AI Planning for Organizing Personal Schedules

Pece Milosev
Netcetera
Skopje, Macedonia

Nevena Ackovska
II, PMF, UKIM
Skopje, Macedonia

Abstract

Automated planning and scheduling is one of the main branches of AI. It was a very hot topic back in the nineties, and theoretically it hasn't changed much in the last several years. In practice, it has proven to be a crucial component of many industrial and scientific processes, but has not yet become integral part of our daily lives. On the other hand, due to the advances in the information technology in the last decade, we are witnessing a drastic change in the daily human-computer interaction. The internet has become main information source, and with the growth of the mobile industries, more and more people are using their smartphones as their tool for retrieving this information. In this paper we will analyze the possibilities of using these advances to make a planning system that will optimize our daily activities.

I. Introduction

In the past few decades, with the rise of the technology, the daily life of the ordinary people has significantly changed. On the expense of easier physical life, the mankind is aggressively pushed towards more complicated mental activities. Due to the globalization, it is quite normal to see one person in his office today, preparing for his tomorrow's business meeting on the other side of the globe, and discuss with him about the ski weekend he is planning to have with his family once he is back. All of this increases the need of a non-trivial activity planning, and this need is without doubt expanding to the vast majority of people, which is also indicated by the increased demand of personal assistants [6][9].

On the other hand, the development of new technologies has provided us with many tools that can assist our daily lives. Mobile computing devices, like PDAs and smartphones, are definitely one category of tools that is widely used today for communication and information access [5][14].

Although the problem of organizing personal schedules might seam like a trivial timetabling problem, there are few issues like *personalization, environment and plan execution uncertainty, etc.*; that raises the need of various other techniques like *machine learning and continuous planning* to be applied. The goal of our research is to formally define the problem of organizing personal schedules, and check the possibilities of solving this problem by either using a general-purpose or domain-specific planning techniques.

II. Formalization of the problem

We can describe the problem as a finite set of events E, which need to be scheduled within a finite set of time intervals T.

$$E = \{e_i | i \leq n; i, n \in \mathbb{N}\} \tag{1}$$

$$\begin{gathered} T = \{[t_{2j-1}, t_{2j}] | j \leq m; p < q \Rightarrow t_p \leq t_q\} \\ j, m, p, q \in \mathbb{N} \end{gathered} \tag{2}$$

The events here represent different things that need to happen at some time (e.g. meetings, actions, etc). If we take a closer look at these events, we will see that, in order to make the schedule, we are more interested in their properties, rather than the exact kind of the events.

The social aspect, for example, is a property that we do not consider when dealing with artificial agents. However, since our final agent executing the plan, wold be a human being, it is very important to take this property into account. Regarding the social aspect, we can define the set of events E as union of three subsets:

$$E = E^p \cup E^h \cup E^o \tag{3}$$

We will put into E^p all events that are related to the user's professional life. E^h will contain all events that are related to the user's hobbies or interests. Finally, all events that cannot fit into the previous two subsets can be placed into E^o. Such definition of the events set E will allow the system to better suite the users preferences regarding the social aspect of the events.

The space-time constraints are another very important property emerging from the physical laws of our world. We can partition the events set E into four subsets: *fixed place - fixed time, fixed place - variable time, variable place - fixed time, variable place - variable time*.

$$E = E^{(f,f)} \cup E^{(f,v)} \cup E^{(v,f)} \cup E^{(v,v)} \tag{4}$$

$$\begin{gathered} \forall a, b, c, d \in \{f, v\} : (a, b) \neq (c, d) \\ \Rightarrow E^{(a,b)} \cap E^{(c,d)} = \emptyset \end{gathered} \tag{5}$$

Since the output of the system should be a valid schedule of the events in the set E, we can define the solution as a function S over the events.

$$\forall e \in E : \exists t : S(e) = t \tag{6}$$

Where $t_p \leq t \leq t_q$ such that $[t_p, t_q] \in T$. We will additionally define a probability function $P(S)$ as function that gives us the probability of success of the schedule S. Obviously we would like to maximize this function.

In practice, of course, other properties might need to be analyzed as well, depending on the target group the system would be designed for.

A. Scheduling aspect of the problem

Scheduling addresses the problem of how to perform a given set of actions using a limited number of resources in a limited amount of time [11].

