

Intelligent Employee Task Scheduler

Gagandeep Garg

Meerut Institute of Engineering and Technology
N.H.58, Delhi-Roorkee Highway, Baghpat Bypass
Road Crossing, Meerut, Uttar Pradesh 250005

Utsav Dikshit

Meerut Institute of Engineering and Technology
N.H.58, Delhi-Roorkee Highway, Baghpat Bypass
Road Crossing, Meerut, Uttar Pradesh 250005

ABSTRACT

In this paper we try to solve task scheduling problem. For this a heuristic is used. The main objective of this paper is to reduce the burden of assigning tasks to employees by task manager. Parameters of employee availability, task deadline, employee free time are considered. It has become the first most priority for both the sectors of economy that is public and private sectors, to schedule the tasks provided to its employees. This research paper displays a framework and the steps required to implement that framework has been illustrated. Also model and algorithm with example to solve task scheduling problem has been included.

Firstly the initial solution is provided to the proposed algorithm. The algorithm halts or stops execution after termination criteria is met.

Keywords

Task Scheduling, Task Assignment, Heuristics, Task Deadline

1. INTRODUCTION

The task scheduling problem (TSP) is related with a number of domains which include scheduling day, and workload assignment. In order to maximize profits workforce task scheduling is very important without much sacrifice in the staff's efficiency. The scheduling problem aims of determining efficient individual work assignment to a set of already computed subjects to organizational and legal limits. Instance of this task scheduling problem are generally NP-Hard. With the help of some mathematical programming we can find the ideal solution in some minor cases. Although heuristic approaches are required to reach the optimal solution possible in reasonably lesser time (real case).

In everyday life problems workforce/tasks scheduling problems can also be seen. A number of applications include scheduling of airline crews, call-centre receptionists, public transportation, hospital staff and industrial plants. This research paper mainly focuses on the task assignment

problem. This scheduling problem mainly consists of determining the optimal individual work assignment to a set of already determined subjects to organizational and legal constraints.

A heuristic algorithm is developed and discussed in this research paper to solve the problem of assigning task to the employees. Intelligent Employee task scheduler is a java swing GUI based desktop enabled intelligent task scheduling system. Its platforms include a desktop version application that holds employee and business data.

In the today scenario managers of small workforce schedule his/her employees manually. In a nut-shell the main aim of them is to meet feasibility constraints hence scheduling employees when they are available at work. Although their solution is difficult to obtain yet there are many possible solutions this is a largely subjective process.

Scheduling more employees with additional constraints increases the difficulty of the scheduling methods are proving inadequate. So there is a need to build an algorithm that automates the task scheduling. The algorithm must be designed in such a way that provides cost saving optimizations.

Currently all scheduling software are providing the facility of minimizing laboring costs. Usually it is the responsibility of task manager to assign tasks to their employees manually according to their availability, task deadline and on many more parameters but by considering all these parameters our algorithm assign tasks to the employees intelligently. On a large scale, reducing employee laboring costs give a significant financial profit.

This research paper is motivated by corporate companies in which companies get daily tasks from different client which companies have to complete before deadline.

