International Journal of Business, Human and Social Sciences
ISSN: 2517-9411
Vol:14, No:9, 2020

Resource Allocation and Task Scheduling with Skill Level and Time Bound Constraints

Salam Saudagar, Ankit Kamboj, Niraj Mohan, Satgounda Patil, Nilesh Powar

***Abstract*—**Task Assignment and Scheduling is a challenging Operations Research problem when there is a limited number of resources and comparatively higher number of tasks. The Cost Management team at Cummins needs to assign tasks based on a deadline and must prioritize some of the tasks as per business requirements. Moreover, there is a constraint on the resources that assignment of tasks should be done based on an individual skill level, that may vary for different tasks. Another constraint is for scheduling the tasks that should be evenly distributed in terms of number of working hours, which adds further complexity to this problem. The proposed greedy approach to solve assignment and scheduling problem first assigns the task based on management priority and then by the closest deadline. This is followed by an iterative selection of an available resource with the least allocated total working hours for a task, i.e. finding the local optimal choice for each task with the goal of determining the global optimum. The greedy approach task allocation is compared with a variant of Hungarian Algorithm, and it is observed that the proposed approach gives an equal allocation of working hours among the resources. The comparative study of the proposed approach is also done with manual task allocation and it is noted that the visibility of the task timeline has increased from 2 months to 6 months. An interactive dashboard app is created for the greedy assignment and scheduling approach and the tasks with more than 2 months horizon that were waiting in a queue without a delivery date initially are now analyzed effectively by the business with expected timelines for completion.

***Keywords*—**Assignment, deadline, greedy approach, hungarian algorithm, operations research, scheduling.

I. Introduction

ASSIGNMENT and scheduling is concerned with the allocation of a limited number of resources among competing tasks over time and with a set of constraints on the resources [1]. It is a decision-making process aiming to optimize one or more objective by considering the processes taking place and their interactions with the environment. It occurs in almost all manufacturing and production systems, transportation industry, healthcare and education industry. Assignment involves assigning the tasks to a resource who will be responsible for completion of those tasks and Scheduling refers to specifying the timeline for completion for each task and sequencing the task completion based on business priority [2]. An evenly distribution of workload among the resources in terms of number of working hours and effective scheduling to meet the deadlines for task completion, is of utmost importance for any organization. Whether the industry wants to efficiently assign resources for ongoing tasks or for a series of smaller tasks, having a method for making these assignments based on business constraints and scheduling it in a manner to meet the deadlines and prioritizing certain tasks as per management input is extremely important.

The Cost Management team at Cummins does a lot manual work of the assignment and scheduling of different tasks for a limited number of resources. If the number of tasks is higher than the available resources, then manual allocation of tasks is complicated and time consuming and it is observed by the Cost Management team that the scheduling period can only provide a horizon of over 2 months. Any real-life assignment and scheduling problem need to achieve the optimal solution over some criteria such as, minimization of time/cost and/or equalization of the amount of work for resources etc. The business requirement is to solve assignment and scheduling problem with equal distribution of workload among the resources and the tasks should be completed as per priority and deadline along with the other constraints provided.

In this paper, a greedy algorithm for the assignment and scheduling problem is developed where number of tasks is greater than the number of available resources. The focus of this work is to provide a strategy for imbalance task and resource scheduling with constraints from the Cost Management team at Cummins. The difficulty of task allocation increases when the number of employees required to complete the tasks are very few and tasks are large in number. The aim of developing such a strategy is to facilitate a scalable, fast, and efficient solution. The strategy focuses on utilizing information about the skill level for each resource corresponding to each task, estimated time required for completion of a task and prioritizing tasks as per business priority and deadline.

The rest of the paper is organized in the following manner: Section II describes the literature review. Data pre-processing and methodology of greedy assignment and scheduling algorithm are explained in Section III. Experimental evaluation and results are presented in Section IV. Finally, the paper ends with conclusions and future work in Section V.

