Human-Centered Decision Support for Agenda Scheduling

Stephanie Rosenthal
Carnegie Mellon University
Pittsburgh, PA
srosenth@andrew.cmu.edu

Laura M. Hiatt
Naval Research Laboratory
Washington, DC
laura.hiatt@nrl.navy.mil

ABSTRACT

Over the course of a day, people often attend many appointments, like meetings or doctor's appointments, with time and location constraints, as well as perform other tasks like stopping by the grocery store with only location constraints. Optimally scheduling the day's agenda typically requires a high cognitive load, where people reason about all their constraints at once. An agent that aids a person in scheduling the day's agenda can potentially reduce the stress associated with scheduling but must be able to 1) perform fast updates and 2) produce new agendas that can be readily understood by the people they are helping. In order to understand how people reason about agenda changes, we first performed a study where participants are asked to perform a series of scheduling tasks and captured their update strategy both subjectively (self-reporting) and objectively (by tracking their reasoning). Our results that show that people use spatial cues and meeting time information to reduce the rescheduling task to a more reasonable size. We then present a novel heuristic for adding tasks to agendas that targets rescheduling to clusters of appointments that are spatio-temporally near the new task. We show that this heuristic approach always finds the optimal solution, while greatly reducing rescheduling time, and performs rescheduling in a way that is similar to our participants' strategies.

KEYWORDS

agenda scheduling; modeling humans; decision support agent

ACM Reference Format:
Stephanie Rosenthal and Laura M. Hiatt. 2020. Human-Centered Decision Support for Agenda Scheduling. In *Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), Auckland, New Zealand, May 9–13, 2020,* IFAAMAS, 8 pages.

1 INTRODUCTION

In the real world, lots of activities are time- and location-constrained, but many are not. Over the course of a day, we attend many *appointments* like meetings or doctor's appointments with specific locations to navigate to and specific times to arrive and leave, and we also have other *tasks* to complete that only have location constraints like taking money of out the ATM or stopping by the grocery store before dinner. Note that these tasks are not necessarily unforeseen or dynamic; going to the grocery store could be a weekly activity without an exact scheduled time. Creating a valid *agenda* of times and locations to complete all of the activities requires searching through the *time gaps* between set appointments to find those that are long enough to navigate to, complete, and navigate away from each task location without disrupting the remaining appointments.

Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), B. An, N. Yorke-Smith, A. El Fallah Seghrouchni, G. Sukthankar (eds.), May 9–13, 2020, Auckland, New Zealand. © 2020 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

The cognitive overhead of performing such scheduling tasks is very high for individuals [2, 18, 19]. People often create physical artifacts, elaborate plans, and routines around scheduled activities to avoid forgetting their agenda, but often still forget especially when unexpected events (especially unscheduled tasks) come up [1, 7]. Many tools have been created to help people with scheduling (e.g., [3, 5, 21, 23]). One challenge to developing activity scheduling support algorithms and extending them to schedule tasks is that current optimal schedulers take exponential time in the number of tasks to complete. In particular, they attempt to place new tasks within an agenda in every possible time gap in order to minimize the overall agenda length. These schedulers seek agenda length optimality regardless of search time, which is NP-Hard [10]. While scheduling agendas is hard for computers, we notice that humans can perform the task relatively quickly while maintaining near-optimality. The speed is especially important given that new tasks that arise during the day tend to be unplanned and must be resolved quickly in order to keep up the pace of execution [1]. In order for us to build usable, deployable decision-support systems to reduce people's cognitive load on scheduling tasks, we must be able to add new tasks to the agenda as quickly and effectively as humans. A related goal is to be able to do so in a way that is explainable so that people can understand and interpret them.

Towards this end, we conducted both human-subject and simulated experiments to understand both how people add tasks to existing agendas, and how optimal planners can quickly perform the same task. We conducted a human-subject experiment in which we asked participants to add tasks to several daily agendas and collected their strategies similar to [11] for meeting scheduling, [15] for collaborative planning, and [4] for routing. We found that many people solve the problem by trying to fit tasks into any available time gap before the end of the last meeting. Importantly, a set of participants also used a self-described "location-based" strategy that focused on the few appointments during the day that were closest in proximity to the new task; this strategy resulted in those participants searching fewer time gaps between appointments while maintaining accuracy at finding valid times for scheduling tasks.

Leveraging the successful location-based strategy of human participants, we propose a heuristic for searching agendas that focuses on the spatial proximity to existing appointments and tasks in addition to their time constraints. The approach identifies the portion of the agenda that is spatially the most similar to the new task, and then performs rescheduling with that smaller group of activities and the new task. Intuitively, this heuristic targets activities on the agenda that are likely to support the new task because spatial-closeness implies that navigation time between the activities is shorter so the time gaps can be smaller yet still fit the new task.

Through extensive testing, we demonstrate that by focusing on a small group of activities, the time to reschedule can be greatly reduced compared to full optimal scheduling when there is a valid time gap for new task to fit in, while maintaining optimal agenda length (time to complete the agenda). This is because appointments within an agenda cannot be moved, so the overall length of the optimal agenda will not change as long as a task can fit in a time gap before the end of the last meeting. As a result, the optimal scheduler performs unneeded computation for many agendas, creating opportunities to utilize heuristic strategies to target rescheduling to places where the task can intuitively be inserted. Additionally, we found that our study participants' location-based strategy analytically correlates well with our heuristic algorithm, demonstrating that our technique of using spatial-proximity of the scheduled appointments to the new task is similar to what people do in practice. Due to these similarities, we conclude that our heuristic not only reduces rescheduling time, but also changes the agenda in a similar way that people do – in relation to the tasks around it.

