I J C R M
International Journal of Contemporary Research In Multidisciplinary

Research Article

Python-Based Approach to Minimise Rental Cost in Two-Stage Flow Shop Scheduling with Transportation Time

Sakshi [1*], Aditi [2], Deepak Gupta [3]
[1, 2] PG Student, Department of Mathematics, Maharishi Markandeshwar University, Mullana Haryana, India
[3] Professor & Head, Department of Mathematics, Maharishi Markandeshwar University, Mullana Haryana, India

Corresponding Author: * Sakshi

DOI: https://doi.org/10.5281/zenodo.21281593

Abstract

The present paper presents a straightforward approach to solving a two-stage flow shop scheduling problem with Python. The aim is to identify a desirable strategy that incurs minimal total cost of renting machines, based on a given rental policy, and taking into account the time needed to transport between the two stages. This is done by a simple algorithm developed and run in Python. Also, a case study is given to demonstrate the effectiveness of the proposed method.

Manuscript Information

- **ISSN No:** 2583-7397
- **Received:** 12-05-2026
- **Accepted:** 30-06-2026
- **Published:** 09-07-2026
- **IJCRM:**5(SP1); 2026: 106-111

- **Plagiarism Checked:** Yes
- **Peer Review Process:** Yes

How to Cite this Article

Sakshi, Aditi, Gupta D. Python-Based Approach to Minimise Rental Cost in Two-Stage Flow Shop Scheduling with Transportation Time. Int J Contemp Res Multidiscip. 2026;5(SP1):106-111.

Access this Article Online

www.multiarticlesjournal.com

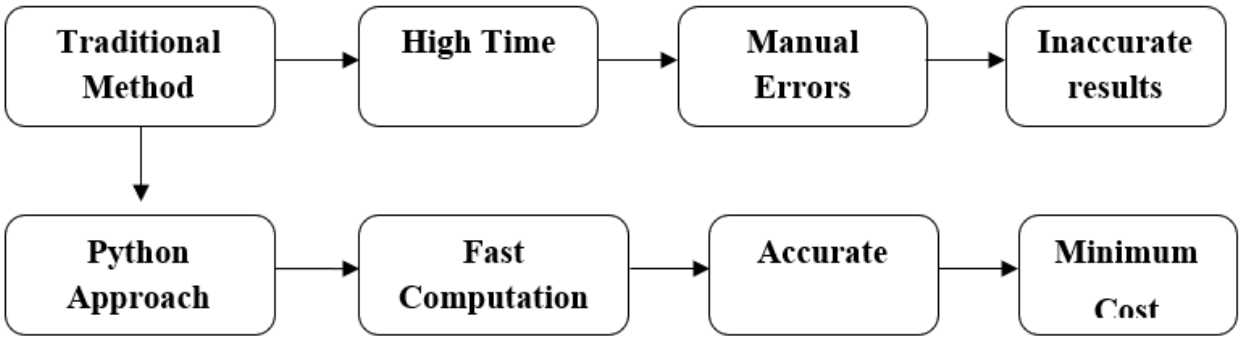
KEYWORDS: Flowshop Scheduling, Elapsed time, Johnson's Rule, Optimal Sequence, Rental Cost, Transportation time.

1. INTRODUCTION

In a flow shop scheduling problem, several jobs are processed on a number of machines in a set order. From a commercial point of view, businesses look to produce goods at the lowest cost to sell at a competitive price and still be profitable. Scheduling is the allocation of resources for the jobs in a given time to get better performance. In shop scheduling, there are several jobs to be processed in a certain sequence on a number of machines. Reducing production costs allows firms to lower prices, increase customer satisfaction, and increase competitiveness. When the cost of production decreases, companies are able to sell their products at a lower price, make their customers happier and become more competitive in the market. To managers, this assists them in making smarter decisions regarding how they can plan production and utilize resources. This implies that they will be able to maintain the control of operating cost and enhance productivity in general. Economically, the production systems that are good assist in managing the limited resources better. When firms reduce unnecessary expenses such as additional machine rental fees and wastage of transportation time, the firms operate more easily and deliver more output.

