Journal of Mathematical Problems, Equations and Statistics

E-ISSN: 2709-9407
P-ISSN: 2709-9393
JMPES 2022; 3(1): 72-77

www.mathematicaljournal.com
Received: 12-01-2022
Accepted: 16-02-2022

Dr. Ambika Bhambani
M.A, M.Phil, Ph.D.,
Associate Professor, Department of Mathematics, Jesus and Mary College, University of Delhi, Chanakyapuri, New Delhi, India

Corresponding Author:
Dr. Ambika Bhambani
M.A, M.Phil, Ph.D.,
Associate Professor, Department of Mathematics, Jesus and Mary College, University of Delhi, Chanakyapuri, New Delhi, India

Scheduling models for economical & budget travel with optimal rental cost

Dr. Ambika Bhambani

Abstract
Whether one is planning for a short trip or as a full-time or long-term traveller, an entirely new set of problems and complications arises into the already difficult task of managing ones finances. In this article, the fundamentals of money management and financial planning for travellers, have been covered based on Mathematical Model of Scheduling.
The problem is solved with the objective of obtaining a schedule of the given jobs which minimizes a certain performance measure, Total Rental Cost, for three Machines.

Keywords: Traveller, rental cost, scheduling, elapsed time, idle time, branch and bound, optimal sequence

Introduction
The tourism industry, also known as the travel industry, is linked to the idea of people travelling to other locations, either domestically or internationally, for leisure, social or business purposes. Tourism is intangible product as it is related to feeling and experience. It cannot be seen, tasted, measured before they are received.
Knowing how much money the trip will cost, formulating a budget, and sticking to it and always having a plan B are some of the most important aspects that you need to consider before you go on extensive trips.
One should Shortlist travel destinations based on preferences and budget. After one make a decision about travel destination, one can now use the financial information and the length of one's trip to come up with a rough estimate of the overall costs of one's trip.
Before departing for one's travels, it is extremely important to get a good grasp of your current financial situation. Otherwise, financial issues might start to unravel while you are on the road, and it is going to be extremely difficult.
Apart from financial planning, managing ones money efficiently can help one to avoid unnecessary expenses, by adopting simple money management tips. A lot of unexpected things can happen while traveling and more often than not, things can change along the way. The best thing to do is to always have a backup plan in case something goes wrong.
By planning a holiday in advance, one can use the time in hand to hunt for deals and discounts. One can save under the following heads by grabbing the best deals:

Transportation
Keep an eye out for airline ticket deals to your preferred destination for economical travel. *Midweek flights* are generally less expensive. One can sign up for travel alerts from flight and travel websites to grab such deals. You can opt for converting some of your air miles to get a discount on your airfares.

For local sightseeing and excursions
Avoid renting a car or taxi if the city has a good public transport system. Ask locals about the distance to destinations and the best way to reach there. Buy passes/ reserve before. It is cost effective. Local tours are cheaper at the place and costlier online.

Food
Do thorough research to identify places where you can get the best deals on food.

Stay

Compare the price of your hotel room on online portals to book it at the best rates. Check if the hotel you opt for provides free airport pick up and drop. One can also explore non-traditional accommodations to save on vacation expenses. Explore options like hostels, vacation rentals, etc. Staying in less popular places could be a good option once you have done in-depth research from safety and accessibility perspective. The accommodation is required for those travelling to a different location and staying overnight.

This paper attempts on minimizing the total cost to be spent for the entire trip, based on the concept of the Mathematical model, under the heading minimizing total rental cost, focusing on transportation cost, (Both for long distance is for destination & also for local sightseeing), Food & Accommodations (Like, Holiday cottages or hotels or, camps) & Here *Machines* have to be interpreted as Resources, may be Transportation, Food & Accommodations.

And *Jobs* to be interpreted in terms of number of destinations to be covered in the trip.

The Total Rental Cost to be defined as the Cost spent, during the entire trip, including Transportation, Food & Stay.

Minimizing total rental cost

The only possibility for the traveller is to take these machines (here transportation, accommodation & food), on rent. Food package may be included, along with accommodation, with the stay, in a particular destination.

