2020

Case Study Submission

WAREHOUSE ASSIGNMENT & SEQUENCING PROBLEM

VAIBHAV SINGH

Model -1

Ignoring transition time between tasks

Assumptions

1. All workers are identical having same speed (10 feet/min)
2. No task splitting is allowed
3. All tasks are available at time t = 0
4. All workers are available at time t = 0
5. Worker can execute only one task at time
6. Worker can immediately start the next task on completion of first task

Formulation

Input Data

- workers
  - m = number of identical workers
- tasks
  - d_j= time needed to finish j^{th} task from start point to its end point

  (distance required to be travelled for finishing j^{th} task from start point to its end point / speed of worker)

  - n = no of tasks to be finished

Decision Variables

- $x_{ij} = \begin{Bmatrix} 1, if\ task\ j\ is\ assigned\ to\ worker\ i \\ 0, otherwise \end{Bmatrix}\ \ for\ i = 1\ to\ m\ and\ j = 1\ to\ n$
- $C_{max} = The\ time\ at\ which\ last\ job\ is\ finished\ by\ the\ last\ working\ woker$

Objective Function

$$Minimize\ \ C_{max}$$

Subject to constraints

$$\sum_{i=1}^{m} x_{ij} = 1 \quad for\ j = 1\ to\ n \qquad \qquad \dots (1)$$

$$\sum_{j=1}^{n} x_{ij} * d_{ij} \leq C_{max} \quad for\ i = 1\ to\ m \qquad \qquad \dots (2)$$

Results

We were able to solve the model optimally using exact algorithm. Following assignments were made and resulted in completion of all the tasks in **42 minutes**

Worker	Assigned Tasks	Max Completion Time
ARMSTRONG	Task_3, Task_9, Task_14, Task_16, Task_23	41
BELL	Task_1,Task_4,Task_8, Task_11, Task_24, Task_40, Task_42	42
CAMPBELL	Task_13, Task_19, Task_26, Task_33, Task_50, Task_52	42
DIXON	Task_2, Task_29, Task_35, Task_36, Task_37	40
ELLIOT	Task_7, Task_17, Task_20, Task_21, Task_22, Task_53	40
FISHER	Task_12, Task_18, Task_34, Task_44, Task_46, Task_54,Task_55	42
GRAHAM	Task_5, Task_6, Task_32, Task_41, Task_47, Task_48	42
HAMILTON	Task_10, Task_15, Task_25, Task_27, Task_30,Task_31, Task_51	42
IRVINE	Task_3, Task_9, Task_14, Task_16, Task_23	42

Model-2

Considering transition time between tasks

Assumptions

1. All workers are identical having same speed (10 feet/min)
2. No task splitting is allowed
3. All tasks are available at time t = 0
4. All workers are available at time t = 0
5. Worker can execute only one task at time
6. Worker can immediately start transit towards the next task on completion of first task
7. Assuming all the workers start from bottom left corner of the warehouse (0,0)

Formulation

Input Data

- workers
  - m = number of identical workers
- tasks
  - d_{ij}= time needed to finish j^{th} task from start point to its end point + transit time to reach j^{th} task's starting point from end point of i^{th} task
    (distance / speed of worker)
  - n = no of tasks to be finished
  - node i = 0 represents starting point
  - node i = 1 to n represent j^{th} task

Decision Variables

- $x_{ij} = \begin{Bmatrix} 1, if\ task\ j\ is\ executed\ after\ task\ i \\ 0, otherwise \end{Bmatrix}\ \ for\ i = 1\ to\ n\ and\ j = 1\ to\ n$
- $u_j = comletion\ time\ of\ task\ j\ for\ j = 1\ to\ n$
- $C_{max} = maximum\ completion\ time$

