Nigerian Journal of Engineering Science Research (NIJESR), 2(2): 27-37
Copyright@ Department of Mechanical Engineering, Gen. Abdusalami Abubakar College of Engineering, Igbinedion University, Okada, Edo State, Nigeria.
ISSN: 2636-7114
Journal homepage: www.nijesr.iuokada.edu.ng

Analysis and Application of Branch and Bound Technique for Minimizing the Make-Span of a Flow Shop

[1]*Oghoghorie, O., [2]Orhorhoro, E.K., [3]Erameh, A.A.

[1]Department of Mechanical Engineering, Benson Idahosa University, Benin City, Nigeria (ooghoghorie@biu.edu.ng)
[2]Department of Mechanical Engineering, College of Engineering, Igbinedion University, Okada, Nigeria (ejiroghene.orhorhoro@iuokada.edu.ng)
[3]Department of Mechanical Engineering, College of Engineering, Igbinedion University, Okada, Nigeria (erameh.andrew@iuokada.edu.ng)

**Corresponding author name: Oghoghorie Oghenekevwe (ooghoghorie@biu.edu.ng; +2348065259539)*

Manuscript History
Received: 26/06/2019
Revised: 25/10/2019
Accepted: 03/11/2019
Published: 15/01/2020

Abstract: *How to schedule jobs on machines subject to certain constraints to optimize some objective functions is actually a scheduling problem. Scheduling specifies when and on which machine each job is to be executed. This research work describes the algorithm for branch and bound techniques for minimizing the make-span of a flow shop. The flow shop problem with make span criterion can be shown by* $n/m/F/c_{max}$ *or equivalently* $F//c_{max}$*, where both show a (n-job, m-machine) flow shop problem with make span criterion that can be defined as completion time at which all jobs complete processing or equivalently as maximum completion time of jobs. The objective of this present study is to obtain the optimal schedule of all jobs which minimize the total elapsed time using branch and bound technique. The optimal sequence is job1—job2—job4—job6—job5—job3. Finally, it was realized from the research that the total elapsed time or make-span is 8657.5mins.*

Key words: *Flow shop, Branch and bound techniques, Optimal sequence, Jobs, Machines*

INTRODUCTION

Mouka foam involves in the production of six different types of products for customers. The major challenge of the company is to determine the order in which the different jobs must be processed to minimize the make-span. The way in which orders are received from customers is in a random process. At present, there is no scientific approach for determining the order in which the different jobs must be scheduled for production using its facilities. Increase in cost of production and inability to meet customers' demands are some of the challenges being faced by the company since the scheduling is done at the discretion of the production manager without the application of any technique. Thus, there is a great need to develop a technique that can minimize the make-span of producing the different jobs. The uses of sequence-dependent scheduling are commonly found in most manufacturing environments. For instance, in Mouka Foam Company, machines must be adjusted whenever there is change in dimensions of end products. Thus, the process is a multi-stage production system (i.e., flow shop). In this situation, a sequence-dependent setup times play a major role and must be considered when modeling the problem.

A flow shop as the name implies is a multi-stage production system with more than one parallel machine at each stage and all products going through the system unidirectional (i.e., stage 1, then stage 2, and so on) (Balasubramanian and Grossmann, 2002; Chen *et al.*, 2014). The area of flow-shop problems, scheduling theory has been strongly influenced by Johnson's early works (Pranzo, 2004). There are thousands of research work on different optimal procedures and heuristics for solving the flow shop scheduling problem and its variants (Nawaz *et al.*, 1983). The regular flow-shop problem consists of two main elements (Balasubramanian and Grossmann, 2004);

i. A group of M machines and
ii. A set of N jobs to be processed on this group of machine