In our case, the set of actions is generalized to the set of events, while the main resource that needs to be shared between them is the user. For example, some common scheduling problems [11][21] might include:

- Educational Timetabling
- Transport Timetabling
- Employee Timetabling and Rostering
- Sports Timetabling
- Machine scheduling

Although all of these problems are at least NP-Complete, in practice many domain specific algorithms have been developed that successfully solve them [11][21][19][2].

In order to be able to take the timetabling approach, for solving our problem, first we need to transform the time from its continuous form into a discrete set of time windows. This means that we need to define the set T mentioned earlier. Although this is a trivial task for some of the problems mentioned above, since the set T is simply defined as a set of working times for the appropriate institution, it is much harder to define the "working time" of the user.

We can represent the user's will to do some task in some period of the day as a probability function $W(t)$, with 0 representing that the user will certainly not like to do the task at that moment and 1 representing that the user thinks that this is the best moment for the task. More precisely, the user might have different preferences regarding the task's properties. That is, for the social aspect defined above, we will have to define three such functions - one for each of the sets E^p, E^h and E^o. Since this does not affect the final complexity of the problem, we will continue our analysis on the single function $W(t)$. One example[1] of how this function might look like is shown on Figure 1.

The time can be transformed into a discrete form by finding a constant $w0$ which will represent the minimum acceptable value of the function $W(t)$. We will consider the problem solved only if (7) holds. For the example in Figure 1, this would mean that all events are scheduled between $t_0 \approx 6:00$ and $t_1 \approx 21:30$.

$$\forall e \in E : W(S(e)) \geq w_0 \tag{7}$$

Let us now define a value $p' = MIN(W(S(e)))$. In order the plan to succeed the user needs at least to be willing to do the tasks, therefore $P(S) \leq p'$. On the other hand, the minimum acceptable value for p' is w_0, since otherwise it would mean that some events are scheduled outside of the preferable time window. As we want to set the constant w_0 to its minimum value, in order to maximize the time window, we would choose it to be equal to the minimum probability of success of our plan, obtained by $S(e)$.

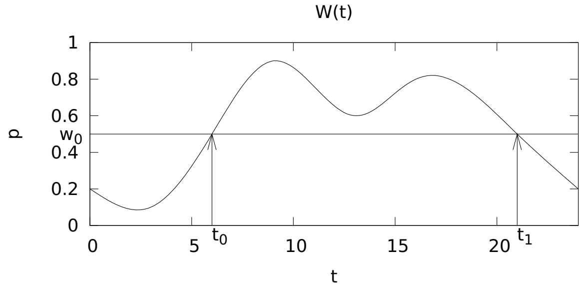

Figure 1: Example of the function $W(t)$ indicating that the user likes his tasks schedule between 6:00 and 21:30

So far, we have only considered the time and the user's will to accomplish some task as the main constraints for the scheduling problem. However, as we have already discussed above, all events happen at some place in some point of the time. Even more, we have defined different types of events regarding their space-time constraints. In order to build a successful system, we would need to include the space in the current model of the problem. Because we try to maximize the probability of success of our plan, we define a function $D(x)$ which gives us the probability that the user will manage to accomplish some task which is x distance units away from his current position. Theoretically, this function would have value of 1 only for the tasks happening at the user's absolute current position and 0 for all other tasks. Since such a function is impractical to work with, we would need to relax the definition of what does the user's current position means. This relaxation is highly dependent on the length of the time unit we would define for the task. For example, we can say that the user is not in the right place to attend a company meeting if he is two blocks away, but we would say that he is on the right place to visit Disney World if the user is anywhere around Orlando, Florida. This is mainly due to our subconscious reasoning about the different lengths of the time units. In order for an artificial system to be able to make such reasoning, we would need to provide this parameter explicitly. Therefore, we will have to define our function as $D(x, t)$ where the time parameter t will define the length of the single time unit for the task.

As $D(x, t)$ monotonically increases regarding the time parameter t, let us simplify our discussion by choosing a fixed[2] value for $t = t_k$. Now we can define our function $D(x) = D(x, t_k)$ and analyze the rest of the properties of

[1] The example shows an arbitrarily chosen function. In practice this function can be roughly defined by the user and then refined by applying different AI techniques, like machine learning, that would analyze the user's acceptance of the produced schedules.

[2] Check the next section for a discussion on how we can make use of different lengths of the time unit in order to come up with a better plan.

