Student: Helena Mihaljčić, 117/22

Link ka Github repozitorijumu projekta: <https://github.com/HelenaMihaljcic/OperacionaIstrazivanja.git>

*Planiranje rasporeda rada*

*Greedy vs PuLP*

1. Problem planiranja radnog rasporeda

Kompanija želi da kreira nedeljni raspored za zaposlene. Glavni cilj je minimizovati ukupne troškove rada, istovremeno osiguravajući dovoljnu pokrivenost svih smjena i poštujući ograničenja vezana za zaposlene i smjene.

Detalji problema

* *Zaposleni (i):*
* Svaki zaposleni ima fiksnu satnicu (Ci).
* Imaju maksimalan broj sati koje mogu raditi nedeljno (Si).
* Mogu imati zahtjeve za određene smjene u određenim danima (Di,j,k∈{0,1}).
* *Smjene (k):*
* Nedelja ima 7 dana (j), svaki dan po 3 smjene (jutarnja, popodnevna, večernja).
* Svaka smjena ima potreban broj zaposlenih (Nk) i trajanje u satima (Tk).

Npr. Jutarnja: 2 zaposlena, 4 sata; Popodnevna: 3 zaposlena, 6 sati; Večernja: 1 zaposleni, 4 sata.

* *Ograničenja:*
* Jedna smjena dnevno: Nijedan zaposleni ne može raditi više od jedne smjene dnevno.
* Maksimalni sati: Ukupan broj sati dodeljenih zaposlenom ne smije preći njegov nedeljni maksimum.
* Pokrivenost: Svaka smjena mora biti u potpunosti pokrivena potrebnim brojem zaposlenih.
* Dostupnost: Zaposleni rade samo smjene za koje su dostupni.

*Primjer podataka*

* Zaposleni:
* Alis: $15/sat, maks. 20h/nedelju. Dostupna sve smjene, osim ponedeljka uveče.
* Bob: $20/sat, maks. 25h/nedelju. Dostupan sve smjene.
* Čarli: $18/sat, maks. 15h/nedelju. Dostupan samo jutarnje i popodnevne smjene.
* Smjene:
* Jutarnja: 2 potrebna, 4h trajanje.
* Popodnevna: 3 potrebna, 6h trajanje.
* Večernja: 1 potreban, 4h trajanje.

2. Matematički model (Linearno programiranje)

Ovaj problem je modelovan kao problem celobrojnog linearnog programiranja (Integer Linear Programming - ILP).

*Varijable odluke*

* xi,j,k∈{0,1}: Binarna varijabla koja je 1 ako zaposleni i radi smjenu k u danu j, inače 0.

*Parametri*

* I: Skup svih zaposlenih.
* J: Skup dana u sedmici (npr. J={1,2,…,7}).
* K: Skup smjena (jutarnja, popodnevna, večernja).
* Nk: Broj radnika potreban za smjenu k.
* Tk: Trajanje smjene k u satima.
* Ci: Satnica zaposlenog i.
* Si: Maksimalan broj sati sedmično za zaposlenog i.
* Di,j,k∈{0,1}: Dostupnost zaposlenog i za smjenu k u danu j.

*Funkcija cilja*

Minimizovati ukupne troškove rada:

*Ograničenja*

1. Pokrivenost smjena: Za svaki dan j i smjenu k, mora biti dodeljeno tačno Nk radnika.
2. Maksimalno jedna smjena dnevno po zaposlenom:
3. Maksimalni broj sati sedmično:
4. Dostupnost zaposlenih: Zaposleni mogu biti dodeljeni samo smjenama za koje su dostupni.

3. Implementacija i pristupi rješavanju

Za rješavanje problema korišćena su dva pristupa: PuLP (linearno programiranje) za optimalno rješenje i pohlepni algoritam za brzo, heurističko rješenje. Podaci za testiranje su generisani dinamički i sačuvani u JSON fajlovima.

*Generisanje instanci*

Kreirane su tri instance problema sa nasumično generisanim parametrima za zaposlene (satnice, maksimalni sati, dostupnosti) i smjene (trajanje je fiksno, dok je potreban broj radnika nasumično određen uz garantovanu izvodljivost , npr. ako su samo 3 radnika dostupna za večernju smjenu, zahtjev neće biti postavljen na 5). Ove instance su sačuvane u JSON fajlovima (instance\_1.json, instance\_2.json, instance\_3.json).

PuLP (Optimalni pristup)

* Model je implementiran korišćenjem PuLP biblioteke u Pythonu.
* PuLP koristi CBC solver (podrazumijevani open-source rješavač) za pronalaženje optimalnog rješenja, što znači da će pronaći raspored s minimalnim ukupnim troškovima, poštujući sva ograničenja.
* Robustnost: Model uključuje "slack" varijable koje se kažnjavaju visokim troškom. Ovo omogućava solveru da pronađe rješenje čak i ako je potpuna pokrivenost nemoguća (npr. zbog nedostatka zaposlenih), mada se takva rješenja označavaju kao "Neizvodljiva" u izvještaju.
* Vrijeme izvršavanja: PuLP, kao optimalni rješavač, obično zahtijeva duže vrijeme izvršenja, posebno za veće instance.

*Pohlepni pristup (Heuristički pristup)*

Cilj pohlepnog pristupa je pronaći brzo rješenje, koje nije nužno optimalno.