Each of the N jobs has the same ordering of machines for its process sequence. Each job can be processed on one and only one machine at a time (which means no job splitting), and each machine can process only one job at a time. Each job is processed only once on each machine. Operations are not pre-emptable and set-up times of operations are independent of the sequences and therefore can be included in the processing time. The scheduling problem is to specify the order and timing of the processing of the jobs on machines, with an objective or objectives respecting above-mentioned assumptions (Bassett *et al.*, 1996; Ben-Tal and Nemirovski, 2000; Bonfill *et al.*, 2005; Jia and Ierapetritou, 2000). Francesco and Raffaele (2008) introduces an additive branch-and-bound algorithm for two variants of the pickup and delivery traveling salesman problem in which loading and unloading operations have to be performed either in a last-In-First-Out (LIFO) or in a First-In-First-Out (FIFO) order. He makes use of two relaxations within the additive approach via the assignment problem and the shortest spanning r-arborescence problem. The quality of the lower bounds was further improve by a set of elimination rules applied at each node of the search tree to remove from the problem arcs that cannot belong to feasible solutions because of precedence relationships. Also, three integer programming formulations and a branch-and-cut algorithm for the Traveling Salesman Problem with Pickup and Delivery with LIFO Loading (TSPPDL) (TSPPDL) were introduced by Cordeau *et al.* (2008). This approach is based on the Traveling Salesman Problem with Pickup and Delivery (TSPPD) formulation of Ruland and Rodin (1997) and relies on an exponential number of constraints to impose the LIFO policy. Several families of valid inequalities are also used to strengthen the formulation. Exact separation procedures are used to identify violated subtour elimination constraints, precedence constraints and LIFO constraints, while heuristic separation procedures are used for the other families of inequalities. This algorithm is able to solve most instances with up to 43 vertices and some instances with 51 vertices in less than 60 minutes of computing time. However, in this present research work, a scheduling technique was applied to determine the order in which the jobs manufactured by Mouka foam company are produced in order for the make-span of flow shop to be minimized.

METHODOLOGY

A. Flow Shop

In a flow-shop, the work in a job is broken down into separate tasks called operations, and each operation is performed at a different machine. In this context, a job is a collection of operations with a special precedence structure. In particular, each operation after the first has exactly one direct predecessor and each operation before the last has exactly one direct successor, as shown in the flow chart (Fig. 1). Thus, each job requires a specific sequence of operations to be carried out for the job to be complete (Kallrath, 2000).

Foaming → Fork-lift truck → Curing → Slicing → Taping → Nyloning

Fig. 1 Flow chart

Data Collection

The data used in this research work was collected from Mouka Foam Nigeria Limited. The data used was based on the major activities in each stage of foam production. These are:

A. *Machines*

i. Foaming Area (Stage 1, Machine 1)
ii. Fork-lift truck (Stage 2, Machine 2)
iii. Curing Hall (Stage 3, Machine 3)
iv. Slicing (Conversion) (Stage 4, Machine 4)
v. Tape edge Section (Stage 5, Machine 5)
vi. Nyloning section (Stage 6, Machine 6)

B. Processing Time

Table 1 shows the data of six (6) jobs to six (machines) flow-shop problem with the respective processing time of the jobs on each machine.

Table 1 Processing time of jobs on each machine

	Jobs					
Machines	1	2	3	4	5	6
M_1	1.5	2.0	1440	3.0	8.0	3.0
M_2	1.7	2.5	1440	3.2	12.0	3.2
M_3	1.9	2.8	1440	3.5	13.0	3.5
M_4	2.1	2.6	1440	3.6	13.5	3.8
M_5	2.5	3.2	1440	3.8	15.0	4.0
M_6	4.2	2.2	1440	5.0	20.0	5.0

Data Analysis

In this research work, branch and bound approach was adopted. The branch and bound approach is a useful method for solving many combinatorial problems. The method involves two fundamental procedures. Namely;

i. Branching which is the process of partitioning a large problem into two or more sub-problems
ii. Bounding is the process of calculating a lower bound on the optimal solution of a given sub-problem.

The branching procedure replaces an original problem by a set of new problems that are:

i. Mutually exclusive and collectively exhaustive sub-problems of the original,
ii. Partially solved versions of the original, and
iii. Smaller problems than the original.

The bounding procedure calculates a lower bound on the solution to each sub-problem generated in the branching process. It is based on the assumption that;

i. Suppose that at some intermediate stage a complete solution has been obtained that has an associated performance measure Z.
ii. Suppose also that a sub-problem encountered in the branching process has an associated lower bound $b > Z$.

