



Predaja zadaće

Rok za predaju zadaće je četvrtak 7.11.2024. do 12:00 sati.

U zadaćama iz Heurističkih algoritama promatrat ćemo problem određivanja rasporeda na sveučilištima (engl. University course timetabling problem) u kojem je cilj rasporediti dana predavanja u određene vremenske intervale i odgovarajuće učionice uz zadovoljenje danih uvjeta. Konkretno, inačica problema koju ćemo promatrati je raspoređivanje kolegija nakon upisa studenata (engl. Post Enrolment Course Timetabling - PECT).

Svaki problem iz ovog skupa se sastoji od sljedećih informacija:

- skup od n predavanja koji se trebaju rasporediti u 45 vremenskih intervala (engl. timeslots) - 5 radnih dana, svaki se sastoji od 9 sati;
- skup od r učionica poznatog kapaciteta;
- skup od f svojstava koje učionice mogu zadovoljavati, a predavanja mogu zahtijevati;
- skup od s studenata koji trebaju prisustvovati različitim kombinacijama događaja;
- skup dostupnih vremenskih intervala za svaki od n događaja (tj. neće svi događaji biti dostupni u svim vremenskim intervalima);
- skup zahtjeva na redoslijed (engl. precedence requirements) koji definiraju koji događaji trebaju biti raspoređeni prije kojih.

Cilj rješavanja ovog problema je staviti svako od n predavanja u raspored (dodijeliti svakom od n predavanja jednu od r učionica i jedan od 45 vremenskih intervala) pri čemu treba zadovoljiti sljedećih 5 čvrstih (engl. hard) uvjeta:

1. nijedan student ne može biti prisutan na više od jednog predavanja u isto vrijeme;
2. učionica odabrana za predavanje treba biti dovoljno velika da stanu svi upisani studenti i treba zadovoljavati sva svojstva koja predavanje zahtijeva;
3. samo jedno predavanje može biti raspoređeno u jednoj učionici u jednom vremenskom intervalu;
4. predavanja se trebaju rasporediti samo u one vremenske intervale koji su unaprijed određeni kao dostupni za ta predavanja;
5. gdje je napomenuto, predavanja se trebaju rasporediti u pravilnom redoslijedu unutar tjedna.

S obzirom da je nerealno očekivati da će svi algoritmi zadovoljavati sve dane uvjete, moguće je neka predavanja ostaviti neraspoređena kako bi za ona raspoređena uvjeti



i dalje bili zadovoljeni. U skladu s time, možemo definirati **valjane** (engl. valid) rasporede, u kojima nisu narušeni uvjeti, ali su neka predavanja ostala neraspoređena; te **dopustive** (engl. feasible) rasporede u kojima nisu narušeni uvjeti i sva predavanja su raspoređena.

Dodatno, uz čvrste uvjete, postoje i meki (engl. soft) uvjeti koji bi trebali biti zadovoljeni, ali nije nužno da budu zadovoljeni u završnom rasporedu:

1. studenti ne bi trebali imati predavanje u rasporedu koje se nalazi u zadnjem vremenskom intervalu u danu (tj. vremenski intervali pod rednim brojevima 9, 18, 27, 36 ili 45);
2. studenti ne bi trebali imati 3 (ili više) uzastopna predavanja u jednom danu;
3. studenti ne bi trebali imati samo jedno predavanje u danu.

Svi dobiveni rasporedi moraju biti valjani. Ako u valjanom rasporedu postoje neraspoređena predavanja (tj. raspored nije dopustiv), za taj raspored se računa "udaljenost do dopustivosti" (engl. distance to feasibility). Navedena udaljenost se računa tako da se jednostavno pozbraja broj studenata koji je trebao prisustvovati neraspoređenim predavanjima. Npr. ako u završnom rasporedu 3 predavanja nisu raspoređena, a na njima je trebalo biti 12, 8 i 5 studenata, onda je udaljenost do dopustivosti $(12+8+5) = 25$. Uočite da je za dopustiva rješenja, po definiciji, udaljenost do dopustivosti jednak 0.

Nakon računanja udaljenosti do dopustivosti, razmatra se broj narušenih mekih uvjeta na sljedeći način:

- izbroji one instance gdje neki student ima samo jedno predavanje u danu (npr. ako jedan student ima srijedom i četvrtkom samo po jedno predavanje, ovaj broj će biti jednak 2);
- izbroji one instance gdje student ima više od 2 predavanja uzastopno u danu (3 uzastopna predavanja imaju vrijednost 1, 4 uzastopna vrijednost 2, 5 uzastopnih vrijednost 3, itd.); predavanja na kraju dana koje slijede predavanja odmah sljedeći dan ujutro se ne računaju kao uzastopni događaji;
- izbroji instance u kojima student ima predavanje u zadnjem vremenskom intervalu u danu.

Trošak narušavanja mekih uvjeta (engl. soft cost) je suma tih triju vrijednosti.

Dakle, svaki valjani raspored ima dvije vrijednosti: udaljenost do dopustivosti i trošak narušavanja mekih uvjeta. Kako bi se odredilo koje je rješenje bolje prvo se promatraju udaljenosti do dopustivosti; rješenja s manjim udaljenostima su bolja. Ako su udaljenosti jednake, onda se promatra trošak narušavanja mekih uvjeta; ono rješenje koje ima manji trošak je bolje.

