A Modified Branch and Bound Algorithm for Solving Multi-criteria Sequencing Problem

Dr. Adawiya A. Mohmood Al-Nuaimi, Wasfaa E. Ahmed
Department of Mathematics, College of Science, University of Diyala.
Dr.adawiya@Sciences.uodiyala.edu.iq
dradwiyaaali@gmail.com
alwasfaa97@gmail.com

Abstract

To efficiently solve the three criteria (objectives) sequencing problem on a single machine, we propose the modified branch and bound (MBAB) algorithm in this article. The criteria (objectives) are total completion time($\sum C_j$),total lateness($\sum L_j$), and maximum tardiness (T_{max}) . On a single machine , a collection of n independent tasks (jobs) must be sequenced . Tasks(jobs) j (j=1,2,3,...,n) require processing time p_j and require due date d_j. Conclusions for the modified branch and bound (MBAB) algorithm are formulated based on the outcomes of computing tests.

Keyword : Multi-objective Sequencing , Multi criteria , one machine, Pareto optimal solution ,Efficient algorithms.

1. Introduction

Sequencing shows how limited resources are allocated to jobs across time. It is a process of making decisions with the aim of maximizing (minimizing) one or more goals [1].

Finding a running interval on one or more machines that can perform each operation, also known as a job, in such a way that all requirements are satisfied can be used to define the essential sequencing problem. The resulting solution, known as a sequence, minimizes the objective function in the best manner feasible [2]. In this article, the one machine case is involved tasks(jobs) j (j=1,2,3,..,n) request processing times (P_j) due date (d_j),define completion times ($C_j= \sum_{i=1}^{j} p_i$) for particular sequence of task(job),the lateness criterion is $L_j=C_j-d_j$ and the tardiness criterion is $T_j=\max\{d_j-C_j ,0\}$. The simultaneous multi-criteria problems approach takes into account two or more criteria (objectives) at once. This method generates all viable sequences, from which the most advantageous one for the criteria's objective function value is chosen (objective)[3] . The majority of multi-criteria sequencing issues are NP-herd issues [4]. For a multi-criteria sequencing problem with sequencing dependent setup time, In[5] Eren provided an integer programming model heuristic method for minimizing the weighted sum of total completion time, maximum tardiness, and maximum earliness. Systems that involve manufacturing finesse are dynamic, complicated, and stochastic. Production sequences, which are plans that specify the timing of specific controllable activities (such as the operation of jobs or tasks by employees), are created and updated by many manufacturing firms. It is recommended that production sequences coordinate activities (such as the operation of tasks by resources). Production sequences coordinate tasks to boost output temporarily and cut down on overhead. Using a hierarchical approach, the multi-criteria problem in [6] was solved. An exclusive custom-built simulator is

presented by Angelidis et al. [7] in order to facilitate solution methodologies for sequencing problems in intricate assembly lines prevalent in industrial settings .An efficient technique to locate the set of all efficient solutions for the multi-criteria problem $1//f(\sum C_j, \sum T_j, L_{max})$ was provided by [8]. Al-Nuaimi [9] proposed a method into identify efficient solutions for the multi-criteria problem $1//F(\sum C_j , V_{max} , L_{max})$.Numerous criteria in real sequencing problems frequently conflict with one another, as is well known [1].

2. Methods and Materials

Definition: (Hoogeveen,2005)[2]: A practical sol. (sequence) σ is effective (Pareto optimum or non –dominated) , if there is no feasible solution (sequence) π, with regard to the "performance standard f and g , such that both $f(\pi) \leq f(\sigma)$ and $g(\pi) \leq g(\sigma)$ where at least one inequality is strictly defined.