Then the sub-problem need not be considered any further in the search for an optimum. That is, no matter how the sub-problem is resolved, the resulting solution can never have a value better than Z. When such a sub-problem is found, its branch is said to be fathomed. By not branching any further from fathomed branches, the enumeration process is curtained because feasible solutions of a fathomed sub-problem are evaluated implicitly rather than being constructed explicitly.

A complete solution that allows branches to be fathomed is called a trial solution. It may be obtained at the very outset by applying a heuristic procedure (i.e. a sub-problem method capable of obtaining good solution with limited computational effort); or it can be obtained in the course of the tree search, perhaps by pursuing the tree directly to the bottom as rapidly as possible.

Notation

The illustration of how these concepts are applied in the flow shop problem is shown with the followings notations. When given n jobs to be processed on six stage flow shop scheduling problem, the following notations apply;

A_i = Processing time for job i on machine A
B_i = Processing time for job i on machine B
C_i = Processing time for job i on machine C
D_i = Processing time for job i on machine D
E_i = Processing time for job i on machine E
F_i = Processing time for job i on machine F
C_{ij} = Completion time of job i on machine j
J_r = Partial schedule of r scheduled jobs
$J_{r'}$ = The set of remaining (n-r) free jobs

Mathematical Development

Consider n jobs say i = 1, 2, 3…,n are processed on six machines A, B, C,D,E and F in the order ABCDEF. A job i (i = 1, 2, 3…,n) has processing time A_i, B_i, C_i, D_i, E_i and F_i on each machine respectively. The mathematical model of the problem in matrix form can be stated as shown in Table 2.

Table 2. The matrix form of flow-shop problem

Jobs	Machine A	Machine B	Machine C	Machine D	Machine E	Machine F
i	A_i	B_i	C_i	D_i	E_i	F_i
1	A_1	B_1	C_1	D_1	E_1	F_1
2	A_2	B_2	C_2	D_2	E_2	F_2
3	A_3	B_3	C_3	D_3	E_3	F_3
4	A_4	B_4	C_4	D_4	E_4	F_4
5	A_5	B_5	C_5	D_5	E_5	F_5
--	--	--	--	--	--	--
n	A_n	B_n	C_n	D_n	E_n	F_n

The objective is to obtain the optimal schedule of all jobs which minimize the total elapsed time using branch and bound technique.

Algorithm of Branch and Bound

Step 1: Calculate

(i) $G_1 = t(J_r, 1) + \sum_{i \in J_r'} A_i + \underbrace{min}_{i \in J_r'}(B_i + C_i + D_i + E_i + F_i)$

(ii) $G_2 = t(J_r, 2) + \sum_{i \in J_r'} B_i + \underbrace{min}_{i \in J_r'}(C_i + D_i + E_i + F_i)$

(iii) $G_3 = t(J_r, 3) + \sum_{i \in J_r'} C_i + \underbrace{min}_{i \in J_r'}(D_i + E_i + F_i)$

(iv) $G_4 = t(J_r, 4) + \sum_{i \in J_r'} D_i + \underbrace{min}_{i \in J_r'}(E_i + F_i)$

(v) $G_5 = t(j_r, 5) + \sum_{i \in J_r'} E_i + \underbrace{min}_{i \in J_r'}(F_i)$

(vi) $G_6 = t(J_r, 6) + \sum_{i \in J_r'} F_i$

Step 2: Calculate
G = max [G_1, G_2, G_3, G_4, G_5, G_6], evaluate G first for the n classes of permutations, for these starting with 1, 2, 3……..n respectively, having labeled the appropriate vertices of the scheduling tree by these values.

Step 3: Now explore the vertex with lowest label. Evaluate G for the (n-1) subclasses starting with this vertex and again concentrate on the lowest label vertex. It was a continuous process, until we reach the end of the tree represented by two single permutations, for which the total work duration is evaluated. Thus we get the optimal schedule of the jobs.

Step 4: We prepare in-out table for the optimal sequence obtained in step 3 and it will give the minimum total elapsed time corresponding to the make span.

RESULTS AND DISCUSSION

The analysis of data presented in Table 1 and Table 2 which comprised of six jobs to six machines is carried out by using the branch and bound algorithm given below.