Kako funkcioniše:

1. Smjene se sortiraju po procijenjenoj cijeni po satu (jeftinije smjene prvo).
2. Za svaku smjenu, raspoloživi zaposleni se sortiraju od najjeftinijih do najskupljih.
3. Dodjeljuju se zaposleni redom dok se ne ispuni potreban broj za smjenu, poštujući dnevna i nedeljna ograničenja sati i dostupnost.
4. Ako u bilo kojem trenutku smjena ne može biti pokrivena, algoritam se prekida i označava problem kao "Neizvodljiv".

Vrijeme izvršenja: Pohlepni algoritam je znatno brži od PuLP-a, ali može proizvesti sub-optimalna rješenja ili čak ne pronaći rješenje tamo gdje optimalno postoji.

4. Testiranje i poređenje rezultata

Testiranje je sprovedeno na instancama različitih veličina (10, 50, 200, 1000 zaposlenih), a rezultati su prikazani u tabelarnoj formi, sa prosječnim vrijednostima dobijenim višestrukim pokretanjima.

Napomena o "Satnica ($)" i "Max sati (h)": Prikazani raspon (npr. 10-34) predstavlja minimalnu i maksimalnu vrijednost satnice/sati među zaposlenima u datoj instanci. Procenti zaposlenih sa određenim vrijednostima unutar tog raspona variraju od instance do instance, čak i ako je raspon isti.

Rezultati (Prosječne vrijednosti)

*Instance sa 10 zaposlenih*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Instanca | Satnica ($) | Max sati (h) | PuLP Trošak (USD) | Pohlepni Trošak (USD) | Prosječno PuLP vrijeme (ms) | Prosječno Pohlepno vrijeme (ms) |
| 1 | 10-34 | 15-35 | 1572.0 | 1636.0 | 205.62 | 0.00 |
| 2 | 10-33 | 16-35 | 2884.0 | 3024.0 | 494.61 | 0.20 |
| 3 | 10-33 | 16-33 | 2546.0 | 2598.0 | 157.46 | 0.00 |

*Instance sa 50 zaposlenih*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Instanca | Satnica ($) | Max sati (h) | PuLP Trošak (USD) | Pohlepni Trošak (USD) | Prosječno PuLP vrijeme (ms) | Prosječno Pohlepno vrijeme (ms) |
| 1 | 10-35 | 15-35 | 6874.0 | 7240.0 | 1822.13 | 0.60 |
| 2 | 10-35 | 15-35 | 16588.0 | 17010.0 | 1838.30 | 0.60 |
| 3 | 10-34 | 15-35 | 20168.0 | 21130.0 | 885.35 | 0.60 |

*Instance sa 200 zaposlenih*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Instanca | Satnica ($) | Max sati (h) | PuLP Trošak (USD) | Pohlepni Trošak (USD) | Prosječno PuLP vrijeme (ms) | Prosječno Pohlepno vrijeme (ms) |
| 1 | 10-35 | 8-25 | 59484.0 | Neizvodljivo | 5907.53 | 3.06 |
| 2 | 10-35 | 8-25 | 47766.0 | 49804.0 | 5834.85 | 2.00 |
| 3 | 10-35 | 8-25 | 69564.0 | Neizvodljivo | 8335.63 | 3.00 |

*Instance sa 1000 zaposlenih*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Instanca | Satnica ($) | Max sati (h) | PuLP Trošak (USD) | Pohlepni Trošak (USD) | Prosječno PuLP vrijeme (ms) | Prosječno Pohlepno vrijeme (ms) |
| 1 | 15-25 | 25-35 | 238134.0 | 241856.0 | 26408.56 | 15.41 |
| 2 | 15-25 | 25-35 | 225198.0 | 228628.0 | 29837.83 | 17.33 |
| 3 | 15-25 | 25-35 | 318304.0 | 324512.0 | 49273.94 | 17.02 |

Na osnovu prikazanih rezultata, jasno je vidljiva razlika u performansama i kvalitetu rješenja između PuLP-a i pohlepnog pristupa:

* PuLP (Optimalnost vs. Vrijeme):
* Trošak: PuLP dosljedno pronalazi optimalno (minimalno) rješenje, što se vidi po tome da je njegov trošak uvek jednak ili niži od pohlepnog pristupa (kada pohlepni pronađe rješenje).
* Vrijeme: Vrijeme izvršenja PuLP-a značajno raste s veličinom instance. Za 1000 zaposlenih, može potrajati i do skoro 50 sekundi, što je i dalje prihvatljivo za nedeljno planiranje, ali bi za real-time primjene bilo predugo.
* Izvodljivost: PuLP može pronaći rješenje čak i kada su resursi ograničeni, ali ako nije moguće ispuniti sve zahtjeve za pokrivenost, to će biti jasno naznačeno (ili rješenje može biti "Neizvodljivo" ako slack varijable ostanu veće od nule).
* Pohlepni pristup (Brzina vs. Kvalitet):
* Trošak: Pohlepni pristup je sub-optimalan. Njegov trošak je često veći od PuLP-ovog, jer donosi lokalno najbolje odluke koje ne vode nužno globalno optimalnom rješenju. Razlika u troškovima raste sa složenošću problema.
* Vrijeme: Ključna prednost pohlepnog pristupa je izuzetna brzina. Njegovo vrijeme izvršenja je zanemarljivo malo (u milisekundama), čak i za instance sa 1000 zaposlenih. To ga čini pogodnim za situacije gdje je potrebno brzo dobiti približno rješenje.
* Izvodljivost: U nekim slučajevima (vidljivo kod instanci sa 200 zaposlenih), pohlepni algoritam nije uspio da pronađe izvodljivo rješenje, dok je PuLP to uspio. To se dešava jer pohlepne odluke mogu dovesti do "ćorsokaka" iz kojih nema povratka.