

Matemaattinen Optimointi I – harjoitustyö no. 2085 – Gotham City Police Department

Matemaattinen optimointi I

Turun yliopisto

Sovellettu matematiikka

Tomi Mäntylä

Sisällysluettelo

1. Tehtävänanto.....	1
2. Lähtökohta.....	1
2.1. Ero Rothsteinin ratkaisuun verrattuna.....	2
3. Ratkaisu R-ohjelmalla.....	3
3.1. Tulos ja sen tulkinta.....	4
4. Herkkyysanalyysi.....	4
5. Lähteet.....	7

1. Tehtävänanto

Gotham City:n poliisilaitoksella on ongelma; miten järjestää poliisien viikoittainen työaika niin, että heillä olisi yksittäisiä, ei peräkkäisiä vapaapäiviä mahdollisimman vähän. Poliiseja on kaikkiaan 30, mutta päivittäinen tarve työvoimalle vaihtelee seuraavasti maanantaista sunnuntaihin: 18, 24, 25, 16, 21, 28, 18.

2. Lähtökohta

Ensin minusta näytti kyseessä olevan kokonaislukuoptimointitehtävä, jossa on päätösmuuttujina $30 \cdot 7 = 210$ binäärimuuttujaa. Joka viikonpäivälle pitäisi siis olla tieto siitä, ketkä 30 poliisista ovat töissä. Toisaalta, kukin poliiseista voi vaihtaa työviikkoa kenen tahansa muun poliisin kanssa ilman vaikutusta työssä olevien henkilöiden lukumäärään. Ei siis ole tarvetta yksilöidä kutakin poliisia optimoinnissa. Mahdollisia työvuorolistoja on siis näin ollen $2^{210} / 30!$. Kaikkien näiden mahdollisten vaihtoehtojen läpikäyminen nykyaikaisellakin tietokoneella kestäisi todella kauan.

Miten ongelman voisi ratkaista lineaarisella optimoinnilla? Koska päätösmuuttujat ovat kokonaislukuja, niin Simplex-algoritmi vaikuttaa hyödyttömältä. Marvin Rothstein esittää ratkaisun artikkelissaan vuodelta 1973, jossa käsitellään hyvin samankaltaista ongelmaa sairaalaympäristössä. [1]

Kuten tehtävänannossa olleessa vinkissä kehoitettiin, tarkastellaan vapaapäiviä työpäivien sijasta. Merkitään peräkkäisten vapaapäivien lukumäärää seuraavasti: $x_1 =$ ma-ti, $x_2 =$ ti-ke, $x_3 =$ ke-to, $x_4 =$ to-pe, $x_5 =$ pe-la, $x_6 =$ la-su ja $x_7 =$ su-ma väliset vapaapäivät. Ne ovat myös ratkaisussa käytetyt päätösmuuttujat. Merkitään yksittäisten, ei-peräkkäisten vapaapäivien lukumäärää tiistaista seuraavan viikon maanantaihin u_1, u_2, \dots, u_7 . Kohdefunktiona voidaan käyttää kumpaa tahansa seuraavista:

$$\max \sum_{i=1}^7 x_i \quad \text{tai} \quad \min \sum_{i=1}^7 u_i$$

Ensimmäisessä maksimoidaan peräkkäisten vapaapäivien lukumäärää, jälkimmäisessä minimoidaan ei-peräkkäisten vapaapäivien lukumäärää. Tässä ratkaisussa käytetään ensiksi mainittua tapaa.

Tarvitaan rajoitus, joka takaa, että jokaisena viikonpäivänä on juuri oikea määrä poliiseja vapaalla. Tämä voidaan kirjoittaa muodossa $x_i + x_{i+1} + u_i = b_i$, missä i käy yhdestä seitsemään ja $x_8 = x_1$. Muuttuja b_i kertoo kuinka monta poliisia on vapaalla

tiistaina, b_2 montako poliisia on vapaalla keskiviikkona ja niin edelleen aina seuraavan viikon maanantaihin saakka.