2.1 Branch and Bound method (BAB) [3]

is a broad technique for solving many combinatorial optimization issues. The BAB method is the sequencing solution approach that is employed frequently . This strategy, which can cover an ideal answer(solution) by methodically studying a subset of workable solutions, is a classic illustration of the implicit enumeration methodology. Typically, a search tree with nodes that correspond to this subset is used to describe the operation. A number of new branches emerge from each node of a partially solution, replacing the old one with a set of new, smaller issues(problems) that are mutually incompatible. Two types of branching are frequently employed:

1. Forward branching, in which each job is performed(sequenced) in turn starting at the beginning.
2. Backward branching, in which each job is executed(sequenced) in turn starting at the end.

The BAB approach divides the problem into subsets using a branching procedure and computes bound using a lower bounding procedure in order to minimize an objective function Z for a specific sequencing problem. These techniques are used to eliminate subgroups where no optimal solution has been found. This ultimately yields at least one ideal response(optimal solution). The bounding method is used to determine the lower bound (LB) on each produced sub problem's solution. For each node we determine a (LB), which represents the cost of the sequencing jobs(depending on the the objective function)and the cost of the un sequencing jobs (depending on the derived lower bound) .The upper bound is typically defined as the minimum of the values of all feasible solutions currently found. If this node's value (LB) is less than or equal to the upper limit (UB), then branching from this node. If branching reaches a full sequence of jobs, then this sequence is evaluated and if its value LB is less than or equal the current upper limit (UB), the current upper bound (UB) is reset to take that value. The process is repeated until every node in the search tree has been taken into account, i.e, LB>UB for every node. This UB is an ideal answer(optimal solution) to the problem .

2.2 Complete enumeration method (CEM)[3]

systematically creates sequences while looking for the best outcome. This approach lists every possible feasible, filters out the less desirable sequences, and then only keeps the best options. Even for small size of jobs, it is obvious that employing complete enumeration to search through all feasible sequences to find an optimal sequence is inappropriate.

3. The multi-criteria problem's formulation

The following is the formulation of the multi criteria (multi objective) sequencing problem (P) of total completion time, total lateness , and maximum tardiness.

$\text{Min}\{\sum C_j, \sum L_j, T_{max}\}$ s.t

$C_j=\sum_{i=1}^{j} P_i$ j=1,2,….n

$C_j \geq 0$ j=1,2,….n

$L_j=C_j-d_j$, j=1,2,….n

$T_j=\max\{C_j-d_j,0\}$ j=1,2,….n

$T_j \geq 0$

…problem(P)

3.1 Some fundamental concepts and notation of multi criteria Sequencing

N=Set of jobs

n=Number of jobs

P_j=Processing time for jobsj

d_j=Due date for jobs j

C_j=completion time for jobs j

L_j=lateness for jobs j

$\sum C_j$ =total completion time

$\sum L_j$=total lateness

T_{max}=Maximum tardiness

3.2 Modified Branch and Bound (MBAB) for finding efficient solutions(Pareto optimal) for the problem 1//F($\sum C_j, \sum L_j, T_{max}$) (P)

Step(1):Input date n , p_j and d_j ∀ j=1,2,…..,n .

Step(2):Suppose G=Ø, is the set of efficient solutions, for any sequence σ define $F(\sigma)=(\sum C_{\sigma(j)}, \sum L_{\sigma(j)}, T_{max}(\sigma))$.

Step(3): Set the upper bound (UB) by σ=SPT sequence , where the SPT is sequencing the tasks in non-decreasing order of their processing times . For this order σ, compute F(σ) and the value of $UB=\sum C_{\sigma(j)}+\sum L_{\sigma(j)}+T_{max}(\sigma)$, ∀j=1,2,…,n . Put this UB at the parent node of the search tree .

Step(4): At each node of the search tree of BAB method and for each partial sequence of tasks δ **,** compute a lower bound LB(δ) as follows : LB(δ)= cost of sequencing tasks (δ) , (value of objective function)+ cost of un sequencing tasks obtained by using the two rules SPT for $\sum C_j$ and $\sum L_j$, EDD for T_{max} .

Step(5): Branch from any node with LB≤UB .