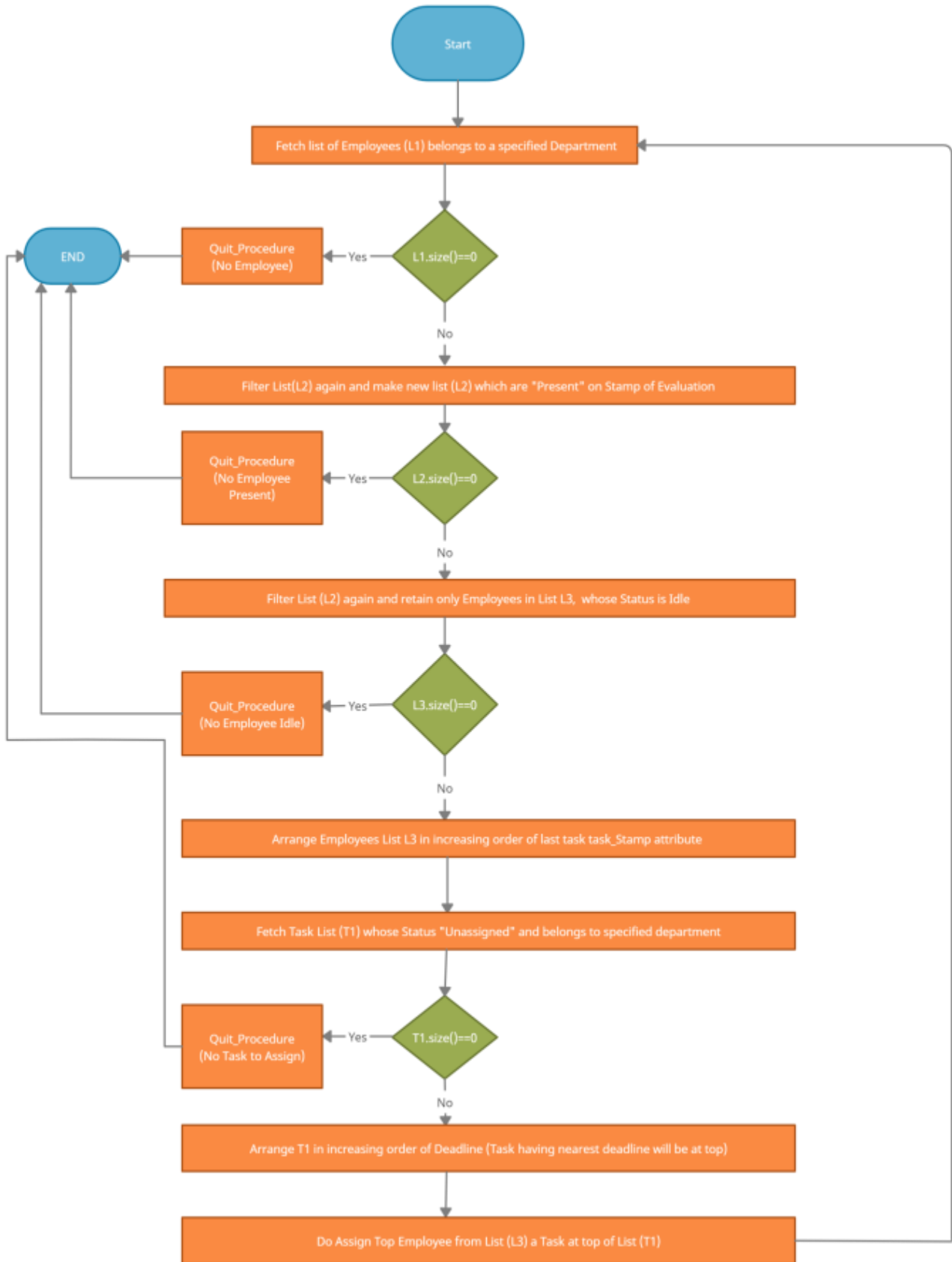


Fig 1: Flowchart for Proposed Algorithm

1.1 Problem with existing system:

Current Applications don't have a procedure for scheduling the tasks to employees automatically. Manual task assignment leads to the burden for project manager. There is a possibility that employee may get two or three tasks at same time before its first task is not completed which increase the burden to the employee. Existing systems have no features like ranking the employees according to their scores, employee login with QR code given from android app etc.

Mostly existing systems do not use any kind of intelligent features. Because of which accuracy is low or inadequate. Automatic task assignment cannot be done with too many constraints associated with both tasks and employees as it may lead to too many possible solution. Due to these too many possible solutions scheduling the task is cumbersome especially in the public and private sector companies.

1.2 Aims & goals of proposed system:

The objective is to reduce the work load of project manager working in companies whose work is to assign various tasks to employees working under him. Our software provide flexibility to project manager to not only assign task automatically but also he(project manager) can assign task to employees manually.

2. DESIGN AND IMPLEMENTATION OF PROPOSED METHODOLOGY

Our task is composed of only one component: a java based desktop GUI application for use of employees, task managers and admin working in a company.

This desktop and GUI based application handles all of the schedule data management which includes: skills, task deadline and availability of employee data. This java based desktop GUI application is used to generate schedule by matching task deadline with workers availability. User can download the java based desktop GUI application for the management of scheduling the employee for a small business. All data involving positions, employees, and schedules would reside in the database but data storage is handled locally.

This java based application gives limited access to all employees for data to be used in schedule generation. Specially, a worker can change their schedule by making present or absent for that day by login and then save changes to the database. Employee is feasible to take the task if he/she is idle and present for that day.

2.1.1 Variables

N: Whole number of employees.

I: Set of Idle employees.

D: Deadline of a task.

E_p : Employees present at that instant.

2.1.2 Input data

T: Tasks which have to be completed by employees.

E_N : New employee added by task manager.

2.1.3 Assumption

1. Shortage of employees is allowed.

2. Extra workers are permitted.
3. The total number of employees either present or absent not need to be known before scheduling process starts.
4. An employee can do only one task at a time and also it is not possible that more than one task can be assigned to the employee when the employee is in working state.
5. It is assumed that all the employees are having same potential.
6. A task is assigned to the employee only when the employee is present for that day and the status of employee is idle.

In a list of employees if there are some employees who are present and idle and all of them are new means they have not yet assigned with any task then the task to be assigned must be to the employee whose idle time is greatest among all.

3. PROPOSED ALGORITHM DESCRIPTION

We have N number of employees in our organization. Our System has a process or a thread which is running continuously in the background and checks the employees E_p who are present at the instant of thread. Since thread can be closed by task manager (using toggle button), at that time our automatic task scheduling algorithm will stop immediately.

Working among N workers we have the following procedure:

1. Firstly, fetch out the list of employees N that are working in the organization.
2. Now prepare a map (Key-Value pair) for N employees based on their availability where each key represent the employee and the value represent their availability.

