GRP_19: Event Planner

Alex Panfilov (21MP27)

Derek Peene (20DSP1)

Adam Qian (17AWQ)

George Salib (20GS36)

Course Modelling Project

CISC/CMPE 204

Logic for Computing Science

October 31, 2022

Abstract

Create an "event planner" that generates a schedule of events based on activity input from the user. We factor in variables such as: weather, availability of group members, rank of activities, etc., to generate an activity itinerary tailored to the group's needs. This means that it will factor in the availability of every single group member, and will create a schedule allowing for all activities to take place in which every single group member can attend if possible. For example, if the weather is bad and there is an availability gap at a given hour, indoor events such as the movies could be good choice to schedule at said time, but if the weather is good, evening walk might be better. To increase complexity, we could create stipulations such as "all of these activities need to be scheduled" to make it not just a best-fit activity planner, but a planner for a list of things. Thus, it could be used not just for groups who want to hang out, but for organizations who need all the meetups to happen for an event at some point.

Propositions

Let it be known that i represents a person, j represents a given activity, and k represents a given time

- $X_{j,k}$ will represent an activity is possible, and is true if the activity j is feasible at a given time k
  - E.g. $X_{1,1}$ is true if activity 1 (going to the movies) is a possible activity at a time 1
- $T_{i,k}$ will represent timing, and is true if person i is available at time k
  - E.g. $T_{2,1}$ is true if person 2 is available at time 1. Time 1 could represent an hour starting at 1pm
  - Hence for every person, there will be a matrix of available times of size p x 24, where p is the number of participants we are working with, and 24 for the hours of the day
- Q_j will represent if the activity j is indoors, in which case weather conditions do not matter
  - E.g. If j is indoors, then Q_j is true
- W_k will represent weather conditions, and is true if the weather conditions are clear for the given time
  - E.g. W_2 could represent clear weather, and is true if it is clear at time 2
- S_k will represent a scheduled event, and is true if an event has been scheduled at a given time k
  - E.g. S_2 could represent has been put into the schedule at time 2

Constraints

1. Availability holds when $A_{j,k} \leftrightarrow (T_{1k} \wedge T_{2k} \wedge ... \wedge T_{nk})$

   This means that an activity can only hold iff every single person is available at a given time, thus everyone has time to attend the event.

   For example, if we say we choose noon time and check that everyone is available at that time, then availability holds, however, even if one person is busy at said time, the availability doesn't hold.

2. An activity holds when $X_{j,k} \leftrightarrow (A_k \wedge (W_k \vee Q_j))$

   This means that an activity can take place iff everyone is available and either the activity is indoors or the weather is clear.

   For example, if at a given time everyone is free, and the activity can take place without weather concerns (i.e. indoors or weather is clear), then activity holds.

3. Only one event can take place at a given time k, thus, $S_k \leftrightarrow (X_{1,k} \vee X_{2,k} \vee X_{3,k}...)$

   This means that a scheduled time slot is only valid iff there is at most one event scheduled at said time

   For example, if at a given time everyone is free, and there is a movie and meeting that needs to be scheduled only one of them can be scheduled and not both.

4. Assuming all activities need to be scheduled, the schedule is complete when $(X_1 \wedge X_2 \wedge ... \wedge X_n)$

   This means that the schedule holds (is complete) iff all of the activities hold (have been scheduled).

   For example, if there is not enough time for all of the activities to take place, then the schedule doesn't hold.

First-Order Extension

We could extend this by creating stipulations such as "all of these activities need to be scheduled" to make it not just a best-fit activity planner, but a planner for a list of things. Thus, it could be used not just for groups who want to hang out, but for organizations who need all the meetups to happen for an event at some point.

FEEDBACK REQUEST

1. Question for TA: We need a way to get avalabilities of people. One way we could do this is by webscraping lettucemeet, which is explained here. We want to know if this is okay to do to get availabilities this way based on our project.

2. Is it possible to have the validity of our propositions and constraints be verified (especially in the python code) to see if they make sense?