Step(6): At the last level of search tree , we get a set of solutions for each application of the definition (2.1), if F(δ) denote the outcome (efficient solution) , then add δ to the set G unless it is dominated by the previously obtained efficient solution in G. Finally we get the set of efficient solutions G .

Step(7): STOP .

Proposition (3.1) : The problem (P) is effectively solved by the SPT sequence .

Proof: First, assume that each processing time is unique. The particular SPT sequence (SPT*) yields the very least amount of $\sum C_j$, hence there isn't a sequence σ≠ SPT* such that

$\sum C_j(\sigma) \leq \sum C_j(SPT^*), \sum L_j(\sigma) \leq \sum L_j(SPT^*)$,and $T_{max}(\sigma) \leq T_{max}(SPT^*)$ ….(3.1)
with at least one strict inequality.
Second, if there are several SPT sequences, jobs with equal processing times are ordered according to the EDD rule; if SPT and EDD are the same, the resulting SPT sequence (SPT*) is used. It should be noted that if σ is an SPT but not an SPT* sequence, it cannot dominate an SPT* sequence because:
$\sum Cj(\sigma) = \sum Cj(SPT*), \sum Lj(SPT*) \leq \sum Lj(\sigma($ and $Tmax(SPT*) \leq Tmax(\sigma)$ …..(3.2)
SPT* sequence is effective as a result.

3.3 Use computational experiments to test issues (problems)

Using MATLAB programming and a Lenovo laptop with 32 GB of RAM, the modified branch and bound (MBAB) algorithm and complete enumeration (CEM) approach are tested on the issue (P) problem to produce efficient solutions. The following test problem is generated: for each job, a discrete uniform distribution-generated integer processing time P_j[1,10]. Additionally, an integer due date dj is created from the discrete uniform distribution [TP(1-TF-RDD/2),TP(1-TF+RDD/2)] where, TP= $\sum_{i=1}^{n} p_i$, for each job j. TF stands for the average tardiness factor, while RDD stands for the relative range of due dates. For both parameters, the values“ 0.2,0.4, 0.6,0.8,1.0” are considered. For each selected value of n, two problems are generated for each of the five values of parameters producing 10 problems . Tables (3.4.1),(3.4.2),(3.4.3),(3.4.4),(3.4.5),(3.4.6),(3.4.7) ,(3.4.8),(3.4.9) compares the average computation time in seconds and average number of effective solutions for the (CEM) and (MBAB) algorithms for the n=3,4,5,6,7,8,9 . Tables(3.4.8)(3.5.1),(3.5.2),(3.5.3),(3.5.4),(3.5.6),(3.5.7),(3.5.8),(3.5.9),
(3.5.10),(3.5.11),(3.5.12) lists the number of efficient solutions and computation durations time in seconds for the (MBAB) method for n=10,....,21.

3.4 Applying (CEM) method and (MBAB) algorithm for 10 examples for n=3,….,9

The (CEM) method and (MBAB) algorithm are applied to the problem (P) in this section to find number of efficient(effective) solutions and time in seconds for both methods, as shown in the following tables :

Table (3.4.1) : number of efficient solutions for the problem (P) by (CEM) and (MBAB) for n=3

n	EX	CEM	Time for(CEM)	MBAB	Time for the (MBAB)
3	1	1	0.0042552	1	0.0112395
	2	1	0.0020187	1	0.0021435
	3	1	0.0008406	1	0.0023719
	4	1	0.0009595	1	0.0021485
	5	1	0.0009729	1	0.0058393
	6	1	0.0012939	1	0.0015904
	7	1	0.0013972	1	0.0015557
	8	1	0.0012837	1	0.0014723
	9	1	0.001134	1	0.0011692
	10	1	0.0009849	1	0.0009849