The following renting policies generally exist :

Policy 1: All the machines are taken on rent at the same time and are returned at the same time.

Policy 2: All the machines are taken on rent at the same time but are returned as and when they are no longer required.

Policy 3: All the machines are taken on rent as and when they are required and are returned as and when the requirement is over.

In the case of two-machines problem, as far as Policy 1 and Policy 2 are concerned, the sequence which minimizes the elapsed time will be the optimal sequence which minimizes the total rental cost. While, for more than two-machines case, only corresponding to Policy 1, the sequence which minimizes the elapsed time, will be the optimal sequence which minimizes the total rental cost.

In this paper, Policy 2 is adopted, best suited for travel cost to be minimized, for three-machines case. It is assumed that all the machines are taken on rent at the same time, i.e., when the processing of the first job of the sequence is started on the first machine and each machine is returned as and when it is no longer required (i.e., when the processing of the last job gets completed on that machine. It has to be applied or repeated, for each destination stay seperately.

The objective under the situation is to obtain a sequence which minimizes the total cost due to hiring of machines.

The Objective Function considered in this paper, is to obtain a sequence of destinations under the situation is to obtain a sequence of destinations which minimizes the total cost due to hiring of machines.

The Total Rental Cost, could be applied to individual travellers or in group) under the headings

1. Three – Stage General Case Problem.
2. Three Stage Specially - Structured Problems.

It deals with those flow shop sequencing problems, in which situations, the machines are taken on rent, for processing n – jobs (Destinations).

It considers the situation of n – jobs (n - destinations), 3 – machines problem (travelling cost, food & accommodation) are taken on rent and where the rental costs of the machines are different. The problem is solved by applying the Branch and Bound technique and the optimal sequence is obtained, which minimizes the total rental cost, covering all destinations.

Section 1: Three - Stage General Case Problem

This section obtains a sequence which minimizes the total rental cost of the machines in three-machines flow shop sequencing problem, under Policy 2.

Situations when the machines are taken on rent for processing of jobs with the rental cost of the machines as different and the objective being that of minimization of total rental cost of the machines, the problem is solved by applying the Branch and Bound technique.

The object of minimizing total rental cost is achieved by minimizing the sum of the idle times of the machine when multiplied by its respective machine rental cost, the sum taken over all the machines. But since the idle time of the first machine is always zero, hence, for three-machines, the problem is to find the sequence which minimizes the sum taken over the last two - machines, of the idle time of the machine multiplied by its corresponding rental cost, or equivalently, which minimizes the sum, taken over the last two machines, of the completion time of the last job on the machine multiplied by its corresponding rental cost.

In the Branch and Bound technique, given in this section, the expressions for determining the lower bound of any partial sequence is obtained. The node with the minimum lower bound is branched further. Branching is continued till a complete schedule is obtained with value less than or equal to all the other nodes, i.e., minimum total rental cost of the optimal sequence cannot be more than the bounds (total rental costs) with any other unbranched node. It has also been shown that substituting the rental cost of second machine to be zero, the problem becomes as of minimization of elapsed time. The sequence which minimizes the elapsed time also minimizes the rental cost and the lower bound in the rental situation match with the bound obtained by Lomnicki [6] multiplied by the rental cost of the third machine.

The method is supported by numerical example at the end.

Let the n - jobs require processing over three - machines A, B and C in the order A, B, C.

Notations

X_i = the processing time of job i on machine X (X = A, B, C).

J_r = any predefined partial schedule of r – jobs.

J'_r = complement schedule of J_r (i.e., the schedule of the remaining jobs which are not in J_r).

$Z(J_r, X)$ = the time at which the last job of the schedule J_r is completed on machine X (X = A, B, C).

C_x = the rental cost per unit time of machine X (X = B, C).

LB $[J_r]$ = lowest possible rental cost corresponding to partial sequence J_r, irrespective of any schedule of jobs in J'_r.

