Technical Disclosure Commons

Defensive Publications Series

November 21, 2018

A System And Method For Task Management

Liya Ophir

Clifford Stein

Christopher Rohrs

Talia Swann Sternberg

Ryan Muller

See next page for additional authors

Inventor(s)
Liya Ophir, Clifford Stein, Christopher Rohrs, Talia Swann Sternberg, Ryan Muller, and Michael Baldwin

A SYSTEM AND METHOD FOR TASK MANAGEMENT

ABSTRACT

A system and method for task management are disclosed. The system may include an electronic device that allows users to enter a list of tasks as input. The system may further use a scheduling algorithm to schedule the entered tasks. The primary output of the system is a schedule assigning tasks for a predetermined interval - days or even hours. The scheduling algorithm may compute an objective function defined by the user and take into account costs for each task. The method then involves minimizing total task costs subject to constraints, while efficiently scheduling the tasks across the time interval. The system is configured to improve via user inputs. The system and method disclosed may help users to schedule work they need to complete each day and in the future. As a result, users may be less overwhelmed by their list of tasks.

KEYWORDS: Task lists, prioritizing tasks, daily schedule, task assistant

BACKGROUND

Many people need to manage lists of tasks to do. For example, students might need to consider all the tasks they need to do for their classes: problem sets, papers, reading, etc. Each task item might have some of the following properties: a day the task can be started, a day the task is due, some notion of priority, some notion of the size of the task, etc. It can be difficult for people to prioritize and allocate their tasks. Should Amy spend an hour working on the 5-point problem set due in two days, or spend the hour studying for the organic chemistry exam in two weeks? Is it OK for Brad to spend only an hour working tonight, or does he need four hours? It’s already a lot of work just to keep track of due dates, but it’s even more work prioritizing what to work on and when. A related problem is procrastination. Can we help Casey to actually start her

work today, rather than put it off until it's past midnight? It may be difficult for people to prioritize and allocate their tasks using just a TODO list.

DESCRIPTION

A system and a method for task management are disclosed. The system may include a computer, a smartphone and a processor. The system allows users to enter tasks. Each task may have an earliest start date, due date, priority and estimated amount of time to complete the task. Tasks may include a list of sub-tasks that may be completed in order by a given day. A scheduling algorithm may be used to schedule the entered tasks. The scheduling algorithm may try to satisfy one or more of the following properties: complete by the due date, don't start tasks before the earliest start date, start task as soon as possible, start high priority tasks sooner than low priority tasks, start large tasks sooner than small tasks, do same amount of work per day when possible, and minimize variation across days. In order to satisfy these properties, the system may need to make tradeoffs or relax constraints. For example, it may be necessary to prioritize a less important task over a more important task because the less important task is due sooner.

The primary output of the system is a schedule assigning tasks for a predetermined interval. The schedule may be viewed for just a given day or for a week. An exemplified task management system as shown in FIG. 1 includes a user entering a list of tasks. The output of the example system is a schedule that assigns tasks to days (or even hours) based on the priorities and constraints.