Step 1: Calculate
For J_1 = (1), then J′ (1) = 1.5, 3.2, 5.1, 7.2, 9.7, 13.9
G_1 = 1.5 + 1456 + 13.3 = 1470.8
G_2 = 3.2 + 1460.9 + 10.8 = 1474.9
G_3 = 5.1 + 1462.8 + 8 = 1475.9
G_4 = 7.2 + 1463.5 + 5.4 = 1476.1
G_5 = 9.7 + 1466 + 2.2 = 1477.9
G_6 = 13.9 + 1472.2 + 0 = 1486.1
G = max [G_1, G_2, G_3, G_4, G_5, G_6],
: . LB (1) = 1486.1

For J_2 = (2), then J′ (2) = 2, 4.5, 7.3, 9.9, 13.1, 15.3
G_1 = 2 + 1455.5 + 12.4 = 1469.9
G_2 = 4.5 + 1460.1 + 10.7 = 1475.3
G_3 = 7.3 + 1461.9 + 8.8 = 1478
G_4 = 9.9 + 1463 + 6.7 = 1479.6
G_5 = 13.1 + 1465.3 + 4.2 = 1482.6
G_6 = 15.3 + 1474.2 + 0 = 1489.5
G = max [G_1, G_2, G_3, G_4, G_5, G_6],
:. LB (2) = 1489. 5

For J_3 = (3), then J′ (3) = 1440, 2880, 4320, 5760, 7200, 8640
G_1 = 1440 + 17.5 + 12.4 = 1469.9
G_2 = 2880 + 22.6 + 10.7 = 2913.3
G_3 = 4320 + 24.7 + 8.8 = 4353.5
G_4 = 5760 + 25.6 + 6.7 = 5792.3
G_5 = 7200 + 28.5 + 4.2 = 7232.7
G_6 = 8640 + 36.4 + 0 = 8676.6

G = max [G_1, G_2, G_3, G_4, G_5, G_6],
:. LB (3) = 8676.4

For J_4 = (4), then J' (4) = 3, 6.2, 9.7, 13.3, 17.1, 22.1
G_1 = 3 + 1454.5 + 12.4 = 1469.9
G_2 = 6.2 + 1459.4 + 10.7 = 1476.3
G_3 = 9.7 + 1461.2 + 8.8 = 1479.7
G_4 = 13.3 + 1462 + 6.7 = 1482
G_5 = 17.1 + 1464.7 + 4.2 = 1486
G_6 = 22.1 + 1471.4 + 0 = 1493.5
G = max [G_1, G_2, G_3, G_4, G_5, G_6],
:. LB (4) = 1493.5

For J_5 = (5), then J' (5) = 8, 20, 33, 46.5, 61.5, 81.5
G_1 = 8 + 1449.5 + 12.4 = 1469.9
G_2 = 20 + 1450.6 + 10.7 = 1481.3
G_3 = 33 + 1451.7 + 8.8 = 1493.5
G_4 = 46.5 + 1452.1 + 6.7 = 1505.3
G_5 = 61.5 + 1453.5 + 4.2 = 1519.2
G_6 = 81.5 + 1456.4 + 0 = 1537.9
G = max [G_1, G_2, G_3, G_4, G_5, G_6],
:. LB (5) = 1537.9

For J_6 = (6), then J' (6) = 3, 6.2, 9.7, 13.5, 17.5, 22.5
G_1 = 3 + 1454.5 + 12.4 = 1469.9
G_2 = 6.2 + 1459.4 + 10.7 = 1476.3
G_3 = 9.7 + 1461.2 + 8.8 = 1479.7
G_4 = 13.5 + 1461.8 + 6.7 = 1482
G_5 = 17.5 + 1464.5 + 4.2 = 1486.2
G_6 = 22.5 + 1471.4 + 0 = 1493.9
G = max [G_1, G_2, G_3, G_4, G_5, G_6],
:. LB (6) = 1493.9

