

Optimizacija profita turističke agencije Turist.co

Informacijska tehnologija i društvo

A. Lucija Babić, B. Lucia Vareško, C. Kaja Radošević, D. Dejana Pivac, E. Adis Hadžić

Fakultet Informatike u Puli
Sveučilište Jurja Dobrile u Puli

Sažetak - Turistička agencija prilikom organizacije izleta želi maksimizirati svoj profit te pritom složiti ponudu koja će ispunjavati želje klijenata. Ideja je ostvariti maksimalan profit uz postavljena ograničenja. Profit se ostvaruje uzimanjem 15% marže od ukupnog iznosa troškova. Programsko rješenje izradili smo u Python-u pomoću PuLP biblioteke te smo također izradili korisničko sučelje u koje se unose podaci potrebni za izračun.

I. UVOD

Naš zadatak bio je prikupiti ulazne podatke i razviti rješenje koristeći metode linearnog programiranja. Odabrali smo problem maksimizacije profita turističke agencije koja se bavi posredovanjem u pružanju usluga smještaja te organizacijom jednodnevnih izleta i degustacija istarskih delicija.

II. OPIS PROBLEMA

Agencija Turist.co nudi web aplikaciju u kojoj voditelj agencije ima mogućnost unijeti broj turista i maksimalnu cijenu koju su oni spremni platiti. Nakon što se podaci unesu, oni se prosljeđuju u program u kojem smo definirali sve uvjete i ograničenja koji nam pomažu u odabiru najoptimalnije ponude.

Zadani su smještaji u Umagu, Poreču, Puli ili Pazinu, a svaki od njih ima određeni kapacitet, udaljenost od zračne luke, koja se nalazi u Puli, te cijenu noćenja. Za sve turiste odabire se optimalni izlet i transfer koji će ih prevoziti. Na izletu će imati mogućnost isprobavanja istarskih delicija uz pratnju vodiča koji je plaćen po satu, a izlet traje otprilike osam sati. Za svaki izlet imamo popis gradova koji će se posjetiti i ukupan broj kilometara na tom izletu. Neke od delicija koje će moći isprobati su vino, maslinovo ulje, kamenice i druge, a cijene variraju ovisno o broju gostiju. Svi turisti će biti smješteni u istom smještaju, ići na isti izlet i probati iste delicije. Optimalno rješenje vraća ponudu koja zadovoljava sva ograničenja koja ostvaruje najveću zaradu.

III. METODE PRIKUPLJANJA PODATAKA

Za potrebe projekta koristili smo se internetom za prikupljanje podataka. Istraživali smo u kojem gradu se nudi koji smještaj te smo provjerili cijene i kapacitet tih smještaja. U slučaju da nije bilo točnih podataka iste smo

procijenili. Također smo proučili koje delicije se nude u kojem gradu kako bismo mogli omogućiti degustiranje. Broj ljudi za koje se organizira izlet i maksimalna cijena koju su spremni platiti su podaci koje je obavezno unijeti prilikom korištenja web aplikacije.

Smještaj karakterizira naziv, grad, udaljenost od zračne luke, kapacitet i cijena noćenja po osobi. Svaki izlet se sastoji od naziva, podatka o ukupnom broju kilometara koji su prijeđeni u obilaženju gradova i, naravno, popis gradova. Degustacija ima svoj naziv, grad u kojem se može isprobati i cijenu ovisno o veličine grupe ljudi za koju se organizira izlet. Podaci su prikazani na Sl. 1 i također su pohranjeni u JSON formatu.

```
"smjestaj": [
  { "naziv": "Apartmani Maric",
    "grad": "Umag",
    "udaljenost": "83",
    "kapacitet": "14",
    "cijena_nocenje": "410" },
  ... još 8
],
"izleti": [
  { "naziv": "Izlet 1",
    "ukupno_km": "86",
    "gradovi": ["Umag",
               "Groznojan",
               "Motovun",
               "Visnjan",
               "Novigrad"] },
  ...
],
"degustacije": [
  { "naziv": "Vino",
    "grad": "Hum",
    "min_1": "50",
    "min_10": "50",
    "min_30": "50",
    "min_50": "50",
    "min_70": "50",
    "cijena": "50" },
  ...
]
```

