ISE 5431 / ISE 4431: Stochastic Operations Research

**Assembly Line Balancing Problem Optimization**

Fall 2021

Instructor: Dr. Nasim Nezamoddini

**Submitted By Group 04:**

Alexander Thaden

Ra'fat Alrawashdeh

VaraAre

**Submitted On:** Dec 7, 2021

**Project Topic:**

This project will be conducted on a production line from ABC’s assembly facility. At this facility, ABC produces product X through several tasks and machines as shown below. We will be using skills learned in ISE 5431 to optimize this assembly line based on the data found.

**Abstract:**

The objective of this study is to optimize the production line tasks for some known production facility, facility ABC, in order to maximize the efficiency of the assembly line at any given time with some probability of a line operator taking a vacation. To formulate this optimization, we utilized a dual stage optimization process to minimize the number of assembly workstations based on cycle time and task time. Our model determined that the Assembly line efficiency for scenario 1 is 91.5% (no worker vacation) while efficiency decreases in scenario 2 to 84.8% efficiency (worker took vacation). This decrease in efficiency highlights the impact that operator' skill level has on the assembly line layout. Based on our results, we propose facility ABC adopts two different assembly line setup scenarios based on whether or not an operator has vacation scheduled. This recommendation will help facility ABC maximize efficiency on any given day.

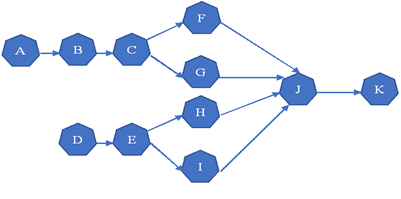
**Introduction:**

In order to perform the study requested by this project, we needed to find some real-world production data to utilize for our optimization. Our group was unable to collect data from a production facility regarding assembly line tasks and task times due to the need for an NDA with any production facility. As a result, we created an example facility, ABC’s assembly facility. This allowed us to create an assembly line from scratch in a way that would allow our group to investigate and optimize the line using a 2-stage optimization. All input data for each task was randomly generated for this problem. The purpose of this investigation is to understand the best way for production facility ABC to optimize their assembly line setup in order to minimize the number of workstations required while ensuring all tasks are completed within the allotted cycle time. The uncertainty in this problem is operator vacation, there is a 50% probability that some operator will be on vacation. This problem is an example of a critical optimization problem that is faced by production facilities worldwide. If we can optimize the order of these assembly line tasks, we can run the assembly facility at optimal efficiency with minimal waste.

**Problem Description:**

How can ABC optimize the production tasks in their assembly line to minimize the number of workstations required while ensuring all tasks are completed within the allotted cycle time, when line workers have some known probability for taking a vacation?

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Task | A | B | C | D | E | F | G | H | I | J | K |
| Minutes | 45 | 11 | 9 | 50 | 15 | 11 | 10 | 9 | 8 | 7 | 8 |



**Methodology:**

The probability of a technician who is working on tasks F, G, H, J and K taking a vacation is 0.5 (P=0.5). When a technician takes a vacation, there is an inherent need for other workers to cover the additional tasks. A table showing the different task times can be found below:

|  |  |  |
| --- | --- | --- |
| Task | Scenario 1 (p=0.5) | Scenario 2 (p=0.5) |
| F | 11 | 13 |
| G | 10 | 14 |
| H | 9 | 15 |
| I | 8 | 16 |
| J | 7 | 9 |
| K | 8 | 15 |

Using this known probability for a worker taking a vacation, the model will be formulated for two scenarios, operator taking a vacation, or not taking a vacation. Once we consider the uncertainty of an operator taking vacation, we can introduce the two stages formulation:

|  |  |
| --- | --- |
| Decision Variable |  |
|  | 1 if task I assigned to station j or 0 if not |
|  |  |
| Objective | |
|  | minimize the number of workstations |
|  |  |
| Parameter | |
| ct | the cycle time which is 50 min |
| Tti | task time of i |
|  |  |
| Set | |
| I | Set of tasks {A,B,C,D,E,F,G,H,I,J,K} |
| J | Set of stations {1 - 11}, the maximum number of station can be is equal to number of tasks |
| h | Set of task' precedence of to task i |