Table(3.4.2) : number of efficient solutions for the problem (P) by (CEM) and (MBAB) for n=4

n	EX	CEM	Time for(CEM)	MBAB	Time for the (MBAB)
4	1	1	0.0011514	1	0.0014854
	2	1	0.000963	1	0.0011194
	3	1	0.0014883	1	0.01212
	4	1	0.0011113	1	0.0013056
	5	1	0.0014088	1	0.0011797
	6	1	0.0016456	1	0.0016771
	7	1	0.0018407	1	0.0015915
	8	1	0.0017198	1	0.0019828
	9	1	0.0014818	1	0.0014772
	10	1	0.001551	1	0.0013448

Table(3.4.3) : number of efficient solutions for the problem (P) by (CEM) and (MBAB) for n=5

n	EX	CEM	Time for(CEM)	MBAB	Time for the (MBAB)
5	1	4	0.0012527	2	0.0208413
	2	2	0.001248	1	0.0081698
	3	1	0.0019138	1	0.0014201
	4	1	0.0015205	1	0.001046
	5	2	0.001464	1	0.0025432
	6	1	0.0022424	1	0.0010356
	7	1	0.0021954	1	0.0014835
	8	1	0.002203	1	0.0014921
	9	1	0.0021998	1	0.001528
	10	1	0.0021887	1	0.0015221

Table(3.4.4) : number of efficient solutions for the problem (P) by (CEM) and (MBAB) for n=6

n	EX	CEM	Time for(CEM)	MBAB	Time for the (MBAB)
6	1	1	0.0049952	1	0.0017774
	2	2	0.0028786	2	0.006693
	3	2	0.0029069	2	0.0095631
	4	2	0.0030242	1	0.0070967
	5	1	0.0049252	1	0.0013779
	6	1	0.0029646	1	0.0013435
	7	1	0.0048532	1	0.0012522
	8	2	0.0033619	1	0.0034748
	9	1	0.0029225	1	0.0007886
	10	1	0.0034394	1	0.0008216

Table(3.4.5) : number of efficient solutions for the problem (P) by (CEM) and (MBAB) for n=7

n	EX	CEM	Time for(CEM)	MBAB	Time for the (MBAB)
7	1	2	0.0261232	1	0.0034177
	2	3	0.0253276	1	0.003816
	3	3	0.02879	1	0.0023465
	4	2	0.0214614	1	0.001713
	5	1	0.0189496	1	0.0007629
	6	3	0.0177483	1	0.0023369
	7	1	0.030994	1	0.0008197
	8	1	0.0427417	1	0.0007675
	9	1	0.0386885	1	0.0012823
	10	1	0.0287049	1	0.001001

Table(3.4.6) : number of efficient solutions for the problem (P) by (CEM) and (MBAB) for n=8

n	EX	CEM	Time for(CEM)	MBAB	Time for the (MBAB)
8	1	1	0.6067839	1	0.0012914
	2	2	0.617432	1	0.0066071
	3	2	1.2039094	1	0.0026036
	4	4	0.2492752	1	0.0016382
	5	2	0.5233006	1	0.0016858
	6	1	0.4358984	1	0.0009598
	7	1	0.6066162	1	0.0008357
	8	3	0.8111084	1	0.0013161
	9	3	0.3869398	2	0.0020356
	10	2	0.3987259	1	0.0031683

Table(3.4.7) : number of efficient

n	EX	CEM	Time for(CEM)	MBAB	Time for the (MBAB)
9	1	2	107.2537648	1	0.0034998
	2	3	61.1174358	1	0.0054181
	3	2	159.6975659	1	0.0021938
	4	7	59.4225943	4	0.0114412
	5	1	104.3272225	1	0.0008412
	6	1	59.7787033	1	0.0010341
	7	1	80.3390191	1	0.0009136
	8	1	89.8322665	1	0.000825
	9	1	49.4533572	1	0.0008211
	10	1	32.4201326	1	0.002273

3.5 Applying (MBAB) algorithm for 10 examples for n=10,….,21

The (MBAB) algorithm is applied to the problem (P) in this section to find number of efficient(effective) solutions and time in seconds , as shown in the following tables :

