Problemi rasporeda u zračnom prometu

Raguž, Ivona

SVEUČILIŠTE U ZAGREBU

PRIRODOSLOVNO–MATEMATIČKI FAKULTET

MATEMATIČKI ODSJEK

Ivona Raguž

PROBLEMI RASPOREDA U ZRAČNOM PROMETU

Diplomski rad

Voditelj rada:
izv. prof. dr. sc. Marko Vrdoljak

Zagreb, rujan 2020.

Ovaj diplomski rad obranjen je dana ____________________ pred ispitnim povjerenstvom u sastavu:

1. ____________________________, predsjednik
2. ____________________________, član
3. ____________________________, član

Povjerenstvo je rad ocijenilo ocjenom _______________.

Potpisi članova povjerenstva:

1. ____________________________
2. ____________________________
3. ____________________________

Hvala profesoru Marku Vrdoljaku na prenesenom znanju tijekom studija te pomoći i sugestijama pri izradi ovog rada!
Hvala mojoj obitelji koja me pratila u ovoj avanturi!

Sadržaj

Uvod

Motivacija za biranje teme iz područja optimizacije potječe još od slušanja kolegija *Uvod u optimizaciju*, a potom i njegova sljedbenika, kolegija *Operacijska istraživanja*. Tada su mi pažnju privukle metode rješavanja cjelobrojnog linearnog programiranja te općenito modeliranje svakodnevnih problema zadaćama linearnog programiranja.

Neki od problema rasporeda s kojima se danas susreće zrakoplovna industrija su problemi određivanja reda letova, problemi raspoređivanja flote, problemi izrade rotacija te mnogi drugi. Problemi rasporeda u zrakoplovnoj industriji daleko su složenije prirode od općenitijih problema rasporeda. Naime, zrakoplovne industrije moraju poštivati ugovorna ograničenja koja definiraju zakonsku strukturu te ispunjavaju sigurnosne aspekte letenja. Glavni problem čijim se rješavanjem bavimo u ovom radu je problem izrade rotacija posade u zračnom prometu – rotacije se odabiru tako da pokrivaju sve letove koji se planiraju, na način da putovanje posade počne i završi u istoj bazi. Od svih mogućih rotacija, nastoji se izabrati optimalan skup. Cilj je minimizirati ukupan trošak letenja uz zadane uvjete.

Ovaj problem u nastavku razmatramo kao optimizacijski problem cjelobrojnog linearnog programiranja, problem minimizacije. Problem formuliramo na dva načina, „tradicionalnim" pristupom (Poglavlje 1) te pristupom zasnovanim na dužnostima opisan u Poglavlju 3. Posezanje za uvođenjem drugačije formulacije problema čini se razumnijim potezom jer prvi pristup zahtjeva popisivanje svih dozvoljenih rotacija što može biti izrazito zahtjevno. Također, linearna relaksacija zadaće u skladu s novom formulacijom daje bolju ogradu na optimalnu vrijednost funkcije cilja izvorne zadaće u odnosu na linearnu relaksaciju zadaće u skladu s tradicionalnim pristupom.

Poglavlje 2 započinje definiranjem pojmova iz područja geometrije poliedarskih skupova koje ćemo koristiti u daljnjim razmatranjima. U nastavku je detaljno objašnjena teorija koja stoji u pozadini generacije stupaca. Korištena je Dantzig-Wolfeova dekompozicija koja se zasniva na teoremu dekompozicije za poliedarski skup. Transformacija polazne zadaće Dantzig-Wolfeovom dekompozicijom rezultirat će zadaćom linearnog programiranja za čije se rješavanje metoda generacije stupaca nameće kao dobar odabir. U novoj zadaći zasnovaj na Dantzig-Wolfeovoj dekompoziciji broj uvjeta je smanjen, a kako je čest slučaj da će novi oblik zadaće linearnog programiranja imati veliki broj varijabli, posežemo za metodom generacije stupaca pri njezinu rješavanju.

Poglavlje 2 nastavlja se opisom još jedne metode rješavanja, Lagrangeove relaksacije. Opisana je motivacija za korištenje ove metode te je prikazana ideja rješavanja Lagrangeove relaksacije preko ekstremnih točaka i ekstremnih zraka, sukladno ideji Dantzig-Wolfeove dekompozicije. Poglavlje završavamo opisivanjem postupaka konveksifikacije i diskretizacije u cilju rješavanja problema cjelobrojnog linearnog programiranja.

Istraživanje teme započela sam čitanjem izvora [1]. Teoriju koja je u pozadini metode generacije stupaca te Dantzig-Wolfeovu dekompoziciju upoznala sam čitajući članak [3], dok sam teoriju Lagrangeove relaksacije proučavala čitajući izvor [2]. Čitanje izvora [4] potaklo mi je zanimanje za drugačijom formulacijom problema.

Moja implementacija metode generacije stupaca u kombinaciji s Dantzig-Wolfeovom dekompozicijom može se pronaći na linku `https://github.com/ivona13/CG-DW`.

Poglavlje 1

Problemi rasporeda u zračnom prometu

1.1 Uvod

U zrakoplovnoj industriji postoji mnogo problema rasporeda i planiranja različite vrste. Zrakoplovne tvrtke troše veliku količinu novca na istraživanje i rješavanje takvih problema. Najčešće područje istraživanja obuhvaća operacijska istraživanja. Problemi rasporeda i planiranja nastoje se ovim tehnikama reprezentirati problemom matematičke optimizacije. Cilj rješavanja ovih problema je smanjenje troškova zrakoplovne indrustrije te povećanje njihovog tržišnog udjela. U procesu planiranja zrakoplovne kompanije postoje četiri uobičajena koraka:

- određivanje reda letova (*engl. timetable*)

  Prvo se kreira raspored letova ovisno o zahtjevima tržišta (uz analizu konkurentnosti). Kako zračni promet postaje sve „gušći", organizacija letova postaje sve zahtjevnija.

- raspored flote (*engl. fleet scheduling*)

  Obzirom na raspored letova, zrakoplovi se dodjeljuju letovima. U zrakoplovnim tvrtkama postoji nekoliko vrsta zrakoplova dodijeljenih različitim flotama, ovisno o njihovu tipu. Održavanje zrakoplova najvažnija je aktivnost u ovom koraku. Zrakoplovi posjećuju skladišta za održavanje i provjeru unutar redovnih razdoblja.

- izrada rotacija (*engl. crew pairing*)

  Nakon što je napravljen raspored letova te su zrakoplovi dodijeljeni letovima, u sljedećem koraku se razmatra posada (obično se sastoji od pilota, kopilota te stjuardesa). Svaka vrsta zrakoplova zahtjeva određeni broj članova posade. Pripadajući članovi posade određene rotacije odabiru se tako da pokrivaju sve letove koji se planiraju – promatraju se letovi koji počinju u bazi posade i vraćaju se u bazu posade ili na zajednički terminal. Kod dodjeljivanja posade određenim letovima nastoje se minimizirati troškovi te posada mora biti kvalificirana za određeni zrakoplov. Raspored rada mora poštovati maksimalno vrijeme udaljenosti posade od svoje baze te zakonske propise o radnom vremenu članova posade zrakoplova, odnosno ne smiju biti na dužnosti dulje od maksimalno propisanog vremena. Dakle, proces planiranja posade obuhvaća proračun potrebnog broja posade što uključuje i upravljanje radnim vremenom i vremenom leta, trajanjem dežurstava i prijevoza, dnevne, tjedne i godišnje odmore te broj uzlijetanja i slijetanja u skladu sa zakonskom regulativom kao i internim aktima kompanije.

- asignacija rotacija članovima posade (*engl. crew assignment*)

  U ovom koraku se svakom članu posade dodjeljuje određena rotacija. To je vrlo složen proces koji može uključivati optimizaciju gdje je cilj neka vrsta „pravednosti".

U nastavku rada obrađuje se problem planiranja posada. Općenito, planiranje posada podrazumijeva rješavanje dva problema – izrada rotacija i dodjela rotacija članovima posada.

Problem izrade rotacija i pronalazak odgovarajućih rješava se na temelju rasporeda letova. Po rješavanju ovog problema, posade se dodjeljuju pronađenim skupinama. Izrada rotacija je najvažniji korak i može doprinijeti velikoj uštedi novca u problemu rasporeda posada.

1.2 Opis problema

Prije detaljnije razrade problema, potrebno je definirati nekoliko tematskih pojmova.

Svaka posada pilotske kabine kvalificirana je za letenje određenog tipa flote ili pak skupa sličnih tipova (*engl. fleet family*). Stoga raspored posade rješavamo za svaki pojedinačni tip posade (i to za one letove koji su već dodijeljeni odgovarajućim tipovima flota).

Problem rasporeda posada za ulaz dobije skup letova koji moraju biti „pokriveni". Pojedinačni letovi se zajedno grupiraju formirajući dužnosti (*engl. duty periods*) – niz uzastopnih letova koji čine radni dan posade. Letovi koji čine niz međusobno su odvojeni kratkim pauzama. Dužnosti potom formiraju rotacije (*engl. pairings*), putovanja posade koja obuhvaćaju jedan ili više radnih dana s pauzama između njih tako da putovanje počinje i završava u istoj bazi. Letovi, dužnosti, parovi i mjesečni raspored sastavni su dijelovi rasporeda posade.

Općenito, raspored posada može sadržavati dnevne, tjedne i mjesečne letove. Problem izrade rotacija posade posade također može biti definiran na dnevnoj, tjednoj ili mjesečnoj bazi kako bi obuhvatio sve moguće vrste letova obzirom na njihovu učestalost. Dnevni problemi pretpostavljaju izvršavanje svih letova svakog dana. Svaku rotaciju obavlja različita posada. Također, niti jedan let se u rotaciji ne može pojaviti više od jednog puta.

Formulacija problema rotacija posade

U problemu izrade rotacija posade cilj je pronaći optimalan skup rotacija iz svih mogućih. Pri izboru rotacije, svaki od letova mora biti obuhvaćen točno jedanput te se troškovi putovanja minimiziraju. Također se moraju poštivati propisana pravila.

Ovaj problem matematički možemo opisati problemom optimizacije, i to problemom cjelobrojnog linearnog programiranja.

Neka je F skup letova koji se moraju obuhvatiti te neka je P skup svih dopuštenih rotacija. Varijable y_p imaju sljedeće značenje:

$$y_p = \begin{cases} 1, & \text{ako je rotacija } p \text{ uključena u rješenje} \\ 0, & \text{inače.} \end{cases} \tag{1.1}$$

Problem izrade posade definiramo problemom linearnog programiranja:

$$\begin{cases} \sum_{p \in P} c_p y_p \to \min \\ \sum_{p: i \in p} y_p = 1, \quad i \in F \\ y_p \in \{0, 1\}, \quad p \in P. \end{cases} \tag{1.2}$$

Formulacija (1.2) zahtjeva eksplicitno nabrajanje svih rotacija. Takvo nabrajanje je zahtjevno zbog velikog broja potencijalnih rotacija. U praksi, takvo popisivanje nije moguće . Prema tome, heuristički optimizacijski pristupi ili metoda generacija stupaca korišteni su pri rješavanju ovog problema.

Brojne zrakoplovne kompanije također zahtjevaju ograničenja na broj radnih sati koji odgovaraju pojedinoj bazi posade. Ovi uvjeti zahtjevaju sljedeće: broj radnih sati provedenih u određenoj rotaciji od strane posade iz određene baze mora biti unutar određenih granica, ovisno o brojnosti posade u određenoj bazi. Takve uvjete nazovimo uvjetima uravnoteženja (*engl. balacing constraints*).

Primjer 1.2.1. *Na primjeru sedam letova ilustrirajmo problem uparivanja posade.*

Let	***Polazište***	***Odredište***	***Početak***	***Kraj***	***Učestalost***
1	*A*	*B*	08 : 00	09 : 00	*svaki dan osim subote i nedjelje*
2	*B*	*C*	10 : 00	11 : 00	*svaki dan*
3	*C*	*D*	13 : 00	14 : 00	*svaki dan osim nedjelje*
4	*C*	*A*	15 : 00	16 : 00	*svaki dan*
5	*D*	*A*	15 : 00	16 : 00	*svaki dan osim subote*
6	*A*	*B*	17 : 00	18 : 00	*svaki dan*
7	*B*	*C*	11 : 00	12 : 00	*svaki dan osim subote i nedjelje*

U nastavku pretpostavljamo da su sve zračne luke (polazišta i odredišta) u istoj vremenskoj zoni.