Step 2 and 3: Now branch from J_1 = (1), since it has the minimum Lower Bound (LB).
For J_2 = (1-2), then J' (1-2) = 3.5, 6, 8.8, 11.4, 14.6, 16.8
G_1 = 3.5 + 1454 + 19.1 = 1476.6
G_2 = 6 + 1458.4 + 15.9 = 1480.3
G_3 = 8.8 + 1460 + 12.4 = 1481.2
G_4 = 11.4 + 1460.9 + 8.8 = 1481.1
G_5 = 14.6 + 1462.8 + 5 = 1482.4
G_6 = 16.8 + 1470 + 0 = 1486.8
G = max [G_1, G_2, G_3, G_4, G_5, G_6],
:. LB (1-2) = 1486.8

For J_3 = (1-3), then J' (1-3) = 1441.5, 2881.5, 4321.5, 5761.5, 7201.5, 8641.5
G_1 = 1441.5 + 16 + 13.3 = 1470.8
G_2 = 2881.5 + 20.9 + 10.8 = 2913.2
G_3 = 4321.5 + 22.8 + 8 = 4352.3
G_4 = 5761.5 + 23.5 + 5.4 = 5790.4
G_5 = 7201.5 + 26 + 2.2 = 7229.7

$G_6 = 8641.5 + 32.2 + 0 = 8673.7$
$G = \max [G_1, G_2, G_3, G_4, G_5, G_6]$,
:. LB (1-3) = 8673.7

For J_4 = (1-4), then J′ (1-4) = 4.5, 7.7, 11.2, 14.8, 8, 18.6, 23.6
$G_1 = 4.5 + 1453 + 13.3 = 1470.8$
$G_2 = 7.7 + 1457.7 + 10.8 = 1476.2$
$G_3 = 11.2 + 1459.3 + 8 = 1478.5$
$G_4 = 14.8 + 1459.9 + 5.4 = 1480.1$
$G_5 = 18.6 + 1462.2 + 2.2 = 1483$
$G_6 = 23.6 + 1467.2 + 0 = 1490.8$
$G = \max [G_1, G_2, G_3, G_4, G_5, G_6]$,
:. LB (1-4) = 1490.8

For J_5 = (1-5), then J′ (1-5) = 9.5, 21.5, 34.5, 48.63, 83
$G_1 = 9.5 + 1448 + 13.3 = 1470.8$
$G_2 = 21.5 + 1448.9 + 10.8 = 1481.2$
$G_3 = 34.5 + 1449.8 + 8 = 1492.3$
$G_4 = 48 + 1450 + 5.4 = 1503.4$
$G_5 = 63 + 1451 + 2.2 = 1516.2$
$G_6 = 83 + 1452.2 + 0 = 1535.2$
$G = \max [G_1, G_2, G_3, G_4, G_5, G_6]$,
:. LB (1-5) = 1535.2

For J_6 = (1-6), then J′ (1-6) = 4.5, 7.7, 11.2, 15, 19, 24
$G_1 = 4.5 + 1453 + 13.3 = 1470.8$
$G_2 = 7.7 + 1457.7 + 10.8 = 1476.2$
$G_3 = 11.2 + 1459.3 + 8 = 1478.2$
$G_4 = 15 + 1459.7 + 5.4 = 1480.1$
$G_5 = 19 + 1462 + 2.2 = 1483.2$
$G_6 = 24 + 1467.2 + 0 = 1491.2$
$G = \max [G_1, G_2, G_3, G_4, G_5, G_6]$,
:. LB (1-6) = 1491.2

Now branch from J_{1-2} = (1-2)
For J_3 = (1-2-3), then J′ (1-2-3) = 1443.5, 2883.5, 4323.5, 5763.5, 7203.5, 8643.5
$G_1 = 1443.5 + 14 + 19.1 = 1476.6$
$G_2 = 2883.5 + 18.4 + 15.9 = 2921.4$
$G_3 = 4323.5 + 20 + 12.4 = 4355.9$
$G_4 = 5763.5 + 20.9 + 8.8 = 5793.2$
$G_5 = 7203.5 + 22.8 + 5 = 7231.3$
$G_6 = 8643.5 + 30 + 0 = 8673.5$
$G = \max [G_1, G_2, G_3, G_4, G_5, G_6]$,
:. LB (1-2-3) = 8673.5