2 RELATED WORK

People spend a large amount of time determining what, when, and where their appointments are, and what other tasks need to be completed in particular places or by particular deadlines. It is challenging for people to manage agendas because there are often many constraints placed on each appointment or task, and because those constraints can affect each other in complicating ways such as determining the time to travel from one location to the next [2, 18, 19]. These challenges are especially complex when scheduling around multiple agendas in a family or in a work environment because each person's agenda affects the times that other group members can schedule their appointments.

In order to remember their agendas, people often create artifacts like calendars, elaborate plans for pickup and dropoff of children at activities, and routines around scheduled activities [7]. However, when unexpected or new events or tasks disrupt the agenda, even with these aids people still can fail to complete all of their required activities [1, 7]. Failing to complete an agenda can have significant consequences for people such as missing important deadlines to complete work items [23]; forgetting items or missing events for school-aged kids [7]; and pushing tasks to another day, cascading changes and potentially more missed events.

Because scheduling meetings at work with many people can be tedious due to conflicting constraints on participants' time, many tools have been created to help people with this task. Microsoft Outlook allows participants to look at each others' schedules using a visual calendar to find times in which everyone is available. Other tools allow people to list their time preferences on their calendar so that an agent can find the highest ranked time for everyone [5], or enable them to create agents to negotiate times on their behalf [3]. Because it is still tedious to list all of one's preferences, still more work allows people to demonstrate their scheduling strategy [11] or provide a schedule workflow to follow [6]. Interestingly, support for scheduling family agendas has focused on supporting communication through calendar annotations for reminders rather than helping to schedule the appointments in the first place [19].

Beyond the challenges of scheduling appointments, unconstrained tasks such as todo list items pose a different challenge. Because they do not have set start and end times but may have deadlines, tasks must be inserted into agendas at opportune times. For example, students typically have many homework assignments to complete. These assignments have deadlines but not set hours, so students must schedule time to work on them during the time gaps between their constrained appointments like class lectures. Tools like [23] help students to prioritize what to work on when and schedule that work time within a calendar. Similarly, when people schedule tours through cities, e.g., for vacation, they have a list of locations that they would like to visit and constraints about when those locations are open and closed. Creating a tour itinerary requires scheduling temporally-unconstrained tasks on an agenda [17, 20].

Scheduling all of these types of appointments and tasks on an agenda is an NP-Hard constraint satisfaction problem that involves modeling the temporal and spatial requirements to ensure that a solution agenda is feasible [10]. Recent work has focused on framing unconstrained or flexibly-constrained tasks within the same constraint satisfaction scheduling framework [21]. Because people may have potentially many appointments during their day, a decision support agent to assist in scheduling tasks should be able to solve scheduling problems more quickly than exponential time. Approaches to efficient scheduling trade off optimality of the solution (i.e., minimizing agenda start time to end time, placing all of the appointments/tasks on the agenda) for speed of finding a solution. For temporally- or spatially-flexible tasks in particular, [21] demonstrates an algorithm for optimizing an agenda that runs in linear time but at the cost of not scheduling all events.

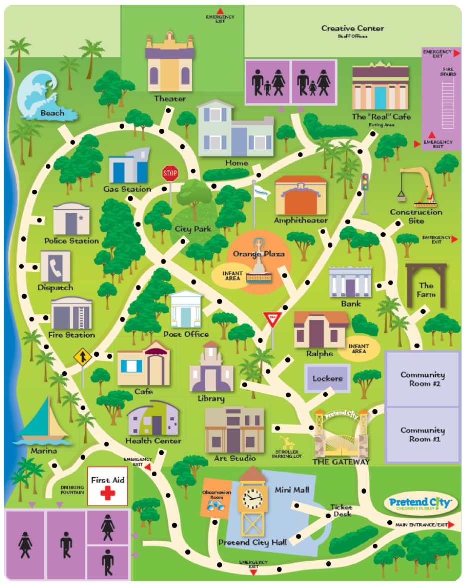
Additionally, a decision support agent should perform the scheduling task in a similar way to people so that they can interpret the schedule changes easily. To the best of our knowledge, no prior work has studied how people schedule new tasks in their agenda. In this work, we first present a study in which we asked participants to schedule a new task in an agenda and we measured their strategies. We found that people spatially cluster appointments to focus on finding time gaps around locations near to the new task (e.g., we stop at the post office when we are already at the grocery store next door). Then, we propose a technique for scheduling temporally-unconstrained tasks that takes advantage of our study finding by clustering the appointments and only rescheduling nearby clusters. Because the clusters can be kept at a constant size, the algorithm runs in constant time compared to linear or exponential time of other algorithms. Additionally, our algorithm always finds the optimal solution, unlike prior approaches.

3 HUMAN SCHEDULING STRATEGIES

In order understand how people perform agenda scheduling tasks, we first performed an experiment in which each participant was asked to add tasks with set locations and durations "at convenient times" within 10 unique 8-appointment agendas. Recall that *appointments* are activities with both location and time constraints, whereas *tasks* have only location constraints; this means that participants had to add a task without moving any of the existing appointments on the agenda. We analyzed the reported and demonstrated strategies that the participants used based on the order that they checked gaps in the agenda to see if the new task fit.

