken sequentially. Such firstly enter the number of machines then number of jobs and finally the job processing time. When all the data are taken then runs the c++ program and get the output.

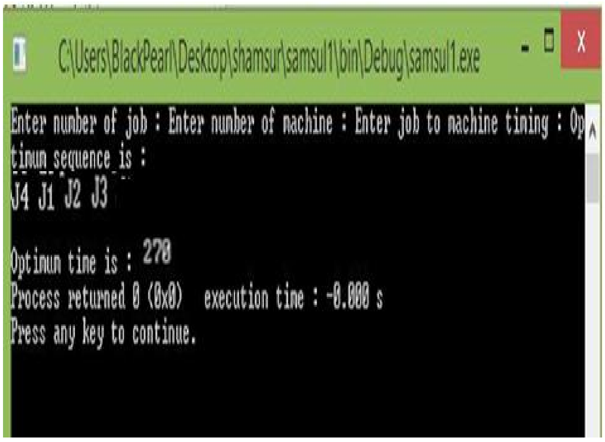
Here,
No of machine is 3
No of job is 4
The Processing time of 3 machine is 77,11, 82 minutes.



Output for Branch &Bound



Output for Tabu search



7. Gantt chart

Devised by Henry Gantt in 1910s, Gantt chart is the representation type of bar chart used to represent a feasible schedule of a scheduling problem. Gantt chart also provides the details about the precedence of operations under taken by the Jobs in various machines. The main advantage of the Gantt Charts is that visualizing the job schedule makes it very easy for the Project Manager to communicate the job schedule to various stakeholders as well as to the project team. Gantt chart is an apt medium for portraying a resultant schedule in a small problem, but a problem with large number of activities that is very difficult to represent the schedule. Gantt does not represent the relative size of work items or the total size of the project. Therefore, it becomes too tough in some cases to compare two projects with the same number of time of completion.

7.1. Reasons for using Gantt chart

Because of the many advantages offered by Gantt charts, thousands of companies use Gantt charts to become more productive, enhance their communications, forecast over the long term and track results. While some a naysayers believe they limit the size of the project that can be tracked, those using Gantt charts note an array of key benefits, including the five listed here. While there are a number of reasons to use Gantt charts below are five key reasons they are often advantageous :

a) **Avoid Completion Confusion**: Gantt charts were created to keep users on track, providing a visual timeline for starting and finishing specific tasks. By providing a visual overview of milestones and other key dates, these charts are thought to offer a more understandable and memorable method of maintaining timescale-based tasks and deliverables whether tracked on a daily, weekly, monthly or yearly basis. Below diagram shows the power of visualization found in Gantt charts. In a glance you can see that the interviews are done, there 50% more to do in training etc.

b) **Keep everyone on the Same Page**: Where there is a visual framework for the work to be done, there are fewer chances for misunderstanding, especially when it comes to highly complex tasks. Using Gantt charts allow all types of stakeholders to have the same information, set mutually understood expectations, and conduct their efforts according to the desired protocol.

c) **Understand Task Relationships**: These charts can make clear how various tasks are interrelated and perhaps rely on the completion of another to meet specific objectives. These task relationships revolve around understanding the timing of each task, which then impacts other tasks listed. This can better assure the optimum work flow, maximized productivity and overall project success. A Gantt charts makes it very easy to visualize related tasks.

d) **Effectively Allocate Resources**: By being able to look ahead on the Gantt chart, users can clearly discern where resources need to be anticipated, allocated or shared to maximize the use of those resources. The more closely the chart is followed, the better chance there is of keeping project costs within budget while also better assuring on-time completion.