Using the above information, we are able to formulate the single stage optimization problem as shown below:

|  |  |  |  |
| --- | --- | --- | --- |
| Minimize | = | | |
| Subject to | = 1 | ; | ϵ I |
| ≤ ct \* | ; | ϵ J |
| ≤ | ; | ϵ I , ϵ pred\_i\_j(i) |
|  | is binary (0,1) | ; | ϵ I , ϵ J |
|  | is binary (0,1) | ; | ϵ J |

**Optimization Constraints:**

**Constraint 1:** make sure that each task i assigned only to one workstation j from the available workstations

**Constraint 2:** make sure that the summation time of tasks i assigned to workstation j will be equal or less to the workstation j multiply by cycle time

**Constraint 3:** make sure that the summation of tasks h which are the precedence tasks to task i assigned to workstation j will be equal or less than the summation of tasks I to make sure that no tasks of h will happen after task i.

After setting up our optimization problem for Scenario 1, we are able to set up the dual stage problem to solve. Two stages will be used as below:

|  |  |
| --- | --- |
| Decision Variable |  |
|  | 1 if task I assigned to station j |
|  |  |
| Objective | |
|  | minimize the number of workstations |
|  |  |
| Parameter | |
| ct | the cycle time which is 50 min |
| Tti | task time of i |
|  |  |
| Set | |
| I | Set of tasks {A,B,C,D,E,F1,G1,H1,I1,J1,K1, F2,G2,H2,I2,J2,K2} |
| J | Set of station {1 - 11}, the maximum number of station can be is equal to number of tasks |
| h | Set of task' precedence of to task i |
| I’ | Set of Scenario 1’ tasks {A,B,C,D,E,F1,G1,H1,I1,J1,K1} |
| I” | Set of Scenario 2’ tasks {A,B,C,D,E,F2,G2,H2,I2,J2,K2} |

Using the above information, we are able to formulate the dual stage optimization problem for Scenario 2 as shown below:

|  |  |  |  |
| --- | --- | --- | --- |
| Minimize | = | | |
| Subject to | = 1 | ; | ϵ I |
| ≤ ct \* | ; | ϵ J |
| ≤ ct \* | ; | ϵ J |
| ≤ | ; | ϵ I , ϵ pred\_i\_j(i) |
|  | is binary (0,1) | ; | ϵ I , ϵ J |
|  | is binary (0,1) | ; | ϵ J |

By solving the dual stage optimization, we will be able to conclude what the optimal number of workstations is, and the best feasible task-assigned-workstation configuration that can be achieved through this optimal setup.

**Results:**

Here, we can see that five workstations should be needed to deal with the uncertainties raised up for the both scenarios as shown on the table below as well as the best route configuration for each scenario. Also, balancing efficiency of the assembly line = total tasks time / (no. of workstation \* cycle time), so for scenario 1 will be 183 / (4\*50) = 0.915 or 91.5% and for the scenario 2 will be 212/(4\*50) = 0.848 or 84.8%

|  |  |  |  |
| --- | --- | --- | --- |
|  | two stages | | |
| no. of stations | 5 | | |
| Configuration | **Task** | | **Station** |
| **Scenario 1** | **Scenario 2** |
| A | A | WS 1 |
| D | D | WS 2 |
| B, C, E, I | B,C,E | WS 3 |
| F,G,H,J,K | F,G,I | WS 4 |
| - | H, J,K | WS 5 |