3.1 Agenda Design

In order to avoid biasing our agendas in any way, we asked a third party to generate eleven agendas of appointments at locations



Schedule	
Theater:	7:00 - 7:30
The "Real" Cafe:	7:45 - 8:45
City Park:	9:15 - 9:45
Mini Mall:	10:00 - 11:00
Art Studio:	11:10 - 12:00
Ralph's:	12:10 - 1:15
Dispatch:	2:00 - 4:00
Marina:	4:15 - 5:15

Figure 1: Participants were given a specific task location and duration to add to each of ten agendas. The map was the same for all agendas.

based on the map in Figure 1 that could represent either one-time appointments or recurring ones. The third party also created one extra task (location and duration) for participants to add to each of those agendas. We required: 1) appointments be at a variety of times within the agenda, near and far from other nearby locations; 2) appointment times be separated by *time gaps* of at least as much time as to travel from one location to the next – the number of minutes to travel from one location to the next equals the number of dots along the shortest path; 3) each new task be a unique location so that we could easily distinguish between the agendas.

One agenda was used as the example explained to participants. The 10 remaining agendas are shown in Table 1; the tasks to add are in Table 2. The agendas are diverse in terms of when tasks can be added and how spatially close those tasks are to other appointments. Task locations range from close by (4 steps) to very far (15+ steps) from the nearest appointment location. Half of the agendas allow for the new meeting to be included in time gaps early in the day (between the first two or second two meetings), and half require the meeting to be in time gaps in the second half of the day, and some have two times. The time gaps when tasks fit into the agendas are darkened in Table 1. The tasks could also be added to the end of the day for a longer (suboptimal) agenda.

3.2 Experimental Setup

We used SurveyMonkey[1] to deploy our survey and collect our results. Participants who consented to take our experiment were shown one example agenda along with our map (Figure 1). They were told that the number of dots along the path from one appointment to another was equal to the number of minutes it would take to travel there. Then, they were given one new task location and duration and were asked to choose the *first* time that it would fit in the agenda without disrupting the existing appointments. Only participants who answered this question correctly could complete the study for payment of $5. This helped us ensure that participants were not just clicking randomly and understood the experiment.

After the example agenda, participants were shown a random ordering of 10 more agendas (Table 1) on the same map and with the instruction to find the *most convenient* time to put the new task (Table 2) without disrupting the rest of the appointments. We used the word "convenient" in order to allow participants to choose any time slot where it would fit or even at the end of the day instead of within the agenda. For each agenda, participants were then asked two questions. First, they used a drop down menu to select the time to add the new task including at the end of the day. Second, they used radio buttons to indicate the order that they checked the time gaps before deciding on a suitable time. This second question allowed us to assess the demonstrated strategy that participants employed when searching the agenda.

After the 10 agendas, participants saw a final page asking them to indicate which strategy they used (e.g., starting from the beginning of the day, choosing the closest locations, choosing the longest time between meetings, or "other" with space for explanation). They then entered their email address to get paid for the study, and saw a final thank you page to indicate that the study was done.

3.3 Participants

Participants were recruited from a small liberal-arts university using mailing lists for students, faculty, and professionals interested in technology, information systems, data analysis, and business at the university. Sixteen people successfully completed the study. While this sample was fairly small, as the results later show, the low variance in participant responses precluded the need for a larger scale study. Being at a university, we assumed that all of them had engaged in scheduling appointments and tasks prior to the study. We did not collect any further demographic information about the participants. We excluded one participant who indicated that they had determined that they did not have to check the agenda if they chose "after the last meeting". This participant was paid but was removed from the study because, while technically correct, they did not provide the data we were looking for in our analysis.

3.4 Analysis and Measures

For each participant, we collected their *responses* for when the new task would fit into the agenda, their reported *order* that they checked the time gaps, and their *reported strategy* from the last page of the study. Participants reported using three main strategies - starting at the *beginning* of the agenda, *closest* appointment to task, *longest* time gap - and all reports of "other" were some combination of these three. Based on the reporting, we also manually coded each agenda with the strategy that they *demonstrated* based on the first time gap in the *order* they reported checking (beginning, closest, or longest). We did mark a demonstration as two or more strategies if we could not distinguish a single strategy. For example, participants who chose a time gap that was both the first of the day and the closest to the task were coded as both beginning and closest.

[1]https://www.surveymonkey.com

Agendas and Their Respective Clusters

A	Library	35	Police	15	Cafe	10	Dispatch	50	Cons. Site	10	Bank	15	Real Cafe	30	Theater
B	Cafe	35	Beach	15	Farm	5	Cons. Site	15	City Park	10	Home	30	Post Office	15	Art
C	Theater	15	Real Cafe	30	City Park	15	Mini Mall	10	Art	10	Ralph's	45	Dispatch	15	Marina
D	Ralph's	30	Library	5	Art	15	Cafe	30	Marina	15	Health C.	15	Amphi.	15	Real Cafe
E	Beach	15	Bank	60	Cons. Site	15	Home	5	Gas	10	Lockers	15	Health C.	15	Lockers
F	Cons. Site	15	Mini Mall	20	Fire	10	Police	15	Home	10	City Park	45	Marina	15	Ralph's
G	Art	15	Obs. Room	15	Library	15	Cafe	15	Marina	45	Cons. Site	10	Orange Plaza	20	Real Cafe
H	Ralph's	5	Library	35	Police	15	Cafe	10	Dispatch	50	Cons. Site	10	Bank	15	Real Cafe
I	Theater	15	Real Cafe	30	City Park	15	Mini Mall	10	Art	10	Ralph's	45	Dispatch	15	Marina
J	Bank	10	Library	20	Police	10	Theater	30	Beach	35	Post Office	10	Amphi.	5	Cons. Site

Table 1: Each agenda included eight appointments with a time gap between each pair. The times when the new task fits into the agenda are darkened. The color will be discussed with respect to our heuristic algorithm.