LB $[J_r, X]$ = the minimum possible completion time of the schedule on machine X with schedule J_r, followed by any schedule of jobs in J'_r.

The problem is to obtain the optimal sequence which minimises LB $[J_r]$. The expression for lower bound for a partial schedule is obtained by assuming that, machines do not wait for any job of J'_r and jobs of J'_r do not have to wait for processing on the remaining machines. The expressions for LB [J_r], LB [J_r, B], LB [J_r, C] are given below:

LB [J_r] = LB [J_r, B] × C_B + LB [J_r, C] × C_C

LB [J_r, B] = Max [Z (J_r, A) + Σ A i + Min B i , Z (J_r, B) + Σ B i]

i ∈ J'_r. i ∈ J'_r i ∈ J'_r

= Max [g_1, g_2] (say)

LB [J_r, C] =

Max [Z (J_r, A) + Σ A i + Min (B i + C i) , Z (J_r, B) + Σ B i + Min C i ,

i ∈ J'_r. i ∈ J'_r i ∈ J'_r i ∈ J'_r

Z (J_r, C) + Σ C i]

i ∈ J'_r

= Max [G_1, G_2, G_3] (Say)

Particular Case: When $C_B = 0$

LB [J_r] = LB [J_r, C] × C_C

This implies that the rental cost corresponding to any partial schedule is directly proportional to completion time of that schedule on machine C (elapsed time).

Thus the sequence which minimizes rental cost will also minimize elapsed time.

Hence, the same sequence can minimize total elapsed time as well as total rental cost provided the rental cost of machine B is zero.

Example 1: Consider the 4 – jobs, 3 - machines problem with processing times, as in Table 1 (a):

Table 1 (a)

Machines → Jobs ↓	A	B	C
1	6	4	8
2	2	5	7
3	3	9	2
4	7	4	2

And the rental costs per unit time for machines B and C as 7 units & 2 units respectively,

i.e., $C_B = 7$, $C_C = 2$.

The lower bound for partial schedule $J_r = 1$ (J'_r = 2, 3, 4) is computed below:

LB [1, B] = Max [Z (1, A) + Σ A i + Min B i , Z (1, B) + Σ B i]

i = 2,3,4 i = 2,3,4 i = 2,3,4

= Max [g_1, g_2]

= Max [6 + (2 + 3 + 7) + Min (5, 9, 4)]

= Max [22, 28] = 28

LB [1, C] = Max [Z (1, A) + Σ A i + Min (B i + C i),

i = 2,3,4 i = 2,3,4

Z (1, B) + Σ B i + Min C i,

Z (1, C) + Σ C i]

i = 2,3,4 i = 2,3,4

i = 2,3,4

= Max [6 + (2 + 3 + 7) + Min (5 + 7, 9 + 2, 4 + 2), 10 + (5 + 9 + 4) + Min (7, 2, 2),

18 + (7 + 2 + 2)]

= Max [18 + 6, 28 + 2, 29] = 30

Thus,

LB [1] = LB [1, B] X CB + LB [1, C] x Cc

= 28 x 7 + 30 x 2 = 196 + 60 = 256

Similarly, the lower bounds for partial schedules $J_r = 2$, $J_r = 3$ and $J_r = 4$ are 220, 237 and 265 units respectively.

Minimum of lower bounds is for vertex 2.

The lower bound for the partial schedule 21 (J'_r = 34) is computed below:

LB [21, B] = Max [Z (21, A) + Σ A i + Min B i, Z (21, B) + Σ B i]

i = 3, 4 i = 3, 4. i = 3, 4

= Max [8 + (3 + 7) + Min (9, 4), 12 + (9 + 4)]

= Max [18 + 4, 25] = Max [22, 25] = 25

LB [21, C] = Max [Z (21, A) + Σ A i + Min (B i + C i),

i = 3, 4. i = 3, 4

Z (21, B) + Σ B i + Min C i, Z (1, C) + Σ C i]

i = 3, 4 i = 3, 4 i = 3, 4

= Max [8 + (3 + 7) + Min (9 + 2, 4 + 2), 12 + (9 + 4) + Min (2, 2), 22 + (2 + 2)]

= Max [(8 + 10 + 6, 12 + 13 + 2, 22 + 4]

= Max [24, 27, 26] = 27

Thus, LB [21] = LB [21, B] x CB + LB [21, C] x Cc

= 25 x 7 + 27 x 2 = 175 + 54 = 229

Similarly, for the partial schedules J_r = 23 and J_r = 24, the lower bounds are 224 and 240 units respectively.