Objective Function

$$Minimize\ C_{max}$$

Subject to constraints

$$\sum_{i=1}^{n} x_{ij} = 1 \quad for\ j = 1\ to\ n \qquad \ldots (1)$$

$$\sum_{j=1}^{n} x_{ij} = 1 \quad for\ i = 1\ to\ n \qquad \ldots (2)$$

$$\sum_{j=1}^{n} x_{0j} = m \qquad \ldots (3)$$

$$\sum_{i=1}^{n} x_{i0} = m \qquad \ldots (4)$$

$$u_j \geq u_i + x_{ij} * d_{ij} - (1 - x_{ij}) * M \qquad for\ j = 1\ to\ n\ and\ i = 1\ to\ n\ and\ i \neq j \qquad \dots (5)$$

$$C_{max} \geq u_i \qquad i = 1\ to\ n \qquad \dots (6)$$

Results

This is an np hard problem, where large instances cannot be solved exactly in polynomial time. Although it is possible to find solution for subset of instances (small instance), as shown below:

Tasks subset:

TASK_ID	x1	y1	x2	y2
Task_1	8	9	1	3
Task_2	7	4	1	1
Task_3	8	9	4	10
Task_4	10	8	3	10
Task_5	9	0	8	6

Employee Subset

EMPLOYEE_ID	EMPLOYEE_NAME
1	ARMSTRONG
2	BELL

Optimal assignment and sequence

EMPLOYEE_ID	EMPLOYEE_NAME	Assignment
1	ARMSTRONG	Task_0 -> Task_2 -> Task_4 -> Task_0
2	BELL	Task_0 -> Task_5 -> Task_3 -> Task_1 -> Task_0

Maximum completion time = 45 minutes (including starting from (0,0) and return to (0,0))

For large instances simulated annealing with TSP is proposed

Model – 3 (Simulated Annealing with TSP)

For large instances

Assumptions

1. All workers are identical having same speed (10 feet/min)
2. No task splitting is allowed
3. All tasks are available at time t = 0
4. All workers are available at time t = 0
5. Worker can execute only one task at time
6. Worker can immediately start transit towards the next task on completion of first task
7. Assuming all the workers start from bottom left corner of the warehouse (0,0)

TSP Formulation

Input Data

- workers
  - m = number of identical workers
- tasks
  - d_{ij}= time needed to finish j^{th} task from start point to its end point + transit time to reach j^{th} task's starting point from end point of i^{th} task
    (distance / speed of worker)
  - n = no of tasks to be finished
  - node i = 0 represents starting point
  - node i = 1 to n represent j^{th} task

Decision Variables

- $x_{ij} = \begin{Bmatrix} 1, if\ task\ j\ is\ executed\ after\ task\ i \\ 0, otherwise \end{Bmatrix}\quad for\ i = 1\ to\ n\ and\ j = 1\ to\ n$
- $u_j = order\ of\ task\ j\ for\ j = 1\ to\ n$

Objective Function

$$Maximize \sum_{\substack{i=1 \\ j=1 \\ i \neq j}}^{n} x_{ij} * d_{ij}$$

Subject to constraints

$$\sum_{i=1}^{n} x_{ij} = 1 \quad for\ j = 0\ to\ n \qquad \dots (1)$$

$$\sum_{j=1}^{n} x_{ij} = 1 \quad for\ i = 0\ to\ n \qquad \dots (2)$$

$$u_j \geq u_i + 1 - (1 - x_{ij}) * n \qquad for\ j = 1\ to\ n\ and\ i = 1\ to\ n\ and\ i \neq j \qquad \dots (5)$$

Simulated Annealing with TSP

Initial Feasible Solution

Using relaxed version (model-1) where transition times were not considered to get the initial allocation of tasks to each worker and then solving TSP (as defined above) for each worker to get the optimal sequence of the assigned tasks

Identify bottleneck worker and bottleneck task

Identify the worker having maximum completion time and the task taking maximum time of that worker

Select nearest neighbours

Randomly select 1 task out of 5 (hyperparameter) nearby task of the selected bottleneck task and determine the worker to whom the task was assigned

Update the solution

Remove the bottleneck task from bottleneck worker and assign to the new identified worker in step above.

Optimize the sequence of the new assignments using TSP.