	Task to Add	Duration	Avg. Gaps Checked	Accuracy
A	City Park	20	4.71	0.73
B	Fire Station	10	4.82	1.0
C	Home	10	4.19	1.0
D	Post Office	10	4.72	0.87
E	Farm	45	3.04	1.0
F	Dispatch	15	5.62	0.87
G	Health Center	10	5.08	0.93
H	Gas Station	10	5.13	0.93
I	Pretend City Hall	10	5.46	0.93
J	Real Cafe	10	5.39	1.0

Table 2: Participants were asked to add a different task location with a set duration to each unique agenda. The right columns show the average number of time gaps checked per agenda out of 7.

From this data, we also computed the:

- *accuracy*, or whether the time gap chosen fit the task,
- *majority strategy*, or the most common strategy each participant demonstrated across the ten agendas, and
- *number of checked time gaps* for each agenda.

We will use this data to test whether reported strategies match demonstrated and majority strategies, whether closest and longest strategies lead to fewer checked time gaps compared to those who employ the beginning first strategy, and, later, whether our heuristic algorithm searches time gaps in a similar order to our participants.

3.5 Results

We report $p < 0.05$ results using * and $p < 0.01$ using **.

3.5.1 Differences in Agenda. We performed a REML Mixed Model analysis to predict the number of checked time gaps based on the agenda, with participant id as a random effect. As expected, we found that there is a significant effect of agenda on number of checked time gaps ($F(9, 478) = 13.69$**). Since many participants checked the agenda from beginning to end, they checked fewer gaps when the task fit early in the day. We performed the same REML analysis with accuracy as the dependent variable and found no significant effects of any of our independent variables including agenda. Participants were accurate at finding valid time gaps irrespective of the agenda (Table 2).

3.5.2 Time Gaps Checked By Strategy. In order to understand the relationship between the number of time gap checks and the strategies that were employed, we performed a Mixed Model analysis with demonstrated strategy, reported strategy, and majority strategy as fixed effects, agenda as repeated effect, and participant id as random effect. We found statistically significant effects for two of the demonstrated heuristic strategies (closest $F(1, 120) = 14.67$**, longest $F(1, 132) = 38.97$**), but not beginning ($F(1, 132) = .94$). We also did not find a significant effect of majority ($F(1, 132) = 0.04$) or reported strategy ($F(1, 132) = 1.73$). The reported and majority strategy were not significant, because we found that participants used different strategies depending on the agenda (i.e., their reported strategy did not correlate with demonstrated one).

When we remove the non-significant effects, we find that the model equation that predicts the number of time gaps[2] is

$$2.63 - 1.72*(\text{longest}) - 1.01*(\text{closest}).$$

In other words, utilizing a longest time strategy decreases the number of time gaps checked per agenda by 1.72, while utilizing the closest strategy lowers the number by 1.01 checks. This finding is confirmed with an ANOVA comparing the three strategies; there is a statistically significant difference between number of time gaps checked by participants using different demonstrated strategies ($F(3, 146) = 18.32$**). A Tukey test shows that using the beginning strategy resulted in 3.54 checks vs. 2.63 for the other strategies (*).

3.5.3 Results Summary. To summarize, participants used strategies of finding closer locations to the new task or longer meeting times with the goal of reducing the number of time gaps to check. Those who employed these strategies were able to reduce the number of checks they performed by 25% on average compared to starting at the beginning of the day. Given the boost in performance for these strategies, we focus on creating a heuristic scheduling algorithm that can mimic the results.

4 HEURISTIC SCHEDULING ALGORITHM

An optimal solver is guaranteed to find the shortest scheduled agenda for a set of appointments and tasks. However, as the number of activities on the agenda increases, so does the time to find a new agenda. The above human participant study suggests two heuristics for directing the search for inserting a task in an existing agenda:

[2] *Longest* and *closest* are binary variables for whether the strategy was demonstrated.

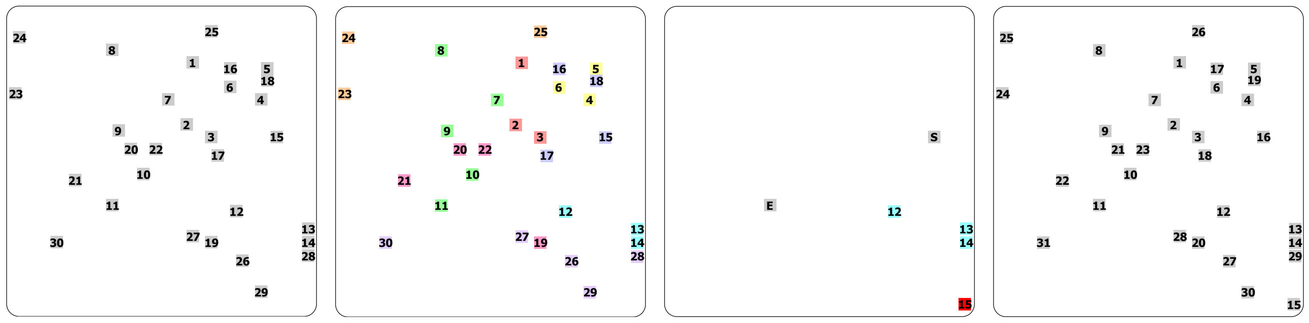

(a) Original agenda (b) Clustered agenda (c) Partial agenda to schedule (d) Final agenda

Figure 2: Agendas at different stages of the rescheduling process. (a) The original agenda, with locations numbered in the order in which they are visited. (b) The original agenda after spatio-temporal clustering, with colors marking the different clusters. (c) The partial agenda that is rescheduled first, including the new task in red, the closest cluster to the new task, and the two surrounding appointments. (d) The final agenda after the new task is added.

one based on time gaps in the agenda, like the *longest* strategy; and one based on spatial proximity, like the *closest* strategy. We focus here on the latter, which we deem to be the more challenging strategy to replicate; we leave the former to future work.