This paper's use of Python provides an efficient approach for implementing Johnson's Rule in two-stage flow shop scheduling. It enables efficient and fast calculation of the optimal job sequence. Python reduces the need for manual calculations that can be tedious and time-consuming. It also minimises the possibilities of human error while calculating the job sequence and makespan. The automation method enhances productivity by giving quick and reliable results within seconds. It is particularly useful for solving industrial scheduling problems that require speed, accuracy and cost-effectiveness. A pioneering approach in flow shop scheduling is the algorithm developed by Johnson [1]. This result was extended by Bagga [2] by designing an optimal algorithm for two-stage flow shop problems with various constraints and criteria. The study to minimize the makespan for two and three stage flow shop scheduling problem was further developed by Jackson [3], Bellman [4], Mitten [5], Cambell [6], Baker [7] and many other researchers. Gupta, J.N.D. [8] developed an algorithm to find the optimal sequence for specially structured flow shop scheduling problem. This work has been extended by many researchers such as Gupta and S. Sharma [9][10], T. P. Singh & D. Gupta [11], D. Gupta and S. Bala [12], S. B. & P. S. D. Gupta [13], Hosseini, L., Tang, S., Mookerjee, V., & Sriskandarajah, C. [14], Gupta, K., Gupta, D., & Goel, S.[15], AlOtaibi, M., El-Rayes, K., & Altuwaim, A. [16] by including additional elements such as related probabilities, job weightages, and other important factors.

This paper, on the other hand, considers the problem of two stages with the objective of minimizing the rental cost of machines, with a consideration for the time taken to transport parts from one stage to the other.



2. PRACTICAL SITUATION

There are many practical scenarios where someone (or a company) needs to complete some task but does not have the required machines or cannot afford to purchase them. In those situations, they rent the equipment. For instance, construction companies rent cranes, diggers and mixers instead of purchasing them. This can help save costs and provides access to new equipment. Python helps in such cases to solve scheduling problems better. It helps in determining the optimal sequence of jobs on rented machines to reduce the total rental cost, while considering the time required to transport jobs from one stage to another. It's also useful if the machines are located far apart or if the jobs are transferred from one stage to another. It saves time, effort and reduces errors and gives quick and accurate results.

3. NOTATIONS

S: Sequence of jobs 1, 2, 3, n

S_v: Sequence obtained by applying Johnson's procedure, where v = 1, 2, 3, etc.

M_r Machine r, where r = 1, 2

b_{ij}: The time required for the ith job in machine Mr.

t_i (S_v): Completion time of the ith job in the sequence S_v on machine M_r.

$T_{i,j\rightarrow k}$: Transportation time of the ith job from machine Mr to machine Mk

$(UT)_i(S_v)$: Usage time of machine Mr, if it starts processing jobs at time Li (Sv), where i = 1, 2

$R(S_v)$: Rental cost for the sequence S_v

4. RENTAL POLICY

Machines are rented as needed and returned when no longer needed. The first machine is rented at the start of the job. The second machine is rented after the first job is completed on the first machine and it is moved to the second machine. The third machine is rented when the first job is completed on the second machine and sent to the third machine.

5. PROBLEM FORMULATION

Let some job i (i = 1, 2, ..., n) be processed on two machines, M_r (r = 1, 2, 3), under a specified rental policy P. Let b_{ij} be the processing time for the ith job on the jth machine and $T_{i,j\to k}$ be the transportation time for the ith job from the jth machine to the kth machine.The mathematical model of the problem in matrix form can be stated as follows:

Table 5.1: Mathematical model

	Machine M_1	**$T_{i,1\to2}$**	**Machine M_2**
i	**b_{i1}**		**b_{i2}**
1	b_{11}	$T_{1,1\to2}$	b_{12}
2	b_{21}	$T_{2,1\to2}$	b_{22}
3	b_{31}	$T_{3,1\to2}$	b_{32}
.	.	.	.
.	.	.	.
.	.	.	.
n	b_{n1}	$T_{n,1\to2}$	b_{n2}

Our goal is to find the sequence {Sk} that minimizes the rental costs for both machines, while taking into account the transportation time between the stages

6. ALGORITHM: Step 1: Introduce two fictitious machines, U and V, with processing times U_i and V_i as follows: $U_i = b_{i1} + T_{i,1\to2}$
$V_i = b_{i2} + T_{i,1\to2}$

Step 2: Apply Johnson's (1954) rule to obtain the sequence S_1.