Za početak razmotrimo problem dnevnog rasporeda. Neke od valjanih dužnosti su sljedeće:

$$\begin{gathered} D_1 = \{1\}, \quad D_2 = \{2\}, \quad D_3 = \{3\}, \quad D_4 = \{4\}, \\ D_5 = \{5\}, \quad D_6 = \{6\}, \quad D_7 = \{7\}, \quad D_8 = \{1,2\}, \\ D_9 = \{1,7,3\}, \quad D_{10} = \{2,3\}. \end{gathered}$$

U nastavku pretpostavimo da su zračne luke A, C, D baze posade. Razmotrimo moguća uparivanja izražena u terminima dužnosti:

$$\begin{array}{lll} P_1 = \{D_4, D_8\} & P_2 = \{D_9, D_5\} & P_3 = \{D_5, D_6, D_{10}\} \\ P_4 = \{D_4, D_6, D_7\} & P_5 = \{D_1, D_7, D_4\} & P_6 = \{D_5, D_7, D_9\}. \end{array}$$

Promotrimo li rotaciju P_6, uočavamo da je let 7 u obzir uzet dva puta - stoga ova rotacija ne može biti rješenje dnevnog rasporeda. Primijetimo da je rotacija P_1 također moglo biti definirano skupom dužnosti $\{D_4, D_1, D_2\}$. Obzirom da obe rotacije imaju istu bazu (baza C), u modelu je moguće pojavljivanje samo jedne od tih rotacija, i to one manjeg troška. U primjeru razmatrajmo rotaciju $\{D_4, D_8\}$.

Neka su cijene uparivanja sljedeće:

$$c_1 = c_2 = c_3 = c_4 = 4, \quad c_5 = 5.$$

Definirajmo problem uparivanja kao problem binarnog (linearnog) programiranja (1.2):

$$
\begin{aligned}
4y_1 + 4y_2 + 6y_3 + 4y_4 + 5y_5 &\to \min \\
y_1 + y_2 \qquad\qquad + y_5 &= 1 \quad (\textit{let } 1) \\
y_1 \qquad + y_3 \qquad\qquad &= 1 \quad (\textit{let } 2) \\
y_2 + y_3 \qquad\qquad &= 1 \quad (\textit{let } 3) \\
y_1 \qquad\qquad + y_4 + y_5 &= 1 \quad (\textit{let } 4) \\
y_2 + y_3 \qquad\qquad &= 1 \quad (\textit{let } 5) \\
y_3 + y_4 \qquad &= 1 \quad (\textit{let } 6) \\
y_2 \qquad + y_4 + y_5 &= 1 \quad (\textit{let } 7) \\
y_1, \quad y_2, \quad y_3, \quad y_4, \quad y_5 &\in \{0, 1\}
\end{aligned}
$$

Zahtjevamo li najmanje tri, a najviše šest radnih sati leta dodijeljenih bazama A i D, te najviše pet sati dodijeljenih bazi C, dobijemo sljedeće uvjete:

$$
\begin{aligned}
3 \le 4y_2 + 3y_5 \le 6 \quad &(\textit{baza A}) \\
0 \le 3y_1 + 3y_4 \le 5 \quad &(\textit{baza C}) \\
3 \le 4y_3 \le 6 \quad &(\textit{baza D})
\end{aligned}
$$

Optimalno rješenje ovog problema uključuje uparivanja 3 *i* 5*. Vrijednost funkcije cilja optimalnog rješenja je* 9.

Restrikcije pri izradi rotacija

U zrakoplovnoj industriji postoje ugovorna ograničenja koja definiraju zakonsku strukturu i troškove letova unutar rotacija. Elementarna odluka u problemima rasporeda je odluka o dodjeli posade određenoj rotaciji. Trošak te dodjele je vrlo složeno odrediti. Posade su plaćene ovisno o vremenskom periodu leta, uz dodatne nadoknade za vrijeme provedeno u odmoru između dvaju letova. Sukladno s time, o troškovima možemo razmišljati u terminima vremena. Cilj rasporeda posade je minimiziranje plaćanja troškova nastalih izvan okvira vremena stvarnog letenja.

Sigurnosni aspekti letenja postavljaju određena pravila. Promatrajući periode zaduženja, pravila i restrikcije referiraju se na kombinacije letova dodijeljenih jednoj posadi. Obično ista posada ostaje zajedno tijekom cijelog perioda zaduženja.

Promotrimo neke od restrikcija. Jedna od esencijalnih je slijedni događaj letova u vremenu i prostoru. Postoji zahtjev na minimalno vrijeme odmora između dvaju slijednih letova (*engl. minimum idle time*, *sit time*), kao i zahtjev na maksimalno dopušteno proteklo vrijeme između dvaju letova (*engl. maximum elapsed time*). Također, postoje strogi propisi koji ograničavaju ukupan broj sati letenja, tj. vremensko ograničenje letenja posade tijekom dužnosti u jednom radnom danu (*engl. total number of flying hours*). Troškovi posade na razini dnevne dužnosti obično se izražava kao maksimum sljedećih triju vrijednosti. Prva vrijednost je vrijeme letenja (*engl. flying time*). Druga vrijednost je udio ukupnog vremena odmora između letova u radnom vremenu (*engl. total elapsed time*). Treća vrijednost je zajamčeni minimalni broj radnih sati. Ovakva struktura plaćanja prvenstveno plaća posadu za njihove letove, no također daje naknadu posadama s vrlo kratkim dužnostima ili dužnostima s dugim razdobljem odmora između letova. Zapišemo li, trošak dužnosti d jednak je

$$c_d = \max\{f^{(d)}_{elapse},\ time_{fly},\ min_{\mathrm{guar}}\}, \tag{1.3}$$

s oznakama:

- $c_d \ldots$ trošak dužnosti d
- $f^{(d)}_{elapse} \ldots$ udio odmora u radnom vremenu (dužnosti)
- $\mathrm{time}_{\mathrm{fly}} \ldots$ vrijeme letenja tijekom dužnosti
- $min_{\mathrm{guar}} \ldots$ minimalno zajamčeno vrijeme letenja.

Legalne rotacije mogu se sastojati od maksimalno *maxduties* dužnosti, a minimalan broj sati odmora između dužnosti je *minrest*. U slučaju dugog leta u trajanju od *maxfly* sati s neadekvatnim kratkim odmorom od *minrest* sati, moguće je posadi dodijeliti kompenzacijski odmor. Trošak rotacije p, u oznaci c_p, jednak je maksimumu triju vrijednosti: prva vrijednost je $min_{\mathrm{guar}} \times$ broj dužnosti u rotaciji (NDP); druga vrijednost je udio ukupnog vremena odmora u radnom vremenu rotacije ($f_{total\_elapsed\_time}$); treća vrijednost je zbroj troškova individualnih dužnosti u uparivanju.

$$c_p = \max\left\{min_{\mathrm{guar}} \cdot NDP,\ f_{\mathrm{total\_elapsed\_time}},\ \sum_{d \in p} c_d\right\} \tag{1.4}$$

Također postoje i ostali uvjeti koji moraju biti ispunjeni, osim pravila vezanih uz rotacije i dužnosti. Takvi su, naprimjer, uvjeti vezani uz baze posada. Za svaku od baza se razlikuje broj ukupnih sati letenja dodijeljenih pojedinoj posadi. Uvedenim ograničenjima na baze nastoji se osigurati otprilike jednak broj radnih sati posada u različitim bazama.

Poglavlje 2

Metode rješavanja problema

Posljednjih godina je veliki naglasak stavljen na metodi grananja i odsijecanja (*engl. branch-and-cut*) pri rješavanju zadaća cjelobrojnog programiranja.

Branch-and-cut pristup koristi dvije tehnike:

- odsijecajuće ravnine (*engl. cutting planes*) – pri rješavanju problema cjelobrojnog linearnog programiranja promatra se relaksirana zadaća (linearna relaksacija – ignoriranje uvjeta cjelobrojnosti) te se ponavlja „odsijecanje" dijelova politopa dodavanjem novih uvjeta u cilju postizanja cjelobrojnog rješenja zadaće. Pronalazak odsijecajućih ravnina podrazumijeva rješavanje problema separacije.

- metoda grananja i ogradivanja (*eng. branch-and-bound*) – pristup „podijeli pa vladaj" gdje se problem „razbija" na potprobleme koje je lakše riješiti

Ukoliko optimalno rješenje relaksirane zadaće cjelobrojnog linearnog programiranja nije dopustivo u smislu originalne zadaće, rješava se potproblem koji nastoji otkriti nejednakosti koje nisu ispunjene. Ako su takve nađene, neke od njih se dodaju relaksiranoj zadaći kako bi otklonili nedopustiva rješenja. Metoda grananja i određivanja cijene (*engl. branch-and-price*) je slična metodi *branch-and-cut*, osim što se fokusira na generaciju stupaca, a ne redaka, kao druga metoda.

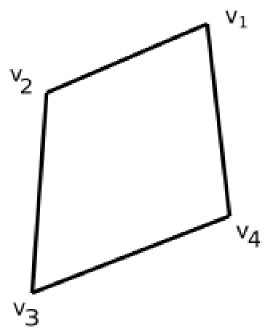
U algoritmu grananja i određivanja cijene, većina stupaca je izostavljena iz relaksirane zadaće – teško je učinkovito riješiti zadaću s velikim brojem varijabli, a i većina njih ne doprinosi optimalnom rješenju (odgovarajuća varijabla u zadaći je jednaka nuli). Kako bismo provjerili optimalnost rješenja takve zadaće, rješavamo novi potproblem, problem određivanja cijene (*engl. pricing problem*) dualne zadaće. Rješavanjem ovog problema želimo utvrditi eventualno postojanje stupaca koji bi ušli u bazu. Ako takvi stupci postoje, rješenje zadaće se optimira. Grananje se odvija u slučaju kada ne postoji stupac koji bi ušao u bazu te optimalno rješenje relaksirane zadaće nije dopustivo.

U nastavku opisujemo metodu generacije stupaca primjenom Dantzig-Wolfeove dekompozicije. Za potrebe opisa metode definiramo pojmove iz područja geometrije poliedarskih skupova. Promotrit ćemo ideju Lagrangeove relaksacije te objasniti vezu između metode generacije stupaca koristeći Dantzig-Wolfeovu dekompoziciju i rješavanja Lagrangeova duala u cjelobrojnom programiranju.

2.1 Geometrija poliedarskih skupova

Definiramo pojmove na kojima se temelji Dantzig-Wolfeova dekompozicija.

Definicija 2.1.1. *Politop je konveksna ljuska konačno mnogo točaka* $v_1, \dots, v_p$.



Slika 2.1: Politop

Definicija 2.1.2. *Kažemo da je* $C \in \mathbb{R}^n$ *konus ako*

$$(\forall x \in C)(\forall t \geq 0) \quad tx \in C.$$

Ako je C ujedno i konveksan skup, za C kažemo da je konveksan konus.

Napomena 2.1.3. *Definicija 2.1.2 označava da za svaku svoju točku x konus S sadrži i zraku iz ishodišta koja prolazi tom točkom.*

Napomena 2.1.4. *Prazan skup također smatramo konusom.*

Definicija 2.1.5. *Konusna kombinacija točaka* $x_1, \dots, x_k \in \mathbb{R}^n$ *je točka oblika*

$$x = \sum_{i=1}^{k} \lambda_i x_i, \text{ gdje su } \lambda_1, \dots, \lambda_k \geq 0. \tag{2.1}$$

Skup svih konusnih kombinacija točaka iz skupa $S \subseteq \mathbb{R}^n$ naziva se konusna ljuska skupa S i označava se s $co\ S$.

$$co\ S = \Big\{ \sum_{i=1}^{k} \lambda_i x_i : \lambda_1, \ldots, \lambda_k \geq 0, x_1, \ldots, x_k \in S \Big\}. \tag{2.2}$$

Propozicija 2.1.6. *Skup $S \in \mathbb{R}^n$ je konveksan konus ako i samo ako je $S = co\ S$.*

Definicija 2.1.7. *Za konus C kažemo da je konačnogeneriran skupom točaka $\{x_1, \ldots, x_k\}$ ako vrijedi $C = co\ \{x_1, \ldots, x_k\}$. Označavamo ga s $C(x_1, \ldots, x_k)$.*

Definicija 2.1.8. *Kažemo da je $x \in K$ ekstremna točka konveksnog skupa K ako ne postoje međusobno različite točke $y, z \in K$ takve da vrijedi $x = \dfrac{y+z}{2}$.*

Primjer 2.1.9. *Promotrimo li politop sa slike 2.1, ekstremne točke skupa su: v_1, v_2, v_3 i v_4.*

Definicija 2.1.10. *Kažemo da je $v \in \mathbb{R}^n \setminus \{0\}$ recesivni smjer skupa K u točki $x \in K$ ako zraka $\{x + tv : t \geq 0\}$ pripada skupu K.*

Definicija 2.1.11. *Kažemo da je recesivni smjer $v \neq 0$ skupa K ekstremna zraka ako ne postoje dva recesivna smjera v_1, $v_2 \in K$, $v_1 \neq \lambda v_2$ za neki $\lambda > 0$ takva da vrijedi $v = \dfrac{1}{2}v_1 + \dfrac{1}{2}v_2$.*

Definicija 2.1.12. *Definiramo recesivni konus skupa $K \in \mathbb{R}^n$ kao skup svih recesivnih smjerova kojem dodajemo i nul-vektor.*