Instance problema koje ćemo rješavati u sklopu zadaća dolaze s Međunarodnog natjecanja u raspoređivanju ¹. Instance su spremljene u direktoriju datasets, a na sljedećim linkovima možete pronaći opis instanci (<https://shorturl.at/4mKP8>) i opis kako treba izgledati rješenje problema (<https://shorturl.at/rj3wz>). Prilikom rješavanja zadataka treba koristiti sve dostupne instance iz direktorija datasets.

¹International timetabling competition <https://www.eecs.qub.ac.uk/itc2007/index.htm>

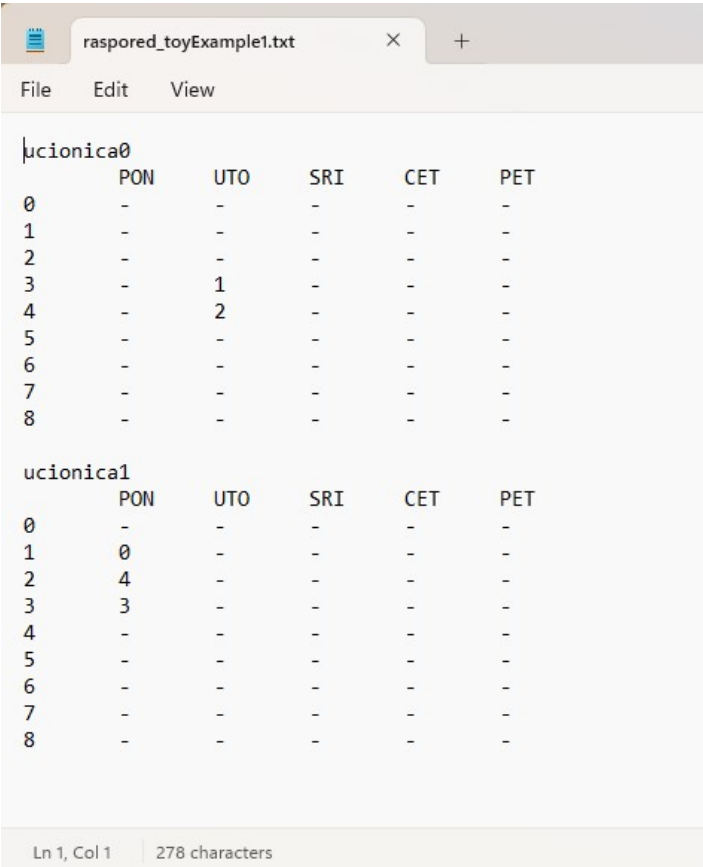


Zadatak 1. Ovaj zadatak u zadaći se sastoji od definiranja stvari koje će se koristiti u svim zadaćama. Za početak je potrebno:

- implementirati parser kojim će se moći u program učitavati instance problema koje će se zatim rješavati određenim pristupom,
- implementirati metodu kojom će se provjeravati je li rješenje dopustivo,
- implementirati metodu koja će od nedopustivog rješenja napraviti valjano (ako je potrebno),
- implementirati metodu kojom će se evaluirati dobiveno rješenje.

U datoteci validator se nalazi file checksln3b.exe koji vam može poslužiti za provjeru je li vaše rješenje valjano nakon što ga riješite svojim pristupom. Bitno je rješenje spremiti u formatu kako je napisano na <https://shorturl.at/rj3wz> i da se naziv rješenja podudara s nazivom input filea. Dodatno, za svaki input file kada dobijete rješenje napravite ispis na kojem će se vidjeti koji je predmet raspoređen u koji vremenski interval po učionicama (kao na slici 1.1). (potrebno pozvati kod svakog pristupa/zadatka u ovoj i narednim zadaćama)

Slika 1.1: Primjer rasporeda



	PON	UTO	SRI	CET	PET
0	-	-	-	-	-
1	-	-	-	-	-
2	-	-	-	-	-
3	-	1	-	-	-
4	-	2	-	-	-
5	-	-	-	-	-
6	-	-	-	-	-
7	-	-	-	-	-
8	-	-	-	-	-

	PON	UTO	SRI	CET	PET
0	-	-	-	-	-
1	0	-	-	-	-
2	4	-	-	-	-
3	3	-	-	-	-
4	-	-	-	-	-
5	-	-	-	-	-
6	-	-	-	-	-
7	-	-	-	-	-
8	-	-	-	-	-



Dodatno, napravite metodu koja će za ostvarena rješenja napraviti tekstualnu datoteku koja će se sastojati od sljedećih stupaca:

- naziv instance,
- udaljenost do dopustivosti,
- trošak narušavanja mekih uvjeta,
- ukupan trošak rješenja (suma 2. i 3. stupca).

(potrebno pozvati kod svakog pristupa/zadatka u ovoj i narednim zadaćama)

Zadatak 2. Definirajte pohlepni pristup za rješavanje ovog problema. Definirajte pohlepni korak i argumentirajte je li on statička ili dinamička heuristika. Implementirajte svoje rješenje, a konačne rezultate zapišite kako je predviđeno Zadatkom 1.

Zadatak 3. Definirajte lokalno pretraživanje za dani problem. Definirajte barem dva različita načina za dobivanje susjedstva. Implementirajte svoje rješenje i usporedite dobivena konačna rješenja prilikom korištenja različitih susjedstva. Implementirajte tehniku odabira susjeda: najbolji poboljšavajući susjed i prvi poboljšavajući susjed. Rješenja zapišite kako je predviđeno Zadatkom 1 i to za oba definirana susjedstva.

Napomena: podzadatke koje treba raspisati predajte u pdf fileu. Rješenja koja dobijete ne moraju biti dopustiva na kraju, bitno je da su valjana i da se vidi na koji način ste implementirali svoja predložena rješenja.

