

Inteligența Artificială - Tema 1

Planificarea activităților sub constrângeri

Descriere

Se considera o problema de planificare, în care se dorește determinarea unui calendar săptămânal de activități. Tipurile de activități pot fi specificate în mai multe moduri:

1. prin **zi** și **interval orar strict** de desfășurare (e.g. miercuri, în intervalul 12:00-13:00)
2. prin **număr de instanțe per săptămână** (e.g. de 3 ori pe săptămână)
3. prin **număr de instanțe pe zi** (e.g. de 2 ori pe zi)
4. prin **definire relativă la alta activitate** (e.g. la cel mult / cel puțin x minute înainte / după activitatea y)
5. prin combinarea modurilor 2 - 4 (e.g. de 3 ori pe săptămână, după masa de prânz)

În afara de modul de specificare, pentru fiecare tip de activitate pot fi definite următoarele tipuri de constrângeri:

- **intervale preferate:** e.g. oricând între 16:00 - 20:00
- **intervale exclude:** e.g. oricând, mai puțin între 06:00 - 08:00 și 20:00 - 23:59
- **distanță minimă** între **două instanțe** ale **aceleiași tip de activitate** (e.g. minim o zi distanță între două instanțe de *mers la sala*).
- **distanță minimă** între **două activități** de **tipuri diferite** (e.g. exerciții fizice la minim 1 ora înainte sau după masa)

Pentru fiecare tip de constrângere se definește un cost al încălcării acesteia. Costurile se calculează în modul următor:

- Pentru număr de instanțe per săptămână: $C_{missing_instance_week} \times nr_{missing_instances}$
- Pentru număr de instanțe pe zi: $C_{missing_instance_day} \times nr_{missing_instances}$
- Pentru definire relativă la alta activitate:
 - $C_{relative} \times diff_{seconds}(activity.start, relative_interval.start)$ dacă $activity.start \leq relative_interval.start$, unde $relative_interval$ este intervalul tampon în interiorul căruia trebuie să înceapă activitatea relativă.
 - $C_{relative} \times diff_{seconds}(activity.start, relative_interval.end)$ dacă $activity.start > relative_interval.end$, unde $relative_interval$ este analog cazului de mai sus
- Pentru interval preferat:
 - $$C_{preferred_interval} \times \frac{||activity.interval|| - ||intersect(activity.interval, preferred.interval)||}{||activity.interval||}$$
- Pentru intervale exclude:
 - $$C_{excluded_interval} \times \frac{\sum_{excluded} ||intersect(activity.interval, excluded.interval)||}{||activity.interval||}$$

- Pentru distanta minima intre doua instante de activitate:

○

$$C_{activity_distance} \times ||intersect(activity2.interval, [activity1.end, activity1.end + buffer_interval])|| \\ \text{daca } activity2.start \geq activity1.end$$

○

$$C_{activity_distance} \times ||intersect(activity2.interval, [activity1.start - buffer_interval, activity1.start])|| \\ \text{daca } activity2.end < activity1.start$$

Dandu-se o lista de tipuri de activitate specificate folosind restrictiile de mai sus, se cere planificarea acestora pe durata unei saptamani, cu un cost minim de restrictii incalcate.

Pentru rezolvarea temei se cere sa combinati urmatoarele idei: ordonare a variabilelor, ordonare a valorilor, cale-consistenta, satisfacere partiala a restrictiilor.

Nota generala: activitatile se pot planifica **doar in intervalul 07:00 - 23:59** in fiecare zi, iar **doua activitati nu pot ocupa un slot orar in acelasi timp**.

Structura unui fisier de intrare

Fiecare instanta a problemei este descrisa intr-un fisier tip yaml. Formatul este urmatorul:

```
costs:
  c_missing_instance_week: 10
  c_missing_instance_day: 5
  c_relative: 20
  ...
activity_list:
  - activity1:
      name: activity1
      scheduling_type: exact_interval
      interval:
        day: 2
        start: 12:30
        end: 13:15

  - activity2:
      name: activity2
      scheduling_type: nr_instances
      instances_per_week: 7
      instances_per_day: 1
      duration:
        value: 1
        unit: hour
      preferred_intervals:
        - interval
          start: 08:00
          end: 10:00

  - activity3:
      name: activity3
      scheduling_type: relative
      instances_per_week: 3
      instances_per_day: 1
      duration:
        value: 10
        unit: min
      after:
        activity_type: activity2
        relative_within:
          value: 15
          unit: minutes
      minimal_distance_from:
        activity_type: self
        value: 1
        unit: day

  - activity4:
      name: activity4
```

```

scheduling_type: nr_instances
instances_per_week: 3
instances_per_day: 1
duration:
  value: 30
  unit: minute

excluded_intervals:
- interval:
  day: 6
  start: 00:00
  end: 23:59
- interval:
  day: 7
  start: 00:00
  end: 23:59

minimal_distance_from:
  activity_type: activity3
  value: 2
  unit: hour

```

Exemplul de mai sus contine cuvintele cheie prin care sunt definite costurile si constrangerile posibile prezentate mai sus.

Fiecare *tip de activitate* este definit printr-un atribut *name*.