Continuing in this way, the Branch and Bound technique is applied for evaluations of relevant lower bounds and the scheduling tree is formed.

Both are shown respectively in Table 1 (b) and Figure 1 below:

Table 1 (b)

J_r	For Machine B g_1	g_2	For Machine C G_1	G_2	G_3	LB[J_r, B]	LB [J_r, C]	LB [J_r]
1	22	28	24	30	29	28	30	256
2	22	24	24	26	26	24	26	220
3	22	25	24	27	31	25	31	237
4	22	29	29	31	30	29	31	265
21	22	25	24	27	26	25	27	229
23	22	24	24	26	28	24	28	224
24	22	26	29	28	26	26	29	240
2314	–	–	–	–	–	24	30	228
2341	–	–	–	–	–	24	32	232

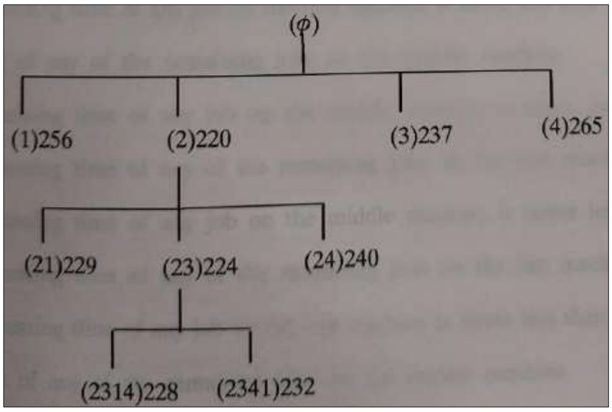

Figure 1: Scheduling tree

Hence, the optimal sequence is 2 – 3 - 1 – 4 with minimum rental cost as 228 units.

In the example, it is to be noted that the optimal sequence for minimizing the total rental cost is 2 - 3 - 1 – 4, while the sequence which minimizes the total elapsed time is 2 – 1 - 3 – 4.

Section 2: **Three - Stage Specially - Structured Problems**

This section deals with the situation of n – jobs, three - machines problem when the processing times of the jobs on machines follow some well - known restrictions and the object is to minimize the total rental cost of the machines, when all the machines are assumed to be equally - costly. The restrictions on the processing times of the jobs considered by Johnson [5] and Szwarc [7] are being relaxed in this section. The relaxed restrictions on the processing times is any one of the following:

Processing time of any job on the first machine is never less than the processing time of any of the remaining jobs on the middle machine.

[It will hold, as transportation time is not less than time spent on food].

Processing time of any job on the middle machine is never less than the processing time of any of the remaining jobs on the first machine.

[Will not hold].

Processing time of any job on the middle machine is never less than the processing time of any of the remaining jobs on the last machine.

[Will not hold].

Processing time of any job on the last machine is never less than the processing time of any of the remaining jobs on the middle machine.

[Will hold, as Accommodation stay time will never be less than the time spent on food].

In Travelling, generally Case 1 & 4, above will be relevant.

In case only one / two machines are to be taken on rent, then the same procedure can be applied assuming the rental cost of the non - rental machines as zero.

Algorithm with illustration of numerical example is considered for each two types of relevant special case.

Let n - jobs require processing over three-machines A, B and C in the order A, B, C. If A_i, B_i, C_i (i = 1, 2,., n) be the processing time of job i on machines A, B and C respectively and Z (i, A), Z (i, B), Z (i, C) be the completion times of job i (i = 1, 2,., n) on machines A, B and C respectively, then optimal sequence is to be obtained which minimizes

[Z (n, B) + Z (n, C)].