Sl. 1 Podaci u JSON formatu

Podatke za transfer prikupili smo analizom ponuda različitih prijevoznika. Na raspolaganju imamo pet prijevoznih sredstava od kojih svaki ima kapacitet i cijenu po kilometru. Podaci su pohranjeni u JSON formatu i prikazani su na Sl. 2.

```

"transfer": [
  { "vrsta": "Taxi",
    "kapacitet": "4",
    "cijena_km": "8" },
  { "vrsta": "Taxi kombi",
    "kapacitet": "8",
    "cijena_km": "12" },
  { "vrsta": "Mini bus",
    "kapacitet": "20",
    "cijena_km": "20" },
  { "vrsta": "Midi bus",
    "kapacitet": "40",
    "cijena_km": "35" },
  { "vrsta": "Bus",
    "kapacitet": "60",
    "cijena_km": "45" }
],

```

Sl. 2 Podaci u JSON formatu

IV. METODE RJEŠAVANJA

A. Podproblem minimizacije cijene transfera

Odabir transporta riješen je kao problem minimizacije koji bira najmanju cijenu transfera ovisno o broju ljudi.

Problem ima funkciju cilja koja zbraja umnoške konstante i varijabli odluke (1).

$$Z = \text{Min} \sum_{i=1}^m c_i x_i$$

i = indeks vozila

c_i = cijena po kilometru

x_i = količina odabranog vozila

Svako vozilo ima svoju cijenu po kilometru, stoga ona predstavlja konstantu u izračunu. Varijabla odluke označava koliko će tog vozila biti odabrano. Samim time varijable odluke ne smiju biti negativne i moraju biti cijeli broj (2).

$$x_1, x_2, \dots, x_m \geq 0 \text{ i cjelobrojne}$$

Obavezno je dodati ograničenje za kapacitet. Svako vozilo ima svoj kapacitet, što znači da kapacitet po vozilu ne smije preći ukupan broj ljudi.

$$\sum_{i=1}^m k_i x_i \leq \text{broj_ljudi}$$

i = indeks vozila

k_i = kapacitet vozila

x_i = količina odabranog vozila

Rješenje ovog potproblema koristimo u izračunu glavnom problema.

B. Glavni problem ponude

Glavni problem ponude je problem maksimizacije odabira najpovoljnije kombinacije smještaja, izleta i degustacije.

Glavni problem prima 2 inputa: broj ljudi koji žele organizirati putovanje i maksimalan iznos koji zasebno mogu platiti. Rezultat funkcije cilja daje nam ukupnu cijenu troškova za sve turiste kojoj je odmah dodana marža od 15% kako bi mogli uključiti ograničenje za maksimalan iznos plaćanja. Prije korištenja glavne funkcije cilja, zbog jednostavnosti, potrebno je posebno napraviti izračun za svaki smještaj, izlet i degustaciju.

Za izračun ukupne cijene smještaja potrebno je rješenje podproblema transfera, koje nam govori koliko je potrebno platiti prijevoz po kilometru za odabrani broj ljudi. S obzirom da za turiste moramo organizirati prijevoz od zračne luke do smještaja i obratno, potrebno je dva puta pomnožiti rezultat podproblema i ukupnom broja kilometra udaljenosti smještaja i zračne luke. Također, potrebno je pomnožiti broj ljudi sa cijenom noćenja i to pridodati prethodnom rezultatu. Smještaj karakterizira indeks i (3)

$$c_j = \text{transfer} * \text{ukupno_km} + \text{vodič_satnica} * \text{trajanje}$$

Izračun troškova izleta je sličan. Također je potrebno rješenje podproblema jer imamo podatak o ukupno prijednim kilometrima za svaki izlet. Također potrebno je pridodati statičnu cijenu vodiča koja je 400kn, odnosno vodič je plaćen 50kn/h dok jednodnevni izlet traje 8h. Izlet karakterizira indeks j (4).