Accordingly, we propose a heuristic for adding tasks to agendas that focuses on the spatial proximity of the new task to appointments and tasks already on the agenda. The approach identifies the temporal portion of the agenda that is spatially the most similar to the new task, and then performs targeted rescheduling with that smaller group of activities and the new task. This approach draws upon and expands other work that proposes opportunistic rescheduling algorithms to try to minimize the amount of rescheduling that must be done. Foundational work by [12] and [13], for example, focused on finding similar jobs in job rescheduling tasks but doesn't consider spatial constraints. Similarly, [22] focuses on constraining vehicle routing tasks spatially for rescheduling but without time constraints.

Intuitively, our heuristic finds portions of the agenda that are likely to support the new task; because spatial-closeness implies that navigation time between the activities is shorter, smaller time gaps in the agenda are more likely to fit the new task. Additionally, because the number of activities to reschedule is much smaller than the entire agenda, the speed of solving the new agenda should greatly decrease. We next discuss the components of our heuristic scheduling algorithm, provide an example illustrating how it works, and then present results to compare it with an optimal scheduler.

4.1 Algorithm Design and Implementation

There are three components to the heuristic scheduling algorithm: (1) clustering the existing agenda so that tasks and appointments are binned according to spatio-temporal proximity to the new task; (2) finding the optimal solution to the subset of activities (including the new task) within the cluster closest to the new task; and (3) expanding the search area if no solution is found, repeating the process until a valid agenda is reached. The heuristic approach is described in prose below; pseudocode also appears in Algorithm 1.

```
Data: current_agenda, new_task
Result: new_agenda
1 clusters = warped-k-means(current_agenda);
2 new_task_cluster = clusters.cluster(new_task);
3 while partial_agenda == null do
4     partial_agenda =
        solve-optimally(new_task_cluster.tasks() ⋃ new_task);
5     new_task_cluster.add(new_task_cluster.prior-cluster(),
        new_task_cluster.next-cluster());
6 end
7 return current_agenda.update(partial_agenda);
```

Algorithm 1: Pseudocode for heuristic scheduler.

4.1.1 Spatio-Temporal Clustering. Our approach requires a clustering in both space and time. Space alone is not sufficient, since activities spatially near one another may not be scheduled close in time. Similarly, time alone is not sufficient, since activities scheduled near to one another may not be spatially close. To cluster the activities, we thus use Warped K-Means [16], an unsupervised clustering algorithm that considers not only spatial similarity but also the sequence of the data points[3]. This consideration of sequence provides an implicit consideration of time in the classification. Distance between data points can be provided by any appropriate distance function based on the point's x and y coordinates[4]. To algorithmically determine the number of clusters, we incrementally increase the number of clusters until the average silhouette of the clusters is no longer increasing. The number of clusters that results in the maximum silhouette at that point is used as the number of clusters for the heuristic scheduler.

Thus, the first step of our heuristic scheduling algorithm when it receives an existing agenda (Figure 2(a)) and a new task is to cluster

[3]We used the implementation of Warped K-Means provided by the authors at https://luis.leiva.name/wkm/

[4]While technically any distance function can be used, because of K-Mean's emphasis on the location of cluster centroids, appropriate distance functions will be in the X-Y space.

the existing agenda using Warped K-Means (Figure 2(b)). It then classifies the new task into an existing cluster to find the closest cluster for that task and proceeds with the next step, below.

4.1.2 Schedule Optimization. To generate an agenda optimally, we use a Mixed Integer Programming paradigm adopted from Kara and Derya ([14]). It provides a formulation that for minimizing agenda length given both tasks and appointments. To solve the problem, we use Google's OR-TOOLS[5] wrapper to the MIP solver COIN-OR Branch and Cut [8]. We use this solver because it has been successful with the problem formulation we consider here [14], and because the OR-TOOLS wrapper is publicly available, widely-used and effective.

Given the new task's cluster, our algorithm extracts a partial agenda consisting of only the activities in that cluster (Figure 2(c)). It then attempts to optimally schedule that partial agenda with the new task, constrained by the partial agenda's surrounding locations, start and end times. Intuitively, this means the algorithm is checking to see if the new task can be fit into that partial agenda. If it can, the resulting partial agenda is used to update the overall agenda, generating a solution to the problem (Figure 2(d)).

4.1.3 Expanding Incrementally. If it cannot fit the new task into the first cluster, the heuristic incrementally expands the partial agenda to include the surrounding clusters, stopping when it finds a solution. If a solution is not found within a subset of the agenda, the upper bound on the overall agenda's end time is dropped in order to allow the new task to potentially add on to the end of the current agenda. Thus, the algorithm guarantees that the shortest possible solution is always found.

5 EVALUATION

In order to assess the heuristic scheduler, we performed two experiments. The first experiment compared the performance of our heuristic scheduler against an optimal rescheduling algorithm. The second experiment compared the heuristic's performance with the strategies that people used in the human-subject experiment.

5.1 Comparison to Optimal Scheduling

5.1.1 Agenda Design. To compare the different scheduling approaches, we generated random agendas of 30 activities each. To successfully satisfy the agenda, the agent is required to "complete" each activity by going to its location. The first activity was considered the agent's starting place; the agent must also return to the starting location after executing all tasks.