II. Literature Review

In the field of operations research, the developments on task assignment and scheduling problems is a prominent area of research. There were assignment algorithms developed for a

Salam Saudagar is Data Science Intern, Ankit Kamboj is Sr. Data Scientist and Nilesh Powar is an Analytics Director in Advanced Analytics Team, Cummins Technologies India Pvt Ltd, Pune, India (e-mail: salamsaudagar@gmail.com, ankitkamboj007@gmail.com, nilesh.powar@gmail.com).

Niraj Mohan is Cost Improvement Leader and Satgounda Patil is Program Manager-Purchasing in Global Analytics Center, Cummins Technologies India Pvt Ltd, Pune, India (e-mail: niraj.mohan@gmail.com, sat_patil2003@yahoo.com).

set of independent periodic tasks in 1978, by Dhall and Liu [3]. The comparative study on the production scheduling problem was done using metaheuristics techniques like Genetic Algorithm, Neural Network, and Fuzzy Logic [4]. If there are constraints to prioritize the tasks, then the traditional assignment and scheduling algorithms cannot be used and based on the assumptions and the objective function, a generic task allocation problem is formulated differently. In distributed systems, some well-known ways for task assignment involves minimization of the sum of task processing cost using a graph- theoretic solution called Ford–Fulkerson algorithm [5].

The Hungarian method, developed by Harold Kuhn in 1955, is a "Combinatorial Optimization Algorithm" that solves assignment problems. The objective in assignment problem is to minimize the cost or time of completing a certain number of tasks by a number of resources. Assignment problem has an important characteristic that only one task is assigned to one resource, i.e. n tasks are assigned to n resources [6]. When there are equal number of resources and tasks, it is considered as a balanced assignment problem otherwise it is referred as an unbalanced assignment problem. An imbalance in the number of resources and tasks can be fixed by including dummy resources or tasks as needed and if there are dummy resources included, then the tasks assigned to dummy resources are not considered for assignment [7]. In a special case of unbalanced assignment problem, the number tasks are more than the available resources and multiple task needs to be assigned to each resource, m tasks need to be assigned to n resources (m > n) i.e. all the m tasks should be assigned to exactly one of the resources. This special case is solved using a variant of the Hungarian algorithm which is explained in the methodology section of this paper.

Greedy algorithm is used in optimization problems for finding the local optimal choice at each stage with the goal of determining the global optimum. It sometimes provides a good approximation to a globally optimal solution for example for Dijkstra's algorithm, which is used to find the shortest path through a graph. However, it may happen that the greedy strategy may not give an optimal solution as it makes decisions that seem best at any one step, without regard to the overall problem. Greedy algorithm is also useful for some problems that do not have an efficient solution and it may provide an effective solution that is nearly optimal. A greedy algorithm is applicable when a problem exhibits a greedy choice property, i.e. an optimal solution can be achieved by making local optimal greedy choices [8]. In 2013, Gevezes and Pitsoulis proposed a solution of the quadratic assignment problem for a cloud-computing data center using a greedy task scheduler to analyze energy-efficient task scheduling [9]. The usage of the iterated greedy algorithm for task assignment was proposed by Mohan and Gopalan in 2014, for heterogeneous computing problems using parallel processing paradigm [10].

III. Methodology

For any task assignment and scheduling problem there are constraints depending on a business problem. In this scenario, the following constraints are in place: a) No resource may process more than one task at any given time. b) Once the task gets started, it must be processed to completion. c) A known and finite time is required to complete each task and the time intervals for processing are independent of the order in which the tasks are performed. d) Holidays and leaves/vacation taken by resource need to be incorporated while scheduling tasks. e) All tasks are prioritized based on deadlines and given priority. f) Spillover of the task to next day/week is allowed.