The following two relevant special cases out of four, have been dealt herewith, redefined as Case 1 & Case 2:

Case 1: Min $A_i \geq$ Max B_j

Case 2: Min $C_i \geq$ Max B_j; $\forall$ i & j; i $\neq$ j

Case 1: Min $A_i \geq$ Max B_j; $\forall$ i & j; i $\neq$ j

i.e., $A_i \geq B_j$; $\forall$ i & j ; i $\neq$ j;

The following algorithm yields the optimal sequence:

Algorithm 1:

Step 1: Read A_i, B_i, C_i (i = 1, 2,., n).

Step 2: Compute

$G_i = A_i + B_i$ and $H_i = B_i + C_i$ (i = 1, 2,., n).

Step 3: Obtain the sequence by applying Johnson's two - machines algorithm on machines G and H. Call this sequence as S_1.

Step 4: Let T be the processing time of the last job of S_1 on machine B.

Step 5: Obtain other sequences, if any, by shifting that job to the end of the sequence S_1 which has the processing time on machine B less than T.

Step 6: Obtain the sum of the completion times on the last two machines through the sequence S_1 and through all other sequences obtained in Step 5 .

Step 7: The sequence which gives the minimum sum of completion times is an optimal sequence.

Example 2

Consider the 4 - jobs 3 - machines problem with processing times, as in Table 2:

Table 2

Machines → Jobs ↓	A	B	C
1	9	1	7
2	5	4	6
3	7	4	3
4	6	2	8

Note that $A_i \geq B_j$; $\forall$ i & j; i $\neq$ j;

Since $G_i = A_i + B_i$ and $H_i = B_i + C_i$; $\forall$ i (Step 2),

Therefore, Table 3 below, provides the processing times, on machines G and H.

Table 3

Machines → Jobs	G	H
1	10	8
2	9	10
3	11	7
4	8	10

From Table 3, the sequence S_1 for 2 - machines problem (corresponding to machines G and H) is 4 - 2 – 1 – 3. (Step 3). Since corresponding to job 3, T= B_3 = 4 (Step 4) and other B_i 's, i.e., B_1, B_2 and B_4 have processing times 1, 4 and 2 respectively (which are less than or equal to 4), therefore, the sequences 4 – 2 – 3 - 1; 4 – 1 - 3 - 2 and 2 - 1 – 3 – 4 are also need to be considered (Step 5) to obtain the minimum of the sum of the completion times of the last job on machines B and C. For determining the optimal sequence, these four sequences are enumerated completely and completion times In - Out Tables, for each of the four sequences, are shown as in tables, *viz.*,Table 4, Table 5, Table 6 and Table 7 respectively (Step 6):

[1] 4 – 2 – 1 – 3

Table 4

Machines → Jobs ↓	A In - Out	B In - Out	C In – Out
4	0 - 6	6 - 8	8 - 16
2	6 - 11	11 - 15	16 - 22
1	11 - 20	20 - 21	22 - 29
3	20 - 27	27 - 31	31 - 34

Thus, total sum = 31 + 34 = 65 units.

[2] 4 – 2 – 3 - 1

Table 5

Machines → Jobs ↓	A In-Out	B In-Out	C In–Out
4	0 - 6	6 - 8	8 - 16
2	6 - 11	11 - 15	16 - 22
3	11 - 18	18 - 22	22 - 25
1	18 - 27	27 - 28	28 - 35

Thus, total sum = 28 + 35 = 63 units.

[3] 4 – 1 – 3 – 2

Table 6

Machines → Jobs ↓	A In - Out	B In – Out	C In – Out
4	0 - 6	6 - 8	8 - 16
1	6 - 15	15 - 16	16 - 23
3	15 - 22	22 - 26	26 - 29
2	22 - 27	27 - 31	31 - 37

Thus, total sum = 31 + 37 = 68 units.