Step 3: Generate sequences S_2, S_3, ..., S_v from S_1 by keeping the jobs that have a higher processing time than the first job of S1 in the same order, and leaving the rest in their original sequence.

Step 4: Create an in-out table for each sequence generated in step 3.

Step 5: Calculate the total completion time tij(S_v) for each sequence Sk (k = 1, 2, ..., r).

Step 6: Determine the utilization time $(UT)_2(S_v)$ for the second machine using the formula: $(UT)_2(S_v)) = tij(S_v) - b_{11}(S_v)$

Step 7: Compute the rental cost as: $R(S_v) = \Sigma\, b_{i1} \times C_1 + (UT)_2(S_v) \times C_2$

Step 8: Find the minimum $R(S_v)$ for k = 1, 2, ..., n. Let the sequence S_p be the one with the minimum rental cost $R(S_p)$. This sequence Sp is the optimal sequence with the corresponding rental cost R(Sp).

7. PYTHON PROGRAM

```
# Two-Stage Flow Shop Scheduling with Transportation Time and Rental Cost
from math import inf

def print line(widths):
 print ("+" + "+". join(["-"*(w+2) for w in widths]) + "+")

def print row (row, widths):
print ("|" + "|". join ([f" {str(val). ljust(w)} " for val, w in zip (row, widths)]) + "|")

# ---------- INPUT ----------
n = int (input ("Enter number of jobs: "))
jobs = list (range (1, n+1))
a = []
b = []
t = []

for i in range(n):
 print (f"\nJob {i+1}")
 a.append(int (input ("Processing time on M1 (ai): ")))
 t.append(int (input ("Transportation time (Ti): ")))
 b.append(int (input ("Processing time on M2 (bi): ")))

C1 = float (input ("\nEnter rental cost per unit time for Machine 1: "))
C2 = float (input ("Enter rental cost per unit time for Machine 2: "))
# ---------- STEP 1: FICTITIOUS MACHINES ----------
g = [a[i] + t[i] for i in range(n)]
h = [b[i] + t[i] for i in range(n)]
print ("\nFictitious Machine Table")
widths = [5,5,5]

print_line(widths)
print_row(["Job","g","h"], widths)
print_line(widths)
for i in range(n):
   print_row([i+1, g[i], h[i]], widths)
print_line(widths)

# ---------- STEP 2: JOHNSON RULE ----------
remaining = set(range(n))
sequence = [None]*n
left = 0
right = n-1
while remaining:
   min_val = inf
   min_job = None
   on_g = True
   for i in remaining:
      if g[i] < min_val:
         min_val = g[i]
         min_job = i
```

```
            on_g = True
        if h[i] < min_val:
            min_val = h[i]
            min_job = i
            on_g = False
    if on_g:
        sequence[left] = min_job
        left += 1
    else:
        sequence[right] = min_job
        right -= 1
    remaining.remove(min_job)

johnson_seq = [j+1 for j in sequence]
print ("\nJohnson Sequence:", johnson_seq)

# ---------- STEP 3: GENERATE N SEQUENCES ----------
def shift_sequences(base):
    seqs = []
    for i in range(len(base)):
        new_seq = [base[i]] + [x for j,x in enumerate(base) if j != i]
        seqs.append(new_seq)
    return seqs
sequences = shift_sequences(sequence)
# ---------- STEP 4: COMPUTE RENTAL COST ----------
def evaluate(seq):
    m1_start = []
    m1_finish = []
    time = 0
    for j in seq:
        start = time
        finish = start + a[j]
        m1_start. append(start)
        m1_finish. append(finish)
        time = finish

    UT1 = time
    m2_start = []
    m2_finish = []
    prev_finish = 0
    for idx, j in enumerate(seq):
        arrival = m1_finish[idx] + t[j]
        start = max (arrival, prev_finish)
        finish = start + b[j]
        m2_start. append(start)
        m2_finish. append(finish)
        prev_finish = finish
    UT2 = m2_finish [-1] - m2_start [0]

    RC = C1 * UT1 + C2 * UT2

    return UT1, UT2, RC, m1_start, m1_finish, m2_start, m2_finish
best = None
best_seq = None
for idx, seq in enumerate (sequences, 1):
    UT1, UT2, RC, m1s, m1f, m2s, m2f = evaluate(seq)

    seq_jobs = [j+1 for j in seq]
    print (f"\nSequence S{idx}: {seq_jobs}")
    widths = [5,8,8,8]
    print_line(widths)
    print_row(["Job","M1","M2"], widths)
    print_line(widths)

 for i, j in enumerate(seq):
        m1_range = f"{m1s[i]}-{m1f[i]}"
        m2_range = f"{m2s[i]}-{m2f[i]}"
        print_row ([j+1, m1_range, m2_range], widths)
    print_line(widths)
    print (f"UT1 = {UT1}, UT2 = {UT2}, Rental Cost = {RC}\n")
    if best is None or RC < best:
        best = RC
        best_seq = seq_jobs
print ("=====================================")
print ("Optimal Sequence:", best_seq)
print ("Minimum Rental Cost:", best)
print ("=====================================")
```