A. Data Processing

1) Raw Data (Given)

In operations research, cost matrix is a matrix with unique tasks in a row and resources in a column. We suppose that there are n tasks and n resources available with different skills. Then the cost of performing task i by the resource j is represented by cij entry in the cost matrix [11]. Below are the different data sources that are used to formulate a cost matrix.

i: Skill Matrix

The skill matrix depicts the skill level for each resource corresponding to each task. The skill level can be 2, 3 or 4, where a higher value represents a higher skill level. Also, skill level is inversely proportional to the time required to complete a task, i.e. a resource with skill level 4 will take lesser amount time for completion of a specific task as compared to a resource with skill level 2, Table I.

ii: Baseline Effort Estimate Matrix

The baseline effort estimate matrix shows the time required for completion of a task, where there are specific tasks in rows and skill along with the type of approach in columns. Within each skill level, there are three different approaches in which a task could be completed; say a resource with skill level 3 can complete a specific task with detailed approach or a quick approach. The type of approach is named as Approach1, Approach2, Approach3, where Approach3 will be a more detailed way of doing a task as compared to Approach2 and Approach1, which are less detailed and quick way of completing a task respectively. For a specific skill level, the time required for completion of a task with Approach1, Approach2, Approach3 is in increasing order respectively i.e. Approach3 will have the highest and Approach1 with the lowest time for task completion. The time in the below baseline effort estimate matrix is in hours, Table II.

iii: Start Date, Holidays and Leaves

Scheduling of the assigned task is done using the Start dates for each resource. Holidays and vacation/leaves for each resource are also incorporated while scheduling the tasks.

iv: Task-Approach-Part Number Matrix

This matrix has task name, type of approach, part number, deadline and priority for project completion. In this study there are 10 different tasks, (Task1 to Task10), 3 different approaches (Approach1, Approach2, Approach3) and 400 different part numbers. Hence the concatenation of Task Name, Approach and Part No gives the unique Task_id for which assignment and scheduling of the resources needs to be done. The prioritization of Task_id is done first by priority and followed by the deadline of task completion, Table III.

TABLE I
SKILL MATRIX

Task Name	Resource1	Resource 2	Resource 3	Resource 4	Resource 5	Resource 6	Resource 7
Task1	3	3	4	2	3	3	3
Task2	4	4	3	2	3	3	3
Task3	2	2	2	2	2	2	2
Task4	3	3	3	2	3	3	3
Task5	2	2	2	2	2	2	2
Task6	3	3	4	2	3	3	3
Task7	2	3	3	3	3	2	3
Task8	3	3	3	4	3	3	2
Task9	4	3	3	3	3	3	3
Task10	2	3	3	3	3	3	4

TABLE II
BASELINE EFFORT ESTIMATE MATRIX

Task Name	Skill 2 - Approach1	Skill 2 - Approach2	Skill 2 - Approach3	Skill 3 - Approach1	Skill 3 - Approach2	Skill 3 - Approach3	Skill 4 - Approach1
Task1	16.7	43.3	66.6	13.9	36.1	55.5	12.5
Task2	15.9	41.3	63.6	13.3	34.5	53.0	11.9
Task3	13.5	35.1	54.0	11.3	29.3	45.0	10.1
Task4	12.6	32.8	50.4	10.5	27.3	42.0	9.5
Task5	10.5	27.3	42.0	8.8	22.8	35.0	7.9
Task6	9.6	25.0	38.4	8.0	20.8	32.0	7.2
Task7	7.7	19.9	30.6	6.4	16.6	25.5	5.7
Task8	6.9	17.9	27.6	5.8	15.0	23.0	5.2
Task9	6.6	17.2	26.4	5.5	14.3	22.0	5.0
Task10	5.7	14.8	22.8	4.8	12.4	19.0	4.3

TABLE III
TASK-APPROACH-PART NUMBER MATRIX

Task Name	Deadline	Approach	PartNo	Priority
Task9	2020-08-21	Approach1	1813	No
Task10	2020-07-19	Approach2	4525	No
Task1	2020-05-07	Approach1	6551	No
Task7	2020-11-06	Approach1	1005	Yes
Task9	2020-11-30	Approach3	6198	Yes
Task8	2020-05-20	Approach3	3018	No
Task7	2020-09-10	Approach1	5652	No
Task4	2020-05-12	Approach1	1721	No
.	.	.	.	.
.	.	.	.	.