For J_4 = (1-2-4), then J′ (1-2-4) = 6.5, 9.7, 13.2, 16.8, 20.6, 25.6
$G_1 = 6.5 + 1451 + 19.5 = 1477$
$G_2 = 9.7 + 1455.2 + 16.3 = 1481.2$
$G_3 = 13.2 + 1456.5 + 12.8 = 1482.5$
$G_4 = 16.8 + 1457.3 + 9 = 1483.1$

G_5 = 20.6 + 1459 + 5 = 1484.6
G_6 = 25.6 + 1465 + 0 = 1490.6
G = max [G_1, G_2, G_3, G_4, G_5, G_6],
:. LB (1-2-4) = 1490.6

For J_5 = (1-2-5), then J′ (1-2-5) = 11.5, 23.5, 36.5, 50, 65, 85.
G_1 = 11.5 + 1446 + 19.1 = 1476.6
G_2 = 23.5 + 1446.4 + 15.9 = 1485.8
G_3 = 36.5 + 1447 + 12.4 = 1495.9
G_4 = 50 + 1447.4 + 8.8 = 1506.2
G_5 = 65 + 1447.8 + 5 = 1517.8
G_6 = 85 + 1450 + 0 = 1535
G = max [G_1, G_2, G_3, G_4, G_5, G_6],
:. LB (1-2-5) = 1535

For J_6 = (1-2-6), then J′ (1-2-6) = 6.5, 9.7, 13.2, 17, 21, 26.
G_1 = 6.5 + 1451 + 19.1 =1476.6
G_2 = 9.7 + 1455 + 15.9 =1480.8
G_3 = 13.2 + 1456.5 + 12.4 =1482.1
G_4 = 17 + 1457.1 + 8.8 = 1482.9
G_5 = 21 + 1458.8 + 5 = 1484.8
G_6 = 26 + 1465 + 0 = 1491
G = max [G_1, G_2, G_3, G_4, G_5, G_6],
:. LB (1-2-6) = 1491

Now branch from J_{124} = (1-2-4)
For J_4 = (1-2-4-3), then J′ (1-2-4-3) = 1446.5, 2886.5, 4326.5, 5766.5, 7206.5, 8646.5.
G_1 = 1446.5 + 11 + 19.5 = 1477
G_2 = 2886.5 + 15.2 + 16.3 = 2918
G_3 = 4326.5 + 16.5 + 12.8 = 4355.8
G_4 = 5766.5 + 17.3 + 9 = 5792.8
G_5 = 7206.5 + 19 + 5 =7230.5
G_6 = 8646.5 + 25 + 0 = 8671.5
G = max [G_1, G_2, G_3, G_4, G_5, G_6],
:. LB (1-2-4-3) = 8671.5

For J_5 = (1-2-4-5), then J′ (1-2-4-5) = 14.5, 26.5, 39.5, 53, 68, 88
G_1 = 14.5 + 1443 + 19.5 =1477
G_2 = 26.5 + 1443.2 + 16.3 = 1486
G_3 = 39.5 + 1443.5 + 12.8 =1495.8
G_4 = 53 + 1443.8 + 9 = 1505.8
G_5 = 68 + 1444 + 5 = 1517
G_6 = 88 + 1445 + 0 =1533
G = max [G_1, G_2, G_3, G_4, G_5, G_6],
:. LB (1-2-4-5) = 1533

For J_6 = (1-2-4-6), then J′ (1-2-4-6) = 9.5, 12.7, 16.2, 20, 24, 29
G_1 = 9.5 +1448 + 73.5 = 1531
G_2 = 12.7 + 1452 + 61.5 =1526.2
G_3 = 16.2 + 1453 + 48.5 = 1517.7

G_4 = 20 + 1453.5 + 35 = 1508.5
G_5 = 24 + 1455 +20 = 1499
G_6 = 29 + 1460 + 0 = 1489
G = max [G_1, G_2, G_3, G_4, G_5, G_6],
:. LB (1-2-4-6) = 1531