8. NUMERICAL ILLUSTRATION

Suppose we have 5 jobs that need to be processed on 2 machines, say M_1 and M_2. We also know the time required to complete each job on each machine. The rental cost per unit time of the machines is 7 units for M_1 and 10 units for M_2. We need to schedule the jobs in the best possible way to minimise the total rental cost of the machines.

Table 8.1: Numerical Illustration

Jobs	Machine M_1	$T_{i,1\to2}$	Machine M_2
1	6	2	8
2	11	5	8
3	9	2	14
4	11	3	7
5	10	6	8

Solution: Result according to our Python implementation code:

Step 1: Input Data

Firstly, the number of jobs, their processing times, Transportation time and their rental cost C_1 and C_2 on Machine A and Machine B are taken.

Step 2: Fictitious machine table

Table 8.2: Fictitious machine table

Jobs	Machine G	Machine H
1	8	10
2	16	13
3	11	16
4	14	10
5	16	14

Step 3: Apply Johnson's (1954) rule to obtain the sequence S_1.

S_1: < 1, 3, 5, 2, 4>

Step4: Obtain sequences S_k ; k= 2,3,4,5 from S_1as
S_2: < 3, 1, 5, 2, 4>
S_3: < 5, 1, 3, 2, 4>
S_4: < 2, 1, 3, 5, 4>
S_5: < 4, 1, 3, 5, 2>

Step 5: - In-out tables for all the sequences

Table 8.3: Sequence S_1 In-Out

Jobs	Machine M_1	Machine M_2
1	0-6	8-16
3	6-15	17-31
5	15-25	31-39
2	25-36	41-49
4	36-47	50-57

$(UT)_1(S_1) = 47$, $(UT)_2(S_1) = 49$, Rental Cost = 819.0

Table 8.4: Sequence S_2 In-Out

Jobs	Machine M_1	Machine M_2
3	0-9	11-25
1	9-15	25-33
5	15-25	33-41
2	25-36	41-49
4	36-47	50-57

Table 8.5: Sequence S_3 In-Out

Jobs	Machine M_1	Machine M_2
5	0-10	16-24
1	10-16	24-32
3	16-25	32-46
2	25-36	46-54
4	36-47	54-61

$(UT)_1(S_3) = 47$, $(UT)_2(S_3) = 45$, Rental Cost = 779.0

Table 8.6: Sequence S_4 In-Out

Jobs	Machine M_1	Machine M_2
2	0-11	16-24
1	11-17	24-32
3	17-26	32-46
5	26-36	46-54
4	36-47	54-61

$(UT)_1(S_4) = 47$, $(UT)_2(S_4) = 45$
Rental Cost = 779.0

Table 8.7: Sequence S_5 In-Out

Jobs	Machine M1	Machine M2
4	0–11	14–21
1	11–17	21–29
3	17–26	29–43
5	26–36	43–51
2	36–47	52–60