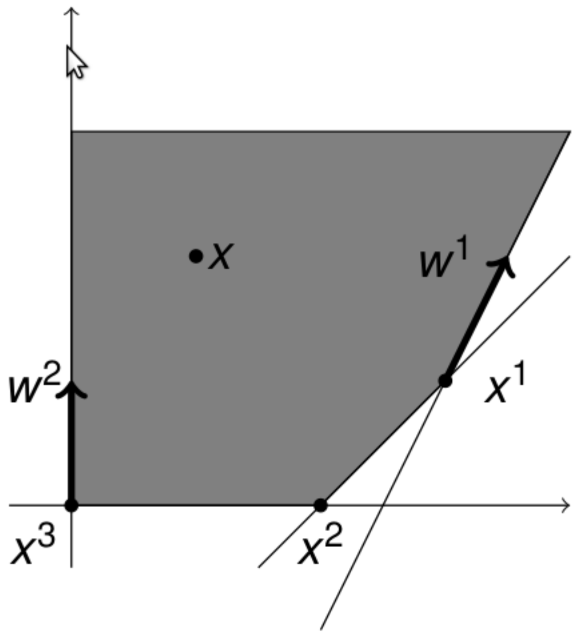
Definicija 2.1.13. *Skup $\{x \in \mathbb{R}^n \mid Ax \leq b,\ A \in M_{m,n}(\mathbb{R}),\ b \in \mathbb{R}^m\}$ nazivamo poliedarski skup.*

Teorem 2.1.14. *Skup $K \subseteq \mathbb{R}^n$ je poliedarski ako i samo ako postoji politop P i konačnogenerirani konus C takvi da je*

$$K = P + C. \tag{2.3}$$

Napomena 2.1.15. *U gornjem rastavu, konus C je jednoznačno određen – radi se o recesivnom konusu poliedarskog skupa K.*

Primjer 2.1.16. *U nastavku navodimo primjer poliedarskog skupa te njegovih ekstremnih točaka i zraka.*



Slika 2.2: Poliedarski skup

Točke x^1, x^2 *i* x^3 *su ekstremne točke poliedarskog skupa sa slike 2.2. Sa* w^1 *i* w^2 *označene su ekstremne zrake skupa.*
Opća reprezentacija proizvoljne točke x *poliedarskog skupa sa slike 2.2 je sljedeća:*

$$x = \lambda_1 x^1 + \lambda_2 x^2 + \lambda_3 x^3 + \mu_1 w^1 + \mu_2 w^2,$$
$$0 \leq \lambda_i \leq 1, \quad i = 1, 2, 3$$
$$\sum_{i=1}^{3} \lambda_i = 1$$
$$\mu_i \geq 0, \quad i = 1, 2.$$

2.2 Generacija stupaca

U brojnim primjenama, matrica uvjeta zadaće linearnog programiranja je rijetko popunjena, s uočenim strukturama. Obično se u takvim primjenama podskupovi varijabli i uvjeta pojavljuju u nezavisnim grupama, formirajući strukture. Problemi rasporeda u zračnom prometu jedni su od takvih primjera linearnog programiranja.

Promotrimo sljedeću zadaću linearnog programiranja:

$$\begin{cases} c^T\lambda \to \min \\ A\lambda \geq b \\ \lambda \geq 0. \end{cases} \tag{2.4}$$

S J označimo skup svih indeksa varijabli linearnog programiranja (λ je vektor varijabli linearnog programa). Neka skup $\mathcal{A} = \{a_j \mid j \in J\}$ čine stupci matrice uvjeta A. U svrhu demonstriranja metode generacije stupaca, zapišimo zadaću (2.4) kako bismo istakli stupce matrice A:

$$\begin{cases} \sum_{j\in J} c_j\lambda_j \to \min \\ \sum_{j\in J} a_j\lambda_j \geq b \\ \lambda_j \geq 0, \quad j \in J. \end{cases} \tag{2.5}$$

Želimo li zadaću riješiti simpleks metodom, u svakoj njenoj iteraciji tražimo nebazičnu varijablu koja će postati bazičnom i poboljšati vrijednost funkcije cilja (u slučaju zadaće (2.4), smanjiti je). Nameće se pitanje o kriteriju ulaska varijable u bazu. Računamo trošak svake varijable λ_j, $j \in J$.

Neka je $u \geq 0$ vektor dualnih varijabli. Želimo naći indeks $j_0 \in J$ takav da vrijedi:

$$\begin{aligned} \overline{c}_j &:= c_j - u^T a_j, \quad j \in J \\ j_0 &:= \operatorname{argmin}\{\overline{c}_j \mid j \in J\}. \end{aligned} \tag{2.6}$$

Ako su sve gornje razlike nenegativne, u je dopustiva točka dualne zadaće. U slučaju da vrijede i uvjeti komplementarnosti, u i λ su optimalne točke dualne, odnosno primarne zadaće. Ovakav pristup nije pogodan niti efikasan za skupove J velikog kardinaliteta, tj. za zadaće linearnog programiranja s velikim brojem varijabli.

U praksi, rješavanje zadaće (2.5) započinje inicijalizacijom malog skupa stupaca $J' \subseteq J$, formirajući reduciranu zadaću, (*engl. restricted master problem*). Uz pretpostavku da postoji dopustiva točka reducirane zadaće, označimo s $\overline{\lambda}$, $\overline{u}$ optimalnu točku primarne, odnosno dualne zadaće. Kada su stupci a_j, $j \in J$, prešutno dani kao elementi skupa $\mathcal{A}$ i kada se koeficijenti c_j mogu izračunati iz a_j koristeći komplementarnost, tada potproblem

$$\overline{c}^* := \min\{c(a) - u^T a \mid a \in \mathcal{A}\} \tag{2.7}$$

daje odgovor o biranju sljedećeg stupca. Ovisno o različitim vrijednostima $\overline{c}^*$, moguće je zaključiti sljedeće:

- Ako je $\overline{c}^* \geq 0$, tada su svi koeficijenti $\overline{c}_j$ nenegativni pa je trenutna baza optimalna, tj. $\overline{\lambda}$ je optimalna točka izvorne zadaće.

- Ako je $\overline{c}^* < 0$, tada reduciranoj zadaći dodamo stupac koji je rješenje problema (2.7) te ponavljamo postupak na reduciranoj zadaći, čime smanjujemo vrijednost funkcije cilja reducirane zadaće (vrijednost funkcije cilja zadaće nastale uvođenjem novog stupca je za $\overline{c}^*$ manja od vrijednosti funkcije cilja zadaće prije njegova uvođenja).

Opisanu metodu nazivamo **generacijom stupaca**, a izraz (2.7) nazivamo generatorom stupaca.

Metoda generacije stupaca završi u konačno mnogo koraka i egzaktna je. Za svako rješenje dobiveno u „međukoracima" možemo dati ocjenu kvalitete. Označimo sa $\overline{z}$ optimalnu vrijednost funkcije cilja reduciranog problema. Neka je s $\overline{u}$ označena optimalna točka dualne zadaće zadaći (2.4). Po teoremu jake dualnosti vrijedi:

$$\overline{z} = \overline{u}^T b.$$

Kada postoji gornja granica U za optimalno rješenje $\overline{\lambda} \in \mathbb{Z}_+^{|J|}$ polaznog problema (2.4) takva da $\sum_{j \in J} \overline{\lambda}_j \leq U$, tada postoji i donja granica na optimalnu vrijednost funkcije cilja z^* u svakoj iteraciji pri rješavanju reducirane zadaće: $\overline{z}$ ne možemo umanjiti za više od $U \cdot \overline{c}^*$, gdje je $\overline{c}^*$ optimalna točka problema (2.7). Dodatno objašnjeno, $\overline{c}^*$ predstavlja najmanji trošak koji je moguće ostvariti uvođenjem nekog od stupaca problema (2.4). Dakle,

$$\overline{z} + U \cdot \overline{c}^* \leq z^* \leq \overline{z}. \tag{2.8}$$

2.3 Dantzig-Wolfeova dekompozicija

Promotrimo zadaću linearnog programiranja:

$$\begin{cases} c^T x \to \min \\ Ax \geq b \\ Dx \geq d \\ \quad x \geq 0. \end{cases} \tag{2.9}$$

Neka je $P = \{x \in \mathbb{R}_+^n : Dx \geq d\} \neq \emptyset$. Po teoremu dekompozicije za poliedarski skup, svaku točku x skupa P možemo zapisati u obliku sume konveksne kombinacije ekstremnih točaka skupa P i konusne kombinacije ekstremnih zraka skupa P. Zapišimo točku $x \in P$ u takvom obliku:

$$x = \sum_{q \in I_Q} \lambda_q p_q + \sum_{r \in I_R} \mu_r w_r, \quad \sum_{q \in I_Q} \lambda_q = 1, \quad \lambda \in \mathbb{R}_+^{|\mathbb{I}_Q|}, \quad \mu \in \mathbb{R}_+^{|I_R|}, \tag{2.10}$$

gdje je $Q = \{p_q \mid q \in I_Q\}$ skup ekstremnih točaka skupa P te $R = \{w_r \mid r \in I_R\}$ skup ekstremnih zraka skupa P.

Iskoristimo prikaz (2.10) točke x u prikazu linearne zadaće (2.9). Zadaća poprima sljedeći oblik:

$$\begin{cases} \sum_{q\in I_Q} \lambda_q(c^T p_q) + \sum_{r\in I_R} \mu_r(c^T w_r) \to \min \\ \sum_{q\in I_Q} \lambda_q(Ap_q) + \sum_{r\in I_R} \mu_r(Aw_r) \geq b \\ \sum_{q\in I_Q} \lambda_q = 1 \\ \lambda_q \geq 0, \quad q \in I_Q \\ \mu_r \geq 0 \quad r \in I_R. \end{cases} \tag{2.11}$$

Problem (2.11) ekvivalentan je problemu (2.9). Varijable problema (2.11) su težine ekstremnih točaka $p_q, q \in I_Q$ i ekstremnih zraka $w_r, r \in I_R$. Broj varijabli $|I_Q| + |I_R|$ u novoj formulaciji problema može biti jako velik, no broj uvjeta je manji. Riješili smo se uvjeta $Dx \geq d$ i zamijenili ga uvjetom $\sum_{q\in I_Q} \lambda_q = 1$.

Označimo s π i α dualne varijable prvog skupa uvjeta i uvjeta nakon, redom. Uvjeti dualnog problema su:

$$\begin{cases} \pi^T(Ap_q) + \alpha \leq c^T p_q, \quad q \in I_Q \\ \pi^T(Aw_r) \leq c^T w_r, \quad r \in I_R. \end{cases} \tag{2.12}$$

Pomoću dualnog problema provjeravamo kada je baza za dani problem optimalna.

Trošak varijable p_q, $q \in I_Q$ je:

$$c^T p_q - \pi^T(Ap_q) - \alpha = (c - A^T\pi)^T p_q - \alpha. \tag{2.13}$$

Slično, trošak varijable w_r, $r \in I_R$ je:

$$c^T w_r - \pi^T(Aw_r) = (c - A^T\pi)^T w_r. \tag{2.14}$$

Kako problem (2.11) obično ima veliki broj varijabli, posežemo za metodom generacije stupaca pri njegovu rješavanju. Pretpostavimo da u određenoj iteraciji rješavanja problema (2.11) imamo podskup stupaca s indeksima u skupu $I'_Q \cup I'_R$.
Sada reducirana zadaća (2.11) izgleda ovako:

$$\begin{cases} \sum_{q\in I'_Q} \lambda_q(c^T p_q) + \sum_{r\in I'_R} \mu_r(c^T w_r) \to \min \\ \sum_{q\in I'_Q} \lambda_q(A p_q) + \sum_{r\in I'_R} \mu_r(A w_r) \geq b \\ \sum_{q\in I'_Q} \lambda_q = 1 \\ \lambda_q \geq 0, \quad q \in I'_Q \\ \mu_r \geq 0 \quad r \in I'_R. \end{cases} \tag{2.15}$$

Označimo sa z^* optimalnu vrijednost funkcije cilja problema (2.11) te sa $\overline{z}$ optimalnu vrijednost funkcije cilja problema (2.15). Jasno je da vrijedi $\overline{z} \geq z^*$.

Neka je sada $(\overline{\pi}, \overline{\alpha})$ optimalna točka dualne zadaće reduciranog problema (2.15). Rješavamo problem određivanja cijena u cilju eventualne generacije novog stupca:

$$\overline{c}_s = \min_{x\in P}(c - A^T\overline{\pi})^T x - \overline{\alpha}, \tag{2.16}$$

gdje je $\overline{\alpha}$ konstanta, pa ju pri praktičnom rješavanju problema minimizacije (2.16) možemo zanemariti.
Postoje tri moguća ishoda rješavanja problema (2.16):

- Ako je $\overline{c}_s < 0$ i vrijednost funkcije cilja problema (2.16) je konačna, optimalna točka zadaće problema je ekstremna točka p_{q_0}, $q_0 \in I_P \setminus I'_P$. Reduciranoj zadaći (2.15) dodajemo novi stupac
$$\begin{bmatrix} c^T p_{q_0} \\ A p_{q_0} \\ 1 \end{bmatrix}$$
te $I'_Q \leftarrow I'_Q \cup \{q_0\}$.