2) Processed Data (Generated)

i: Cost Matrix

Cost matrix is prepared by using the unique Task_id as rows from Task-Approach-Part Number Matrix, resources as columns and the entries in this matrix denote the time required for completion of a specific Task_id by a resource.

For a unique Task_id and resource, say Task3_ Approach2_6506 and resource1, first the skill corresponding to a task name(Task3) and resource1 is fetched from the skill matrix i.e. Skill 2. Then from the baseline effort estimate matrix the task name(Task3) and skill – Approach (Skill 2 - Approach2) is used get the time required for project completion, i.e. 35.1 hours. This process is iterated over all unique Task_id and resource combinations to get a cost matrix and it gives the time required to complete every Task_id by every individual resource. The Task_id in the cost matrix is further sorted first by priority and then by deadline from the Task-Approach-Part Number Matrix. This cost matrix is used for assignment and scheduling, Table IV.

B) Greedy Assignment

The cost matrix is updated by first sorting the Task_id according to the Priority and then by Deadline. Since there are seven resources in this study, the assignment of the topmost seven Task_id is done in a greedy fashion by allocating it to a resource with minimum time for a task completion in updated cost matrix. Further, the subsequent allocations are done in iterative manner where the total time for each resource is updated after each allocation and then a resource selected in a greedy way with minimum total allocated time [8]. The Greedy Assignment ensures that the total time allocated for each resource should be equally distributed Table V.

Below are steps used for greedy assignment:

- Create a time_counter global variable for each of the resources and initialize them a zero
- For first non-zero initialization of time_counter for each resource, assign Task_id to an available resource with minimum time and update the corresponding time_counter by adding the time required to complete the Task_id from cost matrix.
- Then for every Task_id in the cost matrix:
- Iteratively do the assignment in a greedy way by

assigning the available resource with minimum value in time_counter.

TABLE IV
COST MATRIX

Task_id	Resource1	Resource 2	Resource 3	Resource 4	Resource 5	Resource 6	Resource 7
Task3_Approach2_6506	35.1	35.1	35.1	35.1	35.1	35.1	35.1
Task3_Approach1_4250	13.5	13.5	13.5	13.5	13.5	13.5	13.5
Task4_Approach2_5016	27.3	27.3	27.3	32.76	27.3	27.3	27.3
Task3_Approach3_7479	54	54	54	54	54	54	54
Task1_Approach3_4543	55.5	55.5	49.95	66.6	55.5	55.5	55.5
Task1_Approach1_7411	13.875	13.875	12.4875	16.65	13.875	13.875	13.875
Task6_Approach3_5456	32	32	28.8	38.4	32	32	32
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.

TABLE V
GREEDY ASSIGNMENT MATRIX

	Task3_Approach2 _6506	Task3_Approach1 _4250	Task4_Approach2 _5016	Task3_Approach3 _7479	Task1_Approach1 _7411	.	.
Resource 1	35.1	0	0	0	0	.	.
Resource 2	0	13.5	0	0	0	.	.
Resource 3	0	0	27.3	0	0	.	.
Resource 4	0	0	0	54	0	.	.
Resource 5	0	0	0	0	0	.	.
Resource 6	0	0	0	0	13.875	.	.
Resource 7	0	0	0	0	0	.	.

C) Extended Hungarian Assignment

The Hungarian algorithm gives an optimal solution for assignment problems with equal numbers of tasks and resources, i.e. n tasks to be assigned to n resources. In our case the number tasks are more than the available resources and multiple task needs to be assigned to each resource, m tasks need to be assigned to n resources (m > n). One way to handle imbalance in the number of resources and tasks is to include duplicate resources. It is done with a variant of Hungarian algorithm by cloning of the resources to get a square assignment matrix and then the assignment is done iteratively for each n resources i.e. iteratively Hungarian algorithm is applied for each m*m matrix.