$(UT)_1(S_5) = 47$, $(UT)_2(S_5) = 46$, Rental Cost = 789.0
Optimal Sequence: <5, 1, 3, 2, 4>
Minimum Rental Cost: 779.0

9. CONCLUSION

In this paper, a Python-based approach was applied to minimize the total rental cost of machines in a two-stage flow shop scheduling problem with transportation time. The numerical illustration demonstrates the effectiveness of the method by identifying the optimal job sequence and the corresponding minimum rental cost under the given rental policy. The results show that the proposed approach efficiently handles complex and lengthy calculations, significantly reducing manual effort and the possibility of human error. Moreover, the solution is obtained within a very short computational time. These findings confirm that the method not only achieves the stated objective of cost minimization but also provides a practical and reliable tool for solving real-life scheduling problems.

10. FUTURE SCOPE

- The proposed work is applicable to more complex scheduling problems, such as scheduling problems with three or more stages, with a Python-based approach.
- The model can be enhanced by incorporating real world factors like the probability of successfully performing tasks, time spent in shifting items and time spent in preparing machines.
- Future research would consider the machine failure and scheduled maintenance work to ensure the model that is more appropriate to the real situation.
- The approach can be modified to process jobs as they come and rather than assuming that all jobs are available at the beginning.
- It might be useful to include priority levels or different weights for jobs to meet various customer needs.

REFERENCES

1. Johnson SM. Optimal two and three stage production schedule with set up times included. Naval Res Logist Q. 1954;1(1):61-68.
2. Bagga PC. Sequencing in a rental situations. J Can Oper Res Soc. 1969;7:152-153.
3. Jackson JR. An extension of Johnson's results on job lot scheduling. Naval Res Logist Q. 1956;3:201-203.
4. Bellman R. Mathematical aspects of scheduling theory. J Soc Ind Appl Math. 1956;4:168-205.
5. Mitten LG. Sequencing n jobs on two machines with arbitrary time lags. Manage Sci. 1959;5:293-298.
6. Campbell HA, Dudek RA, Smith ML. A heuristic algorithm for n-jobs, m-machines sequencing problem. Manage Sci. 1970;16:630-637.
7. Baker KR. *Introduction to Sequencing and Scheduling*. New York: John Wiley & Sons; 1974.
8. Gupta JND. Optimal schedule for specially structured flow shop. Naval Res Logist. 1975;22(2):255-269.
9. Gupta D, Sharma S, Bala S. n × 2 specially structured flow shop scheduling with transportation time to minimize the rental cost of machines. Int J Math Arch. 2012;3(2):627. Available from: www.ijma.info.
10. Gupta D, Sharma S. Minimizing rental cost under specified rental policy in two stage flow shop, the processing time associated with probabilities including break-down interval

and job-block criteria. Eur J Bus Manag. 2011;3(2):85-103. Available from: www.iiste.org.
11. Singh TP, Gupta D. Minimizing rental cost in two stage flow shop, the processing time associated with probabilities including job block. Reflection DES ERA. 2005;1(2):107-120.
12. Gupta D, Bala S, Singla P. Specially structured two stage flow shop scheduling to minimize the rental cost, processing time each associated with probabilities including weightage of jobs. Int J Res Manag. 2012;2(2).
13. Bala S, Singla P, Gupta D. Minimizing rental cost for specially structured two-stage flow shop scheduling, processing time, setup time associated with probabilities including weightage of jobs. Adv Appl Sci Res. 2012;3(5):2906-2911.
14. Hosseini L, Tang S, Mookerjee V, Sriskandarajah C. A switch in time saves the dime: A model to reduce rental cost in cloud computing. Inf Syst Res. 2020;31(3):753-775.
15. Gupta K, Gupta D, Goel S. Rental cost minimization for two stage parallel machine FSSP with transportation time. In: *Sustainability in Digital Transformation Era: Driving Innovative & Growth*. Boca Raton (FL): CRC Press; 2024. p. 281-285.
16. AlOtaibi M, El-Rayes K, Altuwaim A. Optimizing the renovation scheduling of leased residential buildings to minimize total project cost. J Constr Eng Manag. 2021;147(7):04021074.