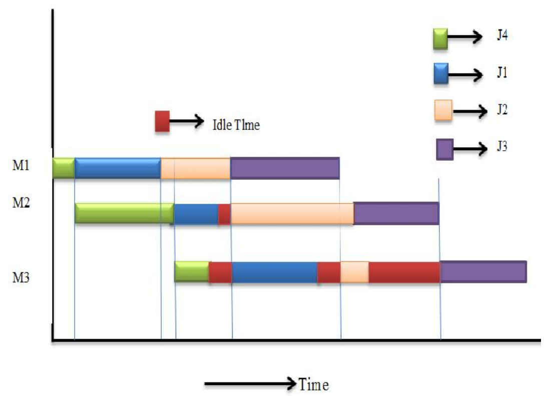


Fig 3 Gantt chart

In fig 3 the flow of jobs after scheduling through Branch and Bound method are shown graphically. It is seen that as jobs are sequenced so the idle time is minimum. Here the red color indicates idle of the machine and other colors indicate machining or working time. Though this gantt chart shows working procedure so much easily, it has also some problems. These are this method is not designed to be the cure-all for an organization's project management ills. The main disadvantage is that Gantt Charts are not well suited for conveying complex dependencies or projects having significant potential variability in completion dates. There are some situations where other tools may indeed be more effective – particularly in scenarios when a particular milestone or critical task is missing because the project manager didn't include. Other limitations include the inability to include certain constraints like time, scope, and costs. Overall, however, Gantt chart advantages have been realized by all types of organizations for applicable applications.

8. Result and discussion

In this paper, applying this Branch and Bound algorithm and tabu search algorithm we can minimize the makespan(total completion time of the last job) of the tasks. We also can reduce waiting time between two jobs which can help for increasing the productivity of the flow shop. To complete above this tasks, first we apply Branch and Bound algorithm then we compare this algorithm with Tabu search algorithm. In Tabu search algorithm we generate the tabu list solution until we can get optimal solutions. We will mainly follow the Branch & Bound method, because it is the most effective for scheduling the total processes in any apparel industry. If we notice to the output Branch & Bound and Tabu search methods, then it is clearly understood that the Branch & Bound method is the best than others. Actually here we use Tabu search method as a parameter which ensures the Branch & Bound method's effectiveness. Here the optimum processing time using Tabu search method and the Branch & Bound is 270 unit times. Finally, the best optimal solution is obtained from Branch & Bound algorithm for n number of jobs and m number of machines through comparing with Tabu search method.

9. Conclusion and Recommendation

In flow shop scheduling, jobs are processed on machines in a set order .Flow shop scheduling problems area class of scheduling problems with a work shop or group shop in or with other resources 1,2…m in compliance with given processing orders. Especially, the maintaining of a continuous flow of processing tasks is desired with a minimum of idle time and minimum of waiting time. Reducing this waiting time we apply Branch and Bound algorithm and Tabu search algorithm in flow shop scheduling problem. The proposed implementation of the tabu search approach uses simple but effective techniques for generating neighborhoods and candidate lists, and a new combined intensifications and diversifications scheme that has not been considered before. These new feature results in an implementation of the tabu search approach. Better results are obtained.

Reducing flow shop scheduling problem some recommendations are provided:

1. Prevent recycling the tabu list in tabu search algorithm &generate the solution until we get our expected solution.
2. Further research should try to remove the total idle time from the overall process.
3. The research also improves the overall productivity of an apparel industry.

References

[1] K.R Baker,Introduction to sequencing and scheduling,viley,,,New York,1974

[2] H.G Campbell,R.A.Dudek,M.L.Smith,A heuristic algorithm for the n-job, m-machine sequencing problem,Management science16/B(1970) 630-637.

[3] E.G,Coffiman,Computer and job Scheduling Theory,Wiley,New York,1976.

[4] D.G.Dannenbring ,An Evaluation of flow shop sequencing heuristics,Management science 23(1977) 1174-1182

[5] S.French, Sequencing And Scheduling:An Introduction to the Mathematics of the job shop,Ellis Horwood,Chichester,UK 1982.

[6] E.Ignall,L.E.Scharge,Application of Branch and Bound technique to some Flow –shop problem,Operations research13(1965) 400-412

[7] A.H.G Rinnoy Kan,Machine Scheduling Problems:Classification,Complexityandcomputations,Martinus Nijhoff,The Hague,1976

[8] J.N.D Gupta,A functional heuristic algorithm for the flow shop scheduling problem,Operational Research Quarterly 22 (1971) 39-47

[9] T.S Hundal,J.Rajgopal,An extension of palmers heuristic for the flow-shop scheduling problem,International journal of production Research26(1988) 1119-1124