Considering the scenario where the number of tasks is greater than the available resources and multiple tasks needs to be assigned to each resource, below are the steps used in a variant of Hungarian assignment:

- Create the Duplicate members (Clone the members multiple times till a square matrix is formed such that matrix will become n x n.
- Initially take the only m task and perform the optimized assignment to m resources. Then literately assign next m tasks to m resources till all the tasks are assigned.

The number of occurrences of a resource in the cloned matrix might not be same for all the resources, i.e. in the proposed study there are 400 Task_id and 7 resources, hence there are 57 (57*7 = 399) number of occurrences for each resource in the cloned matrix and the remaining 1 resource is chosen randomly. The proposed variant of Hungarian algorithm optimizes on the total time completion and it is observed that is no evenly distribution of workload among the resources, i.e. the highly skilled resource gets largest number of working hours and the less skilled resources gets lesser assigned working hours.

D) Scheduling

After the Task_id assignment is completed, the scheduling of tasks is done based on the number of working hours in a day and number of working days in a week, which are set to 8 hours and 5 days respectively and can be changed as per the business requirement. The start dates, holidays and leaves for each resource are also considered for properly adjusting the scheduling of all tasks.

The first allocation of tasks for every resource is done as per their start dates and then the subsequent tasks are scheduled immediately after a previous one is completed. It may happen that a Task_id may continue for more than a day and it depends on the time at which the previous Task_id was done on a day, say day_x , number of hours remaining for the day_x and the time required to complete the current Task_id. Finally, the start date and end date for each Task_id is noted for all the resources.

The scheduling can be done with the below steps:

- Create week_days = 5 and daily_hours = 8 as global variables
- Then for every resource in the assignment matrix:
- Schedule the Task_id for each resource as per the start dates, holidays and leaves and time required for completion
- Record the start date and end date for each Task_id
- The subsequent Task_id is scheduled immediately after completion of previous one and the number of hours remaining on the end date of previous Task_id is utilized.

IV. Experimental Evaluation and Results

The Dell Mobile workstation is used for running this model, which has an Intel(R) Core i7-8850H CPU @2.60Ghz and 32 GB RAM with Windows10. The versions of few important libraries used in this analysis are: Bokeh: 2.0.1, Numpy: 1.18.4, Python:3.6.8, Pandas: 1.0.3, Plotly: 4.6.0, Scipy: 1.3.1 (scipy.optimize.linear_sum_assignment), Streamlit: 0.62.0

Gantt Charts are used to visually analyze the tasks scheduled over time by using both start and end dates for a project and was named after its inventor, Henry Gantt (1861–1919) [12]. In this analysis, each Task_id is a combination of task name, type of approach and part number. The task name in Task_id has strong business relevance. Hence, the task names are kept as a legend in the Gantt chart. Since there are hundreds of Task_id, it is not feasible to show all the Task_id in the same plot. As depicted in the Fig. 1, there are multiple task names assigned to each resource which basically means that many Task_id are assigned to each resource, where each Task_id is a combination of task name, type of approach and part number and only task name is used to visually examine the tasks scheduled over time. The results of greedy assignment and scheduling are used for Gantt charts as the business stakeholders need to have an equal allocation of total working hours among all the resources.

The Gantt chart serves as a useful tool to project to the management the task schedules for a period of the next six months. These task deadlines need to be intimated to respective stakeholders who can plan their subsequent activities dependent on these tasks accordingly. Additionally, the resources are aware of the queue of tasks assigned to them over six months and can prepare accordingly to deliver as per timelines. This brings in a high degree of clarity and certainty in project sequence and timelines for all stakeholders.