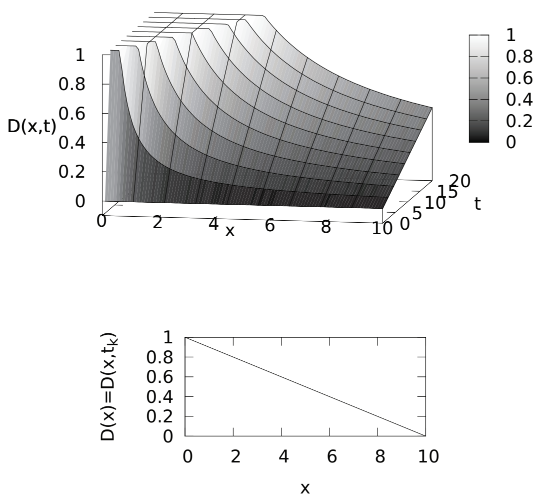

Figure 2: Example of the functions $D(x, t)$ and $D(x)$

this function (Figure 2). Obviously, the function $D(x)$ will monotonically decrease as x increases. We will consider the problem solved only if the probability that the user would be on place for the event exceeds some constant value d_0. Let us now define a probability value $p'' = MIN(X(e))$ where $X(e)$ gives us the distance between the user's current location and the location of the event e. Similarly, to the previous discussion regarding w_0, d_0 can be chosen to be the minimum probability of success of our plan. Finally we can see that the probability of success of the plan $P(S) \leq p' \times p''$. As we have chosen both w_0 and d_0 to be the minimum acceptable probability of success, the system should produce plans with $p' = w_0$ only if $p'' = 1$ and vice verse.

B. Planning aspect of the problem

Planning and scheduling are closely related problems. In a simple decomposition scheme, planning appears to be an upstream problem that needs to be solved before scheduling [11].

If we take a closer look at the two functions defined above, $W(t)$ and $D(x)$, we will see that, beside the other properties, there is one big difference between the two of them. Since the produced plan can not have any impact on the progression of the time, we can consider the function $W(t)$ as independent from the user's previous actions. However, the output of our system would map the user's position to some point in time. This has a direct impact on the probability of the future events. Therefore, besides the scheduling, we need a planning mechanism that will order the events in a way that will maximize the probability of success of the plan.

As the value of $D(x_{j+1})$ depends only on the value of $D(x_j)$ it comes naturally to try solving this problem using some algorithms based on the Markov property [12]. Partially Observable Markov Decision Process (POMDP) is one such algorithm that has been widely used in many various planning domains including robot control, abstract planning and personal assistive tools [11][10][4][8]. On the other hand, the cost function on which all Markov Decision Process (MDP) [17] algorithms highly depend might be hard to define, so we should also consider taking a different approach on solving the problem.

Another very popular planning algorithm that have given significant results is the Hierarchical Task Network (HTN) [11]. The HTN generates the plan by starting from the most abstract plan, containing only one general action connecting the initial state with the goal, and then decomposes this action into multiple, more precise, actions. The planning process finishes once all the general actions are decomposed, or a satisfactory plan is produced. Now, we have seen in the previous section, how our reasoning regarding some task depends on the length of the time unit used. We can make use of this property and run a HTN to first come with a plan regarding the more abstract tasks (in a manner of the length of the time unit) and then decompose this task by introducing the more concise tasks. If we use the same example from above, a HTN would first come with a general plan of going to Orlando next week, then refine this plan by introducing the tasks of visiting Disney World for the weekend, and finally make a plan of living the apartment one hour earlier for the airport, in order to have time to visit the Apple's shop and perhaps by a new fancy gadget.

Before continuing our discussion on which algorithm to use, let us first examine the complexity of the problem. The function $D(x)$, as it is currently defined, does not give us much useful information about the final plan, but it rather provides information regarding the possibility to reach each of the events in E from the current position. If we order the events, we can define a function $D(E)$ that provides the possibility to reach all of the events in E starting from the first event in the array (Figure 3). It turns out that in order to find the best permutation, we would need to provide a search in an n-dimensional space, where n is the number of events in E. Although not complete, meta-heuristic search algorithms have shown to be very efficient in solving problems where the search space grows exponentially [13].

Finally, the probability of success of the event $P(e_{i+1})$ depends on the probability of success of the previous event, $P(e_{i+1}) < P(e_i)$. This means that if there were too many events, it would make no sense to make the plan for all of them, as the success of the events at the end of the array would depend on too many preconditions. That is, we can define a value d_1 and only make the plan for the events where $P(e) \geq d_1$, thus reducing the search space in order to run a complete search algorithm. The probability functions $W(t)$ and $D(x)$ can further be used as a metaheuristics for guiding the search.