Each activity's location was randomly selected on a 50x50 grid with the condition that no two activities were in the same location. Each activity had the negligible duration of 1; note that other durations could be used without changing our results. A specified percentage of activities designated as appointments and were given start and end times at which the agent needed to be at that task's location. These appointment times were constrained to make sure that the agenda always had a valid solution, using Manhattan distance to calculate the travel time between tasks. We considered appointment times for [20%, 100%] of tasks, with increments of 10%, with 100 agendas per appointment percentage. This resulted in a total of 9 sets of 100 agendas each. For each agenda, a new task duration and location was randomly generated, also with the constraint that it was not in the same location as any existing activity.

[5]http://developers.google.com/optimization/

5.1.2 Experimental Setup. For each agenda, we considered two rescheduling strategies: a full optimal reschedule; and rescheduling using our heuristic scheduler. We recorded the time it took for each algorithm to find a solution. We also recorded the final agenda length ultimately found by each rescheduling strategy.

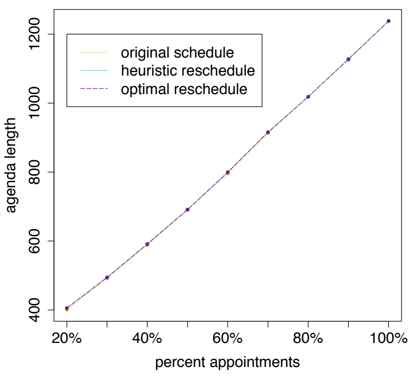
5.1.3 Measures. We measured three variables:

- *agenda length*, the total duration of the agenda.
- *agenda scheduling time*, the time it took to schedule (or reschedule) the tasks and appointments of the agenda.
- *agenda distance*, the distance of the rescheduled agendas when compared to the original agenda. To calculate this, we use Definition 4.2 of [9], which defines plan and schedule similarity metrics.

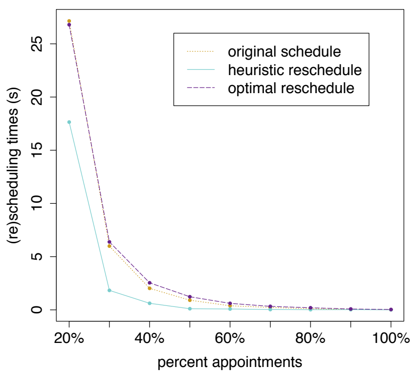
5.1.4 Results. The results show a clear benefit of the heuristic rescheduling strategy vs. optimally rescheduling. Figure 3(a) shows the length of the agenda using heuristic rescheduling vs. optimal rescheduling. Not surprisingly, the length of the plan increases with the number of appointments, because fewer and fewer tasks can be "fit in" between appointments. Importantly, as described earlier, the heuristic rescheduling strategy always finds the optimal solution. Figure 3(b), however, shows that it typically finds the optimal solution in much less time. The computation time generally decreases for all algorithms as the percentage of tasks decreases and percentage of appointments increases because there are fewer tasks to schedule. However, the heuristic rescheduling strategy consistently uses less time to reschedule than both the original (optimal) agenda construction, as well as optimally rescheduling.

Together, these two results occur because appointments within an agenda cannot be moved, so the overall length of the optimal agenda will not change as long as a task can be fit in any time gap before the end of the last appointment. As a result, the optimal scheduler often performs unneeded computation for many agendas, while the heuristic strategy targets its computation to places where the task can intuitively be inserted. Overall, these two graphs highlight our early performance claims of the paper: that our heuristic finds the best solution in less time.

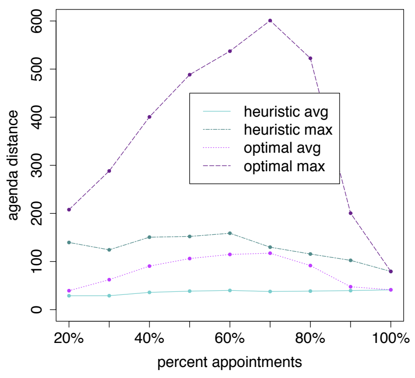
Additionally, we analyzed the agendas produced by the two rescheduling conditions to see which generated agendas that were most similar to the original agenda, with the assumption that closer is better for human understandability of the new agenda. The heuristic rescheduling strategy far outperformed the optimal rescheduling strategy in this regard, both when considering the maximally distant agenda for a given percentage of appointments, as well as the average distance of agendas for a given percentage of appointments. Intuitively, this is because the heuristic strategy moved around fewer tasks than the optimal strategy, which sometimes completely regrouped tasks in between appointments. This indicates that the agendas generated by our approach should be more understandable by human partners than those generated by the optimal strategy.



(a) Agenda length



(b) Agenda scheduling time



(c) Agenda distance

Figure 3: (a) Length of the original agenda after our heuristic rescheduling, and after optimally rescheduling. Note that the heuristic rescheduling agenda length always equals the optimal rescheduling agenda length. (b) Scheduling time to generate the original agenda; to add the new task using our heuristic rescheduling; and add the new task using optimal rescheduling. (c) Agenda similarity between the original agenda and the new heuristic agenda or the new optimal agenda. The graph shows both the maximum distance of the agendas with those percent appointments, as well as the average.

5.2 Validation of Heuristic Strategy

In order to further validate that people use similar strategies to our heuristic and execute that strategy in a similar order to our heuristic, we compared our heuristic scheduler to the strategies of our human-subject participants on the original experimental agendas.