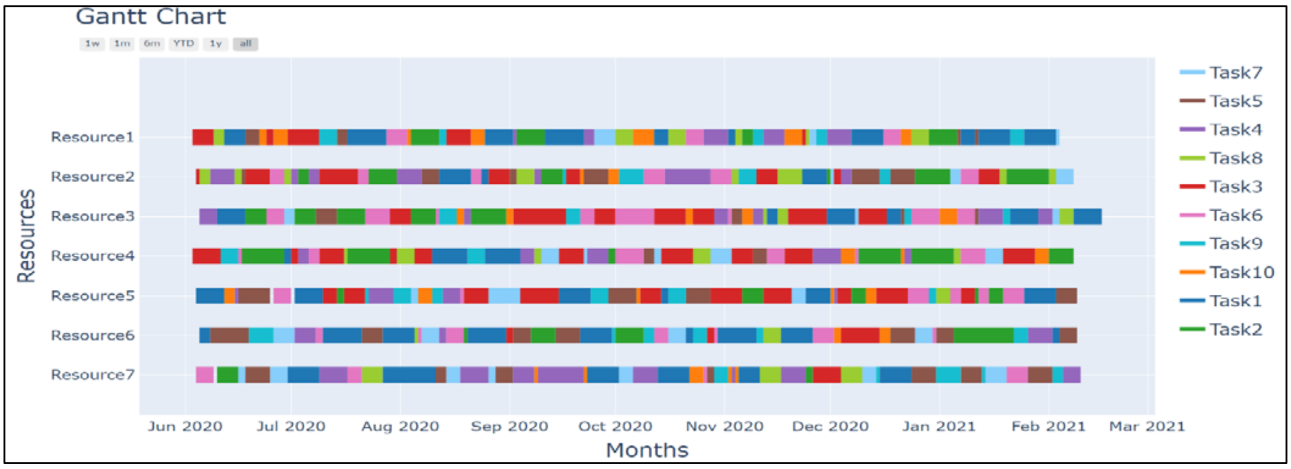


Fig. 1 Gantt Chart for the tasks scheduled for resources

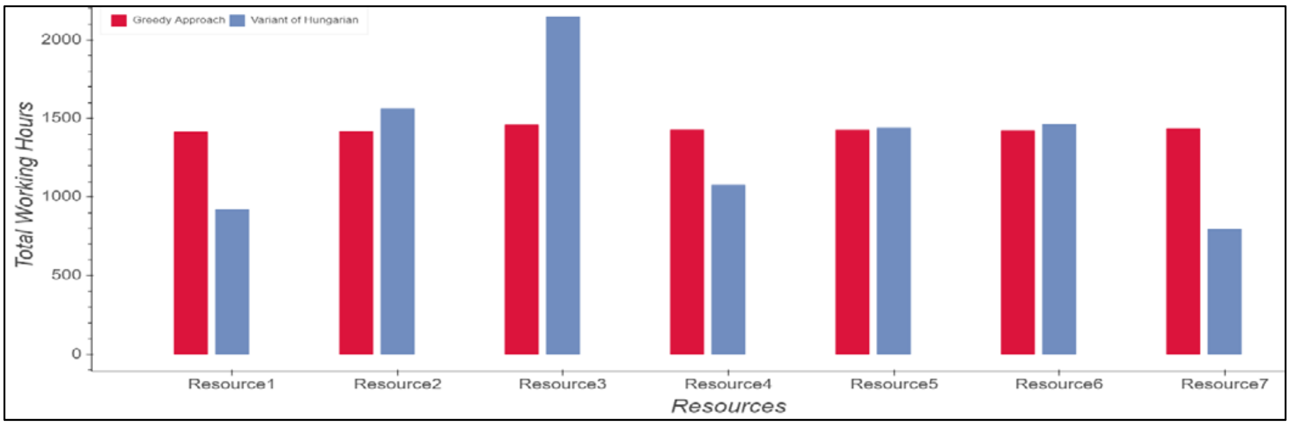
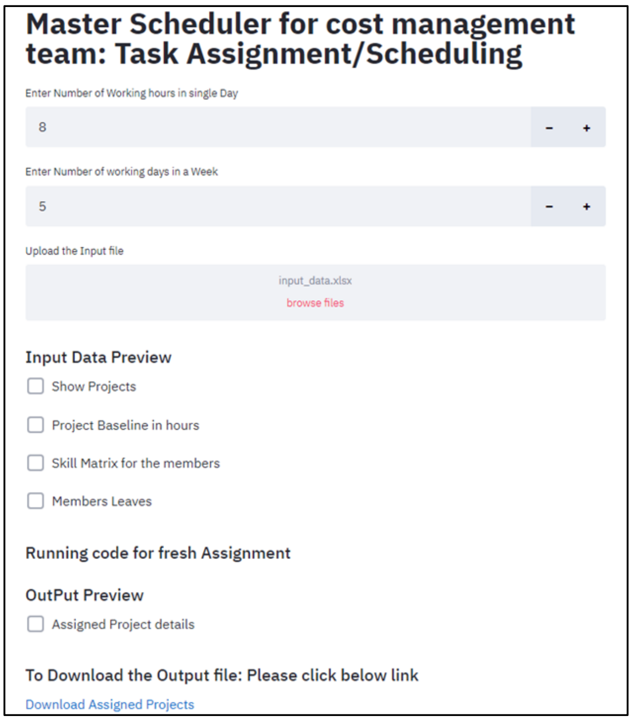