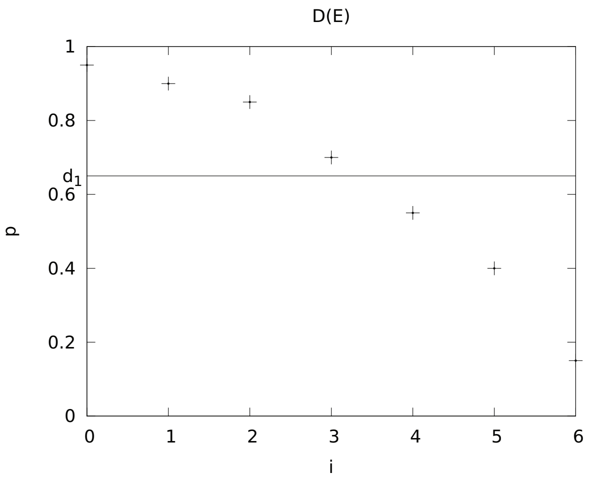

Figure 3: Example of the functions $D(E)$

III. Knowledge and information retrieval goes mobile

So far, we have mentioned the mobile computing devices few times, but didn't provided any information on their role in solving the problem defined above. One of the crucial points of our problem is that we try to plan and schedule the *personal* activities of the user. So we need a way to make the system as "close" to the user as possible, allowing him to define new tasks (and modify old ones) on-the-fly. As we have already mentioned, the number of people that use their smartphones as main tool for communication and information access raises rapidly [16][15][18]. On the other hand the processing power that modern smartphones can offer is comparable with the one of an average PC from the beginning of this decade. Targeting the system to be run on these devices is the best way to achieve this "close" relation with the user.

When building an intelligent agent, one of the most important things that we have to consider is the knowledge of the agent about its environment. Obviously, grater knowledge in most of the cases leads to a better plan [11][20], so we would like to model our system such that it would be capable of gaining knowledge regarding the user's environment. For example, a system that is capable of retrieving the railway schedule from the internet might automatically suggest the user when he would need to leave the office in order to catch the train, opposite to a system that does not have this information and is only capable of including user's final destination into the plan. One of the biggest advantages the smartphones have over a stationary computers, is the location awareness [3]. As we have seen in the previous section, location is the key factor for the success of the resulting plan. Even more, the location awareness can provide the system with a partial ability of confirming the plan execution, which will in turn enable plan modifications if the system notices that the plan is not entirely respected. It is already shown that this is one of the crucial issues in a calendar scheduling process [7].

Since analyzing the information retrieval and machine learning techniques, that can be used to refine the produced plans, is out of scope for this paper, we will only provide a diagram that will identify the key components of the system and their mutual integration (Figure 4).

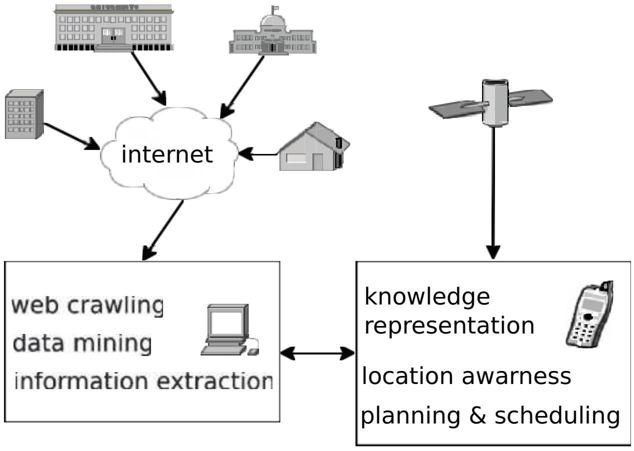

Figure 4: System overview

Finally, we should not underestimate the benefits of a good user interface design. Studies show that good user interface leads to better human-computer interaction, which results with a higher usability of the applications [1]. In our case, this would mean that the user would be willing to spend more time interacting with our system, thus helping the system to come up with a better plan.