5.2.1 Heuristic Algorithm Implementation. The main components of the heuristic scheduling algorithm, including spatio-temporal clustering and cluster-based heuristic schedule optimization, performed the same in the human-subject domain as they did for the larger grid agendas tested above. The single substantive difference for this task was the distance measure between locations – Manhattan distance is not appropriate because of the non-grid-like roads in the map. We used a Sammon function[6] to transform the appointments' X-Y coordinates on the map to a new coordinate space in which the Manhattan distance approximates travel distance.

Our heuristic algorithm, as before, spatially clustered the appointments when scheduling the new task. Each transition to a new color in Table 1 shows a transition to a new cluster. For example, agenda A had four clusters, each with two appointments while agenda H had 5 clusters. An average of 2.5 appointments were determined to be in each cluster. When adding the new task, the heuristic scheduler checked the closest cluster to the task first (pink), then expanded in both earlier and later directions to include more clusters. The first expanded cluster was blue, then green, and then yellow until it found to find a valid agenda. The darker square(s) in each row is the time gap where the new task fits (end of the day time gap not shown).

5.2.2 Results. Our heuristic algorithm found a correct time gap in the first cluster in 8 out of 10 agendas. Agendas A and F required three clusters to be checked as can be seen in the darkened spaces colored green on Table 1. We report $p < 0.05$ results using * and $p < 0.01$ using **.

We compared the order that participants checked the different time gaps to our heuristic algorithm. We performed a linear regression to predict the participant's time gap search order based on the heuristic cluster order, as well as the strategy that was used (beginning, closest, and longest) and the interactions of the heuristic cluster order with the strategy. The $R^2 = .27$, and the model was found to predict the participant order statistically significantly ($F(7, 494) = 25.76^{**}$). In particular, all variables tested were statistically significant - intercept ($t(1) = 3.47^{**}$), heuristic cluster ($t(1) = 1.95$), beginning ($t(1) = -2.35^{**}$), closest ($t(1) = -6.0^{**}$), longest strategies ($t(1) = 10.4^{**}$), and interactions cluster-beginning ($t(1) = 3.78^{**}$), cluster-closest ($t(1) = 5.45^{**}$), cluster-longest ($t(1) = -9.82^{**}$). This indicates that the cluster heuristic does predict the participant order, but the predictions change depending on which strategy was used.

A graph of the heuristic cluster vs. participant order correlations for those who use the closest strategy is shown in Figure 4. Each line is the regression for one agenda. Note that participants did not employ the strategy at all on two agendas. All but one agenda have positive correlations between the cluster and participant orders. The negative correlation is only supported by 3 data points from a single participant. These results indicate that the order that the heuristic chooses to evaluate time gaps does correspond to the participants' closest location strategy.

6 DISCUSSION AND CONCLUSIONS

In this paper, we have introduced an algorithm for human-centered decision support for agenda scheduling. We began by running a human participant experiment that investigated the ways that

[6]Specifically, the Sammon library in R.

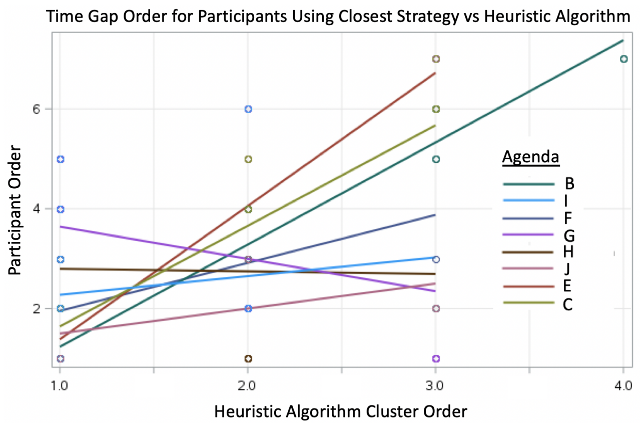


Figure 4: There is a positive correlation between the heuristic cluster order and the participants' order for evaluating time gaps in agendas for all but one agenda.

humans develop and modify agendas on their own. We then took inspiration from human strategies to create an automated, optimal, heuristic rescheduler for updating agendas to add new tasks that is both similar to how people doing scheduling, as well as fast and efficient. This results in a system that can support people in a timely manner without sacrificing optimality.

One of the key reasons why the heuristic scheduler is so successful is because it takes advantage of a characteristic of agendas that people, whether consciously or unconsciously, take advantage of while rescheduling. In agendas with appointments, there is bound to be free time in the agenda where extra tasks can be added; the need for full replans is very rare. This flexibility can be taken advantage of by either looking at activities that are spatially close to the new task, as we do here, or at large temporal gaps in the agenda, as we will do in future work.

There are two other main aspects of work that we will focus on in the future. The first is studying how explainable and understandable people find the agendas generated by the heuristic scheduler. While we have every reason to believe that our approach performs well with this criteria, there is always the chance that human participants did not accurately represent their reasoning strategy or process, implying that our heuristic does not match as well with human performance as our results currently indicate. The second is expanding our validation to other types of agenda modification, such as deleting a task or selecting the location and time that a task can be performed. For example, if the task is going to the ATM, there are many possible ATMs that one could go to depending on locations and times of appointments on the agenda (i.e., [21]). As before, our current heuristic can extend to these cases naturally; however, we still plan to perform additional evaluations to make sure that the understandability extends to these cases as well.

ACKNOWLEDGEMENTS

This work was partially funded by the Office of Naval Research (LH). It was also made possible through an Education Partnership Agreement (EPA-NRL-ONR-18-001) between Carnegie Mellon University, the Office of Naval Research and the Naval Research Laboratory.

We also want to thank Shelley McGee, Alex Mendelsohn and Simon Mendelsohn for their contributions to this work.