Atributul `scheduling_type` indica modul de definire al tipului de activitate:

- *exact_interval* - prin interval exact
- *nr_instances* - prin numar de instante pe saptamana si per zi
- *relative* - relativ la alte activitati

Pentru activitatile definite prin **interval exact**, singurul atribut care se defineste in mod suplimentar este intervalul in sine. Pentru un interval se definesc:

- ziua (*day*): <numar intre 1 - 7> pentru cele 7 zile ale saptamanii
- data de inceput (*start*), exprimata in formatul ora:minute
- data de final (*end*), exprimata in formatul ora:minute

Pentru activitatile definite prin **numar de instante**, attributele importante sunt: `instances_per_week`, `instances_per_day` si `duration`. Daca este definit doar `instances_per_day`, se considera implicit ca activitatea trebuie sa aiba instante *in fiecare zi din saptamana*, de atatea ori cat spune `instances_per_day` ($7 * instances_per_day$).

Folosite in combinatie (ca in exemplul dat), se vor considera `instances_per_week * instances_per_day` instante de activitate (i.e. cate *instances_per_day* instante in fiecare zi din cele specificate de *instances_per_week*).

Atributul `duration` defineste durata activitatii prin valoare si unitate de masura (minute, hour).

Pentru activitatile definite **relativ la o alta activitate**, attributele importante sunt `after / before` si `duration`.

In cadrul acestora se defineste *tipul de activitate (numele)* fata de care este relativa si atributul `relative_within`. Cel din urma specifica o valoare si o unitate de timp (*minute, hour*) in care se masoara zona tampon (buffer) de timp in care trebuie sa se incadreze activitatea relativa.

O activitate relativa poate avea definite si attributele `instances_per_week` si `instances_per_day`.

Pentru fiecare activitate definita prin **numar de instante** sau **relativa la alta**, se pot specifica urmatoarele attribute:

- *preferred_intervals*: avand ca valoare o lista cu dictionare ce includ *day*, *start* si *end* pentru intervalul preferat. Daca din definitia intervalului lipseste *day*, atunci se considera ca intervalul orar dat de (*start*, *end*) este zilnic.
- *excluded_intervals*: avand ca valoare o lista cu dictionare ce includ *day*, *start* si *end* pentru intervalul dezagreat. Daca din definitia intervalului lipseste *day*, atunci se considera ca intervalul orar dat de (*start*, *end*) este zilnic.
- *minimal_distance_from*: avand specificat o lista cu dictionare ce specifica:
 - numele activitatii fata de care se cere o distanta (*self* pentru tipul propriu de activitate)
 - valoarea si unitatea de timp (minute, hour, day) care specifica distanta

Structura unui fisier de iesire

Fisierul de iesire va fi tot un fisier *.yaml*, care are urmatoarea structura:

```
planning_summary:
  - {activity_type: <name>, total_instances: <nr_total instances>,
    planned_instances: <nr_planned_instances>
    unplanned_instances: <nr_unplanned_instances>
    }
  - ...
planned_activity_list:
  - activity_instance:
    activity_type: <activity_name>
    interval:
      day: <day index>
      start: <start date in hour:minute format>
      end: <end date in hour:minute format>
  - ....
```

Prima parte a fisierului ofera un rezumat al planificarii pentru fiecare *tip de activitate*: cate instante erau de planificat, cate au fost planificate si cate au ramas nealocate.

Cea de-a doua parte contine o lista de *instante de activitati*, pentru fiecare specificandu-se: tipul activitatii si slotul alocat.

Cerinte

Formulati problema data ca una de satisfacere a restrictiilor. Implementati un algoritm de rezolvare a acestei probleme tinand cont de restrictiile impuse si incercand sa reducati costul restrictiilor incalcate.

Folositi euristici de ordonare a valorilor si variabilelor.

Implementati un algoritm de verificare a proprietatii de cale-consistenta (path-consistency) si analizati cum influenteaza aceasta timp de rezolvare a problemei.

Impuneti diferite marje maxime de timp pentru cautarea solutiei: e.g. 1s, 2s, 5s, 10s si evaluati costul final al solutiei de planificare obtinute in fiecare caz.

In rezolvarea problemei este necesara crearea unui fisier README, in care sa treceti in mod explicit:

- modul in care ati definit variabilele de planificare si domeniul de valori al acestora
- grafice in care sa aratati (si sa comentati) cum se imbunatateste calitatea celei mai bune solutii in functie de varianta de algoritm folosita:
 - cu ordonare a variabilelor vs fara
 - cu ordonarea a valorilor vs fara
 - cu cale-consistenta vs fara
 - cu toate trei euristicile la un loc

Pentru fiecare caz prezentati:

- un tabel in care prezentati costul obtinut, la fiecare marja de timp (1s, 2s, 5s, 10s)
- un grafic in care sa aratati rata descresterii costului in functie de timp (fara a mai lua in calcul marjele, i.e. pana ce costul nu mai poate fi imbunatatit)

Baremul temei este urmatorul

- citire date si reprezentare variabile si domeniul de valori (2p)
- implementare generala motor planificare si output rezultat (1.5p)
- implementare euristica de ordonare a variabilelor (1p)
- implementare euristica de ordonare a valorilor (1p)
- implementare cale-consistenta (1.5 p)
- grafice si analiza interpretativa a celor 3 euristici (3p - cate 1p pentru fiecare caz)

- Se acorda maxim 2p bonus pentru:
 - implementarea unor euristici suplimentare de ghidare a cautarii pentru cazul particular al problemei date (1p)
 - analiza comparativa a acestor euristici cu cele 3 propuse initial (1p)