Jotta kunkin poliisin mahdolliset ei-peräkkäiset vapaapäivät eivät osuisi samalle viikonpäivälle, tarvitaan seuraava rajoitus:

$$\left(\sum_{i \neq j} u_j\right) - u_i \geq 0$$

Tämä takaa sen, että yksikään u_i ole suurempi kuin muiden u :n summa. Jos päinvastainen väite olisi tosi, se johtaisi tilanteeseen jossa joku u_i :n määräämänä päivänä vapaataan viettävä poliisi ei voisi viettää toista vapaapäiväänsä minään muuna päivänä. Rajoituksen ansiosta voidaan u_i :n mukaisena päivänä vapaataan viettävälle poliisille osoittaa aina myös jokin toinen vapaapäivä.

Rothsteinin mukaan vaatimus kokonaislukuratkaisusta toteutuu rajoitusten rakenteen vuoksi, eikä sitä tarvitse erikseen määrätä. Hänen mukaansa tämä johtuu siitä, että kyseinen ongelma on itse asiassa eräs sellainen verkkovirtausongelma (capacitated network problem), jolle voidaan todistaa kokonaislukuratkaisun olemassaolo.

2.1. Ero Rothsteinin ratkaisuun verrattuna.

Edellä olevien rajoitusten lisäksi, Rothstein käyttää muuttujaa q , joka kertoo niiden työntekijöiden lukumäärän, joilla ei ole peräkkäisiä vapaapäiviä. Tätä käytetään rajoituksessa seuraavassa rajoituksessa:

$$\sum_{i=1}^7 x_i + q = n, \text{ missä } n \text{ on työntekijöiden lukumäärä.}$$

Tämä takaa sen, että jokaisella työntekijällä on täsmälleen kaksi vapaapäivää. Siitä

seuraa myös, että $\sum_{i=1}^7 u_i = 2q$.

Ratkaistavassa ongelmassa rajoitetta ei kuitenkaan tarvita, koska huomataan, että vapaapäiviä on 30 poliisille juuri sopivasti eli $12 + 6 + 5 + 14 + 9 + 2 + 12 = 2 \cdot 30 = 60$. Myös päivittäinen tarve poliiseista ja viiden päivän viikkotyöaika täsmäävät, eli viikossa on $18 + 24 + 25 + 16 + 21 + 28 + 18 = 5 \cdot 30 = 150$ henkilötöypäivää. Yksikään poliisi ei siis ole töissä tyhjänpanttina, minkä syynä on joko onnellinen sattuma tai vain sopiva tehtävänanto.

3. Ratkaisu R-ohjelmalla

R-kielinen ohjelma vaatii slam-, ja Rglpk-kirjastot. Ne ovat saatavissa CRAN:sta, <http://cran.r-project.org>. Rglpk käyttää GLPK-kirjastoa, joka sisältää mm. Simplex-algoritmin. Ohjelmassa käytetty komento Rglpk_solve_LP käyttää juuri sitä tässä ratkaisussa.

Edellä olevan perusteella saadaan siis seuraavanlainen optimointitehtävä:

$$\max \sum_{i=1}^7 x_i$$

$$\text{s.t.} \quad x_1 + x_2 + u_1 = b_1 = 6$$

$$x_2 + x_3 + u_2 = b_2 = 5$$

$$x_3 + x_4 + u_3 = b_3 = 14$$

$$x_4 + x_5 + u_4 = b_4 = 9$$

$$x_5 + x_6 + u_5 = b_5 = 2$$

$$x_6 + x_7 + u_6 = b_6 = 12$$

$$x_7 + x_1 + u_7 = b_7 = 12$$

$$u_2 + u_3 + u_4 + u_5 + u_6 + u_7 - u_1 \geq 0$$

$$u_1 + u_3 + u_4 + u_5 + u_6 + u_7 - u_2 \geq 0$$

$$\vdots$$

$$u_1 + u_2 + u_3 + u_4 + u_5 + u_6 - u_7 \geq 0$$

$$x_1 \geq 0, \dots, x_7 \geq 0, \quad u_1 \geq 0, \dots, u_7 \geq 0$$

R-ohjelma on seuraavanlainen:

```
Amatr<-matrix(data=0,nrow=14, ncol=14)
for(i in 1:7){Amatr[i,i]<-1}
for(i in 1:7){Amatr[i,(i%7)+1]<-1}
for(i in 1:7){Amatr[i,i+7]<-1}
for(i in 8:14){Amatr[i,<-c(rep(0,times=7),rep(1,times=7))}]
for(i in 8:14){Amatr[i,i]<-(-1)}

Bvekt<-c(6,5,14,9,2,12,12,rep(0, times=7))
Cvekt<-c(rep(1,times=7),rep(0,times=7))
suunta<-c(rep("=", times=7),rep(">=", times=7))

library("slam")
library("Rglpk")
sol<-Rglpk_solve_LP(Cvekt, Amatr, suunta, Bvekt, max = TRUE, verbose = TRUE)
print(sol)
```

Alussa kuudella ensimmäisellä rivillä määritellään matriisi *Amatr*, joka sisältää rajoitteiden kertoimet. *Bvekt* sisältää rajoitteiden vakiot (ollen siis rhs eli right hand side), *Cvekt* sisältää kohdefunktion kertoimet ja *suunta* sisältää käytettävät vertailuoperaattorit. Ei-negatiivisuusehtoja ei tarvitse kertoa eksplisiittisesti ohjelmassa, koska *Rglpk_solve_LP* olettaa implisiittisesti kaikkien muuttujien olevan ei-negatiivisia. Tämän jälkeen ladataan tarvittavat kirjastot ja kutsutaan optimointialgoritmia. Vastaus tallennetaan *sol*-muuttujaan ja tulostetaan näytölle.

Amatr-matriisi:

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]	[,10]	[,11]	[,12]	[,13]	[,14]
[1,]	1	1	0	0	0	0	0	1	0	0	0	0	0	0
[2,]	0	1	1	0	0	0	0	0	1	0	0	0	0	0
[3,]	0	0	1	1	0	0	0	0	0	1	0	0	0	0
[4,]	0	0	0	1	1	0	0	0	0	0	1	0	0	0
[5,]	0	0	0	0	1	1	0	0	0	0	0	1	0	0
[6,]	0	0	0	0	0	1	1	0	0	0	0	0	1	0
[7,]	1	0	0	0	0	0	1	0	0	0	0	0	0	1
[8,]	0	0	0	0	0	0	0	-1	1	1	1	1	1	1
[9,]	0	0	0	0	0	0	0	1	-1	1	1	1	1	1
[10,]	0	0	0	0	0	0	0	1	1	-1	1	1	1	1
[11,]	0	0	0	0	0	0	0	1	1	1	-1	1	1	1
[12,]	0	0	0	0	0	0	0	1	1	1	1	-1	1	1
[13,]	0	0	0	0	0	0	0	1	1	1	1	1	-1	1
[14,]	0	0	0	0	0	0	0	1	1	1	1	1	1	-1

Bvekt-vektori: [1] 6 5 14 9 2 12 12 0 0 0 0 0 0 0 0

Cvekt-vektori: [1] 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0

Suunta: [1] "==" "==" "==" "==" "==" "==" "==" ">=" ">=" ">=" ">=" ">=" ">=" ">=" ">="

3.1. Tulos ja sen tulkinta

Ohjelma antaa seuraavan tuloksen:

```
$optimum
[1] 28
$solution
[1] 4 0 5 9 0 2 8 2 0 0 0 0 2 0
$status
[1] 0
```

Kuten helposti huomataan, optimiksi saadaan 28 peräkkäistä vapaapäivää. Vektori *solution* kertoo, kuinka paljon vapaita on minäkin päivänä. Poliiseilla voi siis olla seuraavat määrät peräkkäisiä vapaita: ma-ti 4, ke-to 5, to-pe 9, la-su 2 ja su-ma 8 kpl. Yksittäiset vapaapäivät sijoittuvat tiistaille ja sunnuntaille, kaksi kummallekin.