Example,

There are 4 employees working in the organization (E_1, E_2, E_3, E_4) and E_2, E_4 are present then the map prepared from the employees would be
{ E_1 : "A", E_2 : "P", E_3 : "A", E_4 : "P"}

Where,

P->Present

A->Absent

3. Now prepare a list of employees who are present at the time.
4. Again out of these employees prepare a list in which the employees who are idle come in front of the list and who are working come later in the list.
5. Now arrange the tasks according to their deadline present in the task list.

Example,

Suppose there are three tasks T_1, T_2 and T_3 with their respective deadlines D_1, D_2 and D_3

$T_1 \rightarrow D_1 \rightarrow 22/01/2021$ (11:55)

$T_2 \rightarrow D_2 \rightarrow 23/01/2021$ (14:38)

$T_3 \rightarrow D_3 \rightarrow 24/01/2021$ (8:07)

Then, the task list will be

$T = \{T_2, T_1, T_3\}$

- Out of the employees who are idle, prepare a new list and arrange the employees according to their idle time.

Example,

$E_2 \rightarrow 40$ minutes

$E_4 \rightarrow 49$ minutes

Then, list will be

$E = \{E_2, E_4\}$

- Now automatic assign task thread starts its process by taking out task one by one from the list T and assigning it to the employees one by one from list E and changing the status of employees from idle to working. By taking above example the results will be:

$E_2 \rightarrow T_2$

$E_4 \rightarrow T_1$

T_3 will remain exist in the task which will be later assigned.

If two employees are having same idle time then assign task to the employees randomly.

4. HOW SCORES WILL BE CALCULATED

If task is completed successfully then, total score of an employee is the sum of task score to his/her initial score and if the employee could not complete the task then add the task points to the initial score with some deduction in task score according to the delay in task submission suggested by given formulae with given example,

If task is completed within the deadline and is positively reviewed by task manager successfully then,

Total points of employee = Total score of employee before the task + full task score

Else

Total points of employee = Total points of employee before the task + x% of full task score

Where x can be calculated in accordance with delay in submitting the task referring as a “Task Score Deduction Factor”.

5. CONCLUSION

The result shows the actual behavior of the proposed algorithm as: After testing for at least 10 test cases for different problem situations, we obtain the optimal solution. Since this is not possible to make extreme and complete analysis of the performance and working of current proposed algorithm, so there is a need to consider more scenarios.

The algorithm also shows average behavior on the ground of measuring execution time: In almost all tested scenarios the algorithm executes in significantly shorter time with optimal solution.

We believe that a greater number of real world scenarios can be modeled using the framework presented in this paper. This research has been contributed to improve the system for industries and customers which requires the scheduling of tasks to workers/employees.

Finally the design and implementation of this task will make sure that the application will be consumed by various organizations consisting of managers and employees with ease and reducing the manual workload of the managers.

6. RESULT ANALYSIS

We can show the accuracy of the system by taking an example which contains tasks and employees through the assignment of the task from the list of tasks with their associated deadline to the employees from the list of employees with their associated idle time based on our algorithm by the table representation.

Table 1. Tasks with their Deadlines

Tasks	Deadline
T ₁	05/02/2021 (11:22)
T ₂	02/02/2021 (08:01)
T ₃	07/02/2021 (22:40)
T ₄	01/02/2021 (01:25)
T ₅	07/02/2021 (18:20)
T ₆	02/02/2021 (02:02)

Table 2. Employees along with their status and idle time

Employee	Status	Idle Time
E ₁	Working	
E ₂	Idle	25 min
E ₃	Working	
E ₄	Idle	48 min
E ₅	Idle	10 min
E ₆	Idle	168 min

Table 3. Table Mapping Employees with Tasks

Required Mapping	Actual Mapping	Accuracy
$E_6 \rightarrow T_4$	$E_6 \rightarrow T_4$	1
$E_4 \rightarrow T_6$	$E_4 \rightarrow T_6$	1
$E_2 \rightarrow T_2$	$E_2 \rightarrow T_2$	1
$E_5 \rightarrow T_1$	$E_5 \rightarrow T_1$	1

System Accuracy = (Actual Mapping/Required Mapping) * 100 = (4/4) * 100 = 100%

7. REFERENCES

- [1] An integer programming approach to reference staff scheduling
<http://www.sciencedirect.com/science/article/pii/S0306457385900913>
- [2] Balakrishnan, N and Wong, R.(1990) “A network model for the rotating workforce scheduling problem” Volume 20, pp 25-42
- [3] Java : The Complete Reference(10th edition) : java SE 9(jdk 9) Herbert Schildt – Introduction to Swing, pp 1011
- [4] M.R. Garey and D.S. Johnson, Computers and intractability: A Guide to the theory of NP-Completeness. New York: Freeman, 1979.
- [5] Approach for a constraint employee scheduling problem as applied to employees at mall shops
<http://www.sersc.org/journals/IJAST/vol14/1.pdf>