[4] 2 – 1 – 3 – 4

Table 7

Machines → Jobs ↓	A In - Out	B In - Out	C In – Out
2	0 - 5	5 - 9	9 - 15
1	5 - 14	14 - 15	15 - 22
3	14 - 21	21 - 25	25 - 28
4	21 - 27	27 - 29	29 - 37

Thus, total sum = 29 + 37 = 66 units.
The sum of the completion times of the last job on machines B and C for the sequences 4 - 2 – 1 – 3; 4 – 2 – 3 - 1; 4 – 1 - 3 - 2 and 2 - 1 – 3 – 4 are
65, 63, 68 and 66 units respectively.
Minimum of total sums = Min [65, 63, 68, 66] = 63 units (Step 7) and this value is minimum corresponding to the sequence 4 – 2 – 3 – 1.
It is to be noted that, in the example, the sequence which minimizes the total rental cost is 4 – 2 – 3 – 1, which does not minimize the total elapsed time.

Note: In example 2, the completion times of the jobs for the sequences,
4 - 2 – 1 – 3; and 4 – 2 – 3 – 1 are obtained as in Table 8 & Table 9 below respectively.

Table 8

Machines → Jobs ↓	A In - Out	B In - Out	C In – Out
4	0 - 6	6 - 8	8 - 16
2	6 - 11	11 - 15	16 - 22
1	11 - 20	20 - 21	22 - 29
3	20 - 27	27 - 31	31 - 34

Table 9

Machines → Jobs ↓	A In - Out	B In - Out	C In – Out
4	0 - 6	6 - 8	8 - 16
2	6 - 11	11 - 15	16 - 22
3	11 - 18	18 - 22	22 - 25
1	18 - 27	27 - 28	28 - 35

Since the idle time of any machine is obtained by subtracting the processing times of all the jobs on that machine from the time the last job is completed on that machine, therefore, from table 8, for the sequence 4 – 2 – 1 – 3;
Idle time on machine A = 27 - (9 + 5 + 7 + 6) = 0 units.
Idle time on machine B = 31 - (1 + 4 + 4 + 2) = 20 units
Idle time on machine C = 34 - (7 + 6 + 3 + 8) = 10 units
Total idle time = 30 units.
And from Table 9, for the sequence 4 - 2 - 3 – 1;
Idle time on machine A = 27 - (9 + 5 + 7 + 6) = 0 units.
Idle time on machine B = 28 - (1 + 4 + 4 + 2) = 17 units.
Idle time on machine C = 35 - (7 + 6 + 3 + 8) = 11 units.
Total idle time = 28 units.
Since for the sequence 4 – 2 – 1 – 3, the sum of the idle times is 30 units and the total elapsed time is 34 units and for the sequence 4 – 2 – 3 – 1, the sum of the idle times is 28 units and the total elapsed time is 35 units, thus, the sequence which minimizes the idle time on the last machine or which minimizes the total elapsed time need not minimize the sum of the idle times of all the machines.
Thus, the sequence which minimizes the total rental cost does not always minimize the total elapsed time.

Case 2: Min $C_i \geq$ Max B_j; $\forall$ i & j; i $\neq$ j
$C_i \geq B_j$; $\forall$ i & j; i $\neq$ j

The following algorithm yields the optimal sequence:

Algorithm 2 :

Steps 1: Read A_i, B_i, C_i (i = 1, 2,., n).
Step 2: Compute $G_i = A_i + B_i$ and $H_i = B_i + C_i$ (i = 1, 2., n).
Step 3: Obtain sequence S_1 by applying Johnson's rule for A_i ' s and B_i 's
Step 4: Obtain sequence S_2 by applying Johnson's rule for G_i's and H_i ' s
Step 5: If $S_1 \equiv S_2$, then either of them is an optimal sequence. If not, proceed to Step 6.
Step 6: Enumerate all the sequences to find the optimal sequence.

However, Theorem 1 can eliminate many of the sequences which cannot be optimal.

Theorem 1: If job i precedes job j, in the adjacent positions, in both the sequences, then any sequence in which job i immediately follows job j, cannot be an optimal sequence.