- Ako je $\overline{c}_s = -\infty$ (funkcija cilja problema (2.16) je neograničena), optimalna točka problema je ekstremna zraka w_{r_0}, $r_0 \in I_R \setminus I'_R$. Reduciranoj zadaći (2.15) dodajemo novi stupac
$$\begin{bmatrix} c^T w_{r_0} \\ A w_{r_0} \\ 0 \end{bmatrix}$$
te $I'_R \leftarrow I'_R \cup \{r_0\}$.

- Ako je $\overline{c}_s \geq 0$, tada je optimalna točka reducirane zadaće (2.15) optimalna točka zadaće (2.11).

Pomoću izraza (2.8) za gornju i donju ogradu, uzimajući u obzir uvjet konveksnosti, dobijemo sljedeće:

$$\overline{z} + \overline{c}_s \leq z^* \leq \overline{z}, \tag{2.17}$$

uz oznake kao i u izrazu (2.8). Primijetimo kako je donja ograda valjana i u slučaju kada rješavanje potproblema rezultira generiranjem ekstremne zrake, tj. u slučaju $\overline{c}_s = -\infty$.

Primjer 2.3.1. *Riješimo sljedeću zadaću prethodno opisanom metodom:*

$$\begin{cases} -4x_1 - x_2 - 6x_3 \to \ min \\ 3x_1 + 2x_2 + 4x_3 = 17 \\ \quad 1 \leq x_1 \leq 2 \\ \quad 1 \leq x_2 \leq 2 \\ \quad 1 \leq x_3 \leq 2. \end{cases} \tag{2.18}$$

Označimo:

- $P = \{x \in \mathbb{R}^3 : 1 \leq x_i \leq 2, \quad i = 1, 2, 3\}$,
- $c^T = \begin{pmatrix} -4 & -1 & -6 \end{pmatrix}$,
- $A = \begin{pmatrix} 3 & 2 & 4 \end{pmatrix}$,
- $b = 17$.

U sljedećem koraku, skup P reprezentiramo u smislu (2.10). Skup P je ograničen i skup njegovih ekstremnih točaka je

$$Q = \{(1, 1, 1), (1, 1, 2), (1, 2, 1), (2, 1, 1), (1, 2, 2), (2, 1, 2), (2, 2, 1), (2, 2, 2)\}. \tag{2.19}$$

1. *iteracija:*
Za rješenje metodom generacije stupaca, u početku biramo manji inicijalizacijski skup stupaca na temelju kojeg ćemo generirati nove stupce, ukoliko bude potrebno.
Iz skupa Q izaberimo, naprimjer, dvije ekstremne točke za koje će sljedeća zadaća imati neprazan dopustiv skup. Neka su to $p_1 = (2, 2, 2)$, $p_2 = (1, 1, 2)$.
Rješavamo novu reduciranu zadaću u smislu reprezentacije (2.10):

$$\begin{cases} (-4 \cdot 2 - 2 - 6 \cdot 2)\lambda_1 + (-4 \cdot 1 - 1 - 6 \cdot 2)\lambda_2 \to \min \\ (3 \cdot 2 + 2 \cdot 2 + 4 \cdot 2)\lambda_1 + (3 \cdot 1 + 2 \cdot 1 + 4 \cdot 2)\lambda_2 = 17 \\ \lambda_1 + \lambda_2 = 1 \\ \lambda_1, \lambda_2 \geq 0, \end{cases} \tag{2.20}$$

odnosno:

$$\begin{cases} -22\lambda_1 - 17\lambda_2 \to \min \\ 18\lambda_1 + 13\lambda_2 = 17 \\ \lambda_1 + \lambda_2 = 1 \\ \lambda_1, \lambda_2 \geq 0. \end{cases} \tag{2.21}$$

- *Optimalna točka zadaće (2.21):* $\overline{\lambda}_1^{(1)} = 0.8$, $\overline{\lambda}_2^{(1)} = 0.2$.
- *Optimalna točka dualne zadaće zadaći (2.21):* $\overline{\pi}^{(1)} = -1$, $\overline{\alpha}^{(1)} = -4$.

U nastavku rješavamo problem određivanja cijena. Problem je definiran s

$$\min_{x \in P} (c^T - \overline{\pi}^{(1)T} A)x - \overline{\alpha}^{(1)}. \tag{2.22}$$

Kako je $\overline{\alpha}$ *konstanta, promatramo funkciju cilja potproblema danu s*

$$\overline{c}^{(1)} := c^T - \overline{\pi}^{(1)T} A = \begin{pmatrix} -4 & -1 & -6 \end{pmatrix} - (-1) \cdot \begin{pmatrix} 3 & 2 & 4 \end{pmatrix} = \begin{pmatrix} -1 & 1 & -2 \end{pmatrix}.$$

Rješavamo sljedeću zadaću:

$$\begin{cases} -x_1 + x_2 - 2x_3 \to \min \\ 1 \leq x_1 \leq 2 \\ 1 \leq x_2 \leq 2 \\ 1 \leq x_3 \leq 2. \end{cases} \tag{2.23}$$

Optimalna točka zadaće (2.23) je $x^{(1)*} = (2, 1, 2)$, *optimalna vrijednost funkcije cilja je* $\overline{c}^{(1)*} = -5$. *Vrijedi:*
$\overline{c}^{(1)*} - \overline{\alpha}^{(1)} = -5 - (-4) = -1 < 0 \implies p_3 := (2, 1, 2)$ *je nova ekstremna točka, odnosno generiramo novi stupac*

$$\begin{bmatrix} -4 \cdot 2 - 1 - 6 \cdot 2 \\ 3 \cdot 2 + 2 \cdot 1 + 4 \cdot 2 \\ 1 \end{bmatrix} = \begin{bmatrix} -21 \\ 16 \\ 1 \end{bmatrix}.$$

2. *iteracija:*
Nova reducirana zadaća izgleda ovako:

$$\begin{cases} -22\lambda_1 - 17\lambda_2 - 21\lambda_3 \to \min \\ 18\lambda_1 + 13\lambda_2 + 16\lambda_3 = 17 \\ \lambda_1 + \lambda_2 + \lambda_3 = 1 \\ \lambda_i \geq 0, \quad i = 1, 2, 3. \end{cases} \tag{2.24}$$

- *Optimalna točka zadaće (2.24):* $\overline{\lambda}_1^{(2)} = 0.5$, $\overline{\lambda}_2^{(2)} = 0$, $\overline{\lambda}_3^{(2)} = 0.5$.

- *Optimalna točka dualne zadaće zadaći (2.24):* $\overline{\pi}^{(2)} = -0.5$, $\overline{\alpha}^{(2)} = -13$.

U nastavku rješavamo problem određivanja cijena. Problem je definiran s

$$\min_{x \in P} (c^T - \overline{\pi}^{(2)T} A)x - \overline{\alpha}^{(2)}, \tag{2.25}$$

gdje je funkcija cilja potproblema dana s

$$\overline{c}^{(2)} := c^T - \overline{\pi}^{(2)} A = \begin{pmatrix} -4 & -1 & -6 \end{pmatrix} - (-0.5) \cdot \begin{pmatrix} 3 & 2 & 4 \end{pmatrix} = \begin{pmatrix} -2.5 & 0 & -4 \end{pmatrix}.$$

Rješavamo sljedeću zadaću:

$$\begin{cases} -2.5x_1 - 4x_3 \to \min \\ \qquad 1 \leq x_1 \leq 2 \\ \qquad 1 \leq x_2 \leq 2 \\ \qquad 1 \leq x_3 \leq 2. \end{cases} \tag{2.26}$$

Optimalna točka zadaće (2.26) je $x^{(2)*} = (2, 2, 2)$*; optimalna vrijednost funkcije cilja je* $\overline{c}^{(2)*} = -13$. *Vrijedi:*
$\overline{c}_s'^* - \overline{\alpha} = -13 - (-13) = 0 \implies$ *generacija stupaca staje, jer je postignuto optimalno rješenje.*

Koja je optimalna točka polazne zadaće (2.18)? Optimalna točka zadaće (2.18) je

$$\begin{aligned} x^* &= \overline{\lambda}_1^{(2)} p_1 + \overline{\lambda}_2^{(2)} p_2 + \overline{\lambda}_3^{(2)} p_3 \\ &= 0.5 \cdot (2, 2, 2) + 0 \cdot (1, 1, 2) + 0.5 \cdot (2, 1, 2) \\ &= (2, 1.5, 2). \end{aligned} \tag{2.27}$$

2.4 Dijagonalna blok-struktura u Dantzig-Wolfeovoj dekompoziciji

Na prvi pogled se čini da formulacija zadaće linearnog programiranja u obliku Dantzig-Wolfeove dekompozicije u kombinaciji s generacijom stupaca ne olakšava rješavanje zadaće. No, Dantzig-Wolfeova dekompozicija može biti primjenjiva na paralelno rješavanje manjih potproblema umjesto na rješavanje originalnog problema. Promotrimo u kojim situacijama se Dantzig-Wolfeova dekompozicija može odvijati paralelno.

Prisjetimo se, rješavamo sljedeću zadaću linearnog programiranja:

$$\begin{cases} c^T x \to \min \\ Ax \geq b \\ Dx \geq d \\ x \geq 0. \end{cases} \tag{2.28}$$

Promotrimo slučaj gdje matrica D ima blok-dijagonalnu strukturu:

$$D = \begin{pmatrix} D^1 & & & \\ & D^2 & & \\ & & \ddots & \\ & & & D^k \end{pmatrix}, \quad d = \begin{pmatrix} d^1 \\ d^2 \\ \vdots \\ d^k \end{pmatrix}. \tag{2.29}$$

Definirajmo skupove

$$P^i = \{x \in \mathbb{R}^+_{n_i} : D^i x \geq d^i\}, \quad i = 1, \dots, k, \tag{2.30}$$

gdje je $D^i \in \mathbb{R}^{k_i \times n_i}$.
Svaki od skupova (2.30) može, neovisno o drugim skupovima, biti reprezentiran u skladu s (2.10):

$$\begin{aligned} P^i = \{x \in \mathbb{R}^{n_i}_+ : x = \sum_{q \in I_{Q^i}} \lambda^i_q p^i_q + \sum_{r \in I_{R^i}} \mu^i_r w^i_r, \quad \sum_{q \in I_{Q^i}} \lambda^i_q = 1, \\ \lambda^i \in \mathbb{R}^{|I_{Q^i}|}_+, \quad \mu^i \in \mathbb{R}^{|I_{R^i}|}_+\}, \end{aligned} \tag{2.31}$$

gdje je $Q^i = \{p^i_q \mid q \in I_{Q^i}\}$ skup ekstremnih točaka skupa P^i te $R^i = \{w^i_r \mid r \in I_{R^i}\}$ skup ekstremnih zraka skupa P^i.

Sukladno prikazu (2.11), zadaću (2.28) reprezentiramo na sljedeći način:

$$\begin{cases} \sum_{i=1}^{k} \Big(\sum_{q \in I_{Q^i}} \lambda^i_q (c^T p^i_q) + \sum_{r \in I_{R^i}} \mu^i_r (c^T w^i_r) \Big) \to \min \\ \sum_{q \in I_{Q^i}} \lambda^i_q (A p^i_q) + \sum_{r \in I_{R^i}} \mu^i_r (A w^i_r) \geq b \\ \sum_{q \in I_{Q^i}} \lambda^i_q = 1, \quad i = 1, \dots, k \\ \lambda^i_q \geq 0, \quad q \in I_{Q^i}, \quad i = 1, \dots, k \\ \mu^i_r \geq 0 \quad r \in I_{R^i}, \quad i = 1, \dots, k. \end{cases} \tag{2.32}$$

Označimo s π dualnu varijablu koja odgovara prvom skupu uvjeta te s α^i dualnu varijablu koja odgovara i–tom uvjetu tipa jednakosti. Sukladno (2.16), i–ti potproblem koji određuje generaciju novih stupaca definiran je s

$$\overline{c}^{i*} = \min\left\{(c^{iT} - \overline{\pi}^T A^i)x^i - \overline{\alpha}^i \,:\, D^i x^i \geq d^i,\ x^i \geq 0\right\}, \quad i \in \{1, \ldots, k\}. \tag{2.33}$$

Algoritam generacije stupaca staje u trenutku kada postane $\overline{c}^{i*} \geq 0$ za svaki $i \in \{1, \ldots, k\}$. U suprotnom, ako barem jedan od $\overline{c}^{i*}$ ne zadovoljava uvjet nenegativnosti, postoji mogućnost uvođenja nove ekstremne točke ili ekstremne zrake iz skupova P^i u reduciranu zadaću.

Po teoremu jake dualnosti, vrijedi:

$$\overline{z} = \overline{\pi}^T b + \sum_{i=1}^{k} \overline{\alpha}^i. \tag{2.34}$$

Sada možemo izvesti zaključke o granicama optimalne vrijednosti funkcije cilja originalne zadaće:

$$\overline{z} + \sum_{i=1}^{k} \overline{c}^{i*} \leq z^* \leq \overline{z}, \tag{2.35}$$

gdje je $\overline{z}$ optimalna vrijednost funkcije cilja reduciranog problema, dok z^* predstavlja optimalnu vrijednost funkcije cilja originalnog problema.