Degustaciju Iznos za degustaciju dobivamo umnoškom

$$c_i = 2 * \text{transfer} * \text{udaljenost} + \text{ljudi} * \text{noćenje}$$

broja ljudi i odabrane cijene degustacije za tu grupu. Cijena degustacije bira se običnim izrazom "ako", "onda" s obzirom da nije potrebno optimizirati rješenje, već odabrati traženu cijenu. Cijene degustacije se smanjuju što je grupa veća, time je simuliran "popust na količinu" koje mnogi proizvođači nude. Degustaciju karakterizira indeks k (5).

$$c_k = \text{ljudi} * \text{cijena}$$

Funkcija cilja je kombinacija svakog smještaja sa svakim izletom sa svakom degustacijom, povećana za 15% kako bi se dobio točan iznos uplate. Traži se kombinacija koja daje maksimalnu cijenu uz pridržavanje ograničenja (6).

$$Z = \text{Max} \sum_{i=1}^m \sum_{j=1}^n \sum_{k=1}^t c_{ijk} x_{ijk} * 1,15$$

i = indeks smještaja
 j = indeks izleta
 k = indeks degustacije
 c_{ijk} = ukupna cijena kombinacije
 x_{ijk} = status odabira (0 ili 1)

Varijable odluke, kao i u podproblemu, moraju biti cijeli broj. No, za ovaj problem potrebno je naglasiti da varijabla odluke može poprimiti samo vrijednosti 0 ili 1 - što označava status odabira (7).

$$x_{111}, x_{112}, \dots, x_{mnt} = 0 \text{ ili } 1, i \text{ cjelobrojne}$$

Prvo ograničenje odnosi se na kapacitet smještaja. Svaki smještaj ima svoj kapacitet koji ne smije biti manji od ukupnog broja turista. Kapacitet smještaja potrebno je dodati uz svaku varijablu odluke (8).

$$\sum_{i=1}^m \sum_{j=1}^n \sum_{k=1}^t y_{ijk} x_{ijk} \leq \text{broj_ljudi}$$

i = indeks smještaja
 j = indeks izleta
 k = indeks degustacije
 y_{ijk} = kapacitet smještaja i
 x_{ijk} = status odabira (0 ili 1)

Činjenica je da nije svaka kombinacija moguća. Za svaki izlet navedeni su gradovi među kojima mora biti grad u kojem je smještaj turista i grad u kojem je organizirana degustacija. Za onu kombinaciju koja nije moguća, konstanta z poprima vrijednost 0, dok za onu moguću poprima vrijednost 1. Za to je u konačnoj implementaciji rješenja izrađena funkcija, no u excel-u je potrebno ručno odrediti vrijednosti. Slijedi izraz (9).

$$\sum_{i=1}^m \sum_{j=1}^n \sum_{k=1}^t z_{ijk} x_{ijk} = 1$$

i = indeks smještaja
 j = indeks izleta
 k = indeks degustacije
 z_{ijk} = mogućnost kombinacije (0 ili 1)
 x_{ijk} = status odabira (0 ili 1)

S obzirom da mi organiziramo jedno putovanje, potrebno je odabrati samo jednu kombinaciju smještaja, izleta i degustacije. Samim time, potrebno je ograničiti da suma varijabli odluke mora biti jednaka jedinici (10).

$$\sum_{i=1}^m \sum_{j=1}^n \sum_{k=1}^t x_{ijk} = 1$$

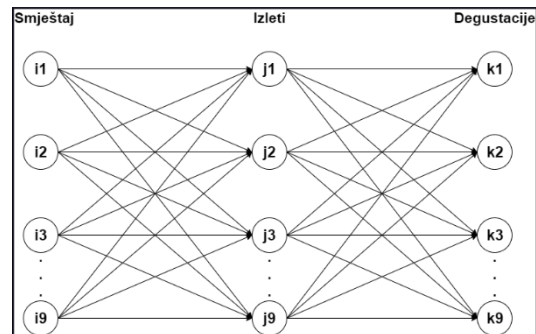
i = indeks smještaja
 j = indeks izleta
 k = indeks degustacije
 x_{ijk} = status odabira (0 ili 1)

Moramo uzeti u obzir input koji unosimo za maksimalnu cijenu koju turisti pojedinačno žele platiti. Ograničenje se dodaje na sam rezultat funkcije cilja, koji mora biti manji ili jednak toj unesenoj cijeni. Cijena mora biti pomnožena s brojem ljudi, s obzirom da rješenje funkcije cilja daje ukupan iznos a ne pojedinačan (11).