Fig. 2 Comparison between Greedy Approach and a variant of Hungarian Approach

Another comparison of the proposed greedy approach is done with a variant of Hungarian assignment in respect of getting an equal allocation of working hours among the resources. Fig. 2 shows that the proposed greedy approach gives equal distribution of total working hours compared to Hungarian assignment.

Finally, for the ease of running the code for business users, an interactive dashboard app is created in the streamlit python package. The user interface of this app gives users the flexibility to choose number of working hours in a day and number working days in a week, Fig. 3. Then the input excel file needs to be uploaded with the raw data of skill matrix, baseline effort matrix, Task-Approach-Part Number Matrix as different sheets. The input excel sheet also contains start date, holidays and vacation days for each resource as separate sheets. Once the input sheet is uploaded, the greedy algorithm code automatically runs at the backend and the output file of assigned tasks is generated.


Master Scheduler for cost management team: Task Assignment/Scheduling

Enter Number of Working hours in single Day: 8

Enter Number of working days in a Week: 5

Upload the Input file: input_data.xlsx

Input Data Preview

- Show Projects
- Project Baseline in hours
- Skill Matrix for the members
- Members Leaves

Running code for fresh Assignment

OutPut Preview

- Assigned Project details

To Download the Output file: Please click below link

Download Assigned Projects

Fig. 3 Interactive dashboard app

V. Conclusions and Future Work

In this paper the greedy assignment and scheduling approach of allocating tasks to resources is demonstrated for the scenario where the number are tasks to be allocated are higher than number of resources and the tasks are allocated in a greedy manner to have an equal allocation of total working hours for each resource.

Using the proposed approach, it is feasible to schedule the tasks for over 6 months. In contrast, by using the manual method the business users are only able to analyze scheduled task with 2 months horizon because the complexity of task assignment increases a lot due to the multi-fold parameters of skill, effort baseline, holidays, vacations, number of hours available and so on for task scheduling. Hence, manually scheduling beyond a two-month horizon while ensuring equal allocation of hours is not feasible. As an outcome, projects which were not in the 2 months window and were waiting in queue without a delivery date are now assigned a scheduled date beyond two months thus benefitting the business with clarity of timelines.

To further enhance the schedule visibility of the tasks, a Gantt Chart is deployed so that the execution schedules are more visual and easier to grasp by management as well as the resources who need to execute the tasks. The Gantt Chart dashboard makes the assigned task schedules visible for a time span of six months. The extended scheduling period also helps identify tasks which will miss the deadline because of resource constraints. This information is used to plan for surge demand months through task outsourcing to third parties and avoid impending delays. Also, a variant of Hungarian algorithm is compared with the proposed model of greedy approach and the latter gives an equal allocation of working hours among the resources which depicts the effectiveness to solve the business problem.