2.5 Lagrangeova relaksacija

Problem rasporeda u zračnom prometu, kako je već utvrđeno, primjer je cjelobrojnog linearnog programiranja. Promotrimo zadaću s uvjetima tipa jednakosti:

$$\begin{cases} c^T x \to \min \\ Ax = b \\ \quad x \in X, \end{cases} \tag{2.36}$$

gdje je $X = P \cap \mathbb{Z}_+^n$ te $P = \{x \mid Dx \geq d\} \subseteq \mathbb{R}^n$ poliedarski skup. X je u ovom slučaju diskretan skup.

Čest pristup rješavanju problema (2.36) je Lagrangeovom relaksacijom. Lagrangeova relaksacija je metoda dekompozicije: ideja ovog pristupa je skup uvjeta „razbiti" na „jednostavnije" i „složenije", a potom „složenije" uvjete izbaciti iz skupa uvjeta i prebaciti u funkciju cilja. Zadržavanjem „jednostavnijih" uvjeta, rješavanje relaksacijskog problema postaje jednostavno. Katkad je optimalno rješenje relaksacijskog problema ujedno i optimalno rješenje originalnog problema.

Direktna primjena Lagrangeove relaksacije na problem (2.36) podrazumijeva relaksiranje uvjeta jednakosti $Ax = b$, odnosno dodavanje funkcije uvjeta u funkciju cilja s pridruženim vektorom **u** čije komponente nazivamo *Lagrangeovim multiplikatorima*. Zapišimo izrečeno u terminima linearnog programiranja:

$$\begin{cases} c^T x + u^T(Ax - b) \to \min \\ \qquad x \in X. \end{cases} \tag{2.37}$$

Problem (2.41) nazivamo *Lagrangeovom relaksacijom* problema (2.36). Funkciju cilja problema možemo zapisati kompaktnije:

$$L(u) = \min_{x \in X}\{c^T x + u^T(Ax - b)\}. \tag{2.38}$$

Funkciju L u varijabli u nazivamo *Lagrangeovom funkcijom*. Eliminiranje uvjeta jednakosti u zadaći (2.36) može, a i ne mora rezultirati optimalnom točkom koja nije dopustiva u (2.36). No, unatoč različitim scenarijima dopustivosti rješenja, motiv za korištenje Lagrangeove relaksacije je sljedeći: rješenje Lagrangeovog duala daje ocjenu odozdo za originalni optimizacijski problem. Iskažimo ocjenu odozdo za optimalno rješenje z^* problema (2.36).

Lema 2.5.1. *Neka je u Lagrangeov multiplikator, L Lagrangeova funkcija te z^* optimalno rješenje originalnog problema. Tada vrijedi:*

$$L(u) \leq z^*. \tag{2.39}$$

Dokaz. Neka je x rješenje problema (2.36). Tada vrijedi:

$$z^* = \min\{c^T x \mid Ax = b,\ x \in X\} = \min\{c^T x + u^T(Ax - b) \mid Ax = b,\ x \in X\}.$$

Micanje uvjeta $Ax = b$ ne može dovesti do povećanja funkcije cilja (eventualno je može smanjiti), pa će vrijediti:

$$L(u) = \min\{c^T x + u^T(Ax - b) \mid x \in X\} \leq z^*.$$

□

Traženje najbolje moguće ocjene iz prethodne leme rezultira rješavanjem *Lagrangeove dualne zadaće* $\mathcal{L} = \max_u L(u)$. Sada možemo postaviti još jedan niz nejednakosti:

$$L(u) \leq \mathcal{L} \leq z^* \leq c^T x. \tag{2.40}$$

Promotrimo zadaću linearnog programiranja s uvjetima nejednakosti:

$$\begin{cases} c^T x \to \min \\ Ax \geq b \\ \quad x \in X, \end{cases} \tag{2.41}$$

gdje je $X = P \cap \mathbb{Z}_+^n$ te $P = \{x \mid Dx \geq d\} \subseteq \mathbb{R}^n$ poliedarski skup.

Relaksiramo uvjet $Ax \geq b$ i Lagrangeovim multiplikatorima $u \geq 0$ reguliramo odstupanje od uvjeta. Lagrangeova funkcija je oblika

$$L(u) = \min_{x \in X} \{c^T x + u^T(b - Ax)\}. \tag{2.42}$$

Označimo sa $z^* = \min c^T x$ optimalno rješenje problema (2.41). Pronađimo donju ogradu na z^*.

$$L(u) \leq \min\{c^T x + u^T(b - Ax) \mid Ax \geq b, x \in X\} \leq z^* \tag{2.43}$$

Prva nejednakost vrijedi jer micanjem uvjeta $Ax \geq b$ iz zadaće neće doći do povećanja vrijednosti funkcije cilja. Druga nejednakost vrijedi zbog uvjeta komplementarnosti koje zadovoljava optimalna točka zadaće, odnosno

$$u^{*T}(b - Ax^*) = 0 \tag{2.44}$$

Najbolju ogradu na z^* dobit ćemo rješavanjem *Lagrangeovog dualnog problema*

$$\mathcal{L} = \max_{u \geq 0} L(u). \tag{2.45}$$

Pretpostavimo da nam je poznat vektor optimalnih multiplikatora u^* zadaće (2.45). Rješavanjem zadaće (2.38), osiguran je zahtjev $x \in X$. Uvjet komplementarosti osigurava $u^{*T}(b - Ax^*) = 0$, no uvjet $Ax^* \geq b$ mora biti provjeren kako bismo potvrdili optimalnost. Ako uvjet nejednakosti nije zadovoljen, tada x nije optimalna točka zadaće (2.41).

Lema 2.5.2. *Ako je za neki u x optimalna točka zadaće dobivene Lagrangeovom relaksacijom zadaće (2.41) i takva da vrijedi:*

- *x je dopustiva točka zadaće (2.41)*
- *x zadovoljava uvjet komplementarnosti $u^T(b - Ax) = 0$,*

tada je x optimalna točka originalne primarne zadaće (2.41).

Dokaz. Kako je x optimalna točka relaksacijskog problema zadaće (2.41) za dani u, vrijedi $L(u) = c^T x + u^T(b - Ax)$. Kako po pretpostavci vrijedi uvjet komplementarnosti, slijedi $L(u) = c^T x$. Iz dopustivosti rješenja x (x zadovoljava $Ax \geq b$) u zadaći (2.41) i prethodnih diskusija poznato nam je da vrijedi:

$$L(u) \leq z^* \leq c^T x = L(u). \tag{2.46}$$

Zaključujemo da je

$$L(u) = z^* = c^T x. \tag{2.47}$$

Također, iz nejednakosti $L(u) \leq \mathcal{L} \leq z^*$ i (2.47) proizlazi jednakost $L(u) = \mathcal{L}$.

Zaključujemo da je x optimalna točka zadaće (2.41) čija je optimalna vrijednost funkcije cilja jednaka z^*. Također, u je optimalna točka Lagrangeovog problema (2.45). □

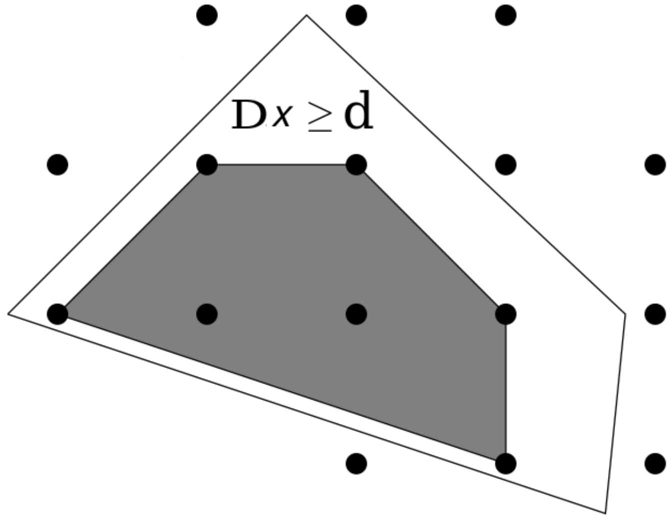
Česti pristupi rješavanju problema Lagrangeove relaksacije su metode subgradijenata te linearno programiranje. Promotrimo rješavanje problema linearnim programiranjem. Zamijenimo li skup X skupom conv$X$, optimalna vrijednost funkcije cilja zadaće (2.41) neće se promijeniti. Tome svjedoči i sljedeća lema.

Lema 2.5.3. *Svaka ekstremna točka poliedarskog skupa* conv$X$ *leži u* X. *Optimiramo li linearnu funkciju cilja na* conv$X$, *neka točka iz* X *bit će optimalna.*

Zadaću

$$\begin{cases} c^T x \to \min \\ Ax \geq b \\ \quad x \in \operatorname{conv} X \end{cases} \tag{2.48}$$

nazivamo konveksifikacijom zadaće (2.41). Zapis zadaća u smislu skupa conv$X$ omogućuje formulaciju zadaća (2.38) i (2.45) preko ekstremnih točaka i ekstremnih zraka, baš kao i ideja prethodno obrađene Dantzig-Wolfeove dekompozicije.



Slika 2.3: Promotrimo li skup X iz (2.41), conv$X$ = conv($\{x \mid Dx \geq d\} \cap \mathbb{Z}^n$). Područje obojeno sivom bojom predstavlja skup conv$X$.

Taj zapis zadaća preko skupa conv$X$ omogućuje rješavanje Lagrangeovog duala linearnim programiranjem. Za dani vektor multiplikatora $\overline{\pi}$ možemo zapisati:

$$\begin{aligned}
L(\overline{u}) &= \min_{x \in \mathrm{conv}X} c^T x + \overline{\pi}^T (b - Ax) \\
&= \min_{x \in \mathrm{conv}X} c^T x + \overline{\pi}^T b - \overline{\pi}^T Ax \\
&= \overline{\pi}^T b + \min_{x \in \mathrm{conv}X} (c^T - \overline{\pi}^T A)x \\
&= (\overline{\pi}^T b + \overline{\alpha}) + \min_{x \in \mathrm{convX}} (c^T - \overline{\pi}^T A)x - \overline{\alpha} \\
&= \overline{z} + \overline{c}^*
\end{aligned}$$

Dobili smo donju Lagrangeovu ogradu na optimalnu vrijednost izvorne zadaće jednaku onoj dobivenoj rješavanjem reduciranog problema u smislu Dantzig-Wolfeove dekompozicije.

U nastavku ćemo pokazati poveznicu prethodno opisane metode s Dantzig-Wolfeovom dekompozicijom te opisati postupak konveksifikacije.

2.6 Konveksifikacija

Prethodna zapažanja upućuju na jaku povezanost između Dantzig-Wolfeove dekompozicije i Lagrangeove relaksacije.

Ukoliko se dogodi da je skup X iz zadaće (2.41) prazan, tada je vrijednost $\mathcal{L} = \infty$. U suprotnom, prikažimo skup conv$X$ preko skupa ekstremnih točaka Q i ekstremnih zraka R.

Neka je u vektor Lagrangeovih multiplikatora. Tada vrijedi:

$$L(u) = \begin{cases} -\infty, & \text{ako je } (c^T - u^T A)w_r < 0 \text{ za neki } r \in I_R \\ c^T p_q + u^T(b - Ap_q), \text{ za neki } q \in I_Q, & \text{inače,} \end{cases} \tag{2.49}$$

gdje su I_Q i I_R skupovi indeksa ekstremnih točaka i ekstremnih zraka, redom. Obzirom da nam je od interesa da optimalna vrijednost funkcije cilja z^* bude konačna, nastojimo izbjeći prvi slučaj gdje je $L(u) = -\infty$.

Zapišimo (2.45) na sljedeći način:

$$\begin{cases} \mathcal{L} = \max\limits_{u \geq 0} \min\limits_{q \in I_Q} \{c^T p_q + u^T(b - Ap_q)\} \\ \qquad (c^T - u^T A)w_r \geq 0, \quad r \in I_R. \end{cases} \tag{2.50}$$

No, zadaću (2.45) možemo zapisati i u obliku zadaće linearnog programiranja s velikim brojem uvjeta. Kako nastojimo minimizirati izraz

$$c^T p_q + u^T(b - Ap_q) = c^T p_q - u^T(Ap_q - b), \quad q \in I_Q$$

iz prethodne zadaće, postupak je ekvivalentan traženju maksimalne vrijednosti v takve da vrijedi

$$u^T(Ap_q - b) + v \leq c^T p_q.$$

Kako nastojimo izbjeći granicu jednaku $-\infty$ iz izraza (2.49), zahtjevamo

$$c^T w_r - u^T A w_r \geq 0.$$

Poslije ovih razmatranja, zadaću (2.50) zapisujemo na sljedeći način:

$$\begin{cases} v \to \max \\ u^T(Ap_q - b) + v \leq c^T p_q, \quad q \in I_Q \\ u^T A w_r \leq c^T w_r, \quad r \in I_R \\ u \geq 0. \end{cases} \tag{2.51}$$

Prethodna zadaća linearnog programiranja obično ima veliki broj uvjeta, obzirom na prethodna razmatranja s početka poglavlja o mogućem velikom broju ekstremnih točaka i ekstremnih zraka skupa conv$X$ u brojnim primjenama.