Table(3.5.1) : number of efficient solutions by (MBAB) algorithm for n=10

n	EX	MBAB	Time for the (MBAB)
10	1	1	0.0051321
	2	4	0.0427441
	3	2	0.0116663
	4	1	0.0010238
	5	1	0.003729
	6	1	0.0029283
	7	1	0.0011134
	8	1	0.0052489
	9	2	0.0138101
	10	1	0.0018683

Table(3.5.2) : number of efficient solutions by (MBAB) algorithm for n=11

n	EX	MBAB	Time for the (MBAB)
11	1	2	0.0084234
	2	1	0.0020589
	3	1	0.0023231
	4	1	0.002337
	5	1	0.0011556
	6	2	0.0374198
	7	1	0.0010777
	8	1	0.0033191
	9	1	0.0011367
	10	1	0.0011346

Table(3.5.3) : number of efficient solutions by (MBAB) algorithm for n=12

n	EX	MBAB	Time for the (MBAB)
12	1	2	0.166098
	2	2	0.031647
	3	5	0.7495664
	4	2	0.0275717
	5	1	0.0042464
	6	2	0.0479649
	7	1	0.0248549
	8	4	0.2825662
	9	1	0.00096
	10	1	0.0009554

Table(3.5.4) : number of efficient solutions by (MBAB) algorithm for n=13

n	EX	MBAB	Time for the (MBAB)
	1	1	0.0447789
	2	2	0.3861113

13	3	3	0.6144062
	4	1	0.010991
	5	1	0.0025464
	6	3	0.523891
	7	1	0.0008945
	8	1	0.0017173
	9	2	0.1231521
	10	1	0.0009027

Table(3.5.5) : number of efficient solutions by (MBAB) algorithm for n=14

n	EX	MBAB	Time for the (MBAB)
14	1	2	0.2839161
	2	2	0.2308327
	3	2	0.1916798
	4	1	0.0021263
	5	1	0.0046661
	6	1	0.0022118
	7	1	0.0011555
	8	2	0.1393951
	9	1	0.0028769
	10	1	0.0014584

Table(3.5.6) : number of efficient solutions by (MBAB) algorithm for n=15

n	EX	MBAB	Time for the (MBAB)
15	1	2	0.0846995
	2	6	4.6265559
	3	7	1800.62753
	4	2	0.3899676
	5	2	0.0668153
	6	2	5.5490442
	7	1	0.0572077
	8	1	0.0087691
	9	1	0.0007029
	10	1	0.5564029

Table(3.5.7) : number of efficient solutions by (MBAB) algorithm for n=16

n	EX	MBAB	Time for the (MBAB)
16	1	1	0.1645129
	2	2	0.7248574
	3	1	0.0009339
	4	2	0.3374876
	5	2	0.5085743
	6	2	5.6131855
	7	3	1145.873402

	8	2	0.3975424
	9	1	0.3329552
	10	2	28.4223979

Table(3.5.8) : number of efficient solutions by (MBAB) algorithm for n=17

n	EX	MBAB	Time for the (MBAB)
17	1	2	1.0682606
	2	1	0.0009817
	3	2	6.5133897
	4	2	7.2148197
	5	1	0.0007793
	6	3	40.093975
	7	2	6.1338896
	8	2	0.2928579
	9	2	0.5972298
	10	1	0.0007789

Table(3.5.9) : number of efficient solutions by (MBAB) algorithm for n=18

n	EX	MBAB	Time for the (MBAB)
18	1	4	644.5415792
	2	4	32.4633992
	3	7	1813.811823
	4	6	161.2490641
	5	2	12.1718774
	6	1	13.4227455
	7	1	0.0007291
	8	1	2.6814404
	9	1	0.067685
	10	1	0.0007254

Table(3.5.10) : number of efficient solutions by (MBAB) algorithm for n=19

n	EX	MBAB	Time for the (MBAB)
19	1	2	6.3493364
	2	2	12.2968955
	3	4	1804.63072
	4	3	487.4574864
	5	9	1838.034032
	6	1	0.0008773
	7	1	0.0020992
	8	2	1.0152138
	9	1	0.0007863
	10	1	0.4945295