Now branch from J_{1246} = (1-2-4-6)
For J_5 = (1-2-4-6-3), then J' (1-2-4-6-3) = 1449.5, 2889.5, 4329.5, 5769.5, 7209.5, 8649.5
G_1 = 1449.5 + 8 +73.5 = 1531
G_2 = 2889.5+ 12 + 61.5 =2963
G_3 = 4329.5 + 13 +48.5 =4391
G_4 = 5769.5 + 13.5 + 35 =5818
G_5 = 7209.5 + 15 + 20 = 7244.5
G_6 = 8649.5 + 20 +0 = 8669.5
G = max [G_1, G_2, G_3, G_4, G_5, G_6],
:. LB (1-2-4-6-3) = 8669.5

For J_5 = (1-2-4-6-5), then J' (1-2-4-6-5) = 17.5, 29.5, 42.5, 56, 71, 91
G_1 = 17.5 + 1440 +7200 = 8657.5
G_2 = 29.5 + 1440 + 5760 = 7229.5
G_3 = 42.5 + 1440 + 4320 = 5802.5
G_4 = 56 + 1440 + 2880 = 4376
G_5 = 71 + 1440 + 1440 = 2951
G_6 = 91 + 1440 + 0 = 1531
G = max [G_1, G_2, G_3, G_4, G_5, G_6],
:. LB (1-2-4-6-5) = 8657.5
The lower bounds for respective jobs are as shown in Table 3.

Table 3. Lower bounds for respective jobs

Node (J_r)	LB (J_r)
(1)	1486.1
(2)	1489. 5
(3)	8676.4
(4)	1493.5
(5)	1537.9
(6)	1493.9
(1-2)	1486.8
(1-3)	8673.7
(1-4)	1490.8
(1-5)	1535.2
(1-6)	1491.2
(1-2-3)	8673.5
(1-2-4)	1490.6
(1-2-5)	1535
(1-2-6)	1491
(1-2-4-3)	8671.5
(1-2-4-5)	1533
(1-2-4-6)	1531
(1-2-4-6-3)	8669.5
(1-2-4-6-5)	8657.5

The optimal sequence is Job1 — Job2 — Job4 — Job6 — Job5 — Job3.

Step 4:
The in-out tableau is for the optimal sequence obtained in step 3 is as shown in Table 4. It will give the minimum total elapsed time corresponding to the make span.

Table 4. In — out tableau

Job	M_1 In — out	M_2 In — out	M_3 In — out	M_4 In — out	M_5 In — out	M_6 In — out
1	0 — 1.5	1.5 — 3.2	3.2 — 5.1	5.1 — 7.2	7.2 — 9.7	9.7 — 13.9
2	1.5 — 3.5	3.5 — 6.0	6.0 — 8.8	8.8 — 11.4	11.4 — 14.6	14.6 — 16.8
4	3.5 — 6.5	6.5 — 9.7	9.7 — 13.2	13.2 — 16.8	16.8 — 20.6	20.6 — 25.6
6	6.5 — 9.5	9.7 — 12.9	13.2 — 16.7	16.8 — 20.6	20.6 — 24.6	25.6 — 30.6
5	9.5 — 17.5	17.5 — 29.5	29.5 — 42.5	42.5 — 56.0	56.0 — 71.0	71.0 — 91.0
3	17.5 — 1457.5	1457.5 — 2897.5	2897.5 — 4337.5	4337.5 — 5777.5	5777.5 — 7217.5	7217.5 — 8657.5

Hence, the total elapsed time is 8657.5mins. The flow shop branching tree diagram is as shown in Fig. 2.