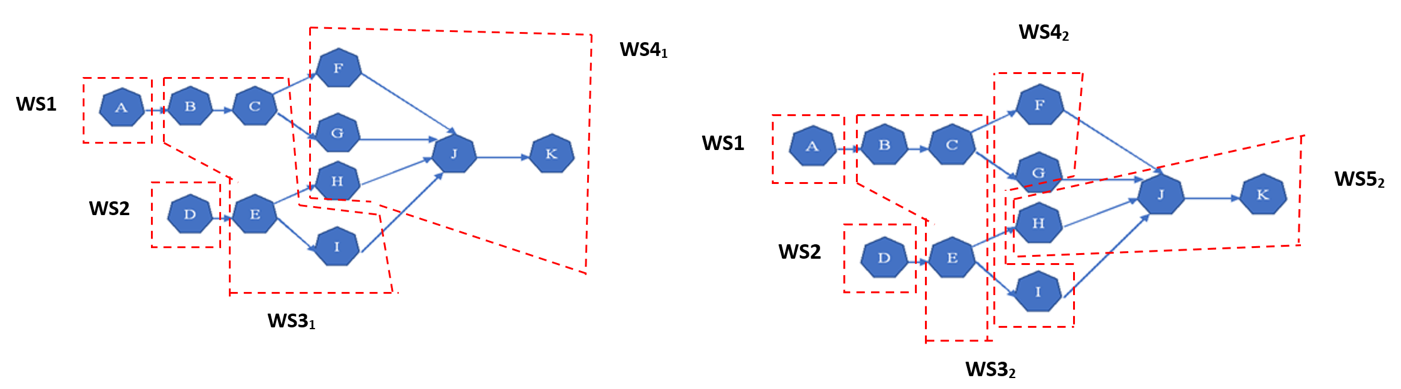


Figure 1: Workstation’s configuration results for the both scenario

**Conclusion:**

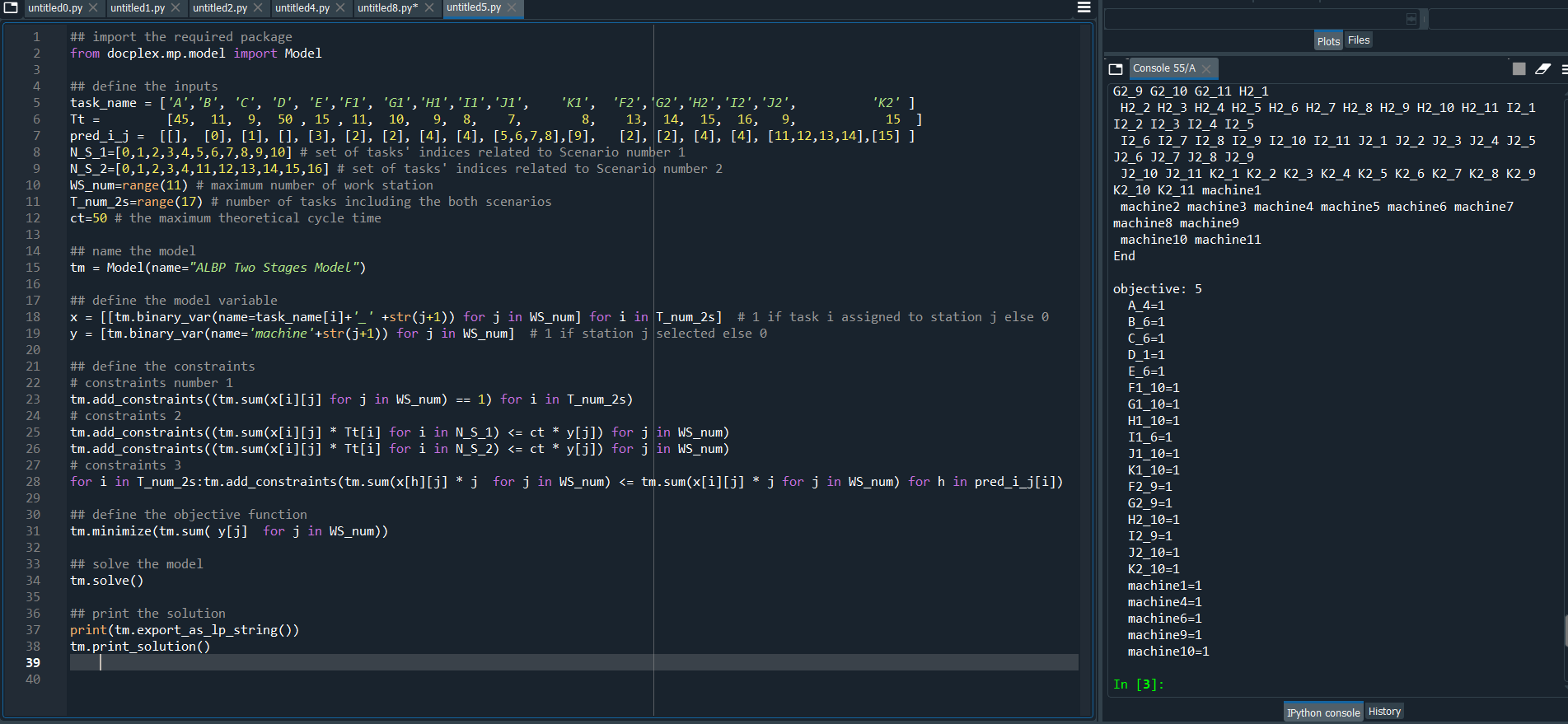
ALBP allocates all tasks to a certain number of stations while optimizing our objective without violating certain constraints such as variability of task time due to operator skill levels. The Assembly line efficiency for scenario 1 is 91.5% while decreasing for scenario 2 which is 84.8%, this shows that the operator skill level has a significant influence on the assembly line layout. The suggestion configuration helps the decision maker to re-route the assembly workstations layout in a way that easily goes from scenario to scenario without doing a major layout modification and to be ready for all uncertainties as well as to introduce a training program for the operators to have the same skill levels.