Table(3.5.11) : number of efficient solutions by (MBAB) algorithm for n=20

n	EX	MBAB	Time for the (MBAB)
20	1	10	1806.771498
	2	2	35.9439358
	3	4	717.4744555
	4	7	1805.598234
	5	9	1816.466427
	6	3	132.1290082
	7	6	1800.564883
	8	1	0.0008801
	9	2	884.9911181
	10	1	0.0008477

Table(3.5.12) : number of efficient solutions by (MBAB) algorithm for n=21

n	EX	MBAB	Time for the (MBAB)
21	1	6	1815.221083
	2	8	1803.506271
	3	7	1805.905807
	4	8	1813.846792
	5	5	1803.263282
	6	1	1.1722857
	7	1	2.3183063
	8	2	1.6256162
	9	2	0.5609131
	10	1	12.2504904

3.6 Conclusion

In this article , we propose a modified branch and bound (MBAB) algorithm to find the number of efficient solution for the multi-criteria sequencing problem(P) on one machine .As a result of our experimentation with test problems to compare the algorithm (MBAB) with complete enumeration method (CEM)we conclude that the(MBAB) algorithm performs quite well for the multi-criteria problem(P) of size n=3,4,…,9 . The (CEM) consumes much time , since it seeks all solutions. Thus the (CEM) solves the multi-criteria problem(P) of size 4P to 9 . Also, from our experimentation results, we show that the (MBAB) algorithm solves the problem (P) efficiently for n=3,4,..,22. The article presented here contributes to the field of multi-criteria sequencing problems . For future research , we recommend the experimentation with the following problems :

1- 1//F($\sum C_j^2$, $\sum T_j$, $\sum L_j$)
2- 1//F ($\sum C_j$, $\sum T_j$, $\sum E_j$, $\sum u_j$)
3- 1//F($\sum C_j$, $\sum u_j$,$\sum T_{max}$)

Reference

[1] Pinedo ,M. Scheduling : theory review algorithms and systems. Springer ,New york,USA.,ISBN-13:9780387789347, 2008pages:671.

[2] Hoogeveen,J.A.,2005.Invited review of multi criteria scheduling. Eur.J.Res.,167:592-623.

[3] Khames A.I.2022. Solving Multiobjective Sequencing Problem on one Machine .M.SC. university of Diyala Colloge of Scines , Dep of Mathmatices .

[4] Akande, S., A.E. OIuleye and E.O. Oyetunji . Reducibility of Some Multi criteria scheduling problems to bi criteria scheduling Problem. Proceedings of the 2014 International conference on"industrial Engineering and operations management bail, January"7-9, 2014, Bail, Indonesia, pp:642-651

[5] Eren,T.,2007. A multi criteria scheduling with sequence dependent Setup times. Appl.Math.Sci.,1:2883-2894.

[6] AI-Nuaimi, A.A.M., Solving a multi-criteria problem in a Hierarchical method. International Journal of Nonlinear Analysis and Applications, 13(1):2671-2674.2022.

[7] Angelidis,E.,D.Bohn and O Rose, 2013.A simulation too for complex assembiy lines with multi-skilled resources . Proceedings of the 2013 Winter simulation conference :simulation :making decisions in a complex word, December 08-11,2013, IEEE Press, Washington, DC.,USA.,ISBN:978-1-4799-2077-8,PP:2577-2586.

[8] AI-Nuaimi, A.A.M. An Algorithm for solving three criteria" "scheduling problem on a single machine, lnt .J. Algricult. Stat. Sci . Vol.14,No.12018, pp.271- 273.

[9] AI-Nuaimi A.A.M.. A proposed algorithm to find Efficient Solutions For Multi -criteria problem, Effect of " Engineering and Applied Scieces2019,14(2):5547-5549