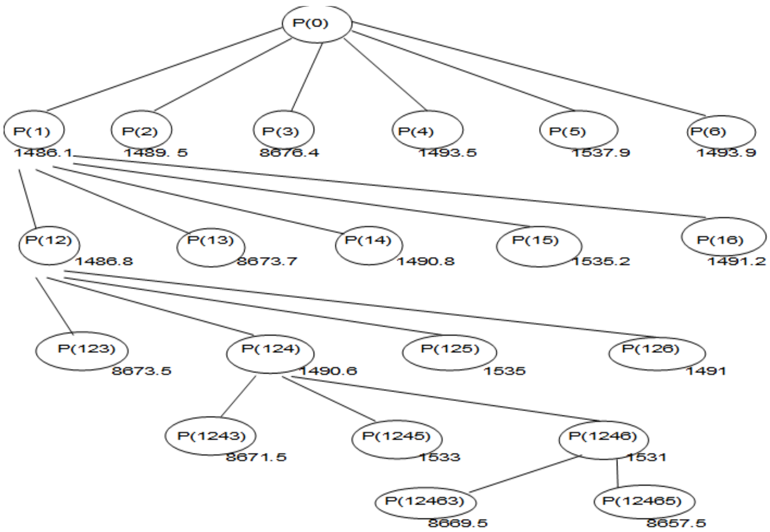

Fig.2 Flow-shop Branching Tree

In this research work, branch and bound approach was used in solving flow-shop problem. We considered flow-shop problem with make-span criterion and the research data were obtained from Mouka Foam Nigeria Limited, Benin City, Nigeria. The data were presented and analyzed. The data comprised the processing time of six jobs (different sizes of foam) to six machines in the production processes of Mouka Foam. It was realized that the optimal sequence of the analyzed data is Job1-Job2-Job4-Job6-Job5-Job3. Finally, it was revealed from the research that the total elapsed time or make-span is 8657.5mins.

CONCLUSION

In this present work, we carried out analysis and application of Branch and Bound Technique for minimizing the make-span of a flow shop. The study mathematical computations and analysis revealed that branch and bound approach is effective tools for solving flow-shop problems. The evaluation has a great contribution to Mouka Foam Nigeria Limited in sequencing its jobs to machines for optimality.

CONFLICT OF INTEREST

There is no conflict of interest associated with this research work.

REFERENCES

Balasubramanian, J. and Grossmann, I. E. (2002). A novel branch and bound algorithm for scheduling flowshop plants with uncertain processing times. *Computers and Chemical Engineering*. 26,41–57

Balasubramanian, J. and Grossmann, I. E. (2004). Approximation to multistage stochastic optimization in multiperiod batch plant scheduling under demand uncertainty. *Industrial and Engineering Chemistry Research*, 43, 3695–3713

Bassett, M. H., Dave, P., Doyle III, F. J., Kudva, G. K., Pekny, J. F., Reklaitis, G. V., Subrahmanyam, S., Miller, D. L. and Zentner, M. G. (1996). Perspectives on Model Based Integration of Process Operations. *Computers Chem Eng*, 20, 821-844.

Ben-Tal, A. and Nemirovski, A. (2000). Robust solutions of linear programming problems contaminated with uncertain data. *Mathematical Programming*, 88, 411–424.

Bonfill, A., Espu˜na, A. and Puigjaner, L. (2005). Addressing robustness in scheduling batch processes with uncertain operation times. *Industrial and Engineering Chemistry Research*. 44, 1524–1534.

Chen, H., Zhou, S., Li, X., & Xu, R. (2014). A hybrid differential evolution algorithm for a two-stage flow shop on batch processing machines with arbitrary release times and blocking. *International Journal of Production Research*,52(19), 5714–5734

Cordeau, J.F., Iori, M., Laporte, G., and Salazar-Gonz´alez, J.J. (2008). A branch-and-cut algorithm for the pickup and delivery traveling salesman problem with LIFO loading. Networks

Francesco, C., and Raffaele, C. (2008). An Additive Branch-and-Bound Algorithm for the Pickup and Delivery Traveling Salesman Problem with LIFO or FIFO Loading,
available at: https://www.researchgate.net/publication/256991198

Jia, Z. and Ierapetritou, M. G. (2004). Short-term scheduling under uncertainty using MILP sensitivity analysis. *Industrial and Engineering Chemistry Research*. 43, 3782–3791

Kallrath, J. (2000). Planning and Scheduling in the Process Industry. *OR Spectrum*. 24, 219-250.

Nawaz, M., Enscore, E. E., and Ham, I. (1983). A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. *Omega*, 11(1), pp.91–95.

Pranzo, M. (2004) .Batch scheduling in a two-machine flow shop with limited buffer and sequence independent setup times and removal times. *European Journal of Operational Research*, 153(3), pp.581–592

Ruland, K.S., and Rodin, E.Y. (1997). The pickup and delivery problem: Faces and branch-and-cut algorithm. *Computer and Mathematics with Applications*, pp.33:1–13