IV. Conclusion

In the past two decades, research in the field of AI planning has resulted in several significant findings, such as the two planning systems, SPSS and Spike, used in the Hubble Space Telescope, the IBM's chess playing computer Deep Blue, as well as many different planners and schedulers used in the industry. However, beside these scientific and industrial breakthroughs, automated planning is still not used as a tool in our daily lives.

In this paper we have given a formal introduction to the problem of organizing personal schedules, which is one of the most obvious fields where automated planning can be used during our daily lives. We have also shortly discussed the several different algorithms that can be used for solving this problem, and analyzed few domain-specific characteristics that can be used as metaheuristics for guiding these algorithms.

Our future work will be focused on elaborating the theoretical framework provided in this paper, and providing a series of experiments which hopefully would lead the further development of automated planning towards its acceptance as a tool for assisting our daily lives.

REFERENCES

[1] Ben Shneiderman, Catherine Plaisant, Maxine S. Cohen, Steven M. Jacobs: “Designing the User Interface: Strategies for Effective Human-Computer Interaction (Fifth Edition).” Pearson (2009)

[2] Bokal Drago, Fijavz Gasper, Harej Bor, Taranenko Andrej, Zagar Klemen: “Hybrid and Modular Approach to Employee Timetabling.” The 7th International Conference on the Practice and Theory of Automated Timetabling (2008)

[3] “The Smartphone GPS Revolution.” http://gps.about.com/od/trends/a/smartphone_gps.htm

[4] Georgios Theocharous, Leslie Pack Kaelbling: “Approximate Planning in POMDPs with Macro-Actions.” Advances in Neural Information Processing Systems 16 (2003)

[5] Giselle Tsirulnik: “Smartphones enhance quality of life for consumers: Platform-A.” http://www.mobilemarketer.com/cms/news/research/3141.html

[6] Greg Holmes: “The Value Of Personal Assistant Services.” http://www.articlesnatch.com/Article/The-Value-Of-Personal-Assistant-Services/1298658

[7] Jean Oh, Stephen F. Smith: “Learning User Preferences in Distributed Calendar Scheduling.” The 5th International Conference on Practice and Theory of Automated Timetabling (2004)

[8] Jesse Hoey, Axel von Bertoldi, Pascal Poupart, Alex Mihailidis: “Assisting Persons with Dementia during Handwashing Using a Partially Observable Markov Decision Process.” National University of Singapore (2007)

[9] Joanna Whitmire: “The Future of Personal Assistant Jobs.” http://www.articlesbase.com/training-articles/the-future-of-personal-assistant-jobs-1452976.html

[10] Joelle Pineau, Geoff Gordon: “POMDP Planning for Robust Robot Control.” International Symposium on Robotics Research (ISRR) San Francisco, CA. (2005)

[11] Malik Ghallab, Dana Nau, Paolo Traverso: “Automated planning: Theory & Practice.” Morgan Kaufmann (2004)

[12] Markov A. A.: “Theory of Algorithms.” Academy of Sciences of the USSR, Moscow (1954)

[13] Martin Pelikan, David E. Goldberg, Fernando G. Lobo: “Time complexity of genetic algorithms on exponentially scaled problems.” Genetic and evolutionary computation conference (2000)

[14] Mika Raento, Antti Oulasvirta, Nathan Eagle: “Smartphones: an emerging tool for social scientists.” Sociological Methods & Research, Vol. 37 (2009)

[15] Mike Cronin: “Smartphones rise from gadget to tool of necessity.” http://www.pittsburghlive.com/x/pittsburghtrib/business/s_668540.html

[16] “Number of people accessing news and information on their mobile device more than doubles.” http://mttnow.com/index.php?pageID=77

[17] R. Bellman: “A Markovian Decision Process.” Journal of Mathematics and Mechanics 6 (1957)

[18] Sarah Perez: “Social Networking Now More Popular on Mobile than Desktop.” http://www.readwriteweb.com/archives/social_networking_now_more_popular_on_mobile_than_desktop.php

[19] Schaerf A.: “A Survey of Automated Timetabling.” Computer Science/Department of Software Technology CS-R9567, CWI - The Netherlands National Research Institute for Mathematics and Computer Science, Amsterdam (1996)

[20] Stuart J. Russell, Peter Norvig: “Artificial Intelligence: A Modern Approach (Second Edition).” Prentice Hall (2002)

[21] “The 7th International Conference for the Practice and Theory of Automated Timetabling.” http://www.asap.cs.nott.ac.uk/patat/patat08/patat08.shtml