Dualna zadaća zadaće (2.51) je oblika

$$\begin{cases} \sum_{q\in I_Q} c^T p_q\lambda_q + \sum_{r\in I_R} c^T w_r\mu_r \to \min \\ \sum_{q\in I_Q} Ap_q\lambda_q - \sum_{q\in I_Q} b\lambda_q + \sum_{r\in I_R} Aw_r\mu_r \geq 0 \\ \sum_{q\in I_Q} \lambda_q = 1 \\ \lambda \geq 0 \\ \mu \geq 0. \end{cases} \tag{2.52}$$

Obzirom na uvjet konveksnosti u zadaći, prethodnu zadaću (2.52) možemo zapisati u obliku

$$\begin{cases} \sum_{q\in I_Q} c^T p_q\lambda_q + \sum_{r\in I_R} c^T w_r\mu_r \to \min \\ \sum_{q\in I_Q} Ap_q\lambda_q + \sum_{r\in I_R} Aw_r\mu_r \geq b \\ \sum_{q\in I_Q} \lambda_q = 1 \\ \lambda \geq 0 \\ \mu \geq 0. \end{cases} \tag{2.53}$$

Zadaće linearnog programiranja koje smo prethodno izveli koriste nam u rješavanju problema (2.45): rješavanje zadaće (2.45) ekvivalentno je rješavanju zadaća (2.51) i (2.53). Rješavanje zadaće (2.53) rezultira multiplikatorima, dok rješavanjem zadaće (2.53) dobijemo, u slučaju cjelobrojnosti točke, dopustivu točku x izvorne zadaće u terminima ekstremnih točaka i ekstremnih zraka.

Zamijenimo li skup X skupom conv$X$ u zadaći (2.41) te u Dantzig-Wolfeovoj dekompoziciji primijenimo sljedeći niz jednakosti

$$\begin{aligned} c_q &= c^T p_q, \quad q \in I_Q \\ c_r &= c^T w_r, \quad r \in I_R \\ a_q &= c^T p_q, \quad q \in I_Q \\ a_r &= c^T w_r, \quad r \in I_R, \end{aligned}$$

dobijemo zadaću:

$$\begin{cases} z^* := \sum_{q\in I_Q} c_q\lambda_q + \sum_{r\in I_R} c_r\mu_r \to \min \\ \sum_{q\in I_Q} a_q\lambda_q + \sum_{r\in I_R} a_r\mu_r \geq b \\ \sum_{q\in I_Q} \lambda_q = 1 \\ \lambda \geq 0 \\ \mu \geq 0 \\ \sum_{q\in I_Q} x_q\lambda_q + \sum_{r\in I_R} x_r\mu_r = x \\ x \in \mathbb{Z}^n_+. \end{cases} \tag{2.54}$$

Stupci ovog problema definirani su preko ekstremnih točaka i ekstremnih zraka skupa conv$X$. Relaksiramo li uvjet cjelobrojnosti varijable x, nema potrebe za uspostavljanjem veze varijable x s λ i μ te možemo izostaviti uvjet $\sum_{q\in I_Q} x_q\lambda_q + \sum_{r\in I_R} x_r\mu_r = x$. Izostavljanjem ovog uvjeta, zadaća (2.54) postaje ekvivalentna zadaći (2.53). Rješavamo ju prethodno opisanom metodom generacije stupaca. No, priroda problema zahtijeva cjelobrojnu optimalnu točku x^*. Kako postići cjelobrojnost varijable x?

2.7 Primjena na problemima cjelobrojnog programiranja

U ovom odjeljku, ideju Dantzig-Wolfeove dekompozicije i generacije stupaca proširit ćemo na probleme cjelobrojnog programiranja, počevši s ograničenim zadaćama cjelobrojnog linearnog programiranja. Odjeljak ćemo zaključiti posebnim slučajem binarnog linearnog programiranja.

U nastavku promatrajmo zadaću linearnog programiranja (2.41). Pretpostavimo da je skup X iz zadaće ograničen. Postoji više različitih pristupa dekompozicije linearnog programa, no u kontekstu Dantzig-Wolfeove dekompozicije, fokusiramo se na **diskretizaciju**, odgovarajući postupak dekompozicije cjelobrojnih linearnih programa. Zapažanja započinjemo sljedećim teoremom.

Teorem 2.7.1 (Teorem reprezentacije cjelobrojnih točaka poliedarskog skupa)**.** *Neka je $P := \{x \in \mathbb{R}^n \mid Dx \geq d,\ x \geq 0\} \neq \emptyset$ te $X = P \cap \mathbb{Z}^n_+$. Tada postoji konačan skup cjelobrojnih točaka $\{p_q\}_{q\in I_Q} \subseteq X$ te konačan skup cjelobrojnih zraka $\{w_r\}_{r\in I_R}$ skupa P tako da vrijedi*

$$X = \left\{x \in \mathbb{R}^n_+ \;\middle|\; x = \sum_{q\in I_Q} p_q\lambda_q + \sum_{r\in I_R} w_r\mu_r, \sum_{q\in I_Q} \lambda_q = 1,\ \lambda \in \mathbb{Z}^{|I_Q|}_+, \mu \in \mathbb{Z}^{|I_R|}_+\right\}. \tag{2.55}$$

Budući da smo pretpostavili da je X ograničen, druga suma u reprezentaciji (2.55) će nestati. Trivijalno, svaka točka $x \in X$ može biti reprezentirana kao jedna cjelobrojna točka iz X. Ovakva reprezentacija cjelobrojnog skupa bit će korisna pri relaksacija uvjeta cjelobrojnosti na λ_q varijablama. Iz uvjeta $\sum_{q \in I_Q} \lambda_q = 1$ te cjelobrojnosti i nenegativnosti varijabli λ_q, $q \in I_Q$, λ_q poprima vrijednosti iz skupa $\{0, 1\}$.

Prikažimo x u skladu s prikazom (2.55) te uvrstimo u zadaću (2.41):

$$\begin{cases} \sum_{q \in I_Q} c^T p_q \lambda_q \to \min \\ \sum_{q \in I_Q} A p_q \lambda_q \geq b \\ \sum_{q \in I_Q} \lambda_q = 1 \\ \lambda_q \in \mathbb{Z}_+, \quad q \in I_Q. \end{cases} \tag{2.56}$$

Relaksiramo li zahtjev cjelobrojnosti varijable λ, prethodna zadaća nalikuje zadaći (2.11) (zahtjevamo ograničenost skupa P u (2.11)). No, postoji bitna razlika između ovih dviju zadaća. Stupci zadaće (2.56) prikazani su preko cjelobrojnih točaka skupa X koje mogu biti i unutarnje i ekstremne točke poliedra P. S druge strane, stupci zadaće (2.11) odgovaraju ekstremnim točkama skupa P.

Kako je prethodno opisano u odjeljku generacije stupaca, rješavanje zadaće započinje podskupom $Q' \subseteq Q$ svih stupaca. Uvođenje novog stupca utvrđuje se rješavanjem problema određivanja cijena

$$\overline{c}^* := \min_{x \in X} \{(c^T - \overline{\pi}^T A)x - \overline{\alpha}\}, \tag{2.57}$$

gdje je $(\overline{\pi}, \overline{\alpha})$ optimalna točka dualne zadaće reduciranoj zadaći. Prethodni problem određivanja cijena je cjelobrojan.

Nameće se pitanje kako nam gornja zapažanja mogu pomoći u rješavanju zadaće cjelobrojnog linearnog programiranja (2.41). Relaksacija zadaće cjelobrojnog linearnog programiranja je idealna postavka za algoritam grananja i ograđivanja (*engl. branch and bound*). Prirodna veza diskretizacije i generacije stupaca dolazi do izražaja pri razmatranju linearne relaksacije cjelobrojne reducirane zadaće u sklopu metode grananja i ograđivanja. Takvu metodu nazivamo cjelobrojna generacija stupaca (poznata i kao *branch and price*).

Posebna vrsta cjelobrojnog linearnog programiranja je binarno linearno programiranje. Pretpostavimo da je uvjet cjelobrojnosti u zadaći (2.41) zamijenjen uvjetom $x \in \{0, 1\}^n$. Označimo s $P := \{x \in \mathbb{R}^n \mid Dx \geq d\}$ te $X = P \cap \{0, 1\}^n$. Skup X je ograničen, no još i više – svaka točka skupa X je ekstremna točka konveksne ljuske skupa X, u oznaci conv$X$. Kompaktnije rečeno, svaka ekstremna točka skupa conv$X$ je cjelobrojna. Označimo s $X^{(LR)} := P \cap [0, 1]^n$ linearnu relaksaciju skupa X. Bilo koja cjelobrojna točka skupa $X^{(LR)}$ je ekstremna točka skupa conv$X$.

Promotrimo diskretizaciju problema binarnog linearnog programiranja te potproblem generacije novog stupca. Potproblem generacije novog stupca je zadaća cjelobrojnog programiranja. Relaksiranje uvjeta cjelobrojnosti postiže se zamjenom skupa X skupom $X^{(LR)}$. Idealan slučaj bi pretpostavljao jednakost skupova conv$X$ i $X^{(LR)}$, što je u praksi vrlo rijedak slučaj. Stoga dodajemo neke rezove (nova ograničenja, uvjete) $\tilde{D}x \geq \tilde{d}$ skupu $X^{(LR)}$ takve da skup $X^{(LR)} \cap \{x \in \mathbb{R}^n \mid \tilde{D}x \geq \tilde{d}\}$ sadrži conv$X$. Cilj je pronaći optimalnu točku relaksirane zadaće (jednostavno rješavanje simpleks metodom) koja će također biti optimalna i u zadaći binarnog potproblema. Jedna od metoda za pronalazak rezova je, već spomenuti, algoritam grananja i ograđivanja.

Poglavlje 3

Drugačija formulacija problema rasporeda

U uvodu je spomenuta „tradicionalna” formulacija problema rotacija posade:

$$\begin{cases} \sum\limits_{p\in P} c_p y_p \to \min \\ \quad \sum\limits_{p:i\in p} y_p = 1, \quad i \in F \\ \qquad y_p \in \{0,1\}, \quad p \in P, \end{cases} \tag{3.1}$$

gdje je $y_p = 1$ ako je rotacija p sastavni dio rješenja, a 0 u suprotnom; F predstavlja skup letova, a P skup rotacija. Formulaciju problema tumačimo na sljedeći način:

- p–ti stupac u i-tom retku ima 1 u slučaju da i–ti let pripada rotaciji p
- c_p predstavlja trošak rotacije p.

Korištenje ove formulacije zahtjeva popisane ili dinamički generirane rotacije. Popisivanje rotacija može biti vrlo zahtjevan posao obzirom na veliki broj pravila koja moraju biti poštivana pri dodjeli letova, a također i zahvaljujući mogućem velikom broju rotacija. Taj proces obično prvo uključuje popisivanje svih dužnosti u rasporedu, a potom i popisivanje svih mogućih načina kombiniranja postojećih dužnosti kako bi formirali rotacije.

Uvidjevši složenost prethodno opisanog procesa, predstavljamo drugačiju formulaciju problema.

3.1 Formulacija zasnovana na dužnostima

Problem rasporeda posade u ovom pristupu dijelimo u dvije faze. Prva faza uključuje odabir dužnosti koje particioniraju letove, dok druga faza podrazumijeva odabir rotacija koje particioniraju prethodno odabrane dužnosti. Ovakav pristup zahtjeva drugačiju formulaciju problema koja u konačnici bira „dobre” rotacije tako što prvo identificira „dobre” skupove dužnosti od kojih se grade rotacije.

Opisana formulacija je dana zadaćom linearnog programiranja:

$$\begin{cases} \sum_{d \in D} b_d x_d + \sum_{p \in P} \hat{c}_p z_p \to \min \\ \sum_{d : i \in d} x_d = 1, \quad i \in F \\ -x_d + \sum_{p : d \in p} z_p = 0, \quad d \in D \\ x_d \in \{0, 1\}, \quad d \in D \\ z_p \in \{0, 1\}, \quad p \in P, \end{cases} \tag{3.2}$$

gdje je:

- $$x_d = \begin{cases} 1, & \text{ako je odabrana dužnost } d \\ 0, & \text{inače} \end{cases}$$

- $$z_p = \begin{cases} 1, & \text{ako je odabrana rotacija } p \\ 0, & \text{inače} \end{cases}$$

- $F \ldots$ skup letova
- $D \ldots$ skup dužnosti
- $P \ldots$ skup rotacija
- $b_d \ldots$ cijena dužnosti d
- $\hat{c}_p \ldots$ dodatni troškovi rotacije p – razlika između stvarnih troškova rotacije p i zbroja troškova dužnosti koje čije rotaciju p (troškovi kao što su plaćeni odmor između dužnosti); vrijednost $\hat{c}_p$ za bilo koju rotaciju p bit će nenegativna obzirom da je definirana kao maksimum triju vrijednosti, od kojih je jedna i zbroj troškova dužnosti te rotacije.