$$\sum_{i=1}^m \sum_{j=1}^n \sum_{k=1}^t c_{ijk} x_{ijk} * 1,15 \leq \max_{\text{cijena}} * \text{ljudi}$$

C. Matematički model

Problem se može definirati kao "three-index assignment problem" [3] gdje je samo jedna kombinacija odabrana. Kao što je već rečeno, kombinaciju čini odabir smještaja, izleta i degustacije. S obzirom da imamo po 9 smještaja, izleta i hotela - ukupnih kombinacija je 729. Na matematičkom modelu vidljiv je prikaz problema.



Sl. 3 Graf matematičkog modela

D. Python PuLP rješenje

Matematički model implementiran je u programskom jeziku Python korištenjem biblioteke PuLP [6]. Izrađeno je jednostavno korisničko sučelje koristeći okruženje Flask. Prva stranica rješenja pita korisnika za unos broja turista i maksimalne cijene koju turisti pojedinačno žele platiti. Ako rješenje problema izlazi iz mogućeg prostora rješenja, korisnik je o tome obaviješten. Suprotno tome, korisnik je prebačen na stranicu rješenja. Stranica rješenja sadrži tablicu svih podataka potrebnih za izračun rješenja, dok se rješenje funkcije cilja nalazi u redu „PRODAJNA CIJENA ARANŽMANA“. Uz to, korisniku je pružena karta na kojoj su vidljive točke zračne luke, smještaja, degustacije i gradova koji će biti obišteni na jednodnevnom izletu. Također, generiran je tekst opisa putovanja.

Podproblem transfera i glavni problem ponude nalaze se u odvojenim .py datotekama. Podaci su spremljeni u posebnu datoteku „podaci.json“ i koriste se pri izračunu. Kao što je objašnjeno, rezultat podproblema koristi se u rješavanju glavnog problema.

Podproblem je definiran kao „Lp_prob1“ u koji se prvo sprema funkcija cilja.

```
24 #f cilja
25 Lp_prob1 += int(transfer[0]["cijena_km"])*x1
26 + int(transfer[1]["cijena_km"])*x2
27 + int(transfer[2]["cijena_km"])*x3
28 + int(transfer[3]["cijena_km"])*x4
29 + int(transfer[4]["cijena_km"])*x5
```

Sl. 4 Funkcija cilja podproblema

Zatim je potrebno pohraniti ograničenja kapaciteta vozila.

```
31 #ograničenja
32 Lp_prob1 += int(transfer[0]["kapacitet"])*x1
33 + int(transfer[1]["kapacitet"])*x2
34 + int(transfer[2]["kapacitet"])*x3
35 + int(transfer[3]["kapacitet"])*x4
36 + int(transfer[4]["kapacitet"])*x5 >= ljudi
```

Sl. 5 Ograničenja podproblema

Na kraju, standardnim procesom pozivanjem funkcije „solve()“ dobivamo konačno rješenje koje spremamo u varijablu „rezultat“ i šaljemo u glavni problem.

```
41 Lp_prob1.solve()
42 rezultat = p.value(Lp_prob1.objective)
```

Sl. 6 Rezultat podproblema

Kod glavnog problema rješavanje je kompleksnije. S obzirom da funkcija cilja zahtjeva sume svake kombinacije, potrebno je korištenje 3 for petlje. Glavni problem je definiran kao „Lp_prob2“ u koji se sprema funkcija cilja.

```
68 #funkcija cilja
69 pom = 0
70
71 for i in range(9):
72     for j in range(9):
73         for k in range(9):
74             var = "x{}{}{}".format(i+1,j+1,k+1)
75             pom += marza(lista_smjestaji[i]+lista_izleti[j]
76                 +lista_degustacije[k])*rijecnik[var]
77
78 Lp_prob2 += pom
```

Sl. 7 Funkcija cilja glavnog problema

Ograničenje na ukupnu cijenu

```
84 #ograničenje na ukupnu cijenu
85 Lp_prob2 += pom <= int(z)*ljudi
```