REFERENCES

[1] Joshua A. Auld. 2011. *Agent-based Dynamic Activity Planning and Travel Scheduling Model: Data Collection and Model Development.* Ph.D. Dissertation.
[2] Sarah Beech, Erik Geelhoed, Rachel Murphy, Julie Parker, Abigail Sellen, and Kate Shaw. 2004. The Lifestyles of Working Parents: Implications and Opportunities for New Technologies. (04 2004).
[3] Pauline M Berry, Melinda Gervasio, Bart Peintner, and Neil Yorke-Smith. 2011. PTIME: Personalized assistance for calendaring. *ACM Transactions on Intelligent Systems and Technology (TIST)* 2, 4 (2011), 1–22.
[4] Tad T Brunyé, Shaina B Martis, and Holly A Taylor. 2018. Cognitive load during route selection increases reliance on spatial heuristics. *Quarterly Journal of Experimental Psychology* 71, 5 (2018), 1045–1056.
[5] Mike Brzozowski, Kendra Carattini, Scott R. Klemmer, Patrick Mihelich, Jiang Hu, and Andrew Y. Ng. 2006. groupTime: Preference Based Group Scheduling. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '06).* 1047–1056.
[6] Justin Cranshaw, Emad Elwany, Todd Newman, Rafal Kocielnik, Bowen Yu, Sandeep Soni, Jaime Teevan, and Andrés Monroy-Hernández. 2017. Calendar. help: Designing a workflow-based scheduling agent with humans in the loop. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems.* ACM, 2382–2393.
[7] Scott Davidoff, John Zimmerman, and Anind K. Dey. 2010. How Routine Learners Can Support Family Coordination. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10).* 2461–2470.
[8] John Forrest, Ted Ralphs, Stefan Vigerske, LouHafer, Bjarni Kristjansson, jpfasano, EdwinStraver, Miles Lubin, Haroldo Gambini Santos, rlougee, and Matthew Saltzman. 2018. coin-or/Cbc: Version 2.9.9. (July 2018).
[9] Maria Fox, Richard Howey, and Derek Long. 2006. Exploration of the robustness of plans. In *AAAI.* 834–839.
[10] Tommy Gärling, Tomas Kalen, Joakim Romanus, Marcus Selart, and Bertil Vilhelmson. 1998. Computer simulation of household activity scheduling. *Environment and planning A* 30, 4 (1998), 665–679.
[11] Matthew Gombolay, Reed Jensen, Jessica Stigile, Sung-Hyun Son, and Julie Shah. 2016. Apprenticeship scheduling: Learning to schedule from human experts. AAAI Press/International Joint Conferences on Artificial Intelligence.
[12] Geir Hasle and Stephen Smith. 1996. Directing an Opportunistic Scheduler: An Empirical Investigation on Reactive Scenarios. *IFIP Advances in Information and Communication Technology* (11 1996), 1–11.
[13] H Henseler. 1995. From reactive to active scheduling by using multi-agents. In *Artificial Intelligence in Reactive Scheduling.* Springer, 12–18.
[14] Imdat Kara and Tusan Derya. 2015. Formulations for minimizing tour duration of the traveling salesman problem with time windows. *Procedia Economics and Finance* 26 (2015), 1026–1034.
[15] Joseph Kim, Christopher J Banks, and Julie A Shah. 2017. Collaborative planning with encoding of users' high-level strategies. In *Thirty-First AAAI Conference on Artificial Intelligence.* 955–962.
[16] Luis A. Leiva and Enrique Vidal. 2013. Warped K-Means: An algorithm to cluster sequentially-distributed data. *Information Sciences* 237 (2013), 196–210.
[17] Kwan Hui Lim, Jeffrey Chan, Christopher Leckie, and Shanika Karunasekera. 2016. Towards next generation touring: Personalized group tours. In *Twenty-Sixth International Conference on Automated Planning and Scheduling.* 412–420.
[18] Pragnesh Jay Modi, Manuela Veloso, Stephen F. Smith, and Jean Oh. 2005. CMRadar: A Personal Assistant Agent for Calendar Management. In *Agent-Oriented Information Systems II.* Springer Berlin Heidelberg, Berlin, Heidelberg, 169–181.
[19] Carman Neustaedter, AJ Brush, and Saul Greenberg. 2009. The calendar is crucial: Coordination and awareness through the family calendar. *ACM Transactions on Computer-Human Interaction (TOCHI)* 16, 1 (2009), 1–48.
[20] Ioannis Refanidis, Christos Emmanouilidis, Ilias Sakellariou, Anastasios Alexiadis, Remous-Aris Koutsiamanis, Konstantinos Agnantis, Aimilia Tasidou, Fotios Kokkoras, and Pavlos S Efraimidis. 2014. myVisitPlanner GR: Personalized itinerary planning system for tourism. In *Hellenic Conference on Artificial Intelligence.* Springer, 615–629.
[21] Ioannis Refanidis and Neil Yorke-Smith. 2010. A constraint-based approach to scheduling an individual's activities. *ACM Transactions on Intelligent Systems and Technology (TIST)* 1, 2 (2010), 1–32.
[22] Paul Shaw. 1998. Using Constraint Programming and Local Search Methods to Solve Vehicle Routing Problems. In *Principles and Practice of Constraint Programming — CP98*, Michael Maher and Jean-Francois Puget (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 417–431.
[23] Luis Zabala, Cristina Perfecto, Armando Ferro, and Juanjo Unzilla. 2001. Integrating Automatic Task Scheduling and Web-based Agenda in a Virtual Campus Environment. *International Conference on Information Technology Based Higher Education and Training* (01 2001).