Objasnimo na što se svaki od uvjeta odnosi. Prvi skup uvjeta ispunjava zahtjev da je svaki od letova „pokriven" točno jednom dužnošću. Drugi skup uvjeta osigurava odabir rotacije tako da ona sadrži dužnost koja obuhvaća neki let.

Formulacija (3.2) ima više redaka i stupaca od formulacije (3.1). U zadaći (3.2) za svaku od dužnosti imamo po jedan stupac, za razliku od zadaće (3.1), no to ne bi trebalo uvelike utjecati na broj stupaca zadaće obzirom da je broj dužnosti manjeg reda veličine u odnosu na broj rotacija.

Teorem 3.1.1. *Linearne relaksacije formulacija (3.1) i (3.2) međusobno su ekvivalentne.*

Dokaz. Varijable z_p iz (3.2) i y_p iz (3.1) reprezentiraju iste rotacije, no u različitom smislu – z_p reprezentira rotaciju u smislu dužnosti koje sadrži, dok y_p predstavlja rotaciju preko letova koje sadrži. Želimo li od optimalne točke jedne zadaće doći do optimalne točke druge, stavimo $z_p^* = y_p^*$, $p \in P$ te $x_d^* = \sum_{p:d \in P} z_p^*$. Odabirom takvih y^*, z^* osiguravamo zadovoljenost drugog skupa uvjeta zadaće (3.2). Ekvivalencija particioniranja letova i dužnosti proizlazi iz sljedećeg niza jednakosti:

$$\sum_{d:i \in d} x_d^* = \sum_{d:i \in d} \sum_{p:d \in p} z_p^* = \sum_{p:i \in P} y_p^* = 1. \tag{3.3}$$

Primijetimo da po definiciji varijabli b_d, $\hat{c}_p$ i c_p vrijedi jednakost

$$\sum_{d \in p} b_d + \hat{c}_p = c_p.$$

Pokažimo jednakost optimalnih vrijednosti funkcija cilja zadaća (3.1) i (3.2).

$$\begin{aligned}
\sum_{p \in P} c_p y_p^* &= \sum_{p \in P} (\hat{c}_p + \sum_{d \in p} b_d) z_p^* \\
&= \sum_{p \in P} \hat{c}_p z_p^* + \sum_{d \in D} \sum_{p:d \in p} b_d z_p^* \\
&= \sum_{p \in P} \hat{c}_p z_p^* + \sum_{d \in D} b_d x_d^*.
\end{aligned}$$

□

Primjer 3.1.2. *Promotrimo primjer iz prvog poglavlja. Za dani primjer napišimo zadaću linearnog programiranja u skladu s formulacijom (3.2). Za početak, odaberimo dužnosti tako da particioniraju letove. Neka su početne dužnosti dane na sljedeći način:*

$$\begin{array}{llll} D_1 = \{1\}, & D_2 = \{4\}, & D_3 = \{5\}, & D_4 = \{6\}, \\ D_5 = \{7\}, & D_6 = \{1,2\}, & D_7 = \{1,7,3\}, & D_8 = \{2,3\}. \end{array}$$

Neka su cijene dužnosti $b_1 = b_2 = b_3 = b_4 = b_5 = 1$, $b_6 = b_8 = 2$, b_7=3. *Neka su uparivanja dana s* $P_1 = \{D_6, D_2\}$, $P_2 = \{D_7, D_3\}$, $P_3 = \{D_4, D_8, D_3\}$, $P_4 = \{D_4, D_5, D_2\}$, $P_5 = \{D_1, D_5, D_2\}$. *Cijene* $\hat{c}_p$ *imaju sljedeće vrijednosti:* $\hat{c}_1 = \hat{c}_4 = 1$, $\hat{c}_2 = \hat{c}_3 = 0$ *te* $\hat{c}_5 = 2$.

Zadaća linearnog programiranja u skladu s formulacijom (3.2) je oblika

$$\begin{array}{rrrrrrrrrrrrrl}
x_1 & +\,x_2 & +\,x_3 & +\,x_4 & +\,x_5 & +\,2x_6 & +\,3x_7 & +\,2x_8 & +\,z_1 & & & +\,z_4 & +\,2z_5 & \to \min \\
x_1 & & & & & +\,x_6 & +\,x_7 & & & & & & & = 1 \quad (\textit{let } 1) \\
 & & & & & x_6 & & +\,x_8 & & & & & & = 1 \quad (\textit{let } 2) \\
 & & & & & & x_7 & +\,x_8 & & & & & & = 1 \quad (\textit{let } 3) \\
 & x_2 & & & & & & & & & & & & = 1 \quad (\textit{let } 4) \\
 & & x_3 & & & & & & & & & & & = 1 \quad (\textit{let } 5) \\
 & & & x_4 & & & & & & & & & & = 1 \quad (\textit{let } 6) \\
 & & & & x_5 & & +\,x_7 & & & & & & & = 1 \quad (\textit{let } 7) \\
-x_1 & & & & & & & & & & & & +\,z_5 & = 0 \quad (\textit{dužnost } 1) \\
 & -\,x_2 & & & & & & & +\,z_1 & & & +\,z_4 & +\,z_5 & = 0 \quad (\textit{dužnost } 2) \\
 & & -\,x_3 & & & & & & & +\,z_2 & +\,z_3 & & & = 0 \quad (\textit{dužnost } 3) \\
 & & & -\,x_4 & & & & & & & +\,z_3 & +\,z_4 & & = 0 \quad (\textit{dužnost } 4) \\
 & & & & -\,x_5 & & & & & & & +\,z_4 & +\,z_5 & = 0 \quad (\textit{dužnost } 5) \\
 & & & & & -\,x_6 & & & +\,z_1 & & & & & = 0 \quad (\textit{dužnost } 6) \\
 & & & & & & -\,x_7 & & & +\,z_2 & & & & = 0 \quad (\textit{dužnost } 7) \\
 & & & & & & & -\,x_8 & & & +\,z_3 & & & = 0 \quad (\textit{dužnost } 8) \\
x_1, & x_2, & x_3, & x_4, & x_5, & x_6, & x_7, & x_8, & z_1, & z_2, & z_3, & z_4, & z_5 & \in \{0,1\}
\end{array}$$

3.2 Dekompozicija problema

Primjenimo li Dantzig-Wolfeovu dekompoziciju na (3.2) te izvršimo li particioniranje uvjeta obzirom na letove

$$\sum_{d:i\in d} x_d = 1, \quad i \in F,$$

dobijemo sljedeći polazni problem:

$$\begin{cases} \sum_{S\in\mathbf{S}} C_S w_S + \sum_{p\in P} \hat{c}_p z_p \to \min \\ \sum_{S\in\mathbf{S}} w_S = 1 \\ -\sum_{S:d\in S} w_S + \sum_{p:d\in p} z_p = 0, \quad d\in D \\ w_S \in \{0,1\}, \quad S\in\mathbf{S} \\ z_p \in \{0,1\}, \quad p\in P, \end{cases} \tag{3.4}$$

gdje S označava skup dužnosti koje particioniraju letove. Stupce pridružene varijablama w_s nazivamo skupovima dužnosti: $w_s = 1$ ako skup S pokriva letove, a inače je 0. Sa $\mathbf{S}$ označavamo skup koji sadrži sve skupove S. Vrijednost C_S jednaka je $\sum_{d\in S} b_d$. Prvi uvjet u zadaći (3.4) govori da je odabran točno jedan skup dužnosti koji particionira letove. Drugi skup uvjeta zahtjeva odabir rotacija koje sadrže dužnosti koje pokrivaju letove. Rješavanju zadaće pristupamo razmatranjem podskupa stupaca (umjesto svih njih), čime formiramo reducirani problem.

Primjer 3.2.1. *Novu formulaciju zadaće linearnog programiranja ponovno demonstrirajmo na primjeru iz prvog poglavlja.*

Neka su dužnosti i particioniranja definirana kao u Primjeru 2. *Primjećujemo da je jedini skup dužnosti koji particionira letove upravo skup* $\{D_1, D_2, D_3, D_4, D_5, D_8\}$*, pa je skup* $\mathbf{S}$ *iz gornje diskusije jednočlan. Prema (3.4), formulacija je oblika:*

$$\begin{array}{llllllll}
7w_1 & +z_1 & & & +z_4 & +2z_5 & \to \min & \\
w_1 & & & & & & = 1 & \\
-w_1 & & & & & +z_5 & = 0 & (\textit{dužnost } 1) \\
-w_1 & +z_1 & & & +z_4 & +z_5 & = 0 & (\textit{dužnost } 2) \\
-w_1 & & +z_2 & +z_3 & & & = 0 & (\textit{dužnost } 3) \\
-w_1 & & & +z_3 & +z_4 & & = 0 & (\textit{dužnost } 4) \\
-w_1 & & & & +z_4 & +z_5 & = 0 & (\textit{dužnost } 5) \\
 & z_1 & & & & & = 0 & (\textit{dužnost } 6) \\
 & & z_2 & & & & = 0 & (\textit{dužnost } 7) \\
-w_1 & & & +z_3 & & & = 0 & (\textit{dužnost } 8) \\
w_1, & z_1, & z_2, & z_3, & z_4, & z_5 & \in \{0,1\} &
\end{array}$$

Općenito, kada govorimo o linearnoj relaksaciji neke zadaće cjelobrojnog programiranja, tada je optimalna vrijednost funkcije cilja relaksirane zadaće barem jednako „dobra" kao i optimalna vrijednost funkcije cilja polazne zadaće – optimalna točka izvorne zadaće je dopustiva točka linearne relaksacije. Obzirom da rješavamo problem minimizacije, optimalna vrijednost funkcije cilja relaksirane zadaće bit će manja ili jednaka optimalnoj vrijednosti funkcije cilja polazne zadaće. Primarna motivacija za korištenjem formulacije (3.4) u odnosu na (3.2) i (3.1) je postizanje bolje ograde linearnom relaksacijom zadaće, te ukoliko je moguće, veća vjerojatnost za postizanjem cjelobrojne optimalne točke zadaće linearne relaksacije.

Linearna relaksacija zadaće (3.4) je

$$\begin{cases} \sum_{S\in\mathbf{S}} C_S w_S + \sum_{p\in P} \hat{c}_p z_p \to \min \\ \sum_{S\in\mathbf{S}} w_S = 1 \\ -\sum_{S:d\in S} w_S + \sum_{p:d\in p} z_p = 0, \quad d\in D \\ w_S \geq 0, \quad S\in\mathbf{S} \\ z_p \geq 0, \quad p\in P. \end{cases} \tag{3.5}$$

Pokažimo da je ograda na optimalnu vrijednost polazne zadaće dobivena rješavanjem zadaće (3.5) veća ili jednaka ogradi dobivenoj rješavanjem linearne relaksacije zadaće (3.2).

Teorem 3.2.2. *Optimalna vrijednost funkcije cilja zadaće (3.5) je veća ili jednaka optimalnoj vrijednosti funkcije cilja linearne relaksacije zadaće (3.2).*

Dokaz. Neka je $(\hat{w}, \hat{z})$ optimalna točka linearne relaksacije zadaće (3.4). Konstruirajmo optimalnu točku (x^*, z^*) linearne relaksacije zadaće (3.2) tako što stavimo $x_d^* = \sum_{S:d\in S} \hat{w}_S$ te $z_p^* = \hat{z}_p$. Iz prethodno definiranog je jasno da je drugi skup uvjeta zadaće (3.2) ispunjen:

$$-x_d^* + \sum_{p:d\in p} z_p^* = -\sum_{S:d\in S} \hat{w}_S + \sum_{p:d\in p} \hat{z}_p = 0, \quad d\in D. \tag{3.6}$$

Posljednja jednakost slijedi iz optimalnosti točke $(\hat{w}, \hat{z})$ u zadaći (3.5). Ispitajmo ispunjenost prvog skupa uvjeta linearne relaksacije zadaće (3.1):

$$\sum_{d:i\in d} x_d^* = \sum_{d:i\in d} \sum_{S:d\in S} \hat{w}_S = \sum_{S\in\mathbf{S}} \hat{w}_S = 1. \tag{3.7}$$

Druga jednakost u prethodnom nizu slijedi iz činjenice da svaki skup dužnosti particionira letove (svaki let je sadržan u točno jednom skupu dužnosti) . Prethodnom diskusijom dobili smo da je (x^*, z^*) dopustiva točka linearne relaksacije zadaće (3.2). Kako je

$$\sum_{p\in P} \hat{c}_p \hat{z}_p = \sum_{p\in P} \hat{c}_p z_p^* \tag{3.8}$$

te

$$\sum_{S\in\mathbf{S}} C_S \hat{w}_S = \sum_{S\in\mathbf{S}} \sum_{d\in S} b_d \hat{w}_S = \sum_{d\in D} \sum_{S:d\in S} b_d \hat{w}_S = \sum_{d\in D} b_d x_d^*, \tag{3.9}$$

optimalne vrijednosti funkcija cilja linearnih relaksacija zadaća (3.2) i (3.4) su jednake.