Example 3
Consider the 4 – jobs, 3 – machines problem with processing times as in Table 10 below:

Table 10

Machines → Jobs ↓	A	B	C
1	5	1	8
2	7	3	9
3	9	2	11
4	3	4	10

Note that $C_i \geq B_j;\ \forall\ i\ \&\ j;\ i \neq j$
Since $G_i = A_i + B_i$ and $H_i = B_i + C_i$ (i = 1, 2., n),
Therefore, Table 11 gives the processing times, on machines G and H,

Table 11

Machines → Jobs ↓	G	H
1	6	9
2	10	12
3	11	13
4	7	14

From Table 10, the sequence for 2 - machines problem (corresponding to machines A and B) is 4 – 2 - 3 - 1 (Step 3) and from Table 11, the sequence for 2 - machines problem (corresponding to machines G and H) is 1 - 4 – 2 - 3 (Step 4).
Let S_1 = 4 – 2 - 3 – 1 and S_2 = 1 - 4 – 2 - 3
Since. $S_1 \equiv S_2$, therefore, comparing these two sequences, since job 4 precedes job 2 and job 2 and job precedes job 3 adjacently, therefore, all these sequences in which job 4 follows job 2 and job 2 follows job 3 adjacently can be dropped out, which can never be the optimal sequences in the enumeration of all possible sequences to obtain the optimal one (By Theorem 1).
Out of 4 ! = 24 possible sequences, the ten sequences, *viz*., 1 - 2 - 4 – 3, 1 – 3 – 2 - 4,
1- 4 - 3 - 2, 2 - 4 - 3 – 1, 2 - 4 - 1 - 3, 3 – 1 - 2 – 4, 3 – 2 - 4 - 1, 3 - 2 - 1 - 4, 4 - 1 – 3 - 2,
4 - 3 - 2 - 1 can be dropped out, since these sequences can never yield to an Optimal sequence. The rest of the sequences can be enumerated separately to yield an Optimal sequence which minimizes the sum of completion times of the last job on machines B and C (Step 6).
It can be verified, hat here in this example, finally the optimal sequences after enumeration are 1 - 4 - 2 - 3 and 4 – 2 – 3 - 1 and the sum of the completion times of the last job on machines B and C for each sequence is 70 units.

Remarks: Although no simple algorithm is ever being presented for the situation where the processing time of any job on the last machine is never less than the processing time of any of the remaining jobs on the middle machine (Case 4), still Algorithm 2 can reduce the number of sequences to be searched for optimality. Many examples of jobs ranging from 4 to 7 are solved and the Table 12 below gives the average number of sequences obtained. In many cases due to Step 5 of the algorithm, the optimal sequence is obtained without any tabulation.

Table 12

Jobs ↓	Average Number of Sequences to be Searched
4	15
5	65
6	467
7	3121

Conclusion
Just add an adventurous attitude, as a travel enthusiast and with plenty of smiles - one go a long way - and that about covers it all, in a budget travel with minimum trip cost !

References
1. Bagga. Sequencing in a Rental Situation, Journal of Canadian Operations Research Society. 1969;7:152-153.
2. Bagga, Ambika. Minimizing Rental Costs in Three – Machines Sequencing Problem, published in, Journal of Indian Association for Productivity, Quality, and Reliability. 1996;21(1):73-77.
3. Bagga, Ambika. Three Machines Sequencing Problem in Rental Situations: Special Cases. 1999;15(1):1-14. IJOMAS, ISSN- 2394-9988.
4. Ignall, Schrage. Applications of the Branch and Bound Technique to some Flowshop Scheduling Problems, Operations Research. 1965;13(3):400-412.
5. Johnsons. Optimal Two and Three Stage Production Schedules with Set – Up – Times Included, Naval Research Logistics. 1954;1:61-78.
6. Lomnicki. A Branch and Bound Algorithm for the Exact Solution of the Three – Machine Scheduling, Operational Research. 1965;16:89-100.
7. Szwarc W. Mathematical Aspects of the 3 × n Job – Shop Sequencing Problem, Naval Research Logistics. 1974;21:145-153.