Inteligență Artificială - Tema 1

Planificare prin satisfacerea parţială a restricţiilor

Descriere

Formulaţi următoarea problemă de planificare a sarcinilor ca una de satisfacere [parţială] a restricţiilor. Dându-se un număr **N** de sarcini de rezolvat (fiecare având o durată d_i, un termen limită t_i şi o listă de sarcini $j_{i1}, \ldots j_{in}$ ce trebuie terminate anterior începerii sarcinii **i**) şi un număr de procesoare **P**, ordonaţi şi distribuiţi sarcinile pe procesoare astfel încât suma depăşirilor limitelor de timp să fie minimă. Un procesor poate rezolva o singură sarcină la un moment dat. Procesoarele sunt identice: fiecare rezolvă sarcina **i** în acelaşi timp d_i. Termenul limită al unei sarcini poate fi depăşit inducând un cost, dar o sarcină nu poate fi executată înainte de terminarea celor care o condiţionează. Pentru rezolvarea temei va trebui să combinaţi următoarele idei: cale-consistenţă, ordonare a variabilelor, ordonare a valorilor, satisfacere parţială a restricţiilor.

Structura unui fişier de intrare

Fiecare instanţă a problemei date este descrisă separat într-un fişier. Prima linie a fişierului conţine două numere întregi separate prin virgulă, acestea reprezentând valorile **N** şi **P** (numărul de sarcini şi numărul de procesoare). Următoarele **N** linii conţin valorile **i**, d_i, t_i, dar şi sarcinile $j_{...}$ ce o condiţionează pe cea curentă pentru cele **N** sarcini, de asemenea separate prin virgulă. Un exemplu de fişier de intrare ar fi:

```
3,3
1,5,9
2,10,10
3,5,8,1
```

Structura fişierului de ieşire

Fişierul de ieşire va avea **P** secţiuni, fiecare corespunzătoare unui procesor. Fiecare secţiune începe cu o linie ce indică numărul de sarcini alocate acelui procesor şi continuă cu indexurile sarcinilor respective şi timpul de începere al fiecăreia (neapărat în ordine crescătoare). Dacă unui procesor nu îi este alocat nimic, secţiunea corespunzătoare lui va avea o singură linie cu valoarea 0. Un exemplu de fişier de ieşire (cu o soluţie la problema descrisă în exemplul de fişier de intrare) ar fi:

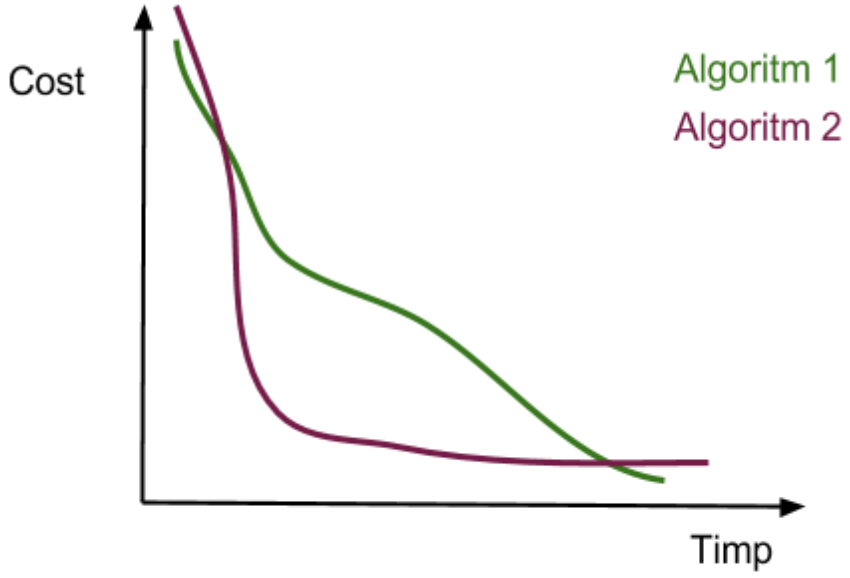
```
0
1
2,0
2
1,0
3,5
```

Evaluarea unei soluții

Costul unei soluții este dată de suma întârzierilor tuturor sarcinilor. Desigur, se dorește găsirea unor soluții cu cost cât mai mic. Dacă o sarcină i începe să fie prelucrată la momentul s_i, atunci costul dat de întârziere este $c_i = max(0, s_i + d_i - t_i)$. Costul unei soluții complete va fi $C = \Sigma_i c_i$. Puteți evalua o soluție folosind scriptul `eval.py`.

Cerințe

Formulați problema dată ca una de satisfacere a restricțiilor. Implementați un algoritm de rezolvare a acestei probleme ținând cont că restricțiile de ordonare a sarcinilor nu pot fi încălcate, dar termenele limită pot fi depășite implicând un cost al soluției respective. Opriți căutarea de oricâte ori costul parțial al unei soluții depășește cel mai bun cost obținut până atunci. Implementați un algoritm de verificare a proprietății de cale-consistență (eng. *path-consistency*) și analizați cum influențează timpul de găsire a celei mai bune soluții. Folosiți euristici de ordonare a valorilor și variabilelor.
Faceți grafice în care să arătați cum se îmbunătățește calitatea celei mai bune soluții descoperite în funcție de varianta de algoritm folosit (cu ordonare a variabilelor / fără, cu ordonare a valorilor / fără; cu cale-consistență de ordin 2, etc.).



Descrieți într-un fișier toate tehnicile și euristicile folosite pentru rezolvarea problemei și însoțiți explicațiile de graficele cerute. Se vor recompensa cu punctaj bonus temele care analizează cât mai multe euristici de ghidare a căutării.

Resurse utile

Cursul al treilea