Use simulated annealing to escape local minima

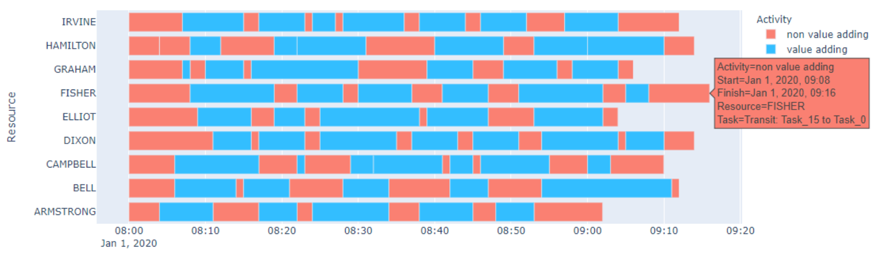
Allow the bad solutions initially to escape local minima

Results and Visualization

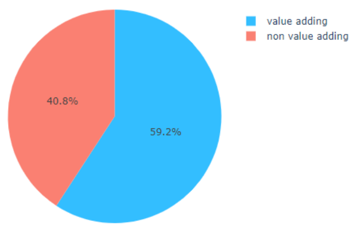
All the tasks were finished in **76 minutes** (including starting from and returning back to starting point – 0,0). Below are the assignments.

Warehouse Assignment Manager

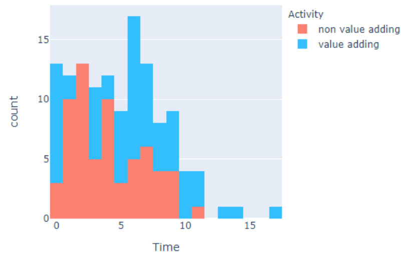
Gantt Chart



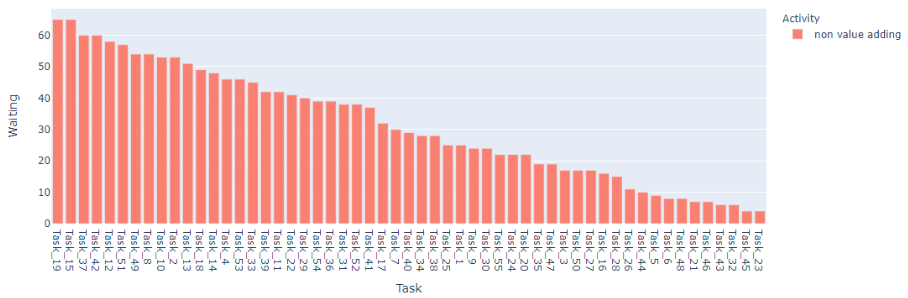
Value Adding vs Non Value Adding



Time distribution



Task Waiting Time (Min)



Assignments

Worker	Assignment Sequence	Total Time(Min)
ARMSTRONG	Task_0,Task_23,Task_3,Task_9,Task_52,Task_14,Task_0	62
BELL	Task_0,Task_43,Task_28,Task_38,Task_39,Task_49,Task_0	72
CAMPBELL	Task_0,Task_32,Task_24,Task_40,Task_17,Task_11,Task_4,Task_42,Task_0	70
DIXON	Task_0,Task_26,Task_50,Task_25,Task_41,Task_33,Task_8,Task_19,Task_0	74
ELLIOT	Task_0,Task_5,Task_35,Task_1,Task_36,Task_2,Task_0	64
FISHER	Task_0,Task_6,Task_20,Task_7,Task_22,Task_13,Task_15,Task_0	76
GRAHAM	Task_0,Task_46,Task_44,Task_16,Task_54,Task_18,Task_12,Task_0	66
HAMILTON	Task_0,Task_45,Task_48,Task_47,Task_55,Task_29,Task_10,Task_37,Task_0	74
IRVINE	Task_0,Task_21,Task_27,Task_30,Task_34,Task_31,Task_53,Task_51,Task_0	72

Task_0 is the dummy task which represent starting and return to bottom left corner of warehouse

To view interactive plot, run show_viz.bat and navigate to http://127.0.0.1:8000

Active internet connection will be required to load JS dependencies.