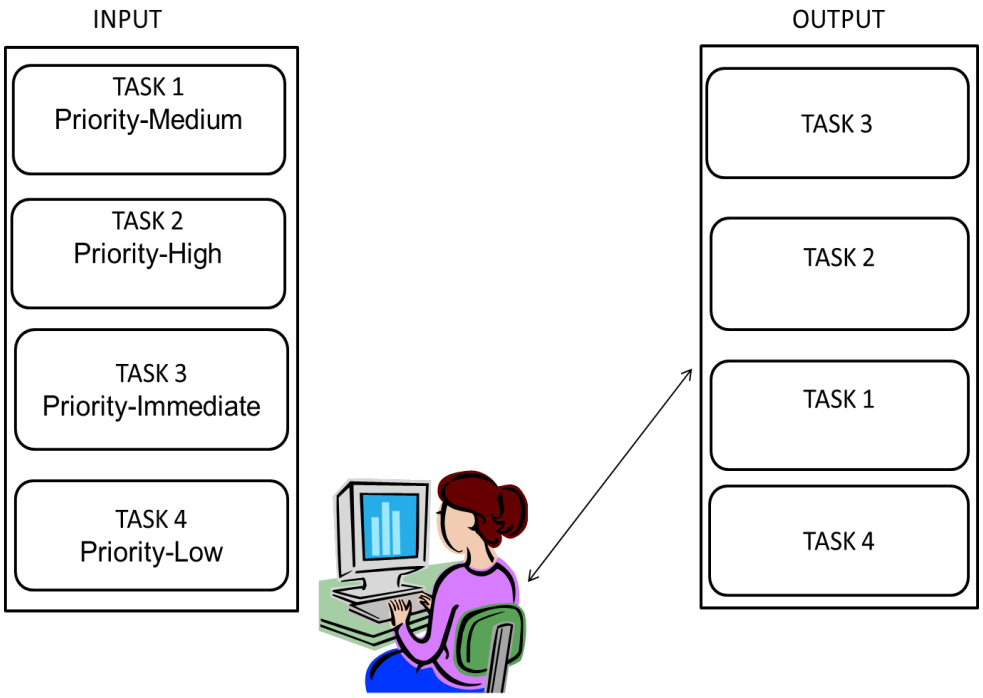

FIG. 1: Task management system

In one implementation, the system may include a scheduling algorithm that initially assigns tasks just before the due date, and then repeatedly assigns tasks to an earlier date in order to smooth the schedule.

Another method as shown in FIG. 2 for scheduling involves the following steps. In step A, the system determines the interval needing scheduling first, based on highest hours of work per day, and the tasks contained within that interval (if no more tasks remain, you're done). The system then (step B) defines an objective function determining the scheduling. The objective function may be a function of the task parameters and the time the task is allocated. Further, the user may specify weights for tasks, and these may be part of the objective function. The task parameters may include one or more of data related to task priority, task size and estimated number of days (or hours) for task completion. For example, an objective function may be defined as priority * size^days, where size is a decay factor related to the size of the task, and

days is the number of days until the due date. In step C, the system determines a cost for each task based on the objective function. Further, step D involves minimizing the sum of task costs subject to constraints, while keeping the work evenly spread across the predetermined interval. The days with the highest work requirement are then scheduled first. Within an interval, tasks are then scheduled and assigned in cost order in step E. Further, the schedule is dynamically updated in step F, so that task order may change as time passes. The system repeats the process including steps A through F as tasks get completed or whenever the user enters a new task.

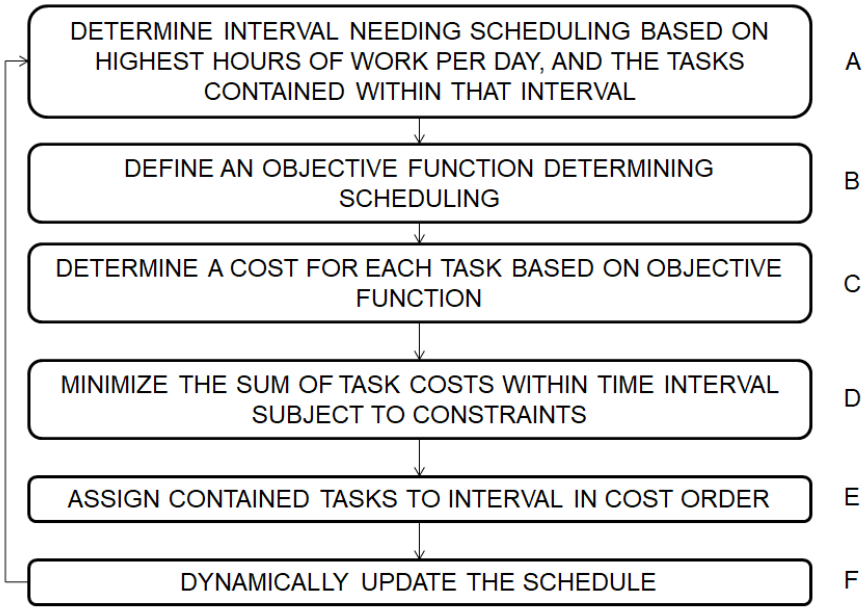

FIG. 2: Method for task scheduling

The system may also provide ways for users to interact with the schedule. For example, a user may edit any task property initially input or mark a task as done or assign ("lock") a task to a certain day or request that the program schedule have more or less work on a given day. The last two features can be expressed by altering the constraints of the scheduling algorithm. For example, locking a task sets the start and due dates of a task. The day constraints may be expressed by assigning a desired number of hours per day and excluding these days from the

required hours of work calculation. Further, the system may automatically learn the cost function for tasks, based on the user's preferences or those of other users within an institution. For example, the system may learn to prioritize a small high priority task over a large low priority task.

The system and method disclosed may help users to know the work they need to complete each day in the future. As a result, users may be less overwhelmed by their list of tasks. The system and method allows users to make informed judgments about re-allocating their time. For example, a user may be able to analyze the effect of doing less work at present. The user may then allocate a predetermined time to complete the work the next day based on the analysis.