4. Herkkyysanalyysi

Ongelman ja ratkaisun erikoisen luonteen vuoksi herkkyysanalyysi on ongelmallinen. Aiemmin mainitun onnellisen sattuman vuoksi voidaan optimi määrittää vain, kun henkilötöypäiviä on $5n$ ja vapaapäiviä $2n$ viikossa, missä n on poliisien lukumäärä.

Oletetaan, että lauantaista tulee entistä rauhattomampi ja tarvittaisiin yksi poliisi lisää. Tällöin tulee osoittaa myös vielä neljä uutta työpäivää ja kaksi uutta vapaapäivää. Nämä neljä uutta työpäivää kannattaa sijoittaa niille työpäiville, jolloin tarve poliiseista on suurin. Nämä ovat siis lauantai, keskiviikko, tiistai ja perjantai. Vapaapäivät kannattaa vastaavasti sijoittaa niille päiville, jolloin tarve poliiseista on muutenkin vähäinen, sunnuntaihin ja maanantaihin. Päivittäinen tarve työvoimalle on siis maanantaista sunnuntaihin seuraava: 18,25,26,16,22,30,18. Koska poliiseja on 31, saadaan vapaapäivien lukumääräksi vastaavasti: 13,6,5,15,9,1,13. R-kielillä:

```
> Bvekt<-c(6,5,15,9,1,13,13,rep(0, times=7))
> print(sol<-Rglpk_solve_LP(Cvekt, Amatr, suunta, Bvekt, max = TRUE))

OPTIMAL SOLUTION FOUND
$optimum
[1] 28

$solution
[1] 4 0 5 9 0 1 9 2 0 1 0 0 3 0
```

Tämä ei kuitenkaan paranna tulosta peräkkäisten vapaapäivien suhteen. Koska keskiviikko ja lauantai vaikuttavat rajoittavilta tekijöiltä, kokeillaan:

```
> Bvekt<-c(6,6,15,9,2,12,12,rep(0, times=7))
> print(sol<-Rglpk_solve_LP(Cvekt, Amatr, suunta, Bvekt, max = TRUE))
```

Saadaan peräkkäisten vapaapäivien suhteen parempi tulos:

```
$optimum
[1] 29

$solution
[1] 4 0 6 9 0 2 8 2 0 0 0 0 2 0
```

Jos työvoiman tarve vähenee, pitäisi sen vähetä vähintään viidellä henkilötyöpäivällä, ennen kuin poliisien lukumäärää voidaan pienentää. Voisimme olettaa, että tarve poliiseista vähenee hiljaisina työpäivinä, kuten maanantaina, torstaina ja sunnuntaina. Olkoon työvoiman tarve maanantaista sunnuntaihin seuraava: 16, 24, 25, 15, 21, 28, 16.

Poliiseja tarvitaan enää 29, jolloin:

```
> Bvekt<-c(6-1,5-1,14,9-1,2-1,12+1,12+1,rep(0, times=7))
> print(sol<-Rglpk_solve_LP(Cvekt, Amatr, suunta, Bvekt, max = TRUE))

OPTIMAL SOLUTION FOUND
$optimum
[1] 26

$solution
[1] 4 0 4 8 0 1 9 1 0 2 0 0 3 0
```

Mikä on selvästi huonompi tulos peräkkäisiä vapaapäiviä ajatellen. Jos taas kiireisimmät päivät helpottavat niin, että jokaisena työpäivänä vähintään viisi poliisia voi olla vapaalla, saadaan:

```
> Bvekt<-c(6-1,5,14-1,9-1,2+3,12-1,12-1,rep(0, times=7))
> print(sol<-Rglpk_solve_LP(Cvekt, Amatr, suunta, Bvekt, max = TRUE))
$optimum
[1] 29

$solution
[1] 5 0 5 8 0 5 6 0 0 0 0 0 0
```

Kukin poliiseista voi siis viettää kaksi peräkkäistä vapaapäivää viikossa. Tämä on paras mahdollinen tulos, mitä ratkaisussa käytetty menetelmä voi tuottaa.

5. Lähteet

- [1] Rothstein, Marvin : "Hospital Manpower Shift Scheduling by Mathematical Programming,". Health Services Research, vol . 8, 1973, sivut 60-66 .