Sl. 8 Ograničenje na ukupnu cijenu

Ograničenje za kapacitete smještaja dodano je na slijedeći način.

```
87 #ograničenje kapacitet smjestaja
88 pom = 0
89
90 for i in range(9):
91     for j in range(9):
92         for k in range(9):
93             var = "x{}{}{}".format(i+1,j+1,k+1)
94             pom += int(smjestaj[i]["kapacitet"])*rijecnik[var]
95
96 Lp_prob2 += pom >= ljudi
```

Sl. 9 Ograničenje kapaciteta

Ograničenje za mogućnost odabira kombinacije

```
98 #ograničenje mogućnosti odabira kombinacije
99 pom = 0
100
101 for i in range(9):
102     for j in range(9):
103         for k in range(9):
104             var = "x{}{}{}".format(i+1,j+1,k+1)
105
106             pom += moze_nemoze(smjestaj[i]["grad"],
107                 izleti[j]["gradovi"],
108                 degustacije[k]["grad"])*rijecnik[var]
```

Sl. 10 Ograničenje za mogućnost odabira kombinacije

Potrebno je postaviti da suma varijabli ćelija mora biti jednaka jedinici.

```
112 #ograničenje sume var. ćelija
113 Lp_prob2 += sum(rijecnik.values()) == 1
```

Sl. 11 Ograničenje sume var. ćelija

```
118 status = Lp_prob2.solve()
119
120 rjesenje = (p.LpStatus[status])
```

Sl. 12 Rješenje glavnog problema

V. TUMAČENJE RJEŠENJA

Rješenje problema predstavlja optimalnu ponudu. Odabrana rješenja moraju smjestiti turiste u isti smještaj i na isti izlet. Rješenje također ovisi o unesenom broju ljudi i cijeni koju su spremni platiti. Tijekom izleta turisti će imati vodiča koji je plaćen po satu, a s obzirom da izlet traje osam sati njegova cijena je fiksna odnosno 400 kn. Ovaj problem se sastoji i od transportnog podproblema minimizacije koji bira najmanju cijenu transfera ovisno o broju ljudi. Rješenje podproblema se množi s 2 jer je turiste potrebno dovesti od zračne luke i obratno. Cijena degustacije varira ovisno o broju ljudi što predstavlja popust na količinu. Na kraju prodajna cijena ovisi o ukupnim troškovima na koje se doda marža od 15%, koja predstavlja profit za agenciju. Ukoliko rješenje nije moguće, web aplikacija vrati obavijest o tome.

VI. ZAKLJUČAK

Metodom linearnog programiranja dolazi se do optimalnog rješenja linearnih funkcija više varijabli (poput maksimiziranja profita ili minimiziranja troškova). Ovo je posebno pogodno za različita poslovanja koja nastoje poslovati profitabilno. Sve se svodi na poznavanje relevantnog matematičkog modela određenog problema kojeg je moguće riješiti uz pomoć raznovrsnih programa i biblioteka.

LITERATURA

- [1] Chand,S. (n.d) *Assignment Problem in Linear Programming: Introduction and Assignment Model* [Article post], [Assignment Problem in Linear Programming : Introduction and Assignment Model \(yourarticlelibrary.com\)](https://yourarticlelibrary.com/Assignment-Problem-in-Linear-Programming-Introduction-and-Assignment-Model) [23.01.2021.].
- [2] Excel Easy (n.d) *Assignment Problem*, [Assignment Problem in Excel - Easy Excel Tutorial \(excel-easy.com\)](https://excel-easy.com/Assignment-Problem-in-Excel-Easy-Excel-Tutorial) [19.01.2021.].
- [3] Frits C. R. Spieksma (2011) *Multi Index Assignment Problems: Complexity, Approximation, Applications* , Department of Mathematics, Maastricht University, [Assignment Problem in Linear Programming : Introduction and Assignment Model \(yourarticlelibrary.com\)](https://yourarticlelibrary.com/Assignment-Problem-in-Linear-Programming-Introduction-and-Assignment-Model) [19.01.2021.].
- [4] <http://www.assignmentproblems.com/doc/MultiAPIntroduction.pdf> [27.01.2021.].
- [5] <https://coin-or.github.io/pulp/> [24.01.2021.]