Promotrimo sada suprotan proces konstrukcije rješenja. Neka je (x^*, z^*) optimalna točka linearne relaksacije zadaće (3.2) takva da je $x^* \in P$,

$$P := \operatorname{conv}\Big\{x \;\Big|\; \sum_{d:i\in d} x_d = 1,\ i \in F,\ x_d \in \{0,1\},\ d \in D\Big\}. \tag{3.10}$$

Svaka cjelobrojna točka skupa P zapravo reprezentira jedan skup dužnosti S (npr., vektor $(1,1,0,0,0,0,0,0)$ definira skup dužnosti $\{D_1, D_2\}$). Kako, je $x^* \in P$, x^* možemo prikazati kao konveksnu kombinaciju vektora „reprezentanata" skupova dužnosti. Dopustivu točku linearne relaksacije zadaće (3.4) konstruirajmo na sljedeći način: neka $\hat{w}_S$ predstavlja težinu vektora koji reprezentira skup dužnosti S u konveksnoj kombinaciji točke x^* te neka je $\hat{z}_p = z_p^*$. Kako je x^* konveksna kombinacija cjelobrojnih točaka iz P, vidimo da vrijedi $x_d^* = \sum_{S:d\in S} \hat{w}_S$ za sve $d \in D$ (za svaki $S \in \mathbf{S}$, $\hat{w}_S$ se množi nulom ili jedinicom, ovisno o koordinati vektora koji reprezentira S). Ponovimo li (3.8) i (3.9) u suprotnom smjeru, dobit ćemo istu optimalnu vrijednost funkcije cilja linearnih relaksacija zadaća (3.2) i (3.4). Promotrimo što se dogodi kada $x^* \notin P$. Tada ne postoje težine w_S skupova dužnosti takve da $x_d^* = \sum_{S:d\in S} \hat{w}_S$ za sve $d \in D$. Stoga, ne postoji optimalna točka $(\hat{w}, \hat{z})$ linearne relaksacije zadaće (3.4) uz uvjet $\hat{z}_p = z_p^*$, $\forall p \in P$. U prvom dijelu dokaza pokazano je da se za svaku dopustivu točku zadaće (3.5) može konstruirati neka dopustiva točka linearne relaksacije zadaće (3.2) tako da se postignute vrijednosti funkcija cilja podudaraju, što nije slučaj i u suprotnom smjeru. To znači da je optimalna vrijednost funkcije cilja linearne relaksacije zadaće (3.2) manja ili jednaka od optimalne vrijednosti funkcije cilja zadaće (3.5). □

Primjer 3.2.3. *Ovaj primjer nastavak je primjera iz prvog poglavlja te prethodnog primjera, gdje smo iskoristili različite formulacije kako bi definirali zadaće cjelobrojnog linearnog programiranja koje opisuju problem.*

Promotrimo optimalne točke linearnih relaksacija prethodno spomenutih zadaća cjelobrojnog programiranja.

Optimalna točka linearne relaksacije zadaće koja opisuje Primjer 1 *u skladu s formulacijom (3.2) je* $x_1 = 0$, $x_2 = x_3 = x_4 = 1$, $x_5 = x_6 = x_7 = x_8 = \frac{1}{2}$, $z_1 = z_2 = z_3 = z_4 = \frac{1}{2}$, $z_5 = 0$. *Optimalna vrijednost funkcije cilja zadaće je* 8.

Promotrimo li sada linearnu relaksaciju zadaće koja opisuje Primjer 1 *u skladu s formulacijom (3.4), optimalna točka te zadaće je* $w_1 = 1$*,* $z_1 = z_2 = z_4 = 0$*,* $z_3 = z_5 = 1$ *s optimalnom vrijednošću funkcije cilja jednakoj* 9*.*

Linearna relaksacija zadaće oblika (3.4) daje bolju ogradu na optimalnu vrijednost funkcije cilja izvorne zadaće – točka $x = (0, 1, 1, 1, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}) \notin P$*, gdje je* P *oblika (3.10).*

Jedina cjelobrojna točka koja ispunjava uvjete

$$\begin{array}{lr}
x_1 + x_6 + x_7 = 1 & (\textit{let } 1)\\
x_6 + x_8 = 1 & (\textit{let } 2)\\
x_7 + x_8 = 1 & (\textit{let } 3)\\
x_2 = 1 & (\textit{let } 4)\\
x_3 = 1 & (\textit{let } 5)\\
x_4 = 1 & (\textit{let } 6)\\
x_5 + x_7 = 1 & (\textit{let } 7)
\end{array}$$

za sve cjelobrojne varijable x_i*,* $i \in \{1, \ldots, 8\}$ *je točka* $(1, 1, 1, 1, 1, 0, 0, 1)$.

Iz teorema 3.1.1 i 3.2.2 zaključujemo da je ograda na optimalnu vrijednost zadaće koja opisuje problem dobivena linearnom relaksacijom zadaće u skladu s formulacijom (3.4) „bolja" u odnosu na ogradu dobivenu linearnom relaksacijom zadaće u skladu s „tradicionalnom" formulacijom. Izvedeni zaključak o ogradama opravdava posezanje za novom formulacijom. Također, zadaća linearne relaksacije opisane nove formulacije u nekim problemima ima cjelobrojnu optimalnu točku, dok to nije slučaj kod tradicionalnog pristupa. No, linearna relaksacija prethodno opisane formulacije zasnovane na dužnostima pokazala se težom za riješiti od linearne relaksacije tradicionalnog pristupa.

Bibliografija

[1] G. Desaulniers, J. Desrosiers, M. M. Solomon, F. Soumis i D. Villeneuve, *A unified framework for deterministic time constrained vehicle routing and crew scheduling problems*, Fleet Management and Logistics (T. Crainic, G. Laporte, ur.), Springer (1998), 57–93.

[2] C. Lemaréchal, *Lagrangian Relaxation*, Springer (2001), 112–156.

[3] M. Lübbecke i J. Desrosiers, *Selected Topics in Column Generation*, Operations Research 53(6) (2005), 1007–1023.

[4] P. Vance, E. Johnson C. Barnhart i G. L. Nemhauser, *Airline Crew Scheduling: A New Formulation and Decomposition Algorithm*, Operation Research (1997), 188–200.

Sažetak

Problemi rasporeda u zračnom prometu primjer su složenijeg tipa rasporeda koji u obzir uzima brojna ograničenja i regulative postavljene od strane zrakoplovne indrustrije. U ovom radu naglasak je stavljen na problemu izrade rotacija posade – niz dužnosti s odmorima između koje pokrivaju letove tako da posada počinje i završava putovanje u istoj bazi. Taj problem matematički opisujemo optimizacijskim problemom, i to problemom cjelobrojnog linearnog programiranja. Cilj je pronaći optimalan skup rotacija uz minimalan trošak tako da svaki od letova bude obuhvaćen točno jedanput. Navodimo standardnu formulaciju problema izrade rotacija te predlažemo metode za njihovo rješavanje.

Linearna zadaća kojom opisujemo problem izrade rotacija posade obično ima veliki broj uvjeta. Broj uvjeta smanjujemo primjenom Dantzig-Wolfeovom dekompozicije zadaće. Dantzig-Wolfeove dekompozicija zasniva se na teoremu reprezentacije poliedarskog skupa. Poliedarski skup prikazujemo preko njegovih ekstremnih točaka i ekstremnih zraka, čime broj varijabli obično postane vrlo velik. Rješavanje zadaće linearnog programiranja s velikim brojem varijabli je složenije, pa se javlja potreba za efikasnom metodom.

Metoda generacije stupaca primjenjuje se na reduciranoj zadaći s manjim podskupom stupaca. U svakoj iteraciji traži nebazičnu varijablu koja će ući u bazu i optimirati funkciju cilja reducirane zadaće. Odluku o generaciji novog stupca dobijemo rješavajući pomoćni problem određivanja cijena. U slučaju blok-dijagonalne strukture matrice uvjeta, formulacija zadaće linearnog programiranja u obliku Dantzig-Wolfeove dekompozicije u kombinaciji s generacijom stupaca može biti primjenjiva na paralelno rješavanje potproblema umjesto na rješavanje originalnog problema. Iza teorije metode generacije stupaca stoji teorija dualnosti.

Još jedan pristup rješavanju problema je Lagrangeova relaksacija. To je pristup koji omogućuje „preoblikovanje" modela uklanjanjem „kompliciranih" uvjeta te njihovim prebacivanjem u funkciju cilja, uz pridruživanje Lagrangeova multiplikatora. Razradom teorije Lagrangeove relaksacije dolazi se do zaključka da je za bilo koju vrijednost Lagrangeovog multiplikatora optimalna vrijednost relaksirane zadaće donja ograda na optimalnu vrijednost funkcije cilja izvorne zadaće. Cilj je pronaći najbolju moguću ocjenu odozdo, tj. pronaći što „kvalitetniji" multiplikator za koji je optimalna vrijednost Lagrangeove relaksirane zadaće što veća moguća – rješavamo Lagrangeovu dualnu zadaću. U nastavku

razmatramo konveksifikaciju Lagrangeova duala, što nam omogućuje primjenu Dantzig-Wolfeove dekompozicije, tj. zapis konveksne ljuske skupa preko njegovih ekstremnih točaka i ekstremnih zraka.

Opisan je pristup rješavanju zadaće cjelobrojnog linearnog programiranja generacijom stupaca.

U konačnici, kako popisivanje rotacija iz prve, tradicionalne formulacije problema može biti vrlo zahtjevno, ponudili smo drugačiju formulaciju problema rotacija posade. Prednost nove formulacije problema je što njegova linearna relaksacija daje „bolju" ogradu na optimalnu vrijednost funkcije cilja izvorne zadaće u odnosu na linearnu relaksaciju zadaće u skladu s tradicionalnom formulacijom.

Summary

Airline scheduling problems are an example of a more complex type of scheduling that takes into account a lot of restrictions and regulations posted by airline companies. In this paper, the accent is on solving the crew pairing problem – grouping series of duties with breaks which cover flights in the way that crew begins and ends the journey at the same base. Mathematically saying, we describe this problem by optimization problem, or to be more precisely, by the problem of integer linear programming. The goal is finding optimal pairings with minimal cost to cover each flight exactly once. We give a standard formulation of the problem and suggest methods for its solving.

The linear problem which is used to describe the crew pairing problem usually has a lot of conditions. We reduce the number of conditions by applying Dantzig–Wolfe decomposition on the problem. Dantzig–Wolfe decomposition is based on representation theorem of polyhedron. We represent polyhedron in terms of its extreme points and extreme rays, which usually makes the number of variables very large. Solving a linear problem with large number of variables is more complex, so there is a need for an efficient method for solving it.

The column generation method is applied on a restricted master problem with a small subset of columns. In each iteration, it looks for a non-basic variable to enter the base and optimize the objective function. We get a decision to generate a new column by solving pricing subproblem. If condition matrix has a block–diagonal structure, the formulation of the linear programming problem in terms of Dantzig–Wolfe decomposition in a combination with column generation can be applied to the parallel subproblem solving instead of solving the original problem. There is the theory of duality behind the theory of the column generation.

Another approach to problem solving is Lagrange relaxation. It is the approach that enables reshaping the model by removing complicated conditions and putting them to the objective function with addition of a Lagrangian multiplier. Elaboration of the theory of Lagrange relaxation leads to the conclusion that for any value of the Lagrange multiplier the optimal value of the relaxed problem is the lower bound for the optimal value of the objective function of the original problem. The goal is to find the best possible lower bound, i.e. to find the multiplier of the highest quality for which the optimal value of

Lagrangian relaxed problem is as large as possible – we solve Lagrangian dual problem. Further, we consider the convexification of the Lagrangian dual, which allows us to apply the Dantzig-Wolfe decomposition, i.e., set formulation in terms of its extreme points and extreme rays. There is a description of an approach for solving the problem of integer linear programming by column generation.

Finally, since the enumeration of pairings from the first, traditional formulation of the problem can be very demanding, we offer a different formulation of the crew pairing problem. The advantage of the new problem formulations is that its linear relaxation provides stronger bound on the optimal value of objective function of the original problem compared to the linear relaxation of the problem according to the traditional formulation.

Životopis

Rođena sam 3. veljače 1997. u Čapljini, Bosna i Hercegovina. Nakon završene osnovne škole, upisala sam Opću gimnaziju u Stocu koju sam završila 2015. godine. Iste godine upisujem preddiplomski studij Matematika na Matematičkom odsjeku Prirodoslovno–matematičkog fakulteta Sveučilišta u Zagrebu, koji sam završila 2018. godine. Iste godine upisala sam diplomski sveučilišni studij Računarstvo i matematika na istom fakultetu. Tijekom studiranja pohađala sam više ljetnih škola iz programskog jezika Java. Tijekom diplomskog studija bila sam stipendist firme Atos.