**Appendices**

Appendix A1

1. ## import the required package
2. from docplex.mp.model import Model
4. ## define the inputs
5. task\_name = ['A','B', 'C', 'D', 'E','F1', 'G1','H1','I1','J1','K1', 'F2','G2','H2','I2','J2', 'K2' ]
6. Tt = [45, 11, 9, 50 , 15 , 11, 10, 9, 8, 7, 8, 13, 14, 15, 16, 9, 15 ]
7. pred\_i\_j = [[], [0], [1], [], [3], [2], [2], [4], [4], [5,6,7,8],[9], [2], [2], [4], [4], [11,12,13,14],[15] ]
8. N\_S\_1=[0,1,2,3,4,5,6,7,8,9,10] # set of tasks' indices related to Scenario number 1
9. N\_S\_2=[0,1,2,3,4,11,12,13,14,15,16] # set of tasks' indices related to Scenario number 2
10. WS\_num=range(11) # maximum number of workstation
11. T\_num\_2s=range(17) # number of tasks including the both scenarios
12. ct=50 # the maximum theoretical cycle time
14. ## name the model
15. tm = Model(name="ALBP Two Stages Model")
17. ## define the model variable
18. x = [[tm.binary\_var(name=task\_name[i]+'\_' +str(j+1)) for j in WS\_num] for i in T\_num\_2s] # 1 if task i assigned to station j else 0
19. y = [tm.binary\_var(name='machine'+str(j+1)) for j in WS\_num] #1 if station j selected else 0
21. ## define the constraints
22. # constraints number 1
23. tm.add\_constraints((tm.sum(x[i][j] for j in WS\_num) == 1) for i in T\_num\_2s)
24. # constraints 2
25. tm.add\_constraints((tm.sum(x[i][j] \* Tt[i] for i in N\_S\_1) <= ct \* y[j]) for j in WS\_num)
26. tm.add\_constraints((tm.sum(x[i][j] \* Tt[i] for i in N\_S\_2) <= ct \* y[j]) for j in WS\_num)
27. # constraints 3
28. for i in T\_num\_2s:tm.add\_constraints(tm.sum(x[h][j] \* j for j in WS\_num) <= tm.sum(x[i][j] \* j for j in WS\_num) for h in pred\_i\_j[i])
30. ## define the objective function
31. tm.minimize(tm.sum( y[j] for j in WS\_num))
33. ## solve the model
34. tm.solve()
36. ## print the solution
37. print(tm.export\_as\_lp\_string())
38. tm.print\_solution()

Appendix A2



*Figure 2: running code after using two stages model*