The future work for this paper is to incorporate reassignment and rescheduling for the originally assigned projects. Tasks could be prioritized and reassigned based on updated business priorities and deadlines. In addition to the equal allocation of assigned working hours, optimization of the total allocated time for all the resources could also be done.

References

[1] R. Aboudi and K. Jornsten, Resource Constrained Assignment Problems: Discrete Applied Mathematics 26) 175-191 North-Holland (1990) https://core.ac.uk/download/pdf/82253919.pdf

[2] G. Georgiadis, A. Elekidis, Optimization-Based Scheduling for the Process Industries: From Theory to Real-Life Industrial Applications, Processes 2019, 7, 438. https://doi.org/10.3390/pr7070438

[3] S. Dhall, C. Liu, On a Real-Time Scheduling Problem, Operations Research Journal Vol. 26, No. 1, (Feb 1978) https://doi.org/10.1287/opre.26.1.127

[4] T. Nehzati, N. Ismail, Application of Artificial Intelligent in Production Scheduling: a critical evaluation and comparison of key approaches, International Conference on Industrial Engineering and Operations Management Kuala Lumpur, Malaysia, (January 22 – 24, 2011) http://ieomsociety.org/ieom2011/pdfs/IEOM007.pdf

[5] H.S. Stone, "Multiprocessor Scheduling with the Aid of Network Flow Algorithms" IEEE Trans. Software Eng., vol. 3, no. 1, pp. 85-93, (Jan. 1977) https://ieeexplore.ieee.org/document/1702405

[6] W. Kuhn, The Hungarian Method for the Assignment Problem, Bryn Yaw College https://web.eecs.umich.edu/~pettie/matching/Kuhn-hungarian-assignment.pdf

[7] G. Korsah, M. Dias, A. Stentz, A Comprehensive Taxonomy for Multi-Robot Task Allocation, International Journal of Robotics Research 32. 1495-1512. 10.1177/0278364913496484. (2013) http://www.cs.cmu.edu/afs/cs/Web/People/gertrude/documents/Korsah_IJRR_Taxonomy.pdf

[8] A. Malik, A. Sharma, V. Saroha. International Journal of Scientific and Research Publications. Greedy Algorithm. Volume 3, Issue 8, ISSN 2250-3153, (August 2013) https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.642.4368&rep=rep1&type=pdf

[9] Z. Dong, N. Liu, R.Cessa, Greedy scheduling of tasks with time constraints for energy-efficient cloud-computing data centers. Journal of Cloud Computing. 4. 10.1186/s13677-015-0031-y, (2015). https://www.researchgate.net/publication/276886824_Greedy_scheduling_of_tasks_with_time_constraints_for_energy-efficient_cloud-computing_data_centers/link/5612c21608aea34aa9299a9f/download

[10] R. Mohan, N.Gopalan, Task Assignment for Heterogeneous Computing Problems using Improved Iterated Greedy Algorithm. International Journal of Computer Network and Information Security. 6. 50-55. 10.5815/ijcnis.2014.07.07, (2014). https://www.researchgate.net/publication/276230469_Task_Assignment_for_Heterogeneous_Computing_Problems_using_Improved_Iterated_Greedy_Algorithm/link/589becd3a6fdcc75417435e0/download

[11] V. Adlakha and H. Arsham, Managing Cost Uncertainties in Transportation and Assignment, Journal of Applied Mathematics

Decision Science, 2(1), 65-104 (1998) http://www.kurims.kyoto-u.ac.jp/EMIS/journals/HOA/JAMDS/Volume2_1/104.pdf
[12] C. Zorlu, Project Overview of Breast Cancer Treatment, Hult Business School (2020) https://www.academia.edu/42775787/PROJECT_OVERVIEW_OF_BREAST_CANCER_TREATMENT